ABSTRACT

Expert System for
the Injection Molding of
Engineering Thermoplastics

by
Tzy-Cherng Jan

Injection molding of engineering thermoplastics is the most widely used manufacturing method in industry. It is a priority to maintain a deviation-free operating environment to ensure high quality, low cost manufacture. An expert system for the injection molding of engineering thermoplastics has been investigated. The system can be used to attenuate the deviations experienced during the injection molding of engineering thermoplastics. The system is coded in C programming language.

The resolution procedures of this system include two stages such as the definition of declarative knowledge and the procedure of corrective actions. In the definition of declarative knowledge, all of the necessary information is collected for firing the inference engine. This information includes the material type, the material manufacturer, the material grade, the recommended operating conditions, the operating conditions, the deviation type, and the correlative weighting factors.

The procedure of corrective action is classified by fishbone diagram into four different levels. These levels include and are ranked as method corrective actions, operating variable corrective actions, mold corrective actions, and material corrective actions.

The rule values of the corrective action in each level are assigned to determine the rank for employing these corrective actions. Among those rule values, the rule values of the method corrective actions, of the mold corrective actions, and of the material corrective actions are determined by the degree of difficulty required to eliminate the deviation and the input of the molding experts. A decision algorithm is developed to calculate the priority weighting factors, rule values, of each operating variable corrective

action. Furthermore, the Pareto principle is introduced to analyze the control parameters of the decision algorithm.

During the interactive procedures of eliminating the deviation, the system provides an explanation function for each step. It allows the system to illustrate the reason for each action to the user. A self-learning mechanism is also developed in this study. This self-learning mechanism based on the response of the resolution results modifies the parameters which influence the sequence of the corrective actions.

The system has been examined by experts in the field of injection molding. It is recognized that the system not only provides reasonable resolution sequences for eliminating the deviation, but also, accurate suggested actions for the user. Furthermore, the resolution actions have been simulated in the injection molding filling package-MOLDFLOW. This confirms that the resolution actions can actually influence the parameters which can eliminate or reduce the deviations.

EXPERT SYSTEM FOR THE INJECTION MOLDING OF ENGINEERING THERMOPLASTICS

by
Tzy-Cherng Jan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
Department of Mechanical and Industrial Engineering
October, 1992

APPROVAL PAGE

Expert System for
the Injection Molding of
Engineering Thermoplastics

by
Tzy-Cherng Jan

Dr. Keith T. O'Brien, Dissertation Adviser
Director of the Plastics Processing Laboratory
Professor of Mechanical Engineering
Department of Mechanical and Industrial Engineering
New Jersey Institute of Technology

Dr. Rong-Yaw Chen, Committee Member
Professor of Mechanical Engineering
Department of Mechanical and Industrial Engineering
New Jersey Institute of Technology

Dr. Ernest S. Geskin, Committee Member
Professor of Mechanical Engineering
Department of Mechanical and Industrial Engineering
New Jersey Institute of Technology

Dr. Nouri Levy, Committee Member
Associate Professor of Mechanical Engineering
Department of Mechanical and Industrial Engineering
New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Name: Tzy-Cherng Jan

Degree: Doctor of Philosophy in Mechanical Engineering

Date: October, 1992

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering, New Jersey Institute of Technology, Newark, NJ, U.S.A., 1992

- Master of Science, New Jersey Institute of Technology, Newark, NJ, U.S.A., 1988

- Bachelor of Science in Marine Engineering, Tamkang University, Tan-Shui, Taiwan, R.O.C., 1983

Major: Mechanical Engineering

Presentations and Publications:

Jan, Tzy-Cherng and Keith T. O'Brien. 1992. "Reducing surface defects in the injection molding of thermoplastics using a hybrid expert system," *the 50th ANTEC of the Society of Plastics Engineers*, Detroit **39:** pp.2214-2217.

Jan, Tzy-Cherng and Keith T. O'Brien. "A user friendly interactive expert system for the injection molding of engineering thermoplastics," Accepted for publication, *International Journal of Advanced Manufacturing Technology*.

Jan, Tzy-Cherng and Keith T. O'Brien. 1991. "Architecture of an expert system for injection molding problems," *the 49th ANTEC of the Society of Plastics Engineers*, Montreal, **38:** pp.439-443.

This dissertation is dedicated to
my beloved parents
Hurng-Puu Jan and Wang Huei-Lan Jan
and my children
Jason T. Jan and Joanne T. Jan
and my wife
Hur Yur-Maan Jan

ACKNOWLEDGMENT

The author would like to express my sincere gratitude to my advisor, Dr. Keith T. O'Brien, for his guidance, friendship, wisdom, criticism, and moral support throughout this research.

Special thanks for Dr. Rong-Yaw Chen, Dr. Ernest S. Geskin, and Dr. Nouri Levy, for serving as members of the committee and for their constructive encouragement and suggestions.

The author would like to express his gratitude to the New Jersey Institute of Technology Foundation, and the AT&T Foundation for their financial support of this work. I would further like to express my appreciation to the Department of Mechanical and Industrial Engineering at New Jersey Institute of Technology for financial support and for use of their facilities.

The author indebted to his parents for their encouragement throughout the period of studying abroad. Finally, special thanks to Yur-Maan, Jason and Joanne for all their patience and understanding during this study.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF SYMBOLS

A_s	is the solid bed area.
A_t	is the cross-sectional area of the channel.
$A_{j,m}$	is the correlative weighting factor of operating variable, j, and influencing physical property, m.
A^*	is the solid bed area ratio.
A_f^*	is the solid bed area ratio at begin of screw rotation.
A_i^*	is the solid bed area ratio at end of screw rotation.
$A_{j,m}^*$	is the modified correlative weighting factor of operating variable, j, and influencing physical property, m, in self-learning mechanism.
$A_{j,m,i}^*$	is correlative weighting factors of influencing physical property, m, and, operating variable, j, after the modification of correlative weighting factor, $E_{i,m}$ and $V_{i,j}$, for particular deviation, i.
$A'_{j,m,i}$	is correlative weighting factor of influencing physical property, m, and operating variable, j, after the modification of correlative weighting factor, $E_{i,m}$.
$B_{j,n}$	is the correlative weighting factor of operating variable, j, and inherent physical property, n.
B_r	is the Brinkman number.
$B_{j,n}^*$	is the modified correlative weighting factor of operating variable, j, and inherent physical property, n, in self-learning mechanism.
$B_{j,n,i}^*$	is correlative weighting factors of inherent physical property, n, and, operating variables, j, after the modification of correlative weighting factor, $V_{i,j}$, for particular deviation, i.

C_1, C_2, C_3 is the experimental constants.

D_b is the barrel diameter.

$E_{i,m}$ is correlative weighting factor of influencing physical property, m, and the particular deviation, i.

$F_{i,h}$ is the highest total frequency percentage of correspond correction action for the specific deviation, i.

$F_{i,j}$ is the total frequency percentage of correspond correction action, j, for the specific deviation, i.

H is the depth of the screw channel.

N is the screw speed.

P_i is the pressure at end of injection stage.

P_a^* is the average dimensionless pressure.

Q_c is the heat received by the cooling medium.

Q_m is the heat removed from the molten material.

$\dot{Q}$ is the constant injection rate.

$\dot{Q}_d$ is the volume flow rate of drop flow.

$\dot{Q}_p$ is the volume flow rate of pressure flow.

Q^* is the dimensionless flow rate.

R_e is the Reynolds number of cooling medium.

T_b is the barrel temperature.

T_e is the ejection temperature.

T_m is the inlet melt temperature.

T_{mold} is the mold temperature.

T^* is the dimensionless temperature.

T_a^* is the average dimensionless temperature.

$V_{i,j}$ is correlative weighting factor of the particular deviation, i, and operating variable, j.

$V^{*}_{i,j}$ is the modified correlative weighting factor of deviation, i, and operating variable, j, in self-learning mechanism.

VA_{j} is the operating condition of the operating variable, j.

$VA_{(j)\text{adj}}$ is the adjusted amount of operating variable, j.

$VA_{(j)\max}$ is the maximum recommended condition of operating variable, j.

$VA_{(j)\min}$ is the minimum of recommended condition of operating variable, j.

$VA^{*}_{(j)\max}$ is the modified maximum recommended operating condition of operating variable, j, in self-learning mechanism.

$VA^{*}_{(j)\min}$ is the modified minimum recommended operating condition of operating variable, j, in self-learn mechanism.

$VA_{(j)\text{new}}$ is the updated operating condition of operating variable, j.

c_{ps} is the specific heat of the material.

e is the flight width.

i is the latent heat of fusion of the material.

$\dot{m}$ is the output of molten material.

n is the power-law exponent.

r_{m} is the molten material density.

s is the screw lead.

t_{c} is the cooling time.

t_{r} is the screw rotation time.

t_{t} is the total cycle time.

ΔP is the actual pressure drop in the disk-shaped cavity.

ΔP_{i} is the isothermal pressure drop in the disk-shaped cavity.

Δp	is the pressure difference across the melt zone.
α	is thermal diffusivity of the polymer.
α_m	is thermal diffusivity of molten polymer.
α_s	is the thermal diffusivity of solid polymer.
β	is the experiment constant.
δ	is the melt film thickness.
ϕ	is the screw helix angle = $\tan^{-1}$ [s/(p D_b)].
λ	is the thermal conductivity of the polymer.
λ_{st}	is the thermal conductivity of the mold material.
λ_m	is the thermal conductivity of molten polymer.
λ_s	is thermal conductivity of solid polymer.
η_m	is the molten material viscosity.
θ	is the melt flow angle.
ρ_e	is the density of molten material at the end of injection stage.
ρ_i	is the initial density.
ρ_m	is the density of molten polymer.
ρ_a^*	is the average dimensionless density.
τ	is the filling time.
υ	is the number of the flights.

CHAPTER ONE
INTRODUCTION

1.1 Introduction

In 1952, William H. Willert [DuBois, 1972][Willert, 1956] developed the concept of the in-line reciprocating screw plasticating injection unit for the injection molding process. It was patented in 1956. The injection molding method has been widely used to manufacture low cost, and high productivity parts with complicated geometries from the plastics. Since, it has become the most popular item of manufacturing equipment.

The injection molding process is a process that the particular solid feedstock is molten by heat applied through the barrel by the band heater, and the screw rotation which generates heat by viscous dissipation. It is then conveyed to the front of screw by the rotation of the screw. After a charge of molten material has accumulated, it is injected into the mold cavity by the forward movement of the screw. While the molten material is injected into the mold cavity, the cooling water or oil circulates around in the halves of mold to solidify the molten material. Once the molten material has been completely solidified, the solidified part is then ejected by the ejector mechanism. The entire process is then completed and cycle itself.

Obviously, the injection molding process is a very complex process. It involves many fields of science and engineering, such as rheology, heat transfer, fluid dynamics, friction, polymer science, and control theory. Since, a mathematical model to represent the whole process does not exist, nor is likely to be developed in this century. Also, it is clear that there are many opportunities for the deviations to occur during the process. Therefore, ensuring a good quality product is also difficult.

However, to ensure a good quality product is the principal objective for the injection molding process. The question arises as to how to ensure a good quality product in the injection molding process. In this program, an expert system technology is introduced to eliminate or reduce the deviations occur during the injection molding process.

1.2 Need for an Expert System

There are three key elements which determine the quality of injection molded products. These are,

- the quality of the raw material,
- the capabilities of the equipment and,
- the skills of the operating personnel.

The quality of the raw materials and the capabilities of the equipment have been developed significantly. However, the skills of the operating personnel have probably peaked, and may even be in decline.

Typically, molding personnel need 10 to 20 years experiences to become expert molding personnel. It is a costly item. The other primary disadvantage regarding expert molding personnel is how to make the experts' resolution skills useful to other molding personnel. Since, the experts' resolution skill comes from their working experience, their resolution skills probably exceed their understanding of the technology. Also, the experts' resolution skills usually involve personal intuition. Sometimes, it is hard to explain the reason to take these actions. The result is that their resolution skills are hard to codify and transfer to other molding personnel.

Furthermore, the experts' resolution skill is only appropriate to particular situations to which their experience relates. For a new situation, their resolution skills probably become invaluable. Also when an expert leaves the plant their resolution skills leave with them. These expert molding personnel limitations make it difficult to manufacture high quality products consistently by injection molding.

The question arises as to how the experts' resolution skill can be made available to other personnel. Usually, a mathematics equation or model to explain the process is developed. However, few experts have the ability to set up the mathematical equations or models. Even, if someone has this ability, it is doubtful that there exists a realistic mathematical equation or model which can be used to explain the entire injection molding process.

To overcome this problem, resort may be made to a computer technology known as an expert system. An expert system consists of two major elements which are the knowledge base and the inference engine. To collect and analyze the experts' experiences, the knowledge base codes the information to make useful statements for the users. Then, the inference engine mechanism is used to fire the resolution sequence strategy. This simulates the experts' experience, and seeks the best resolution statement available in the knowledge base.

1.3 Features of the Dissertation

In this study, the use of an expert system for the injection molding of engineering thermoplastics has been researched. The system can be used to attenuate the deviations experienced during the injection molding of engineering thermoplastics. The system consists of four major elements. These are the user interface, the inference engine, the knowledge base, and the explanation facility.

The user interface is a menu-driver type. Its functions are to collect the necessary information for firing the inference engine, and respond with the suggested action statements to the user. This information includes the material type, the material supplier, the material grade, the operating variables, the type of deviation, the confirmation of correlative weighting factors, the recommended operating conditions, and the resultant suggested actions.

A hybrid type of inference engine was used in this system. This inference engine combines with the backward-chaining and rule-value search. It is based on collected information. It then employs the correlative weighting factors of the specified material, which are determined from four correlations,

- the cross correlation of the operating variables and the deviation,
- the cross correlation of the influencing physical properties and the deviation,
- the cross correlation of the operating variables and the influencing physical properties and,
- the cross correlation of the operating variables and the inherent physical properties,

These are entered into the decision algorithm to calculate the priority of the weighting factor for each operating variable. According to the priority of the weighting factor, the inference engine fires the rule searching strategies to seek the optimum suggested actions from the knowledge base of the expert system.

In this expert system, the suggested action statements can be divided into four different types. These are,

- the method corrective actions,
- the operating variable corrective actions,
- the mold corrective actions and,
- the material corrective actions.

The method corrective action guides to the user to resolve the experienced deviation from the examination of the observed causes of the deviations. These causes can be determined by the view of point during the process. The operating variable corrective action is decided through some specific measurement devices, such as, the thermal coupler, the pressure transducer, and the timer. The amount of the operating variable, which will be adjusted through some specific controllers, shows in the operating variable adjustment suggested actions statements. The mold corrective action statement is the suggestions, which pilot the user to resolve the deviation from the modification of the mold design. The material corrective action is the suggestions which guide the user to resolve the deviation for changing the material type.

For each deviation resolution sequence, the method corrective actions always have the highest priority. The steps in the method corrective actions are based on the degree of difficulty required to eliminate the deviation. This has been determined a priori by experts. Once the method corrective actions have been exhausted, the operating variables corrective actions are then employed by the system. The precedence is based on the priority weighting factors which have been calculated in the inference engine. If the operating variable corrective actions cannot resolve the deviation, then the mold corrective actions will be implemented. As with the method corrective actions, the mold corrective actions are based on the evaluations of experts. Finally, when the mold corrective actions have been exhausted, the material corrective actions are then employed in resolving the deviation by the system. If all corrective actions have been exhausted and the deviation still cannot be resolved, then, the final suggested action statement, "consult with the raw material supplier or the experts" will be presented to the user.

During the interactive procedures of eliminating the deviations, the system provides an explanation function for each step. It allows the system to illustrate why the action was suggested to the user.

The system has been examined by experts in the field of injection molding. It is recognized that the system not only provides reasonable resolution sequences for eliminating the deviation, but also, accurate suggested actions for the user. Furthermore, the resolution actions have been simulated in the injection molding filling package-MOLDFLOW. This confirms that the resolution actions can actually influence the parameters which can eliminate or reduce the deviations.

In the next chapter, the components of injection molding machine and the basic principles of injection molding will be addressed. Then in chapter three, the deviations which occur during injection molding will be introduced. This includes the causes of and the resolution analysis of the deviations. In chapter four, a basic introduction of the expert system is presented. The detailed development procedures of the system are then presented in chapter five. In this chapter, the development procedures of this program is used to illustrate the design procedures of an expert system. Then case studies for the performance of the system is studied in chapters six. The discussion is presented in chapter seven. This includes the deviation resolution procedures and using the injection mold filling package-MOLDFLOW[1] simulates the resolution results. The decision mechanism and self-learning capability of the system are also introduced in the late of chapter seven. .The conclusions and future works of this study are then presented in chapter eight.

---

[1]MOLDFLOW is the registered trademark of MOLDFLOW PLT. Australia.

CHAPTER TWO
PRINCIPLES OF INJECTION MOLDING

2.1 Introduction

Injection molding is the most important method for processing engineering thermoplastics. During the process, the material is melted by heat conducted from the band heaters around the plasticating unit, and by the viscous dissipation imparted by the rotation of the screw in the barrel on the material. Then, the molten material is injected into the mold cavity as the screw moves forward. After the molten material is injected, it solidifies in the mold, which is cooled by water, or oil, which circulate in the cooling channels of the mold. Then, the mold opens, and the solidified part is ejected by the ejector pins. Then the cycle repeats itself.

The Hyatt injection molding machine can be said to be a prototype of the modern injection molding machine [DuBois, 1972]. The original drawing for this machine is shown in Figure 2.1. In the Hyatt injection molding machine, the cellulose acetate was plasticated in a heating cylinder and pushed into the mold by a piston, which was driven by hydraulic pressure. Since heat was supplied only by conduction, the heating times were very long and production rates were slow. Thus industry growth was inhibited.

In 1932, Hans Gastrow [DuBois, 1972] invented the Gastrow torpedo plastics injection unit which is shown in Figure 2.2. The major improvement of this invention is that the material plasticating capability is increased greatly. In the Gastrow torpedo plasticating unit, the material enters from the hopper and a precise quantity of pellets is deposited in the cylinder. Then the plunger moves forward and forces the material across the torpedo and through the nozzle. As the material passes across the torpedo, it is softened by the heat supplied by the barrel heater.

The most important improvement of the injection molding process occurred in 1957 [DuBois, 1972] [Willert, 1957], when William H. Willert patented the in-line reciprocating screw plasticating injection unit. Figure 2.3. shows the prototype drawing of the in-line reciprocating screw plasticating injection unit. Willert used an extrusion screw to replaced the plunger of the Gastrow torpedo, and placed it in a temperature controlled heating cylinder. The major improvement of this invention is that it provides a more uniform mixing of the plastics compounds than was available before by introducing viscous dissipation into the heating component of the cycle. The heating time was dramatically reduced and production rates soared and production cost plummeted.

After Willert's invention, the injection molding process for the manufacture the plastic products has grown steadily. Rosato and Rosato [Rosato and Rosato, 1986] stated that in 1984 approximately 32 percent by weight of all plastics products are manufactured by injection molding. A comparison of the advantages and disadvantages of the injection molding process is presented in Table 2.1. The most significant advantage of the injection molding process is that it can be used to manufacture products economically in unlimited quantities with no finishing operations.

In the following text, the components of the in-line reciprocating screw injection molding machine, the elements of the injection molding method, and the processing variables for injection molding will be discussed.

2.2 Reciprocating Screw Injection Molding Machine

A typical injection molding machine is shown in Figure 2.4. The injection molding machine consists of three major elements. These are,

- the plasticating unit,
- the clamping unit and,
- the mold.

2.2.1 Plasticating Unit

The function of the plasticating unit is to melt the solid plastics pellets in the plasticating cylinder, and inject the molten material toward the mold cavity as the screw moves forward.

There are four common types of plasticating unit used. These are, 1) the one stage plunger type, 2) the pre-plasticating, two stage plunger type, 3) the pre-plasticating, two stage plunger-screw type, and 4) the in-line reciprocating screw type. Today, approximately 95 percent of plasticating units are of the in-line reciprocating screw type [Kroshcwitz, 1987]. In addition, the plunger type and the pre-plasticating types are no longer manufactured. Therefore, the focus of this thesis is directed to the in-line reciprocating screw type.

Figure 2.5 shows a typical in-line reciprocating screw plasticating unit. The operation sequence for the in-line reciprocating screw plasticating unit is illustrated in Figure 2.6. In the first step, the molten material is deposited between the front of the screw and nozzle. Then, when the mold is closed, the plasticating unit will move forward as the hydraulic power is applied to the injection ram. This action causes the molten material to be injected into the mold cavity as shown in Figure 2.6(a). Once the mold cavity has filled, the pressure in the plasticating unit is maintained until the material has solidified in the gates. This action prevents the material from back flowing towards the nozzle, and causes additional material to pack the mold cavity as shrinkage occurs.

Once the material had solidified in the mold gates, the screw begins to rotate and produce a new charge. As it does so, it retracts against the back pressure as shown in Figure 2.6(b).

Once the material for next molding cycle has been prepared, the solidified parts in mold have been ejected, and the mold has been closed again, the next cycle starts. Usually, there is a period of delay for the plasticating unit between the end of the screw rotation and the start of next injection which is shown on Figure 2.6(c). This delay time is termed the soak time or idle time.

Since the reciprocating screw type of plasticating unit has a continuous conveying motion and the plastic is melted down due to the shearing action on the material, the following advantages exist,

- material is mixed more uniformly,
- stock temperature is reduced,
- molded material ranges are greater,
- material flow back in the screw is restricted,
- shot size is controlled more exactly,
- injection pressure loss is minimized,
- volatiles may be removed from the cylinder during plasticating and,
- material is plasticated more effectively.

2.2.2 Clamping Unit

The clamping unit is either hydraulically or mechanically actuated. In industry, one measure of the size of the injection molding machine is the capacity of clamping unit. Also, since the clamping unit consists of a complex mechanical and hydraulic system, it is a major cost factor for the injection molding machine.

There are two major functions for the clamping unit. First, it must create a sufficient clamping force on the mold to keep the mold closed during material injection into the mold cavity, and subsequent packing. Second, it operates the opening and closing of the mold.

In general, there are two common configurations which are used for the clamping unit. These are 1) the hydraulically operated toggle clamping unit (toggle clamping unit), and 2) the fully hydraulic clamping unit.

2.2.2.1 Toggle Clamping Unit. A toggle clamping unit is a mechanical device, used to amplify the force which is generated from a hydraulically actuated cylinder, and to apply it to the mold. Figure 2.7 shows a common double-acting toggle clamping. As the mold is opened, the hydraulic actuating cylinder will retract and pull the crosshead backward. This action causes the moving plate to move away from the stationary plate, and open the mold. As the mold is closing, the hydraulic actuating cylinder will extend, and push the moving plate forward towards the stationary plate. The crosshead is a mechanism system which amplifies the force. In other words, when the crosshead is straightened, the force which is generated by the hydraulic actuating cylinder will be amplified and transferred to the mold. In addition, the crosshead has a facility to lock. It can prevent the mold slipping when the material is injected into the mold cavity.

2.2.2.2 Fully Hydraulic Clamping Unit. The difference between a fully hydraulic clamping unit and a toggle clamping unit is that the fully hydraulic clamping unit is supplies the force from the hydraulic cylinder directly as shown in Figure 2.8. Since, the fully hydraulic cylinder can generate hydraulic forces of great magnitude, most large machines use fully hydraulic clamping units.

Both the toggle and the fully hydraulic clamping unit work well. In Table 2.2, a comparison between the toggle and fully hydraulic clamping unit, is presented based on equivalent operating conditions.

2.2.3 Mold

The mold is the heart of the injection molding process. Its design will directly influence the quality of the part. A mold provides several functions, the creation of the configuration of the parts, the solidification of the material, the cooling of the material, the venting of entrapped air, and the ejection of the parts. An exploded view of a standard mold base is presented in Figure 2.8.

Basically, a mold consists of several of components that include,

- the locating ring and sprue bushing,
- the front clamping plate, A plate, B plate, leader pins, and leader pin bushings,
- the support plate, ejector housing, knock-out plate, return pins, ejector pins, and sprue puller,
- the cooling channels, and venting system and,
- the runner system, gates, cavity and core.

These will now be explained in detail.

2.2.3.1 Locating Ring and Sprue Bushing. The locating ring surrounds the sprue bushing and fixes the sprue bushing by a screw in the front of the clamping plate which maintains the center of sprue adjacent to the center of the nozzle of the injection cylinder. It also guides the nozzle of the injection cylinder to the sprue bushing when material is injected into the cavity. In addition, the sprue bushing provides a passage through the front clamping plate. It provides a channel which allows the injected material passage into the cavity.

Usually, the sprue bushing has the equivalent countershape of the nozzle of the injection cylinder. The diameter of sprue bushing is always greater than the diameter of the nozzle of the injection cylinder, so that when the material solidifies it will not generate a solidified mass larger than the diameter of the sprue bushing. This facility prevents the material sticking in the sprue bushing. Furthermore, to provide the easy removal of the material from the sprue bushing, it is taper. Generally, its taper is 0.5 in/ft.

2.2.3.2 Front Clamping Plate, A Plate, B Plate, Leader Pins, and Leader Pin Bushing. The front clamping plate, which is also called stationary plate, supports the A plate. It is set during the mold operating. A cavity which is outside the configuration of the part is installed in the A plate. There are four leader pins in the A plate. These pins are guided into four corresponding pin bushings, which are installed in the B plate. The B plate is supported by the support plate. In the B plate, a mold core, which is the inside configuration of the part, is installed. The function of the leader pins and bushings is to align the A and the B plates into the correct position relevant to each other, and hence ensure that the cavity and core are aligned.

2.2.3.3 Support Plate, Ejector Housing, Knock-Out Plate, Return Pins, Ejector Pins, and Sprue Puller. The support plate frequently rests on the ejector housing . Usually, the ejector housing is a U-shape structure. The ejector housing has two components. These are the rear clamping plate and the spacer block, which are either as a separate part or as a welded part. The ejector housing is bolted to the B plate. In the inside of ejector housing, there is an ejector plate and an ejector retainer plate, which are screwed together. Sometimes, the combination of the ejector retainer plate and ejector plate is termed knock-out plate. During the mold operating, the knock-out plates slip along on the inside of the ejector housing.

The knock-out plate is supported by the return pins. The return pins are installed in the ejector retainer plate. The holes are drilled through the ejector plate, support plate, and the B plate. This allows the return pins to make contact with the A plate, when the mold is closed, and move the knock-out plate to its backward position. The ejector pins and the sprue puller are installed in the ejector plate. When the mold is opened, the knock-out plate is pushed by a bar in the clamping unit, and to move the knock-out plate forward. As the knock-out plate moves forward, the ejector pins are also moved forward to the outside of the core to eject the part. At the same time, the sprue puller drag the part from the cavity. Then, the part falls out of the mold.

2.2.3.4 Cooling Channel and Venting System. The cooling channel system circulates coolant around the mold halves. It is filled with the cooling water, oil, or air to remove the heat from the molten material and to solidify it. The layout of the cooling channel is very important components of the mold design. Unbalanced cooling results in the deviations for the part, such as, warpage, sink marks, splay marks, surface ripples, distortion, and delamination.

The venting system provides a tunnel to remove the air from inside cavity, and the gas which is generated by heating the material. To prevent the plastic from entering into the venting tunnel, the size of tunnel has to be very small. When the molten material is injected into the mold cavity, it compresses the air inside cavity which causes high pressure air. The high pressure air results in high temperatures. The heat of the air can cause burn spots on the parts. In addition, an improper venting system will cause short shots, poor weld lines, splay marks, and high internal stress of the parts.

2.2.3.5 Runner System, Gates, Cavity and Core. The runner is a passage between the sprue and the gate. Its dimension should be optimized. Too large a runner, requires long cooling times for the molten material in the runner. The results are increased cycle times, increased regrind, and reduced manufacturing productivity. In contrast, too small a runner, causes molten flow difficulties and high pressure drops in the runner. Furthermore, in multicavity molds, the runner must be balanced to ensure that every cavity can be filled at same rate. The different shapes of runners and the balanced runners are illustrated in Figure 2.9.

A gate connects the runner and the mold cavity. Its function is to control the flow rate of molten material into the cavity. In Figure 2.10, the different types of gate designs are presented. Usually, the cross-section of the gates is designed as a narrow and thin section. This design allows the molten material in gates freeze rapidly, and isolate the cavity at end of the injection stage. In addition, the narrow and thin section permits an easy way to disconnect the part from the runner.

The cavity and core provide the inside and outside configurations of the part. When the molten material is injected, the material passes through the sprue bushing, the runner system, and the gates, and fills the space between the cavity and core. Theoretically, a perfect design of mold consists of,

- an accurate fully dimensioned drawing of the parts,
- the tolerances of the parts,
- the shrinkage specification,
- the mold surface finish specification,
- the location of the tapers,
- the location of the ejector pins,
- the location and size of the gates,
- the layout of the cooling channels,
- the layout of the venting system,
- the runner system specification,
- the material type and,
- the size of the injection molding machine.

2.3 Elements of the Injection Molding Process

The injection molding process is a very complex process. It involves many fields of science and engineering, such as rheology, fluid mechanics, heat transfer, friction, polymer science, and control theory. Until today, there is no mathematical model or set of equations to represent the entire process.

Basically, the entire of injection molding process includes four major stages. These are,

- the injection stage,
- the packing stage,
- the cooling stage and,
- the plasticating stage.

A schematic of a single cycle of a reciprocating screw injection molding is illustrated in Figure 2.11. When the injection cycle starts, the mold is closed by the clamping unit as shown on Figure 2.12. At this time, the screw has finished plasticating material and has charged it into the screw chamber which is at front end of screw cylinder.

After the mold unit is closed, the plasticating unit will move forward and touch to the sprue bushing in the mold as shown in Figure 2.13. After these two actions, the screw drive system will force the screw forward causing the material, which is in the screw chamber, to be injected into the mold cavity as shown in Figure 2.14.

During the packing stage additional material will be injected to provide more material to the cavities to compensates for the shrinkage of the material during the cooling stage. After the mold cavity is filled, the cooling stage commences. The cooling water, or oil, circulates in the cooling channels of the mold and cools the part by conduction and convection. When the cooling stage begins, the screw will also begin to rotate, move backward and plasticating the material for the next cycle. As the screw rotates the material is forced into the screw chamber by the screw rotation, until the chamber contains the exact amount of the material require, the shot size. Figure 2.15 and Figure 2.16 show these actions.

Simultaneously with the plasticization, the material in the mold cavities is cooled and shrinks material must be packed into the mold to prevent voids and sink marks. This is called the packing stage. After the material in the mold cavities is solidified and cooled, the clamping unit will open the mold, and the part will be ejected from the mold as shown in Figure 2.17. The cycle is then complete. In the following text, the theoretical study for each stage is presented.

2.3.1 Injection Stage

In the injection stage, the major work is to ensure that the molten material can fill the cavity, and the clamping unit can provide a sufficient clamping force to hold the mold closed. To ensure these performances, the two key controlled parameters, 1) the injection pressure drop, and 2) the clamping force, must be considered. Here, a disk-shaped cavity, as shown in Figure 2.18, is used to illustrate the governing equations for these controlled variables.

The governing equation for the isothmeral injection pressure drop, ΔP_i [Stevenson, 1978], is,

$$\Delta P_i = \frac{K_r}{(1-n)}\left[\frac{360 \cdot \dot{Q} \cdot (1+2n)}{N \cdot \theta \cdot 4\pi \cdot n \cdot r \cdot b^2}\right]^n \tag{2.1}$$

the filling time, τ [Stevenson, 1978], is,

$$\tau = \frac{V \cdot a}{\dot{Q} \cdot b^2} \tag{2.2}$$

and the Brinkman number, B_r [Stevenson, 1978], is,

$$B_r = \frac{K_r \cdot b^2}{\lambda(T_m - T_{mold})}\left[\frac{360 \cdot \dot{Q}}{N \cdot \theta \cdot 2\pi n \cdot r \cdot b^2}\right]^{(1+n)}\left(\frac{r}{b}\right) \tag{2.3}$$

The actual injection pressure drop, ΔP, can be determined from Stevenson's experimental results [Stevenson, 1978], and is,

$$\ln\left(\frac{\Delta P}{\Delta P_i}\right) = 0.337 + 4.7 \cdot \tau - 0.093 \cdot B_r - 2.6 \cdot \tau \cdot B_r \tag{2.4}$$

where,

B_r = the Brinkman number,

N = number of gates,

$\dot{Q}$ = constant injection rate,

T_m = inlet melt temperature,

T_{mold} = mold temperature,

V = part volume,

b = half-thickness of the disc,

n = power-law exponent,

r = radius of the disk,

ΔP = actual pressure drop in the disk-shaped cavity,

ΔP_i = isothermal pressure drop in the disk-shaped cavity,

α = thermal diffusivity of the polymer,

τ = the filling time,

λ = thermal conductivity of the polymer,

θ = melt flow angle.

The calculation of clamping force is similar to the injection pressure drop. The governing equation for the isothermal clamping force, F_i [Stevenson, 1978], is,

$$F_i(r) = \pi \cdot r^2 [\frac{(1-n)}{(3-n)}] \Delta P_i \tag{2.5}$$

From the Stevenson's experimental result, the actual clamping force, F, is determined as,

$$\ln(\frac{F}{F_i}) = 0.372 + 7.6 \cdot \tau - 0.084 \cdot B_r - 3.538 \cdot \tau \cdot B_r \tag{2.6}$$

where,

F = actual clamping force,

F_i = isothermal clamping force.

2.3.2 Packing Stage

The key consideration of the packing stage is to ensure that the additional molten material to prevent shrinkage is packed into the mold. To ensure this performance, there are three key parameters which should be considered. These are 1) the average density, 2) the average pressure, and 3) the average temperature. As with the analysis of the injection stage for disk-shaped cavities, as shown in Figure 2.18, is used to illustrate the governing equations of these key parameters.

The average dimensionless density, ρ_a^* , is given as follows [Kamal and Kenig, 1972],

$$\rho_a^*(\tau^*) = 1 + \frac{b\int_{\tau^*}^{0} Q^* \cdot \rho_a^* \cdot d\tau^*}{\pi \cdot r} \tag{2.7}$$

The average dimensionless temperature, T_a^*, is given as follows [Kamal and Kenig, 1972],

$$T_a^* = 2\pi \int_0^1 \int_0^1 T^*(r^*, z^*)\, r^*\, dr^*\, dz^* \tag{2.8}$$

The average dimensionless pressure, P_a^*,is given as [Kamal and Kenig, 1972],

$$P_a^* = \frac{T_a^* \cdot C_1 \cdot (T_i - T_{mold})}{\left(\frac{1}{\rho_a^* \cdot \rho_i} - C_2\right) \cdot P_i} + \frac{C_1 \cdot T_{mold}}{P_i} \frac{1}{\frac{1}{\rho_a^* \cdot \rho_i} - C_2} - \frac{C_3}{P_i} \tag{2.9}$$

where,

C_1, C_2, C_3 = the experimental constants,

P_a^* = the average dimensionless pressure,

P_i = the pressure at end of injection stage,

Q^* = the dimensionless flow rate,

T^* = the dimensionless temperature,

T_a^* = the average dimensionless temperature,

T_i = the initial molten material temperature,

T_{mold} = the mold temperature,

b = the half-thickness of disc,

r = the radius of the disc,

r^* = the dimensionless radial distance,

z^* = the dimensionless axial distance,

ρ_a^* = the average dimensionless density,

$\rho_e^* = \dfrac{\rho_e}{\rho_i}$,

ρ_e = the density of molten material at the end of injection stage,

ρ_i = the initial density,

τ^* = the dimensionless time.

2.3.3 Cooling Stage

In the cooling stage, one needs to ensure that the part is sufficiently solidified. To achieve this performance, the key parameters are, 1) the cooling time, and 2) the mold temperature. A design of circular cooling channels, as shown in Figure 2.19, is used to illustrate the procedures for calculating these parameters.

To determine the cooling time, t_c, the governing equation is [Ballman and Shusman, 1959],

$$t_c = \frac{a}{2 \cdot \pi \cdot \alpha} \ln[\frac{\pi}{4}(\frac{T_i - T_{mold}}{T_c - T_{mold}})] \tag{2.10}$$

where,

T_e = the ejection temperature,

T_i = the initial molten material temperature,

T_{mold} = the mold temperature,

a = the maximum cavity thickness,

t_c = the cooling time,

α = the thermal diffusivity.

To determine the mold temperature, one needs to consider the heat transfer between the molten material and the cooling medium. From the heat transfer equation, the heat require to be removed from the molten material can be calculated from [Rao, 1991],

$$Q_m = [(T_m - T_e)c_{ps} + i] \cdot \rho_m \cdot \frac{a}{2} \cdot x$$

$$Q_m = [(T_m - T_e)\, c_{ps} + i]\, \rho_m\, \frac{a}{2}\, x \qquad (2.11)$$

And, the heat received by the cooling medium in the cooling time, t_c, is determined as follows,

$$Q_c = t_c [\frac{1}{\frac{1}{\lambda_{st} S_e} + \frac{1}{\chi 2 \pi r_c}}](T_{mold} - T_m) \qquad (2.12)$$

The shape factor, S_e, can be determined as follows [Throne, 1979],

$$S_e = \frac{2 \cdot \pi}{\ln[\frac{2x \cdot \sinh(\frac{2\pi \cdot y}{x})}{\pi \cdot d}]} \qquad (2.13)$$

and the heat transfer coefficient, χ, can be obtained from,

$$\chi = \frac{0.06279}{r_c} R_e^{0.8} \qquad (2.14)$$

where,

Q_c = the heat received by the cooling medium,

Q_m = the heat removed from the molten material,

R_e = the Reynolds number of cooling medium,

T_e = the ejection temperature,

T_m = the molten material temperature,

T_{mold} = the mold temperature,

a = the cavity thickness,

c_{ps} = the specific heat of the material,

i = the latent heat of fusion of the material,

r_c = the radius of cooling channel,

r_m = the molten material density,

x = the distance between cooling channel as shown in Figure 2.19,

y = the distance between cooling channel and cavity as shown in Figure 2.19,

λ_{st} = the thermal conductivity of the mold material.

Then, the mold temperature, t_c, in Eq. (2.10) can be determined iteratively from the heat balance between Q_m and Q_c.

2.3.4 Plasticating

In the plasticating stage, the major functions is to melt the solid plastics pellets, and convey the molten material to the front of the screw chamber.

Melting the solid plastics pellets consists of two phases. One is the heat conduction from the heater band, and, the other is the viscous dissipation imparted by the rotation of the screw in the barrel on the material.

The governing equation for melting the solid material from heat conduction is given as follows [Donovan, 1971].

The melt film thickness, δ, can be determined from,

$$\delta = K\sqrt{t} \qquad (2.15)$$

and, K can obtained from,

$$(i \cdot \rho_m)^K = -2\left\{\frac{(T_m - T_b)\cdot \lambda_m \cdot \exp(\frac{-K^2}{4\cdot \alpha_m})}{\sqrt{\pi \cdot \alpha_m}\cdot erf(\frac{K}{2\sqrt{\alpha_m}})} - \frac{(T_r - T_m)\cdot \lambda_s \cdot \exp(\frac{-K}{4\cdot \alpha_s})}{\sqrt{\pi \cdot \alpha_m}\cdot erf(\frac{K}{2\sqrt{\alpha_s}})}\right\} \quad (2.16)$$

where,

T_b = barrel temperature,

T_m = temperature of molten polymer,

T_r = temperature in middle of solid bed,

t = time,

α_m = thermal diffusivity of molten polymer,

α_s = thermal diffusivity of solid polymer,

δ = melt film thickness,

λ_m = thermal conductivity of molten polymer,

λ_s = thermal conductivity of solid polymer, and,

ρ_m = density of molten polymer.

The governing equation for viscosity dissipation is determined from the solid bed area ratio, A^*, [Donovan, 1978][Tadmor and Klein, 1970] as,

$$A^* = \frac{A_s}{A_t} \qquad (2.17)$$

To determine the ratio of the solid bed area during conductive melting, the following equation applies,

$$A_f^* - A_i^* = A_f^* \frac{[k\sqrt{t_t - t_r + (\delta^2 / K^2)} - \delta]}{H} \tag{2.18}$$

where,

A_s = solid bed area,

A_t = cross-sectional area of the channel,

A^* = solid bed area ratio,

A_f^* = solid bed area ratio at begin of screw rotation,

A_i^* = solid bed area ratio at end of screw rotation,

H = depth of the screw channel,

N = screw speed,

t_r = screw rotation time,

t_t = total cycle time,

β = experiment constant,

δ = melting thickness.

For conveying the molten material, the governing equation is [Bernhardt, 1963],

$$\dot{m} = 6 \cdot \pi^2 \cdot D_b^2 \cdot N \cdot H \cdot (1 - \frac{\upsilon \cdot e}{s}) \frac{\sin\phi \cdot \cos\phi \cdot (1 - \frac{\dot{Q}_p}{\dot{Q}_d}) - (\frac{\delta}{H})}{2} \cdot \rho_m \tag{2.19}$$

and,

$$\dot{Q}_p = \frac{-\pi \cdot D_b \cdot H^3 \cdot (1 - \frac{\upsilon \cdot e}{s}) \sin^2\phi \cdot \Delta p}{12 \cdot \eta_m \cdot L} \tag{2.20}$$

$$\dot{Q}_d = \frac{\pi^2 \cdot D_b^2 \cdot H \cdot N \cdot (1 - \frac{\upsilon \cdot e}{s}) \sin\phi \cdot \cos\phi \cdot \Delta p}{2 \cdot 60} \tag{2.21}$$

where,

D_b = the barrel diameter,

H = the channel depth,

L = the length of melt zone,

N = the screw speed,

$\dot{Q}_d$ = the volume flow rate of drop flow,

$\dot{Q}_p$ = the volume flow rate of pressure flow,

e = the flight width,

$\dot{m}$ = the output of molten material,

s = the screw lead,

Δp = the pressure difference across the melt zone,

ϕ = the screw helix angle = $\tan^{-1}$ [s/(p D_b)],

η_m = the molten material viscosity,

ρ_m = the molten material density,

υ = the number of the flights.

2.4 Variables in the Injection Molding Process

The injection molding process is a science which combines with the fields of rheology, fluid mechanics, heat transfer, friction, polymer science, and control theory. To manufacture a product from the injection molding process, there are many processing variables which need to be considered. Table 2.3 presents the processing variables involved in the process.

According to their characteristics, these processing variables can be classified into three groups,

- the operating variables,
- the material variables and,
- the mold design variables.

The detailed discussion of these variables is presented in the following text.

2.4.1 Operating Variables

The operating variables are adjusted during the process. These variables will directly influence the quality of the products. For the expert system, these variables are the key consideration to resolve the products' deviation. As shown in Table 2.3, these operating variables can be categorized into four groups as follows,

- temperatures,
- pressures,
- times and,
- displacements.

2.4.1.1 Temperatures. The operating temperatures which can be varied include the barrel temperature, the nozzle temperature, and the mold temperature. The major influence of these temperatures is that they control the phase change of the material from solid to melt, and from melt to solid. In addition, they control the viscosity of the melt. For melting the solid plastics, the barrel temperature and the nozzle temperature are key factors. The mold temperature is a key factor in the solidification process.

Insufficient barrel temperature and nozzle temperature will cause incomplete melting during the plasticating stage. As well it results in a high viscosity of the melt during the injection stage. This causes deviations, such as, short shots, surface ripples, pit marks, sink marks, delamination, and voids.

In contrast, too high a barrel temperature and nozzle temperature will cause the material to overheat during the plasticating stage. Also, it results in a low viscosity the melt during the injection stage. The influences of overheat is that the melt is difficult to solidify during cooling stage. With a low viscosity, the melt flows too freely into the cavity. This results in deviations, such as, splay marks, warpage, distortion, and flashing.

The influence of mold temperature is to solidify the molten material. Insufficient mold temperature causes the low viscosity of melt in the cavity. It results in deviations, such as, splay marks, short shots, surface ripples, pit marks, and sink marks. In contrast, too a high mold temperature causes incomplete solidification melt during the cooling stage. It results in parts hard to eject from the cavity, and causes deviations, such as, warpage, and distortion. In addition, a high mold temperature causes a high viscosity of melt in the cavity. It is a major cause for warpage, distortion, voids, and flashing.

2.4.1.2 Pressures. The operating pressure includes the injection pressure, the holding pressure, and the clamping force.

The major influence of the injection pressure is the flow rate of the melt during the injection stage. The injection pressure is proportionally with the flow rate of the melt. Too high an injection pressure causes a high flow rate of the melt. It is a major cause for splay marks, and the flashing.

However, a too low injection pressure causes a low flow rate of the melt. This results in that molten material maybe solidified before the cavity is fulfilled. In addition, it also causes the air trapped and a high surface temperature in somewhere of the cavity. It is the causes of several surface deviations, such as, short shots, surface ripples, pit marks, sink marks, and voids. In addition, the high surface temperature results in an uneven temperature distribution of the part surface. It is the effect of warpage and distortion of the part.

The holding pressure provides a sufficient pressure to prevent the melt flow backward to the nozzle. Furthermore, it provides an additional material to prevent the part's shrinkage. An insufficient holding pressure causes the melt flow backward to the nozzle. Its resultant is the melt drooping at nozzle, and cause the splay marks. It also causes insufficient addition melt injection into the cavity, and results in over-shrinkage of part.

The clamping force provides a force to keep the mold close during the injection and the packing stage. An insufficient clamping force which causes the mold slip during the process, and results the deviations, such as, flashing and inaccurate dimension of the products.

2.4.1.3 Times. The operating times of the injection molding process, include the cycle time, the cooling time, and the decompression.

The cycle time depends primarily on the parts' thickness. The effects of cycle time include the rate of plasticating, and the rate of cooling, during the whole process. A short cycle time which results deficient rate of plasticating and rate of cooling. It's is the effects of several surface deviations, such as, short shot, surface ripples, pit marks, sink marks, and splay marks. Also, the insufficient cooling rate results the uneven solidification of the parts during the cooling stage. This is effect of warpage, and distortion.

The cooling time primarily influences the cooling rate. An insufficient cooling time causes the incomplete solidified part. This results in deviations, such as warpage, and distortion. Furthermore, it causes parts which may be too soft to eject. In addition, the cooling time also controls the plasticating rate. An insufficient cooling time affects the melt incomplete plasticating. It is the effects of several surface deviations, such as, short shot, surface ripples, pit marks, sink mark, splay marks, and voids.

The decompression is to release the pressure which is generated during the material plasticating at the heating cylinder. Therefore, it can prevent nozzle drooling during the mold open time before inject the molten material into the cavity. An insufficient decompression causes the material drooping at the nozzle, and is one of the major effect of splay marks.

2.4.1.4 Displacements. There are three key parameters involved in the displacement operating variables. These are 1) the shot size, 2) the cushion, 3) the screw or ram speed.

The shot size controls the quantity of molten material injected into the cavity. A short shot size causes an insufficient quantity of molten material to fill in the cavity. It's is the cause of the deviations, such as, short shot, surface ripples, pit marks, splay marks, sink marks, and voids. An over shot size causes the molten material over-charge in the cavity. It results the flashing during the process.

The cushion also influences the amount of the molten material charge into the cavity. As well as, it provides an additional material to fill into the cavity during the packing stage. Several surface deviations, such as, short shots, surface ripples, pit marks, splay marks, sink marks, and voids are caused by the over-quantity of cushion. Furthermore, an over-quantity cushion causes the part over-shrinkage. The resultant is an inaccurate dimensional product.

The screw speed influences the material plasticating . Too a rapid screw speed results in incompleted melting during the plasticating stage. This is the causes of deviations, such as short shot, surface ripples, pit marks, splay marks, voids, and delamination. However, too slow screw speed results in the melt overheat. This causes a low viscosity flow of melt during the injection stage. This is the effects of deviations, such as flashing, warpage, and distortion. In addition, too slow screw speed increases the cycle time and reduces the productivity.

2.4.2 Material Variables

Most of the material variables are the parameter which can not be changed during the processing. These variables shall be considered prior the processing. According to their characteristics, these variables can be classified into several groups. These are,

- the physical properties,
- the mechanical properties,
- the electrical properties,
- the optical properties,

- the chemical properties and,
- the material operating variables.

2.4.2.1 Physical properties. The physical properties influence the material performance during the injection molding process, but can not be adjusted from the machine. According to the characteristic of these physical properties, these can be classified into two categories. These are the influence physical properties and the inherent physical properties. The influence physical properties are the parameters which vary with the change of the operating variables and cannot be altered with adjustment to the machine. These includes shear rate, shear stress, and viscosity. The other category, the inherent physical properties, is the parameters which are the constants of the material properties. These include thermal conductivity, thermal diffusivity, specific volume, and specific heat. Complete listings of these physical properties is presented in Table 2.3.

Shear rate is the overall velocity over the cross section of a channel with which molten polymer layers are gliding along each other or along the wall in laminar flow. During the injection molding process, the shear rate controls the molten polymer velocity in the cavity. It influences the flow rate of molten polymer in the cavity. Shear rate is as a function of temperature, pressure, and viscosity.

During the injection molding process, the shear stress will develop in the molten polymer where the layers in a cross section are gliding along each other or along the wall of the mold cavity. An uneven developed shear stress of molten polymer causes the part an uneven shrinkage. It results in warpage and distortion of the products.

Viscosity is a function of temperature, pressure, or shear rate. It is a property of resistance of flow exhibited within the body of a molten polymer. It is expressed in terms of the relationship between applied shearing stress and resulting rate of strain in shear. For the injection molding process, viscosity controls the velocity of and the flow rate of molten polymer.

Specific heat is a thermal physical property. It is a quantity of heat required to raise the temperature of a unit mass of substances 1 unit degree under specific condition. the major influence of specific heat is that it controls the amount of heat required to transfer for melting and solidifying the material.

Specific volume is a property that the volume of the molten polymer divided by that of a water at a standard temperature. Since the molten polymer is a compressible flow, its volume variety will influence the other properties of the melting polymer during the packing stage, and the cooling stage.

Thermal conductivity is another important thermal physical property. It is a property that the quantity of heat conducted per unit time through unit area of a slab of unit thickness having unit temperature difference between its faces. Thermal conductivity influences the time required to melt and solidify the polymer.

Thermal diffusivity as thermal conductivity, controls the time required to melt and solidify the material. It is defined as the quantity of heat diffusion per unit time through unit area of a slab of unit thickness having unit temperature different between material face.

Although these physical properties are not controllable, they are still influence the molten polymer performance during the injection molding process. Therefore, in this program, these properties will be assigned to a form of the corrective weighting factors with parameters such as other physical properties, operating variables, and specific deviation. These physical properties corrective weighting factors includes,

- the influencing physical properties and inherent physical properties,
- the physical properties and the operating variables and,
- the physical properties and the deviation.

These corrective weighting factors indicate the degree of influence for each other. According to these weighting factors, the expert system employs the decision algorithm in the inference engine to fire the rule searching strategy, then, searches the best resolution from the knowledge base to resolve the problem. A detailed definition of these corrective weighting factors is discussed in Chapter 5.

2.4.3.2 Mechanical Properties. The product's final performance primarily depends on the mechanical properties. However, during the injection molding process, these properties will not influence the performance of the molten polymer. These properties include,

- tensile strength and modulus,
- elongation,
- compressive strength and modulus,
- flexural strength and modulus,
- impact strength,
- hardness,
- tensile and flexural creep and,
- tensile fatigue and flexural fatigue.

Tensile strength is a property that allows the maximum force or load, that in pulling direction, to be applied into per unit cross-sectional area, within the gage length of the test specimen. Tensile modulus is the ratio of normal stress correspond for tensile stress less than the proportional stretching.

Compressive strength is a property that indicates the ability of a material to resist a maximum force or load that tends to crush or buckle the material. Compressive modulus is the ratio of normal stress to correspond to compressive stress under the proportional limit stretching.

Flexural strength is the unit resistance to the maximum load before the material fails by bending. It can be used to indicate that the maximum stress can be borne by a surface fibers in a beam in bending. The ratio, within the elastic limit, of the applied stress on a flexural test specimen to the corresponding strain in the outermost fiber of the test specimen is called flexure modulus.

The ability to withstand shock loading is called impact strength. There are several methods to indicate this strength. These are Izod impact test, reverse impact test, and Charpy impact test. Izod impact test is that a notched specimen bar is held at one end and broken by striking, and energy absorbed is measured. Sheet of material is struck by a pendulum or falling object, and reverse side is inspected for damage. This test is called reverse impact test. Charpy impact test is a centrally notched specimen bar to be held at both ends and to be broken by striking the back face in the same plane as the notch.

Hardness is a resistance to surface indentation. It is usually measured by the depth of penetration of a blunt point under a given load suing a particular instrument according to prescribe procedure. There are five different kinds of hardness. These are Barcol hardness, Knoop hardness, Mohs hardness, Rockwell hardness, and Shore hardness.

Barcol hardness is obtained by measuring the resistance to penetration of a sharp steel point under a spring load. Knoop hardness is measured by calibrated machines that force a rhombi-shape, pyramidal diamond indenter having specified edge angles under specified condition into the surface of test material. Mohs hardness indicates the resistance of scratch for a material. Rockwell hardness is a value derived from the increase in depth of an impression as the load on a indenter is increased from a fixed minimum value to a higher value and then return to the minimum value. Measurement of resistance of material to indicate by a spring loaded indenter is called Shore hardness.

Tensile creep is the amount of change in dimension of a specimen under a given tensile load and a specified temperature over a period of time. This does not include the initial instantaneous elastic deformation. Similarly to tensile creep, flexure creep is the amount of change in dimension of a material under a given temperature and a flexure load over a period of time.

The failure or decay of mechanical properties after repeated application of tensile stress is called tensile fatigue. If a flexure stress is applied to this test, then it is called flexure fatigue.

These mechanical properties are independent of the process. It means that during the injection process these properties will not influence to the quality of the products. However, these properties indicate the product can be used in what kind of the condition. For instant, the tensile strength indicates the maximum tensile force can be applied to the product, with no part failure.

2.4.3.3 Electrical Properties. Since polymer are good dielectric materials, they are usually used as insulator in the electrical industrial. To ensure their dielectric performance, these electrical properties employ important factor. For instance, dielectric strength indicate the maximum current can be applied into polymer, and polymer still can perform as standard. Nevertheless, these properties are available in many data sources. Therefore, for designing an electrical using part, these properties can be confirmed prior to the process, and ensure that the dielectric properties perform as expected.

The important electrical properties include arc resistance, resistance, dielectric strength, dielectric constant, volume resistivity, conductivity, and dissipation factor. Arc resistance is the ability to withstand exposure to an electric voltage. It is measured from total time in seconds that an intermitted arc may employ across material surface without rendering the surface conductivity.

Resistance indicates the ability to resist passage of the electric current through the material bulk or on the material surface. Dielectric strength is a property indicating the ability of an insulted material to withstand electric stress. The measurement of dielectric is the average potential per unit thickness at which failure of the dielectric material occur. The ratio of the capacitance of an assembly of two electrodes separated solely by a plastics insulating material to its capacitance when the electrodes are separated by air is called dielectric constant.

Volume resistance is the ratio of the direct voltage applied to two electrodes in contact with or embedded in a test specimen to that portion of the current between them that is distributed through the volume of the test specimen. Electric conductivity is a reciprocal of volume resistance. Dissipation factor is the ratio of the power loss in a dielectric material to total power transmitted through it.

2.4.3.4 Optical Properties. Optical properties may not seem to be very important properties for the part performance. However, when it applies into the optical using, these properties then become a key influence to the part performance. The most important influence of the optical properties is that what is the rate of light can be passed through the material. For instant, the plastic glass lens, it require the light can be passed though as much as possible, therefore, the refractive index is to require as small as possible, and the luminous transmittance is to require as large as possible.

The important optical properties include refractive index, reflective index, gloss, luminous transmittance, haze, yellowness, yellowness index, and color. Refractive index is the ratio of the speed of light in vacuum to the speed of light in the material. A lower refractive index material tend to reduce surface or back reflection. In contrast, a higher refractive index material tend to correspond to higher part brilliance.

Reflective index is the ratio of the amount of light reflect to the total light pass through at a specified surface. Gloss is defined as reflectance of a surface responsible for its lustrous appearance. It indicates the amount of light reflect at the material surface. The reflective amount is proportional to the value of gloss.

Luminous transmittance is the ratio of the intensity of light passing through to that of light incident on the material surface. Haze is defined as the ratio of diffuse luminous transmittance to the total luminous transmittance. It indicates the amount of light that can be transmitted at a specified surface.

Yellowness is defined as deviation in chrome from whiteness or water whiteness in the dominant wavelength range of 570 to 580 nm. Yellowness index is the ratio of the tristimulus value to the primary standard of magnesium oxide. A negative change of yellowness indicates the material tends to blueness. In contrast, a positive change of yellowness indicates the material tends to yellowness.

Since the polymer are often colored in the bulk through the use of appropriate colorants, the color evaluation become an important optical property. The insight of color depends on light spectral temperature, gloss, hue, and background. To evaluate color, there is a standard evaluation method ASTM D 1729 that is developed by American Society for Testing and Materials [Charrier, 1990]. This method is based on using a comparison between the evaluated color sample against the standard color sample in a controlled environment.

2.4.3.5 Chemical Properties. As shown in Table 2.3, the important chemical properties include solubility, flammability, permeability, and chemical resistance. These properties are all aspects of the chemical behavior of the polymer.

Solubility is the ability of solvent action that the polymer in a various solvents and tendency for a solvent to diffuse into and swell the given polymer. It is a very important consideration of many applications. For instant, plasticization of polymer is a very important aspect of solubility. A plasticizer is a chemical added to a polymer. It will influence the polymer processing characteristic and change the polymer mechanical properties. A plasticizer in general lower the temperature resistance of the polymer. It influence the plasticating process of the polymer during injection molding process. Furthermore, a plasticizer added will decrease the mechanical properties such as hardness, tensile strength, and stiffness.

Flammability is the resistance of material to burn. Due to the polymer being widely employed into variety temperature application, the flammability required to be considered cautiously. Flammability in general depends on several factors such as section thickness, ignition ease, thermal conductivity, smoke emission, and toxic emission.

Permeability is a property that allows the passage or diffusion of a gas, vapor, liquid, or solid through a barrier of material without affecting physically or chemically of material. The material barrier property is the primary parameter for affecting the permeability.

The chemical resistance is the ability to resist the attack by chemical, environment, or radiation. It is a dependent on the parameters such as chemical structure, exposure time, exposure temperature, internal stress, external stress, concentration, morphology, crystallinity, molecular weight, part fabrication condition, and part design. An appropriate chemical resistance selection results in the effects such as cracking, warping, swelling, embrittling, etching, dissolving, discoloring, or mechanical properties loss of the products.

2.4.3.6 Material Operating Variables. The material operating variables include the regrind rate, and the moisture content. These variables can be adjusted during the process. For the expert system, these variables are also considered as operating variables.

The regrind rate influences the physical properties and the mechanical properties. For the influence of the physical properties, the regrind rate affects the heat transfer characteristics, and the flow characteristics of viscosity. Therefore, an inappropriate regrind rate results in several surface deviations, such as short shots, surface ripples, pit marks, splay marks, sink marks, and voids. For the influence on the mechanical properties, the regrind rate affects the product performance, and will cause failure of the product in performance.

2.4.3 Mold Design Variables

The mold design variables are the pre-process variables which are considered prior to the start up of the process. Since it is difficult to modify these variables during the operation of the process, it is important to ensure their correctness. Therefore, for the expert system for injection molding, these variables always are the last consideration to resolve the experienced deviations.

Fortunately, today, there exists several computer-aided-design packages, such as, MOLDFLOW[1], C-FLOW[2], CADMOULD[3], and TMconcept[4]. These packages can help the mold designer to determine the optimum mold design variables. A detailed discussion of these variables is presented in the next section.

A complete listing of the mold design variables is presented in Table 2.3. According to their characteristics, these variables can be classified as, dimensions of mold components, and locations of mold design variables.

2.4.3.1 Dimensions of Mold Components. The dimensions of the mold design variables include, 1) the part dimensions, 2) the part tolerances, 3) the size of the cooling channels, 4) the size of the venting channels, 5) the size of the gates, and 6) the size of the runners.

The part dimensions and part tolerances control the outside and the inside configuration of the products. An inaccurate parts dimension results in an off spec. An inaccurate parts tolerance results in failure of the parts to assemble with other parts.
The sizes of the cooling channel influence the cooling rate during the cooling stage and the mold surface temperatures. Inappropriate cooling channel sizes cause insufficient cooling rates. The part cannot be completely solidified, and cannot be ejected from the mold cavity. In addition, the mold temperature depends on the size of the cooling channels. A too small cooling channel results in a too high mold temperature, and causes the material incompleted melting. In addition, it causes an un-even cooling rate. It is the causes of the deviations, such as, flashing, warpage, and distortion. In contrast, a too large cooling channel results in too low mold temperature, and causes a low viscosity melt in the cavity. This causes the material difficult to fill in the cavity. This is the effects of deviations, such

---

[1]MOLDFLOW is the registered trademark of MOLDFLOW PLT., Australia.
[2]C-FLOW is the registered trademark of Advanced CAE Technology, Inc., Ithaca, New York.
[3]CADMOULD is the registered trademark of Institute of Kunststoffe, Germany.
[4]TMconcept is the registered trademark of Plastics & Computer, Inc., Montclair, New Jersey.

as, short shots, surface ripples, pit marks, splay marks, sink marks, voids, and delamination.

The venting channel allows the air to escape from the mold cavity. Too a small size of venting channel, results in air becoming entrapped in the mold cavity. This is compressed air heats up, and causes high surface temperatures of the melt. This results in burned marks on the parts. Furthermore, the air generates a pressure to resist the melt flow into the cavity. This causes several surface deviations, for instance, short shots, surface ripples, pit marks, splay marks, sink marks, voids, and high internal stresses. Too large a size of venting channel, results in melt escaping from the venting channel. This affects the quantity of melt required to fill the cavity.

The gate size determines the melt solidification time. If the gate size is too small, then the melt solidifies at the gate too rapidly. The result is that additional packing of the melt during the packing stage and cooling stage is not possible. This causes deviations, such as short shots, surface ripples, pit marks, sink marks, splay marks, voids, and delamination. Furthermore, a high internal stress occurs near the gate, which is due to the early freezing of melt in the gate. An oversize gate results in a slower freeze of melt in the gate, and thereby increases the cycle time. The productivity plummets.

The runner system must be dimensioned as small as possible to reduced the amount of regrind and the cycle time. The molding operating conditions usually depend on the amount of the regrind. In addition, the regrind molding material performance is uncertain. Furthermore, the amount of regrind has a negative influence on the mechanical properties of the part. Therefore, to ensure high part performance, the amount of regrind must be minimized. Reducing the cycle time increases the productivity. However, a too small size runner system causes large pressure drops in the runner system. This results in a high injection pressure, and in difficulty in filling in the cavity.

2.4.3.2 Locations of the Mold Components. A key component for mold design is the locations of the mold components. These include the layout of the cooling system, the venting system, the runner system, and the gates.

The layout of the cooling system influences the heat removes from the melt. Inappropriate layout causes unbalance cooling of the melt. This generates unbalance thermal stresses in the parts. It is the major causes of warpage and distortion.

Unsuitable layout of the venting system results in trapped air which cannot be removed from the cavity. This causes high pressures to be generated in the cavity. The high pressure generation results in high temperatures. This not only causes burn marks in the part surface, but also creates high internal stresses in the parts.

The major concern for the layout of the runner system is to balance the runner system. Unbalanced runner systems causes uneven filling of the melt into the cavities. Furthermore, inappropriate layout of the runner system results in excessive pressure drops in the runner system. It is therefore difficult to fill in the cavity.

The gate location affects the weld line locations. The welding line is usually the weakest area of the part. Inappropriate weld line locations results in a product which is easy to break or crack. Also, the gate location is usually a stress concentration area. It causes an uneven shear stress distribution. It is one major cause of warpage and distortion.

Table 2.1 Comparison of the advantages and the disadvantages of injection molding.

Advantages	**Disadvantages**
1. High production rates. 2. Volume production. 3. Low labor cost per unit. 4. Parts require little or no finishing. 5. Can be used to manufacture complicated geometry parts. 6. Can be used to manufacture small parts. 7. Easy to decorate. 8. Easy to color. 9. Used material can be reground. 10. Close dimensional tolerances can be maintained. 11. Easy to change molded material. 12. Can be used to manufacture in a combination of different kinds of plastics material. 13. Can be used to manufacture with metallic or nonmetallic inserts. 14. The natural physical properties of the plastics give many advantages such as, high strength-weight rates, corrosion resistance, and clarity.	1. Low profit margins per unit. 2. Continuous operation is often necessary to compete. 3. Machine costs are high. 4. Mold costs are high. 5. Process control may be poor. 6. Quality control is a difficulty. 7. Manpower training is a difficulty. 8. Causes of deviation are unclearly. 9. Lack of knowledge about the long term physical properties of the plastics may result in long term failures.

Table 2.2 Comparison of the toggle clamping unit and the fully hydraulic clamping unit.

Fully Hydraulic Clamping Unit	**Toggle Clamping Unit**
1. Original cost higher.	1. Original cost lower.
2. Maintenance cost low.	2. Maintenance cost higher.
3. Running horsepower higher.	3. Running horsepower lower.
4. Non positive clamp.	4. Positive clamp.
5. Unlimited stroke.	5. Limited stroke.
6. Clamping force readout direct.	6. Clamping force readout not direct.
7. Easy to adjust clamping force.	7. Difficult to adjust clamping force.
8. Mold unit set up easy.	8. Mold unit set up more complicated.
9. Speed control easier.	9. Speed control more difficult.

Table 2.3 Processing variables of the injection molding processing

Operating Variables	Temperatures	Barrel Temperature Mold Temperature Nozzle Temperature
	Pressures	Injection Pressure Holding Pressure Clamping Force
	Times	Cycle Time Cooling Time Decompression
	Displacements	Shot Size Cushion Screw or Ram Speed
Material Variables	Physical Properties	Shear Rate Shear Stress Thermal Conductivity Thermal Diffusivity Specific Volume Specific Heat Viscosity
	Mechanical Properties	Tensile Strength Tensile Modulus Elongation Compressive Strength Compressive Modulus Flexural Strength Flexural Modulus Impact Strength Hardness Tensile Creep Flexural Creep Tensile Fatigue Flexural Fatigue
	Electrical Properties	Arc Resistance Resistance Dielectric Strength Dielectric Constant Volume Resistivity Conductivity Dissipation Factor

Table 2.3 (continued) Processing variables of the injection molding processing.

Material Variables (continued)	Optical Properties	Refractive Index Reflective Index Gloss Luminous Transmittance Haze Yellowness Yellowness Index Color
	Chemical Properties	Solubility Flammability Permeability Chemical Resistance
	Material Operating Variables	Regrind Rate Material Moisture
Mold Design Variables	Dimension	Part Dimensions Part Tolerances Size of Cooling Channel Size of Venting Channel Size of Gates Size of Runner System
	Location	Layout of Runner System Layout of Cooling System Layout of Venting System Location of Gates Number of Gates

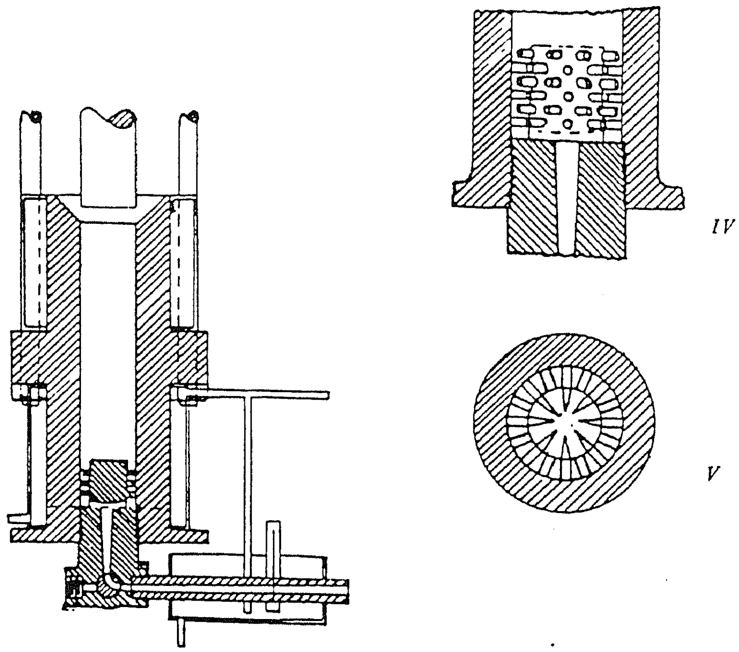


Figure 2.1 The original drawing for the Hyatt injection molding machine. In view V and IV is to illustrated the heat transfer of material. Reproduced from J.H. DuBois, *Plastics History, U.S.A.*, Caners Books, Boston, 1972. p.216.

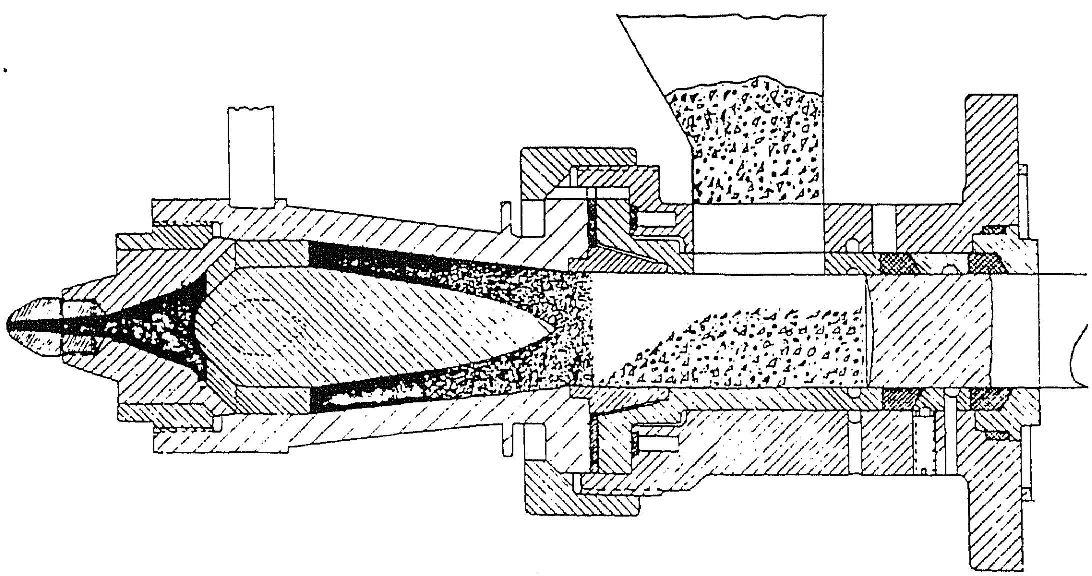


Figure 2.2 The invention of Gastrow torpedo injection molding machine by H. Gastrow in 1932. Reproduced from J.H. DuBois, *Plastics History, U.S.A.*, 1st edition, Caners Books, Boston, 1972. p.222.

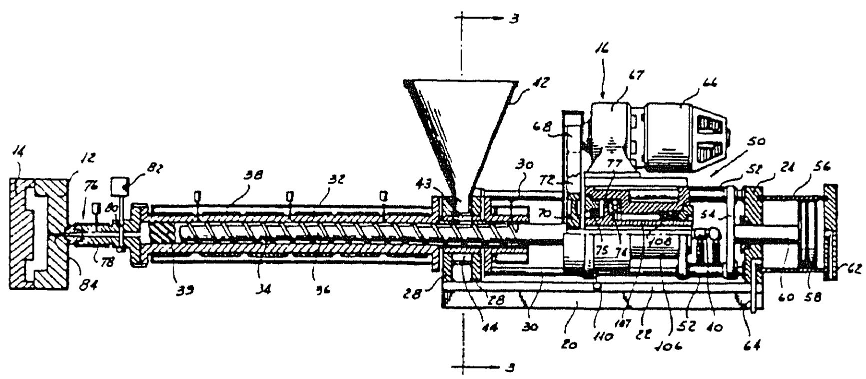


Figure 2.3 The invention of in-line reciprocating screw plasticating injection unit by W.H. Willert in 1957, Reproduced from J.H. DuBois, *Plastics History, U.S.A.*, Caners Books, Boston, 1972. p.230.

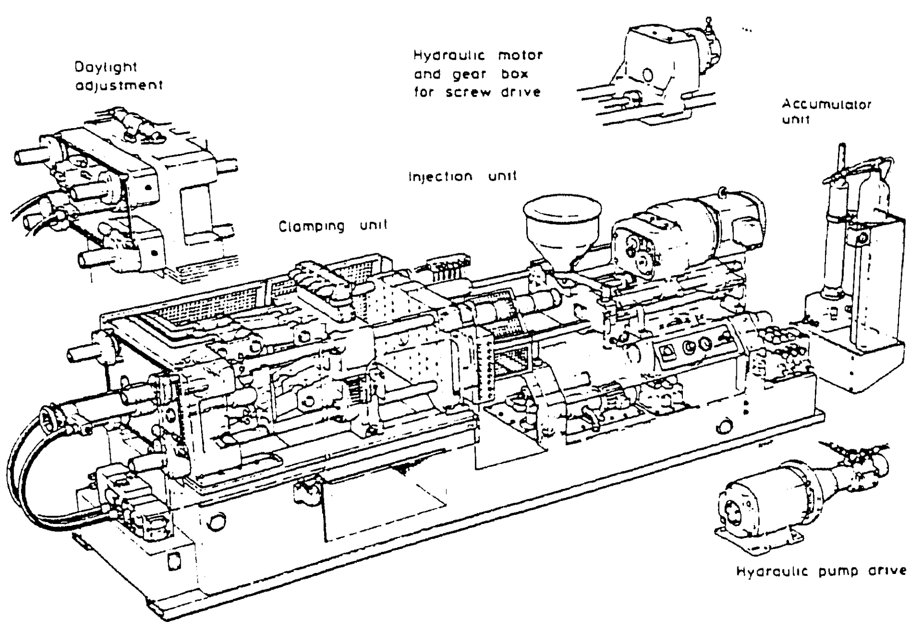


Figure 2.4 A typical injection molding machine. Reproduced from F. Johannaber, Injection Molding Machines, Hanser Publish, New York, 1985, p.14.

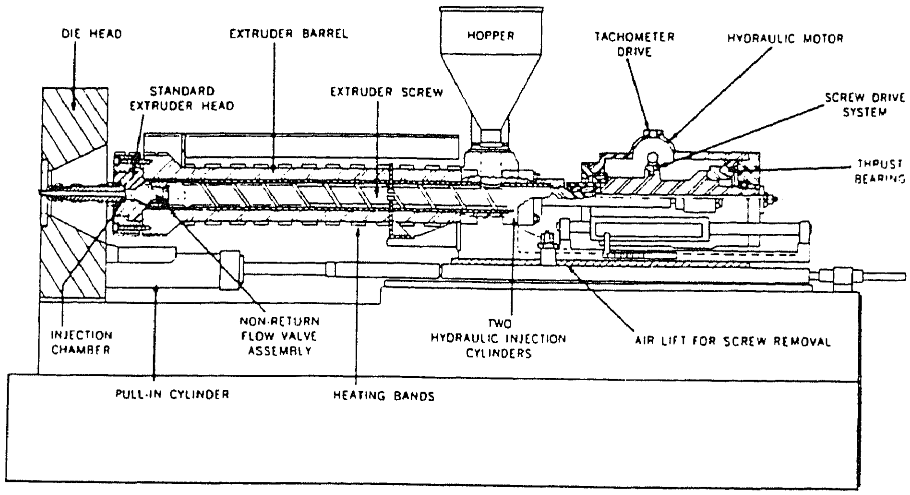


Figure 2.5 A typical in-line reciprocating screw plasticating unit. Reproduced from I.I. Rubin, *Injection Molding, Theory and Practice*, John Wiley & Sons, Inc., New York, 1972. p.5.

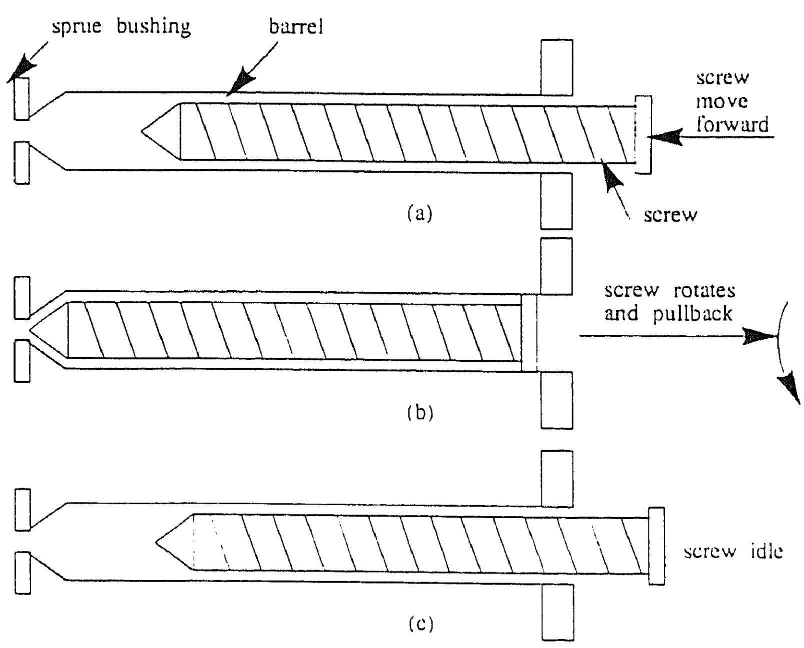


Figure 2.6 A operation sequence of in-line reciprocating screw plasticating unit. (a) injection, screw moves forward, (b) shot preparation, screw rotates and pullback. (c) soak or idle, screw is not moving.

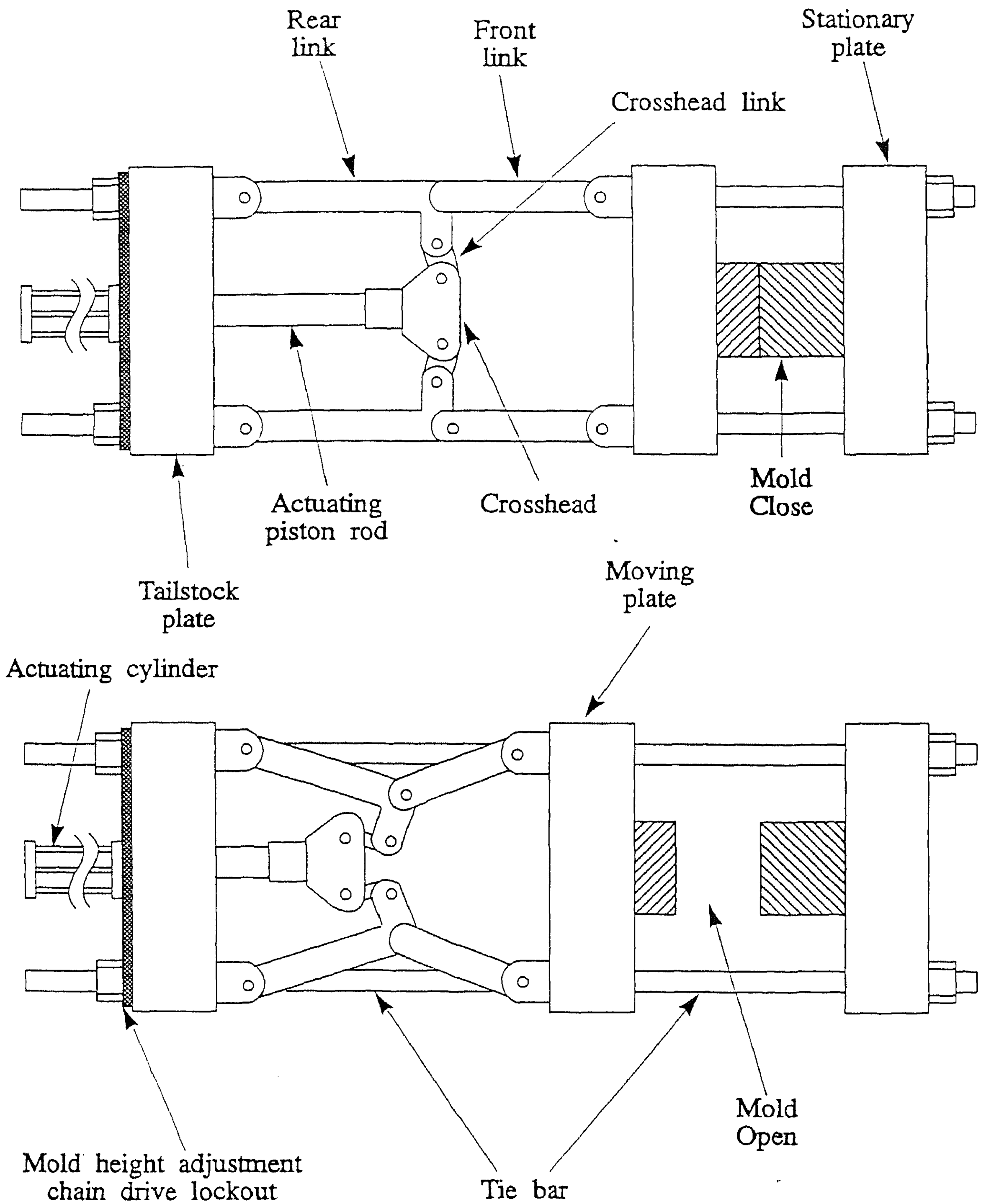


Figure 2.7 Drawing of a typical double toggle acting clamping unit.

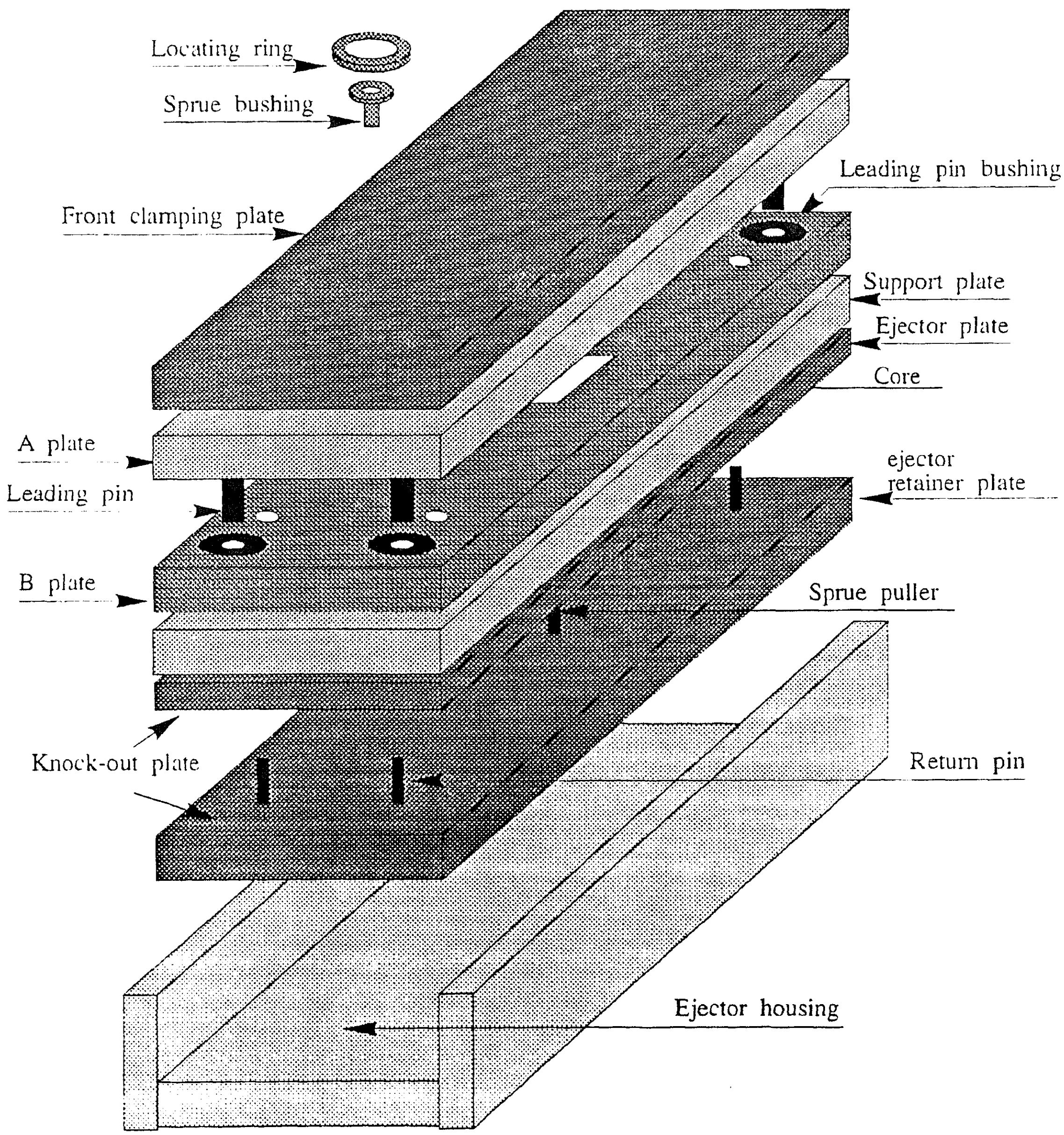


Figure 2.8 Exploded view of standard mold base.

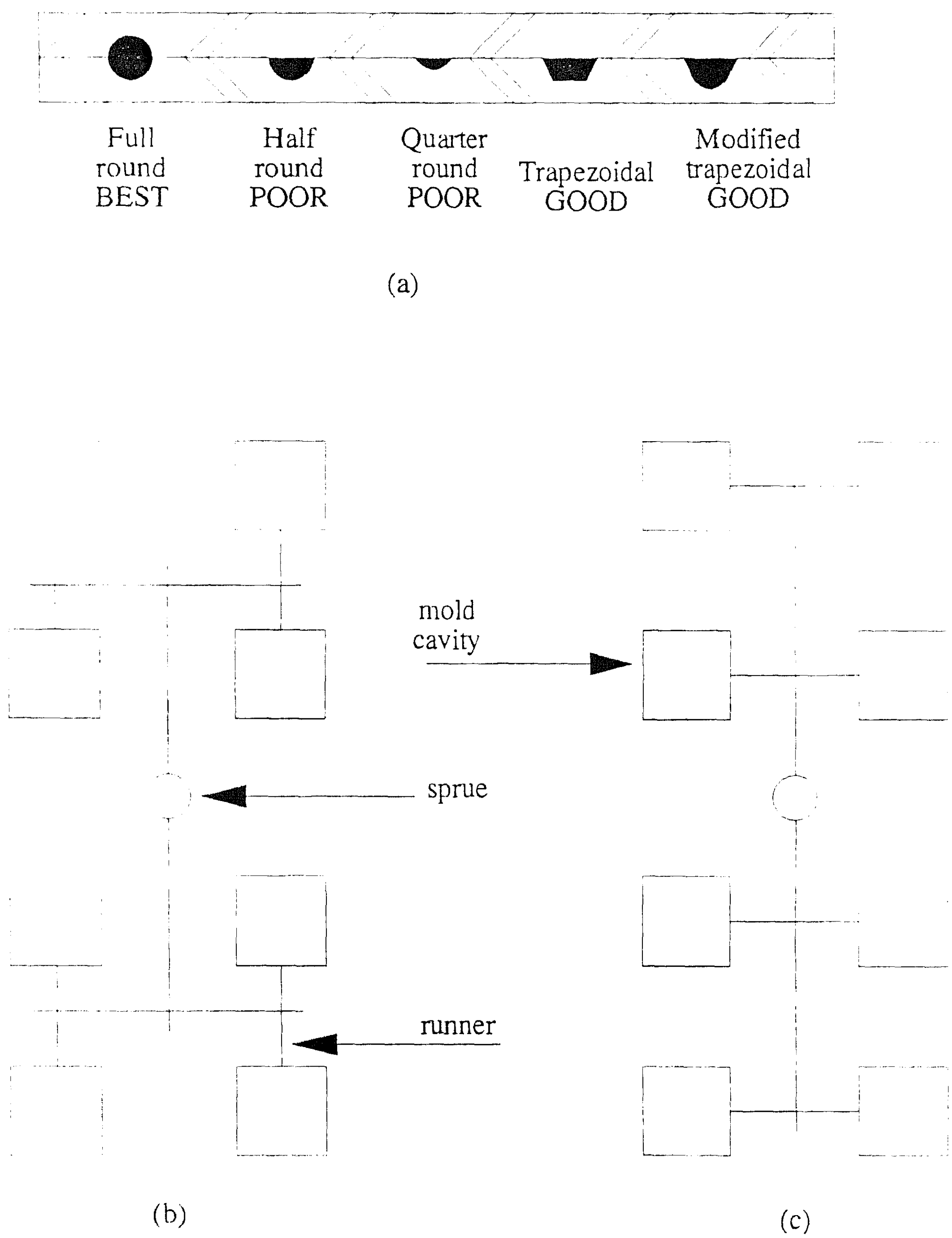


Figure 2.9 Example of different shapes of runner and the runner balance layout. (a) different shapes of runner, (b) the balanced "H" runner layout, (c) the unbalanced "H" runner layout.

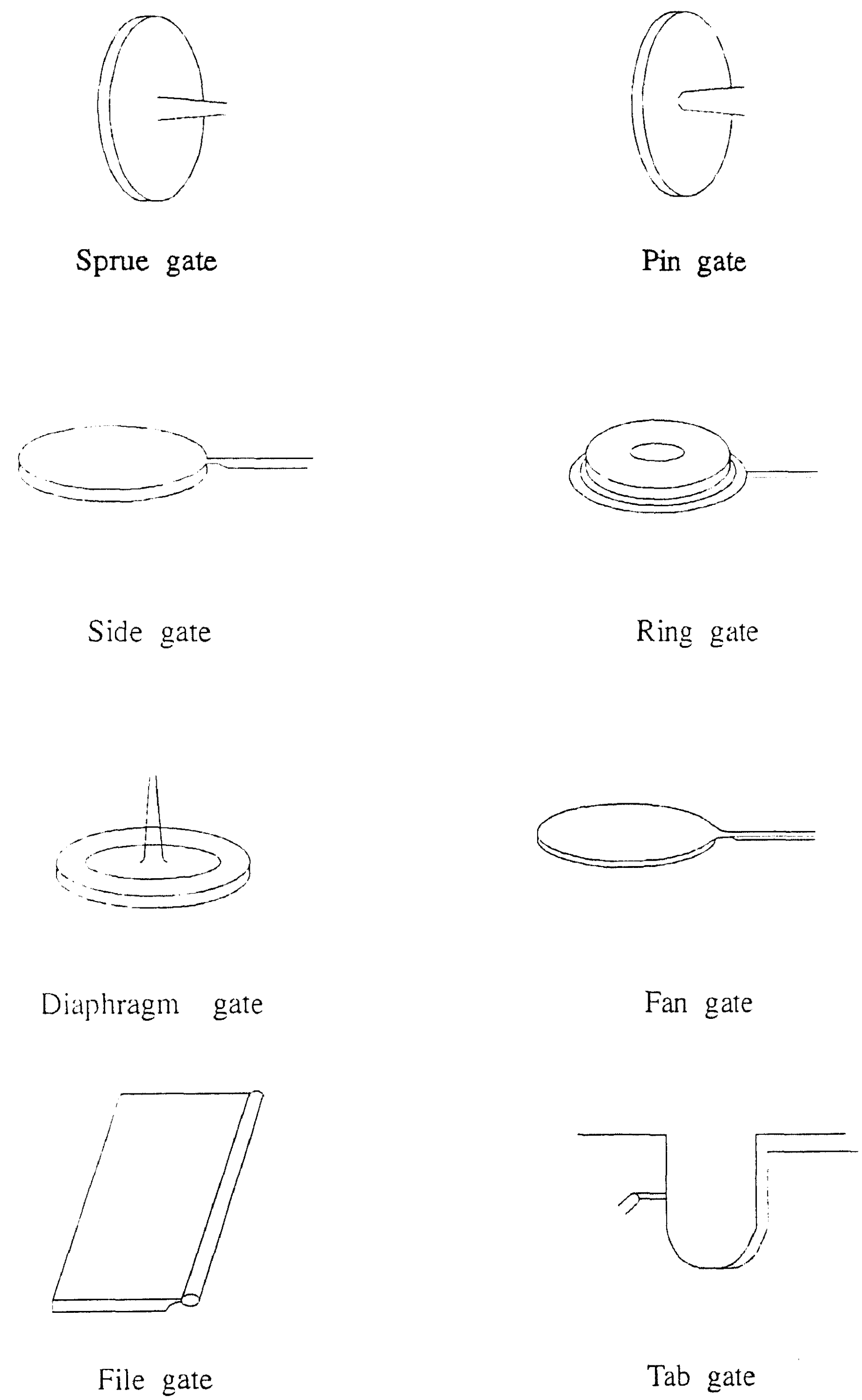


Figure 2.10 Different types of gates.

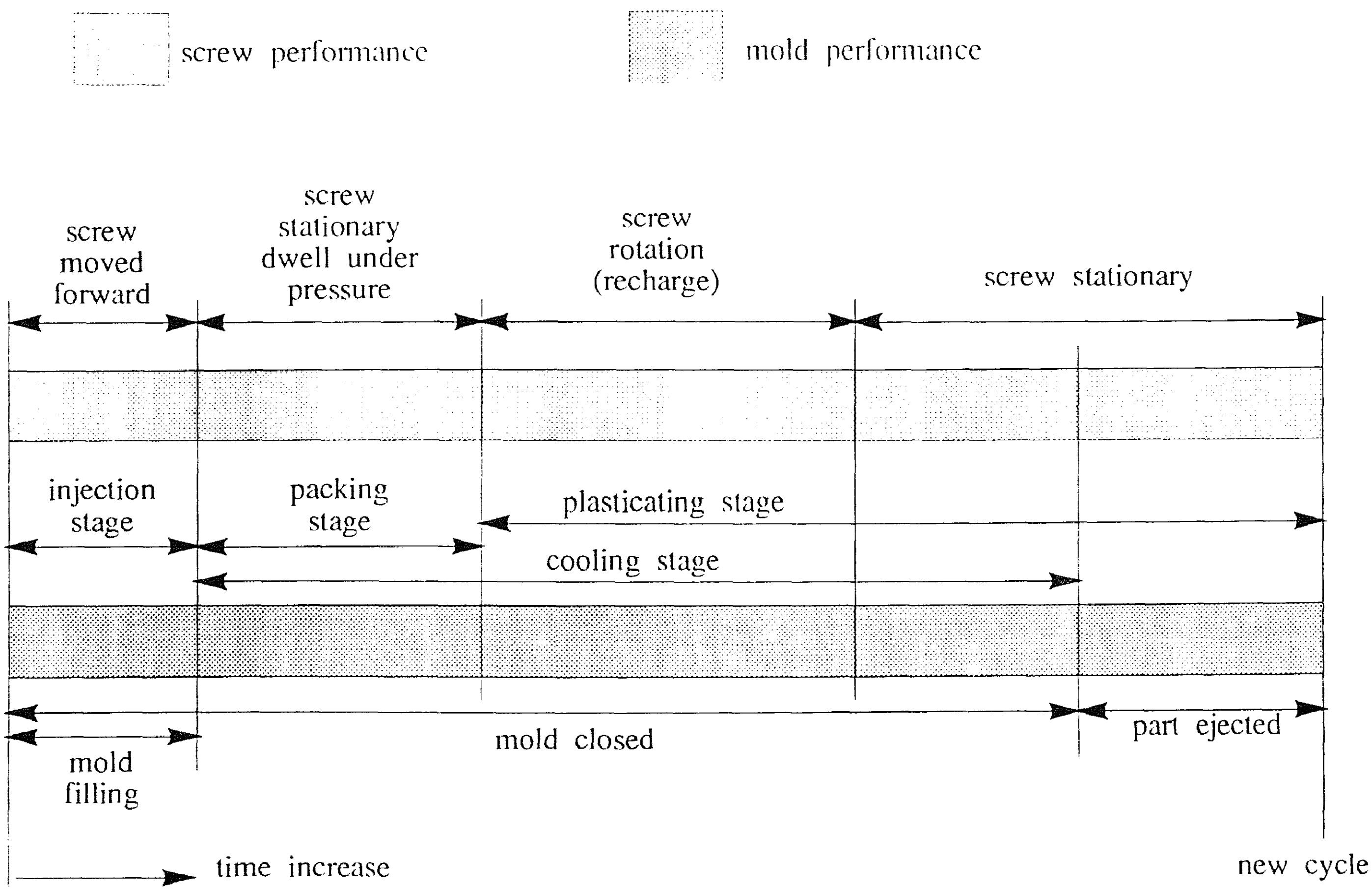


Figure 2.11 Schematic representation of cycle time of a reciprocating screw injection molding process.

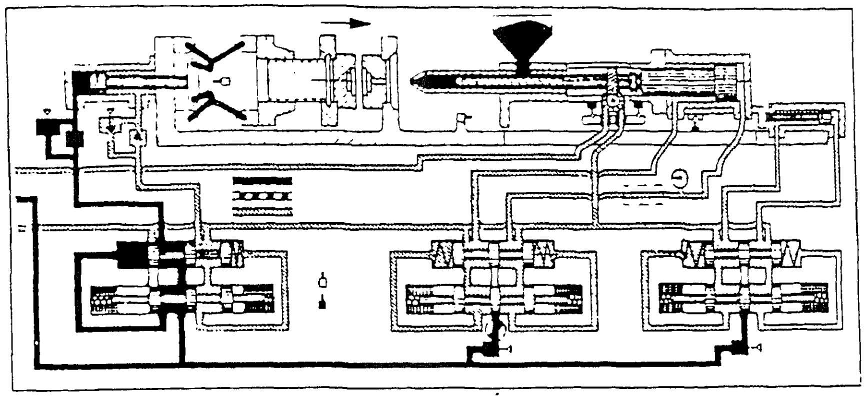


Figure 2.12 The injection cycle start and mold closing by the clamping unit.

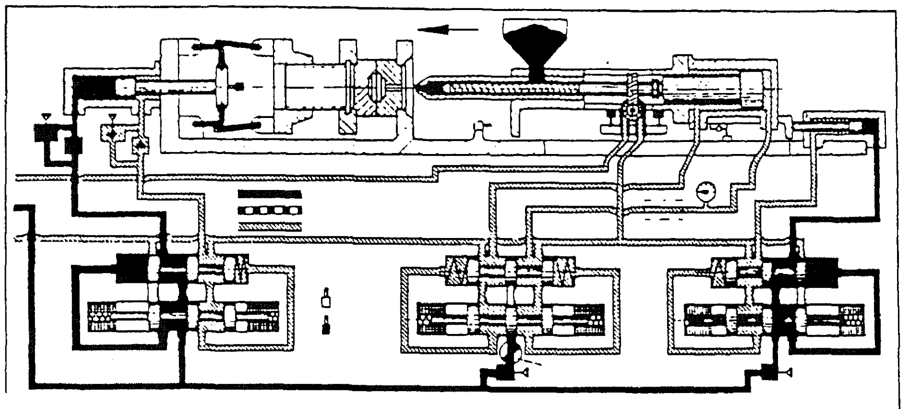


Figure 2.13 The plasticating unit moves forward to the sprue bushing.

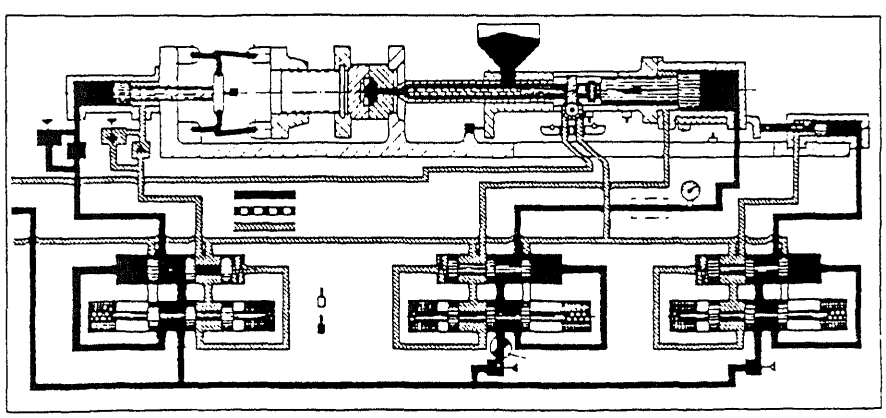


Figure 2.14 The material begins to injection into the mold cavities

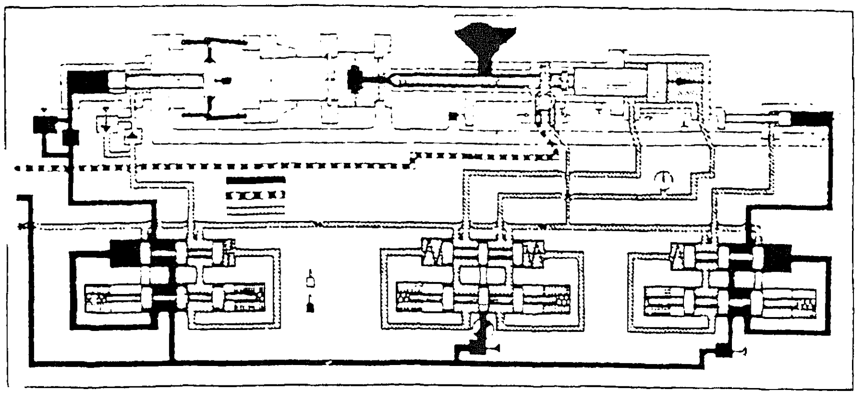


Figure 2.15 The mold cavities are filled; the cooling stage and the packing stage start; the screw starts to rotate.

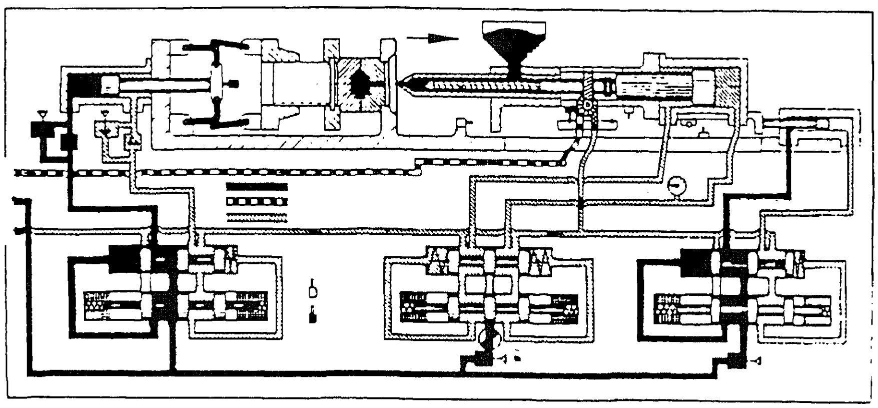


Figure 2.16 The screw chamber is still charging; the plasticating unit starts to move away from the sprue bushing.

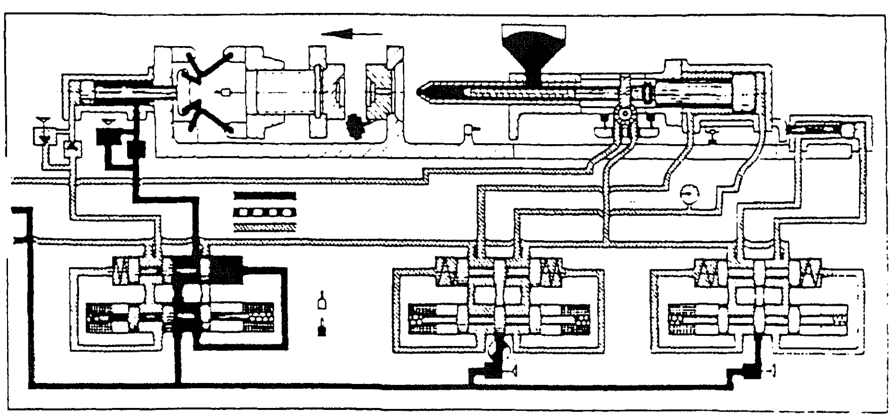


Figure 2.17 The part is ejected. Figure 2.12 to 2.17 are Reproduced from J.H. DuBois and F.W. John, Van Nostrand Reinhold, New York, 1986, pp.118-119.

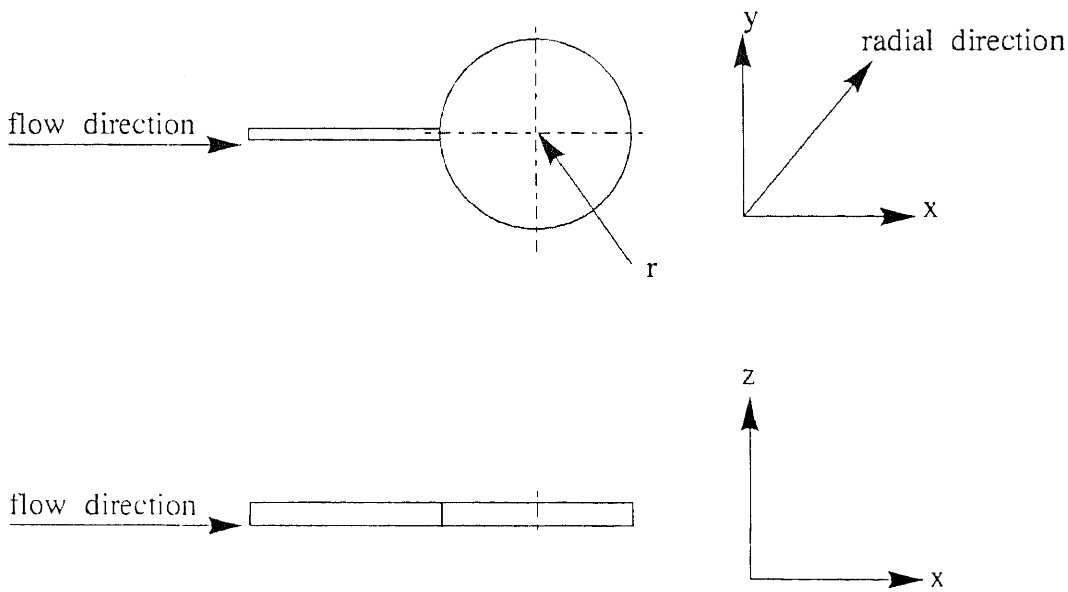


Figure 2.18 Disk-shaped cavity.

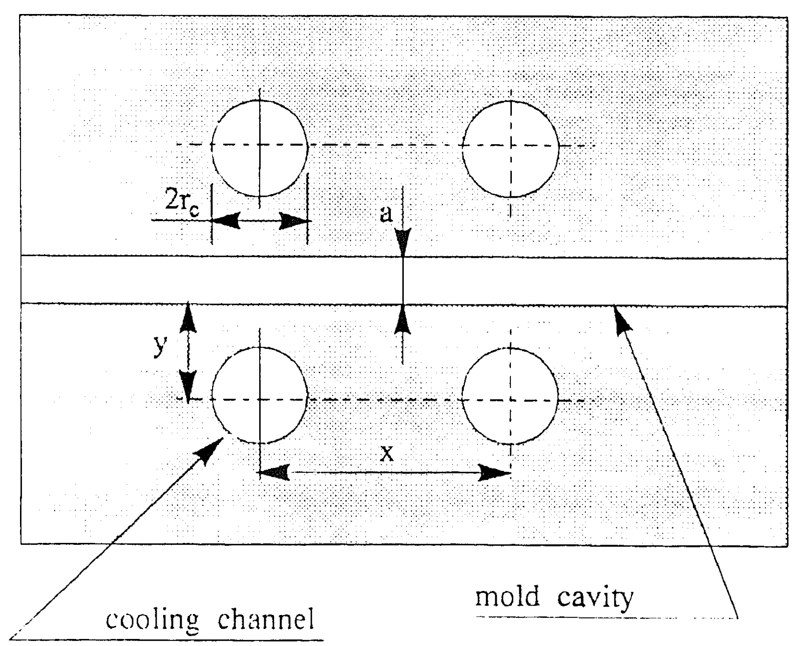


Figure 2.19 Design of circular cooling channel.

CHAPTER THREE
DEVIATION ENCOUNTERED IN THE INJECTION MOLDING PROCESS

3.1 Introduction

To ensure a good quality product is the principal objective for any manufacturing process. It is also true for the injection molding process. However, for the injection molding process, it is somewhat more difficult to achieve. This is due to the complexity of the injection molding process, and the unstable properties of plastics.

The injection molding process, as addressed in chapter two, is a process that melts the material from solid phase to molten phase, and injects the molten material into the cavity, then solidifies the molten phase to solid phase. To accomplish the whole process requires several fields of science. To change the material phase requires heat transfer, the friction, and polymer science. To convey the material requires fluid dynamics, the friction, rheology, and control theory. Obviously, there are many opportunities for deviations to occur at anytime and anywhere during the injection molding process.

To eliminate or reduce the deviations there exist three key rules. First, it is necessary to understand the entire injection molding operation. Second, it is necessary to know how to recognize the deviations. Moreover, it is necessary to realize the causes of the deviations. Following these key rules, deviations can be eliminated or reduced eventually.

To understand the entire process, it is necessary to trace the causes of the deviation, and to adjust the influencing variables and remedy the deviations. To recognize the deviation, the cause of the deviations must be understood. To remove the cause of the deviation, one must comprehend the influence variables involved in the deviations.

To recognize the deviations is a tremendous challenge. To eliminate or reduce the deviation is even greater a challenge. Some of the deviations can be recognized from a specific indicator. For instance, the cavity pressure transducer can indicate if short shots are caused by insufficient injection pressure or not.

However, there also exists a group of deviations that cannot be indicated by instruments. These deviations are usually only recognized after the process has been completed. For example, for warpage, there are no warpage sensors to detect the warpage during the process. It is only observed afterward. Once the deviation has been recognized, the causes of the deviation can be eliminated to remedy the problem. The detail discussion of the possible causes and the corrective actions are addressed in the section 3.3.

3.2 Classification of Deviations

The common deviations occurring in the injection molding process are shown in Table 3.1. From this table, it can be seen that these deviations are classified into two groups. These are,

- surface deviations and,
- dimensional deviations.

In this program, a rectangular molded plaque is used to illustrated the definition of the deviations. These deviations are illustrated schematically in Figure 3.1 to Figure 3.10.

3.2.1 Surface Deviations

As shown in Table 3.1, surface deviations include surface ripples, pit marks, splay marks, sink marks, voids, and flashing. These deviations occur in the surfaces of the parts. The reason to eliminate or reduce that deviation is not only for a good external appearance, but also for the parts' functional performance.

A good external appearance is an essential factor for a good quality product. For any kind of product, the product becomes worthless due to the bad external appearance. This is especially true for the decorative products.

Furthermore, the surface deviations sometimes cause the product to become functionless. This is especially true for a part that requires close conjunction with other parts. For example, a plastic gear is used to demonstrate this. If flashing exists in gear teeth, then, it is obvious that the gear cannot mesh with other gear teeth. In addition, the region in which the surface deviation occurs is also a region of the internal stress concentration. This causes the product to easily break or crack and the part becomes functionless.

3.2.2 Dimensional Deviations

Common dimensional deviations include short shots, warpage, distortion, and delamination. Dimensional deviations are defects that exist when an unacceptable difference between the desired dimension and the actual dimension of the parts exists.

Accurate part dimensions are key components of a good quality product. Inaccurate part dimensions cause the part to become useless. Unfortunately, the effect of the dimensional deviation is that it causes the inaccurate part dimension. It is one of the reasons to eliminate or reduce the dimensional deviations. Furthermore, some of these deviations cause the structure of part deformed. It results in to reduce the life cycle time of the parts. It also reduces the reliability of functional performance of the part. This is another reason to remove the dimensional deviations.

Some dimensional deviations, such as warpage and distortion, are more difficult to treatment. These deviations are not only hard to analyze, but also, to recognize. Nevertheless, other dimensional deviations, like short shots and delamination, are possibly the most easy to address and to recognize. A detailed discussion of causes of these deviations is presented in the section 3.3.

3.3 Causes and Resolutions of Deviations

In this section, the causes of and the resolutions of specific deviations are addressed in detail. In general, there has three sources to acquire all necessary information and to resolve the deviations. These are,

- from printed material, such as, textbook, journal reports, raw material supplier bulletin, and conference reports and,
- consultation with expert molding operators.

In this program, all of these sources of knowledge are considered to be acquired the necessary information. The detailed acquisition procedures of these knowledge sources are discussed in the chapter 5.

3.3.1 Surface Ripples

As shown in the Figure 3.1, surface ripples are wave like marks that appear on the surface. The surface ripples occasionally occur near a weld line region. The surface ripples usually be usually only recognized after the process has been completed. It requires a human inspection to recognize the existence of the defect. A listing of causes and corrective actions is presented in Table 3.2.

There are two major reasons to remove the surface ripples. First, it is for a good external appearance. Second, it is to ensure part performance. Since the value of the part primarily depends on the external appearance, especially for decorative products, it is essential to meet this specification. Furthermore, the region of the surface ripples causes the higher internal stresses. These higher internal stresses cause the part to break or crack easily. Therefore, to ensure the functional performance of the part, the surface ripples must to be eliminated.

The causes of surface ripples include material shortage, slow flow rates, prior material solidification, foreign material stick in mold surface, and instrument reading failure. For correcting the material shortage, one can increase the shot size and decrease the cushion.

The methods to increase the flow rate are as follows,

1. Increase injection pressure,
2. Increase injection speed or use the booster pressure,
3. Increase mold temperature,
4. Increase barrel temperature,
5. Increase nozzle temperature,
6. Increase gate size,
7. Reduce regrind rate,
8. Reduce screw speed and,
9. Increase injection time.

To resolve material solidification, one may use the following,

1. Increase barrel temperature,
2. Increase nozzle temperature,
3. Increase mold temperature,
4. Increase injection pressure,
5. Increase injection speed,
6. Increase gate size,
7. Increase runner size,
8. Increase venting channel size and,
9. Decrease cooling channel size.
10. Reduce screw speed,
11. Increase injection time and,
12. Increase cycle time.

The foreign material stick in mold surface causes the material flow uneven. It results in ripples marks appears in the parts. The foreign component stick in mold surface is caused by the material contains the foreign component, the screw wears compounds with material, and mold surface wear.

The other possible cause for surface ripples is that the instruments reading failure. These instruments indicate the operating conditions such as temperature, pressure, screw speed, and screw position. An incorrect reading of these instruments, it misleads the molding operators into inaccurate operating condition and causes the deviation. To remedy this, it requires to adjust the accuracy of these instruments.

3.3.2 Pit Marks

Pit marks is also an unacceptable surface defect for a good quality product. A schematic of pit marks is shown in Figure 3.2. Pit marks are very normal and quiet often occurs in molded parts. The pit marks are difficult to be detected by sensors. Usually, human inspection requires to recognize this deviation. In Table 3.3, the causes and corrective actions for remedying pit marks are presented.

As with surface ripples, pit marks cause unacceptable part surfaces and create higher internal stresses. The poor surface quality results in ejected parts. The higher internal stresses result in poor tensile performance of the parts.

The causes of pit marks are similarly to the causes of surfaces ripples. Pit marks are caused by a shortage of material, slow flow rates, premature material solidification, and instruments reading failure. The corrective actions of these causes are the same as those which are listed for remedying surface ripples. These are,

For correcting the material shortage, it includes,

1. Increase shot size and,
2. Decrease cushion.

For increasing the flow rate, these are,

1. Increase injection pressure,
2. Increase injection speed or use the booster pressure,

3. Increase mold temperature,
4. Increase barrel temperature,
5. Increase nozzle temperature,
6. Increase gate size,
7. Reduce regrind rate,
8. Reduce screw speed and,
9. Increase injection time.

For resolving material solidification, it includes,

1. increase barrel temperature,
2. increase nozzle temperature,
3. increase mold temperature,
4. increase injection pressure,
5. Increase injection speed,
6. increase gate size,
7. Increase runner size,
8. Increase venting channel size,
9. Decrease cooling channel size,
10. Reduce screw speed,
11. Increase injection time and,
12. Increase cycle time.

For the failure of instrument reading, these are,

1. Check all temperature reading,
2. Check all pressure reading,
3. Check screw or ram speed reading and,
4. Check screw position reading.

Beside these causes, another cause for pit marks is that the material contains contamination. The contamination cannot usually be melted. It will be injected with the material into the cavity, and cause the pit marks. The source of the contamination is manifold. One is debris from mold and screw wear, and another is foreign matter present in the feedstock.

3.3.3 Splay Marks

Splay marks are small bubbles like marks spout away near the gate section as shown in Figure 3.4. Splay marks like as the other surfaced deviations cause unacceptable surface quality, and the strong internal stresses. The causes and remedy methods for splay marks are listed in Table 3.4. According the characteristic of splay marks, it can be classified into two kinds of the splay marks. One is the moisture splay marks, the other is the over-heat material splay marks.

The moisture splay marks are due to material moisture do not to be removed appropriately as it is filled in the cavity. The un-removed moisture in the material causes an uneven flow velocity of the melt in the cavity. It results in splay marks exist. To removed the moisture in the material, it can make by several ways as shown in Table 3.4. First, remove the moisture before the material is processed. To accomplish that, it can pre-heat the material, or dry the material before the process. Second, remove the moisture during the material is processed. To achieve that, there have two ways. One is that use a vented barrel, the other is that reduce the material temperature, to allow moisture escape from the barrel during the plasticating stage. Furthermore, to slow down the material flow velocity, it allows the moisture escape for venting system during the injection stage. To ensure the moisture to remove from the material, it can be achieved by adjusting the variables as follows,

1. Reduce barrel temperature,
2. Reduce nozzle temperature,
3. Reduce mold open time,
4. Reduce injection speed,

5 Reduce injection time,

6 Increase mold temperature,

7. Reduce screw speed.

The over-heat splay marks are due to the material is over-heated, and causes the low viscosity melt. It results in the melt too free flow in the cavity, and generates the splay marks. The over-heat splay usually can be recognized from the material drooling in the nozzle area. To overcome this kind of deviation, it requires to reduce the material temperature and increase the melt viscosity in the cavity. These can be achieved by correcting the following variables. These are,

1. Reduce barrel temperature,
2. Reduce nozzle temperature,
3. Reduce mold open time,
4. Reduce injection time,
5. Reduce cushion,
6. Use nozzle with small orifice,
7. Increase decompression time,
8. Decrease cooling channel size and,
9. Increase venting channel size.

The instrument reading failure, that misleads the operating conditions, is also a cause for splay marks. The following listing can be used to resolve this cause. These are,

1. Check all temperature reading,
2. Check all pressure reading,
3. Check screw position reading,
4. Check screw speed reading.

3.3.4 Sink Marks

Sink marks are similarly as pit marks. It is very common defects during the injection molding process. Figure 3.5 is a schematic to illustrate the sink marks. The sink marks are the spot which shrinkage inward into the part surface. The sink marks often find in the location which are the rib, the boss, and the different thickness section conjunction. Table 3.5 shows the possible causes and the suggested correction actions for sink marks.

During the injection molding process, sink marks can be indicated from the distribution of cavity pressure as shown in Figure 3.11. As other surface defects, sink marks causes the poor surface quality and the strong internal stresses concentrate in some area. This result is that product becomes priceless and its functional performance fails.

The uneven shrinkage of the part is a primarily cause for sink marks. The shrinkage rate is proportionally to the thickness of the part. In conjunction area of the different thickness, the shrinkage is maybe variously uneven. This uneven shrinkage possibly results in sink marks. To prevent uneven shrinkage occurs, it can be corrected by two ways. One is from the part design. When part requires to vary the thickness, it shall concern that to smooth the thickness variety. The other way is from changing the operating variables as follows,

1. Decrease mold temperature,
2. Increase injection pressure,
3. Using the maximum injection speed or/and boost pressure,
4. Increase injection time,
5. Decrease barrel temperature,
6. Decrease nozzle temperature,
7. Increase gate size,
8. Relocate gate nearer heavy section,
9. Increase cooling channel size and,
10. Modify part design.

Furthermore, the additional packing material shortage is also the other cause for the sink marks. To correct the shortage of packing material, it can be achieved by increasing the shot size and decreasing the cushion. Moreover, the instruments mislead the operating conditions, it is also possibly to cause sink marks. To remedy this, it requires to adjust all instruments individual. These instruments include temperature indicator, pressure indicators, and screw position indicator.

3.3.5 Voids

Figure 3.5 shows a schematic to illustrate voids. Voids are the defects that the vacuum bubble like exist inside the part. The effect of the void is that it causes the stronger internal stresses near the region of void. This results in the part easy to fail near these areas. A cavity pressure transducer possibly can be used to indicate the void incident during the process as shown in Figure 3.11.

There are three possible causes for the voids as shown in Table 3.6. One is the additional packing material shortage. To increase the addition packing material, it can be implemented as, for instance, increase shot size, and decrease cushion. The other major cause is that an over-shrinkage inside the part. When cooling stage starts, the molten material is still injected into the mold cavity. At this time, the material has two phases inside the mold cavity. One is the solid phase, the other is the molten phases. When the hot molten phase encountered the cold solid phase, the shrinkage occurs inside the part. It creates the vacuum spots. To avoid this condition, the resolutions are,

1. Decrease mold temperature,
2. Increase injection pressure,
3. Using the maximum injection speed or/and boost pressure,
4. Increase injection time,
5. Decrease barrel temperature,
6. Decrease nozzle temperature,

7. Increase gate size,
8. Relocate gate nearer heavy section,
9. Increase cooling channel size and,
10. Modify part design.

The other cause is instruments mislead the operating conditions. It results in the operating conditions are indicated inaccuracy amount and cause the deviation. The way to resolve this problem includes,

1. Check all temperature reading,
2. Check all pressure reading and,
3. Check screw position reading.

3.3.6 Flashing

As shown in Figure 3.6, flashing is that the excessive material attached to the edge of parting line. In general, lashing only occurs in which the "A" plate and the "B" plate conjunction place. Flashing can be indicated by installing a cavity pressure transducer to find out flashing. An excessive high cavity pressure distribution indicates that flashing possibly exists in the molded parts as shown in Figure 3.11.

Flashing must be removed for two reasons. First, flashing is an unexpected part. Sometimes, it causes the part functionless. For instance, if flashing exists in gear teeth, it is obviously that this gear cannot mesh with other gear teeth. Even, flashing can be remove some kind of the secondary operating. However, it will increase the cost of part. It is due to the extra labor that requires to be invested in to make part acceptable. Second, the flashing sometimes stick in the mold surface. When mold closes up, this flashing part will damage the mold surface, and, eventually wear down the mold surface.

Table 3.7 shows the possible causes and the suggested correction action of flashing. From Table 3.7, it shows that there are three causes for flashing. These include the material quantity over charged into the mold cavity, the melt viscosity too low in the mold cavity, the mold inappropriately closing, and instrument reading failure.

Inappropriate mold closing, it causes that a slit exists between the stationary plates and the moving plates, and allows the melt flow through it. To re-polish the mold surface is the major resolution for this problem. The other resolution is to increase the clamping force to keep the mold close tightly.

The over-charged material forces the mold slip. It results material flow from the slit of the mold. To resolve over-charge material, it can be accomplished as follows,

1. Decrease shot size and,
2. Increase cushion.

Too a low viscosity melt, it results in the melt too free flow, and generates the high material pressure inside the mold cavity. It results in the material pressure greater than the clamping pressure, and causes the mold slip. To overcome this problem, it can modify the variables as follows,

1. Decrease barrel temperature,
2. Decrease mold temperature,
3. Decrease nozzle temperature,
4. Decrease injection pressure,
5. Decrease injection speed,
6. Decrease injection time,
7. Increase clamping force,
8. Increase venting channel size,
9. Decrease gate size,
10. Increase cooling channel size,
11. Decrease venting channel size,
12. Reduce runner size and,
13. Increase screw speed.

The other possible cause for flashing is that the instruments reading failure. These instruments indicate the operating conditions such as temperature, pressure, screw speed, and screw position. An incorrect reading of these instruments, it misleads the molding operators into inaccurate operating condition and causes flashing. To remedy this, it requires to adjust the accuracy of these instruments.

3.3.7 Short Shots

Short shots are an inverse defect of flashing. It is an unacceptable dimension defect. Short shots cause that the part becomes totally useless. Usually, it causes by the insufficient material fill in the mold cavity. The cavity pressure Transducer can indicate short shots during the process, as shown in Figure 3.11.

Figure 3.7 is a schematic to illustrate the short shots. The causes of the short shots include the insufficient quantity of injection material, the material solidified prior of the cavity fulfill, the material flow rate too slow, the foreign material stick in mold surface, and the instrument reading failure as shown in Table 3.8. The resolutions for insufficient quantity of the melt, are shown as following,

1. Check hopper for material supplier,
2. Increase shot size and,
3. Decrease cushion.

The material solidified prior of the cavity fulfill, it persists material appropriate fill in the mold cavity. It possibly causes by the low flow rate of the injection material, the high material viscosity, and the inappropriate cooling rate. The following resolutions can be used to resolve this problem. These are,

1. Increase barrel temperature,
2. Increase mold temperature,
3. Increase nozzle temperature,
4. Increase injection pressure,

5. Using the maximum injection speed or/and boost pressure,
6. Increase gate size,
7. Reduce regrind rate,
8. Increase venting channel size,
9. Decrease screw speed,
10. Increase injection time
11. Increase cycle time.

For resolving material flow rate too slow, it includes,

1. Increase injection pressure,
2. Use the maximum injection speed or/and boost pressure,
3. Increase gate size,
4. Increase barrel temperature,
5. Increase mold temperature,
6. Increase nozzle temperature,
7. Reduce regrind rate,
8. Increase runner size,
9. Increase venting channel size,
10. Increase cooling channel size,
11. Reduce screw speed and,
12. Increase injection time.

The other possible cause for short shots is that the instruments reading failure. These instruments indicate the operating conditions such as temperature, pressure, screw speed, and screw position. An incorrect reading of these instruments, it misleads the molding operators into inaccurate operating condition and causes the deviation. To remedy this, it includes,

1. Check all temperature reading,
2. Check all pressure reading,

3. Check screw or ram speed reading and,
4. Check screw position reading.

3.3.8 Warpage

Warpage is that the part's configuration deformed or shifted to unexpected uni-direction, and causes the part becomes useless. A schematic to illustrate warpage is presented in Figure 3.8.

Warpage is one of the most difficult defect to eliminate or reduce during the injection molding process. It is because that hard to analyze their causes, and hard to recognize theirs exist during the process. The way to recognize warpage is inspecting by human after the process was completed.

Generally, the warpage is due to an uneven internal stress exists inside the parts, and results in the part deformed or bended toward an uni-direction. There are several sources that cause the uneven internal stresses inside the parts. The uneven cooling rate is the major one. The uneven cooling rate can be remedied as follows,

1. Decrease mold temperature,
2. Increase injection pressure,
3. Increase injection time,
4. Increase mold close time,
5. Decrease barrel temperature,
6. Decrease nozzle temperature,
7. Set uniform temperature in both halves of the mold,
8. Jig the part and cool uniformly,
9. Increase screw speed,
10. Increase cooling channel size,
11. Decrease gate size and,
12. Decrease runner size.

The other possible cause is that during the part is ejected from the mold cavity, the ejector force uneven applied in the parts. This results in that a bending moment is generated to the parts, and causes the part deformed or bended toward an uni-direction. The major solution for this kind of cause is to modify the ejector pins layout. The other one is to allow the part solidified enough to persist the bending moment. This can be achieved as indicate in the below,

1. Decrease mold temperature,
2. Increase injection time,
3. Increase mold close time,
4. Relocated ejector pins.

The other cause is instruments mislead the operating conditions. It results in the operating conditions are indicated inaccuracy amount and cause the deviation. The way to resolve this problem includes check all temperature reading, check all pressure reading, check screw or ram speed reading, and check screw position reading.

3.3.9 Distortion

Distortion is the other tough defect to remove during the injection molding process. Distortion is similarly as the warpage. The only different is that their deformed or bending direction not restricts in uni-direction. It possible deform in two-direction or even in three direction. Figure 3.9 shows a schematic of the distortion.

The causes and the suggested corrective actions for the distortion are presented in Table 3.10. As the causes of warpage, distortion is due to an uneven internal stress exists inside the parts, and results in the part deformed or bended toward some direction. To resolve this cause, it can be remedied by adjusting the variables as follows,

1. Decrease mold temperature,
2. Increase injection pressure,
3. Increase injection time,

4. Increase mold close time,
5. Decrease barrel temperature,
6. Decrease nozzle temperature,
7. Set uniform temperature in both halves of the mold,
8. Jig the part and cool uniformly,
9. Increase screw speed,
10. Increase cooling time,
11. Decrease gate size and,
12. Decrease runner size.

The other cause of distortion is that the ejector forces apply un-even into the parts. One of the resolution for un-even ejector forces is that modify the ejector pins layout. The other one is that solidify part hard enough to persist the un-even ejector force. To solidify part hard enough, the resolutions are list in below,

1. Decrease mold temperature,
2. Increase injection time,
3. Increase mold close time,
4. Relocated ejector pins.

The instruments mislead the operating conditions, it is also possibly to cause distortion. To remedy this, it requires to adjust all instruments individual. These instruments include temperature indicator, pressure indicators, screw or ram speed indicator, and screw position indicator.

3.3.10 Delamination

Figure 3.10 shows a schematic to illustrate delamination. It causes the part's dimension inaccurate, and results in that the parts become useless. A cavity pressure transducer can be used to indicate the existence of the delamination as shown in Figure 3.11.

The contamination contains inside the mold cavity or in the material is one of the cause of delamination. The contamination resists the molten material flow through the cavity, and results in the cavity insufficiently fill in. To ensure a clean mold surface, and the material does not contain any contamination, these are the resolutions to remedy this effect.

The molten material solidified prior the cavity fulfill is the other cause for the delamination. The solidified material is the resistor for the path of molten flow. Thus, to increase the flow rate of the molten material, it can prevent this situation occurs. To increase the flow rate, it can be accomplished by adjusting the operating variables as follows,

1. Increase barrel temperature,
2. Increase mold temperature,
3. Increase nozzle temperature,
4. Increase injection pressure,
5. Using the maximum injection speed or/and boost pressure,
6. Increase gate size,
7. Reduce regrind rate,
8. Increase cycle time,
9. Increase runner size,
10. Reduce cooling channel size,
11. Reduce screw speed and,
12. Reduce cushion.

The instrument reading failure, that misleads the operating conditions, is also a cause for delamination. The following listings can be used to resolve this cause. These are,

1. Check all temperature reading,
2. Check all pressure reading,
3. Check screw speed reading,

4. Check screw position reading.

3.4 Nominal Out of Control Operation

According to the definition of the controllability [Ogata, 1970], a controllable system is defined as at time t_0 if it is possible by means of an unconstrained control vector to transfer the process from any initial state $x(t_0)$ to any other state in a finite interval of time. Consider the control states of the deviations in the injection molding process described by,

$$x = u \cdot A \tag{3.1}$$

where,

x = states vector of the deviations in the injection molding process, (n-dimensional vector),

u = control signal, $n \times r$ matrix and,

$A = r \times 1$ matrix, r control variables.

Based on the discussion in section 3.3, the Eq. (3.1) can then employ the causes of the deviations, and rewritten as,

$$\begin{bmatrix} \text{surface ripples} \\ \text{pit marks} \\ \text{splay marks} \\ \text{sink marks} \\ \text{voids} \\ \text{flashing} \\ \text{short shots} \\ \text{warpage} \\ \text{distortion} \\ \text{delamination} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \text{barrel temperature} \\ \text{nozzle temperature} \\ \text{mold temperature} \\ \text{injection pressure} \\ \text{injection speed} \\ \text{injection time} \\ \text{cycle time} \\ \text{mold close time} \\ \text{mold open time} \\ \text{decompression} \\ \text{short size} \\ \text{cushion} \\ \text{regrind rate} \\ \text{screw speed} \end{bmatrix} \tag{3.2}$$

The control states described by Eq. (3.2) is said to be state controllable if it is possible to transfer the control signal matrix into an upper triangular or lower triangular matrix. It is obviously that, in Eq. (3.2), the control signal matrix is not possible to be transferred to an upper triangular or lower triangular matrix. Therefore, the control states of the deviations in the injection molding is nominated as an out control operation. However, in this program, an expert system technique is introduced to overcome this out control operation, and to eliminate or reduce the deviation in the injection molding. In the next two chapter, the basic concept of an expert system, and the development procedures of this program are discussed in detailed.

Table 3.1 The common deviation occur in the injection molding process.

Surface Deviations	1. Surface ripples 2. Pit marks 3. Splay marks 4. Sink marks 5. Voids 6. Flashing
Dimensional Deviations	1. Short shots 2. Warpage 3. Distortion 4. Delamination

Table 3.2 The possible causes and suggested correction actions for surface ripples deviation.

Possible Causes	Suggested Correction Actions
Material Shortage	1. Increase shot size. 2. Decrease cushion.
Material Flow Rate Too Slow in the Cavity.	1. Increase injection pressure. 2. Use the maximum injection speed or/and boost pressure. 3. Increase gate size. 4. Increase barrel temperature. 5. Increase mold temperature. 6. Increase nozzle temperature. 7. Reduce regrind rate. 8. Reduce screw speed. 9. Increase injection time.
Material Solidification Prior Cavity Filling	1. Increase barrel temperature. 2. Increase mold temperature. 3. Increase nozzle temperature. 4. Increase injection pressure. 5. Use the maximum injection speed or/and boost pressure. 6. Increase gate size. 7. Increase runner size. 8. Increase venting channel size. 9. Decrease cooling rate. 10. Reduce screw speed. 11. Increase injection time. 12. Increase cycle time.
Foreign Material Stick in Mold Surface	1. Clean the mold surface. 2. Clean the Hopper. 3. Clean the Screw.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading. 4. Check screw position reading.

Table 3.3 The possible causes and suggested correction actions for pit marks deviation.

Possible Causes	Suggested Correction Actions
Material Shortage	1. Increase shot size. 2. Decrease cushion.
Material Flow Rate Too Slow in the Cavity.	1. Increase injection pressure. 2. Use the maximum injection speed or/and boost pressure. 3. Increase gate size. 4. Increase barrel temperature. 5. Increase mold temperature. 6. Increase nozzle temperature. 7. Reduce regrind rate. 8. Reduce screw speed. 9. Increase injection time.
Material Solidification Prior Cavity Fulfill	1. Increase barrel temperature. 2. Increase mold temperature. 3. Increase nozzle temperature. 4. Increase injection pressure. 5. Use the maximum injection speed or/and boost pressure. 6. Increase gate size. 7. Increase runner size. 8. Increase venting channel size. 9. Decrease cooling rate. 10. Reduce screw speed. 11. Reduce injection time. 12. Increase cycle time.
Material Contains Foreign Component	1. Clean the mold surface. 2. Clean the hopper. 3. Clean the screw.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading. 4. Check screw position reading.

Table 3.4 The possible causes and suggested correction actions for splay marks deviation.

Possible Causes	Suggested Correction Action
Excessive Moisture	1. Reduce barrel temperature. 2. Reduce nozzle temperature. 3. Dry material. 4. Use a vented barrel. 5. Reduce injection speed. 6. Reduce mold temperature. 7. Reduce screw speed.
Drooling at Nozzle	1. Reduce barrel temperature. 2. Reduce nozzle temperature. 3. Reduce mold open time. 4. Reduce injection time. 5. Reduce cushion. 6. Use nozzle with small orifice. 7. Increase decompression time. 8. Increase venting channel size.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw position reading. 4. Check screw speed reading.

Table 3.5 The possible causes and suggested correction actions for sink marks deviation.

Possible Causes	Suggested Correction Action
Uneven Shrinkage in the Part	1. Decrease mold temperature. 2. Increase injection pressure. 3. Use the maximum injection speed or/and boost pressure. 4. Increase injection time. 5. Decrease barrel temperature. 6. Decrease nozzle temperature. 7. Increase gate size. 8. Relocate gate nearer heavy section. 9. Modify part design. 10. Increase cooling channel size.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw position reading.
Material Shortage	1. Decrease cushion. 2. Increase shot size.

Table 3.6 The possible causes and suggested correction actions for voids deviation.

Possible Causes	Suggested Correction Action
Uneven Shrinkage in the Part	1. Decrease mold temperature. 2. Increase injection pressure. 3. Use the maximum injection speed or/and boost pressure. 4. Increase injection time. 5. Decrease barrel temperature. 6. Decrease nozzle temperature. 7. Increase gate size. 8. Relocate gate nearer heavy section. 9. Increase cooling channel size. 10. Modify part design
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw position reading.
Material Shortage	1. Decrease cushion. 2. Increase shot size.

Table 3.7 The possible causes and suggested correction actions for flashing deviation.

Possible Causes	Suggested Correction Actions
Material Over Charge	1. Decrease shot size. 2. Increase cushion.
Material Viscosity too low in the cavity	1. Decrease barrel temperature. 2. Decrease mold temperature. 3. Decrease nozzle temperature. 4. Decrease injection pressure. 5. Decrease injection speed. 6. Decrease injection time. 7. Increase clamping force. 8. Increase venting channel size. 9. Decrease gate size. 10. Increase cooling channel size. 11. Decrease venting channel size. 12. Reduce runner size. 13. Increase screw speed.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading. 4. Check screw position reading.
Mold Wear	1. Modify mold surface. 2. Clean mold surface. 3. Increase clamping force.

Table 3.8 The possible causes and suggested correction actions for short shots deviation.

Possible Causes	Suggested Correction Actions
Material Shortage	1. Increase shot size. 2. Decrease cushion.
Material Flow Rate Too Slow in the Cavity.	1. Increase injection pressure. 2. Use the maximum injection speed or/and boost pressure. 3. Increase gate size. 4. Increase barrel temperature. 5. Increase mold temperature. 6. Increase nozzle temperature. 7. Reduce regrind rate. 8. Increase injection time. 9. Reduce screw speed.
Material Solidification Prior Cavity Filling	1. Increase barrel temperature. 2. Increase mold temperature. 3. Increase nozzle temperature. 4. Increase injection pressure. 5. Use the maximum injection speed or/and boost pressure. 6. Increase gate size. 7. Increase runner size. 8. Increase venting channel size. 9. Decrease cooling rate. 10. Reduce regrind rate. 11. Reduce injection time. 12. Reduce screw speed. 10. Increase cycle time.
Foreign Material Stick in Mold Surface	1. Clean the mold surface. 2. Clean the Hopper. 3. Clean the Screw.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading. 4. Check screw position reading.

Table 3.9 The possible causes and suggested correction actions for warpage deviation.

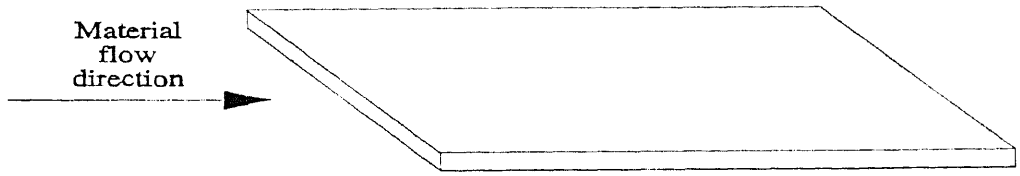
Possible Causes	Suggested Correction Action
Unbalanced Cooling Rate	1. Decrease mold temperature 2. Increase injection pressure. 3. Increase injection time. 4. Increase mold close time. 5. Decrease barrel temperature. 6. Decrease nozzle temperature. 7. Set uniform temperature in both halves of the mold. 8. Jig the part and cool uniformly. 9. Increase cooling channel size. 10. Decrease gate size. 11. Decrease runner size. 12. Increase screw speed.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading.
Unbalanced Ejection Force	1. Decrease mold temperature 2. Increase injection time. 3. Increase mold close time. 4. Relocate ejector pins.

Table 3.10 The possible causes and suggested correction actions for distortion deviation.

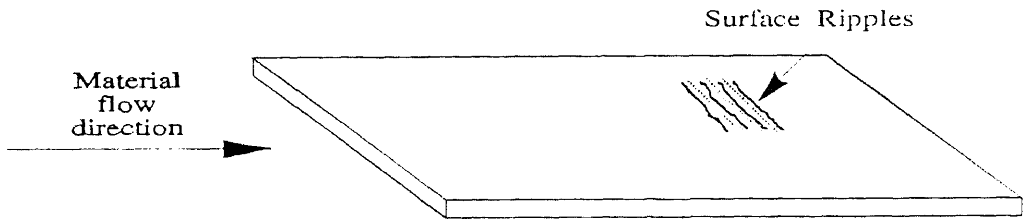
Possible Causes	Suggested Correction Action
Unbalanced Cooling Rate	1. Decrease mold temperature 2. Increase injection pressure. 3. Increase injection time. 4. Increase mold close time. 5. Decrease barrel temperature. 6. Decrease nozzle temperature. 7. Set uniform temperature in both halves of the mold. 8. Jig the part and cool uniformly. 9. Increase cooling channel size. 10. Decrease gate size. 11. Decrease runner size. 12. Increase screw speed.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading.
Unbalanced Ejection Force	1. Decrease mold temperature 2. Increase injection time. 3. Increase mold close time. 4. Relocate ejector pins.

Table 3.11 The possible causes and suggested correction actions for delamination deviation.

Possible Causes	Suggested Correction Actions
Contamination	1. Eliminate contamination from hopper. 2. Eliminate contamination from the mold cavity.
Instrument Reading Failure	1. Check all temperature reading. 2. Check all pressure reading. 3. Check screw or ram speed reading. 4. Check screw position reading.
Material Solidification Prior to Cavity Filling	1. Increase barrel temperature. 2. Increase mold temperature. 3. Increase nozzle temperature. 4. Increase injection pressure. 5. Use the maximum injection speed or/and boost pressure. 6. Increase gate size. 7. Reduce regrind rate. 8. Increase runner size. 9. Increase cycle time. 10. Reduce cooling channel size. 11. Reduce screw speed. 12. Reduce cushion.

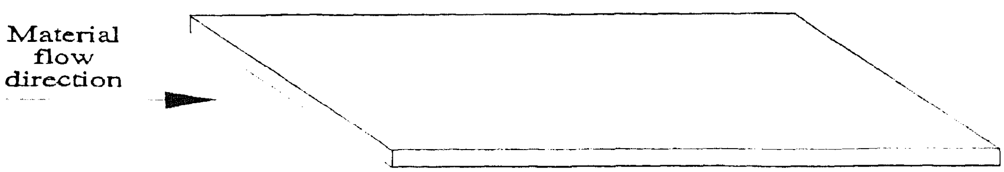


(a)

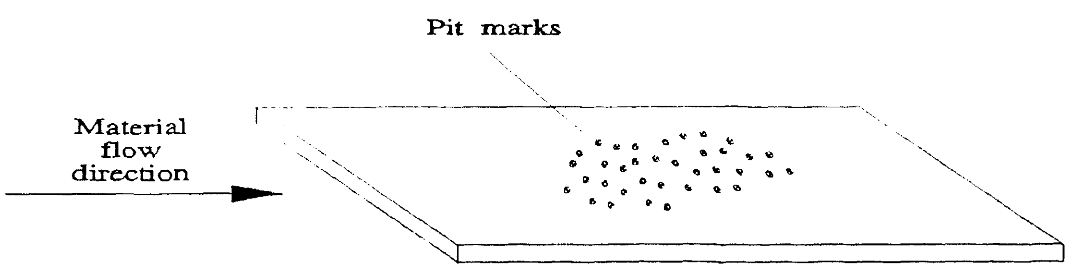


(b)

Figure 3.1 A schematic to illustrate the surface ripples deviation.



(a)



(b)

Figure 3.2 A schematic to illustrate the pit marks deviation.

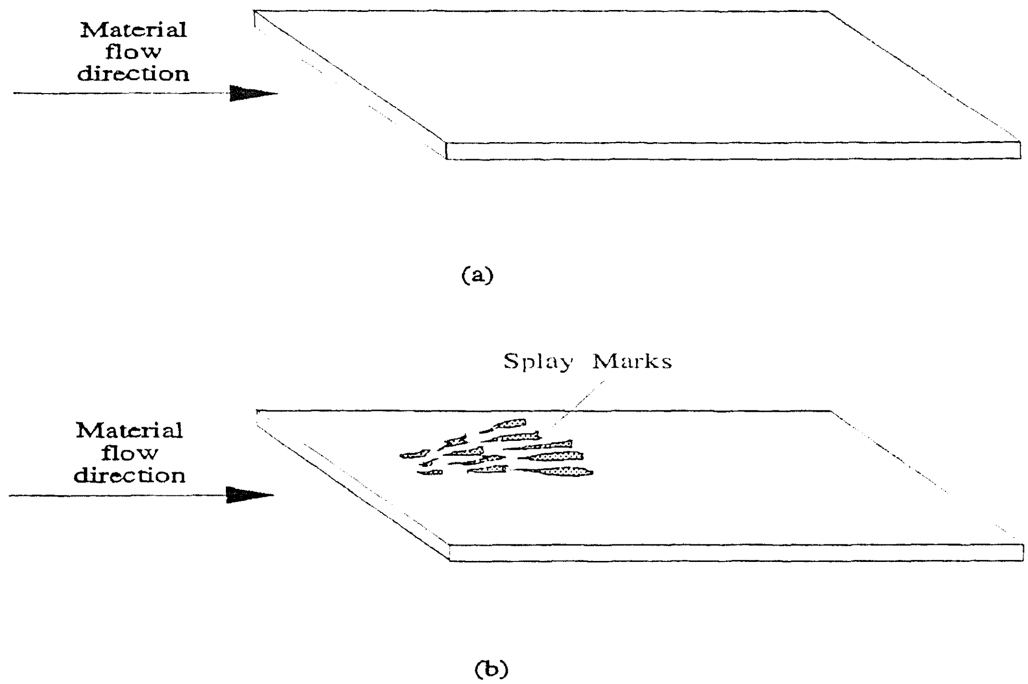


Figure 3.3 A schematic to illustrate the splay marks deviation.

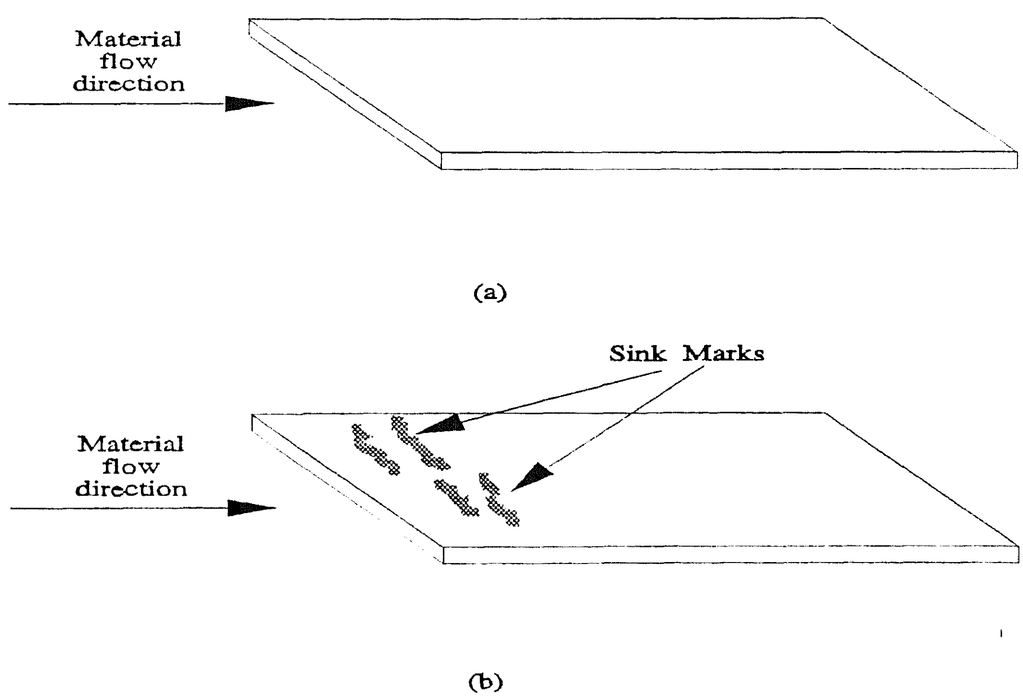


Figure 3.4 A schematic to illustrate the sink marks deviation.

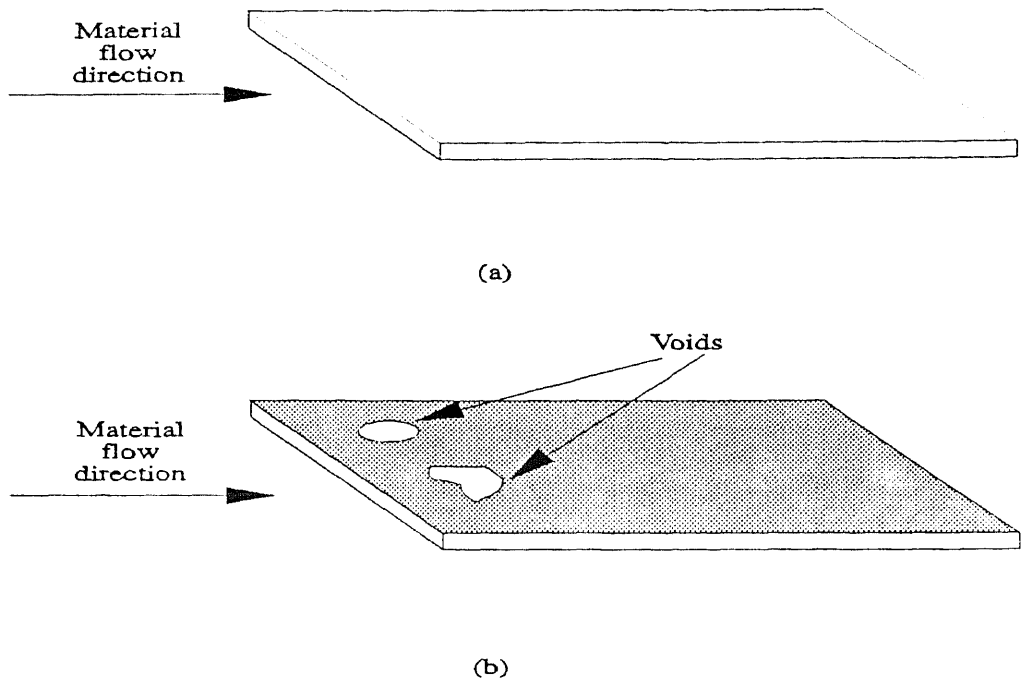


Figure 3.5 A schematic to illustrate the voids deviation.

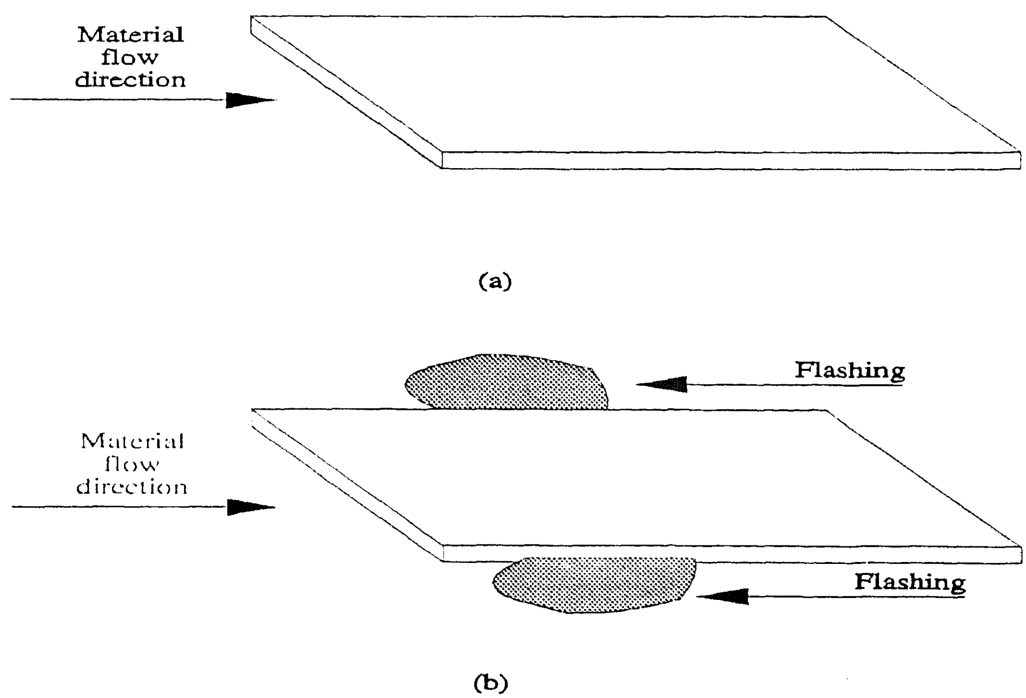


Figure 3.6 A schematic to illustrate the flashing deviation.

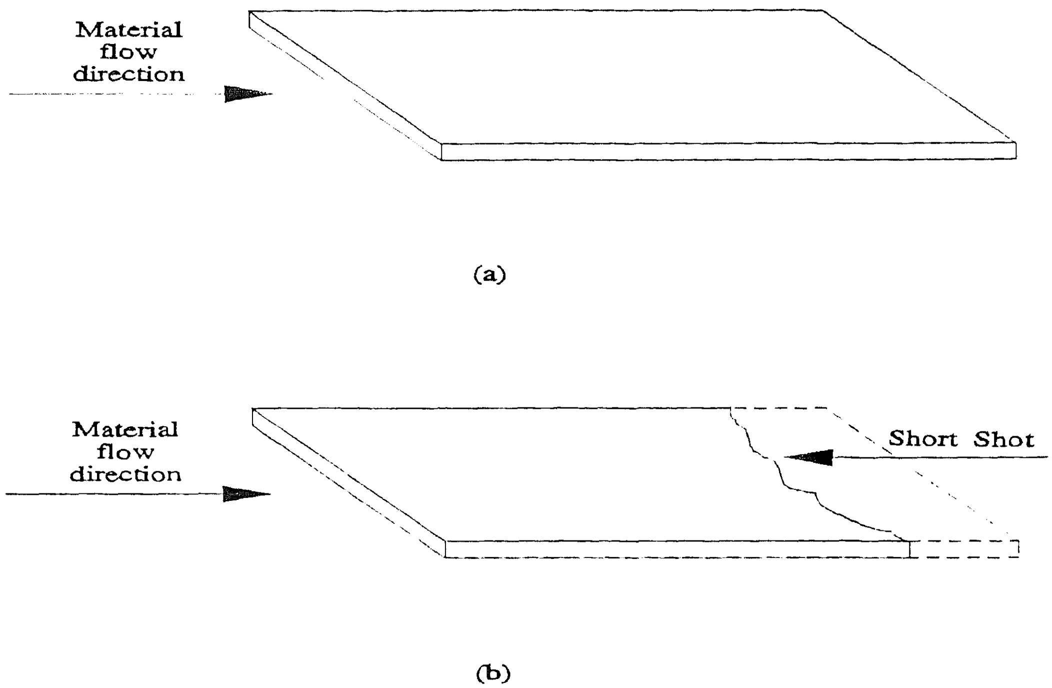


Figure 3.7 A schematic to illustrate the short shots deviation.

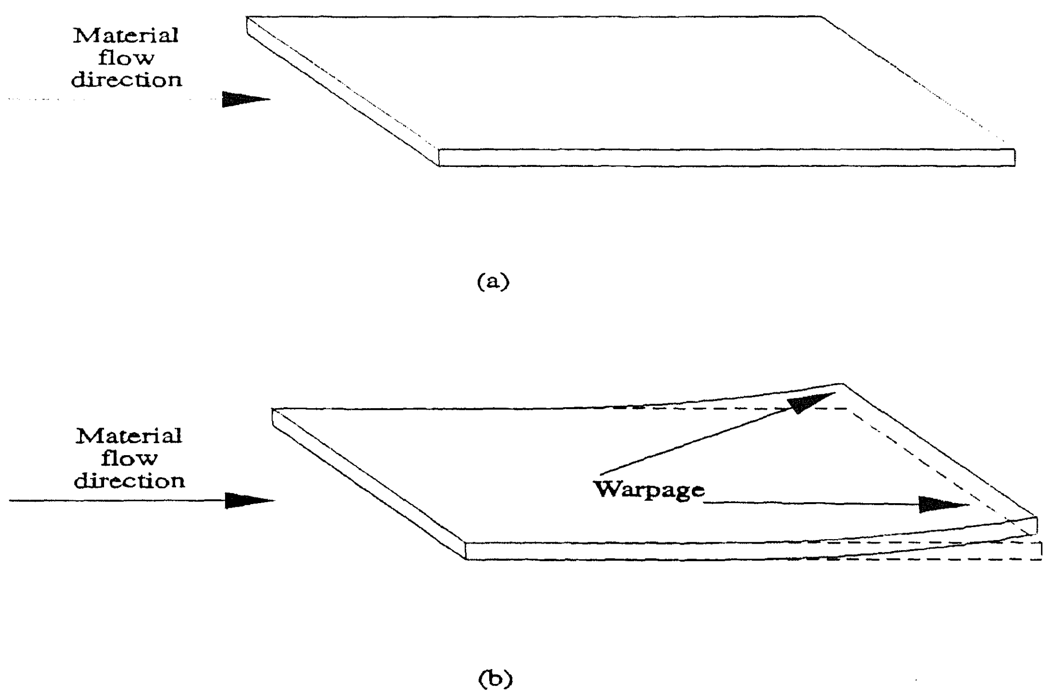


Figure 3.8 A schematic to illustrate the warpage deviation.

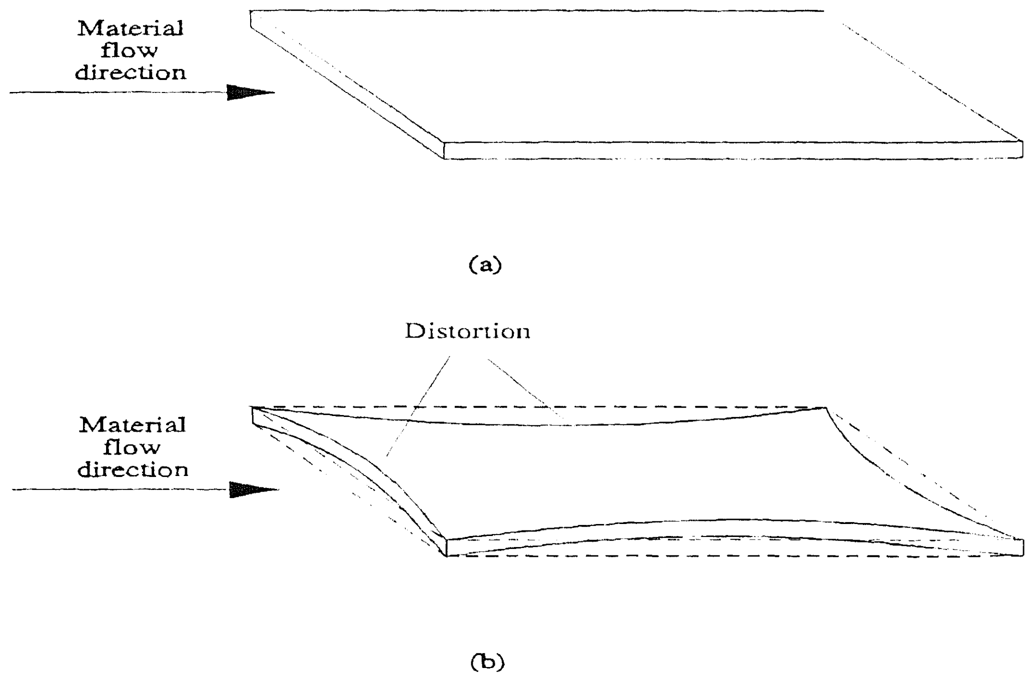


Figure 3.9 A schematic to illustrate the distortion deviation.

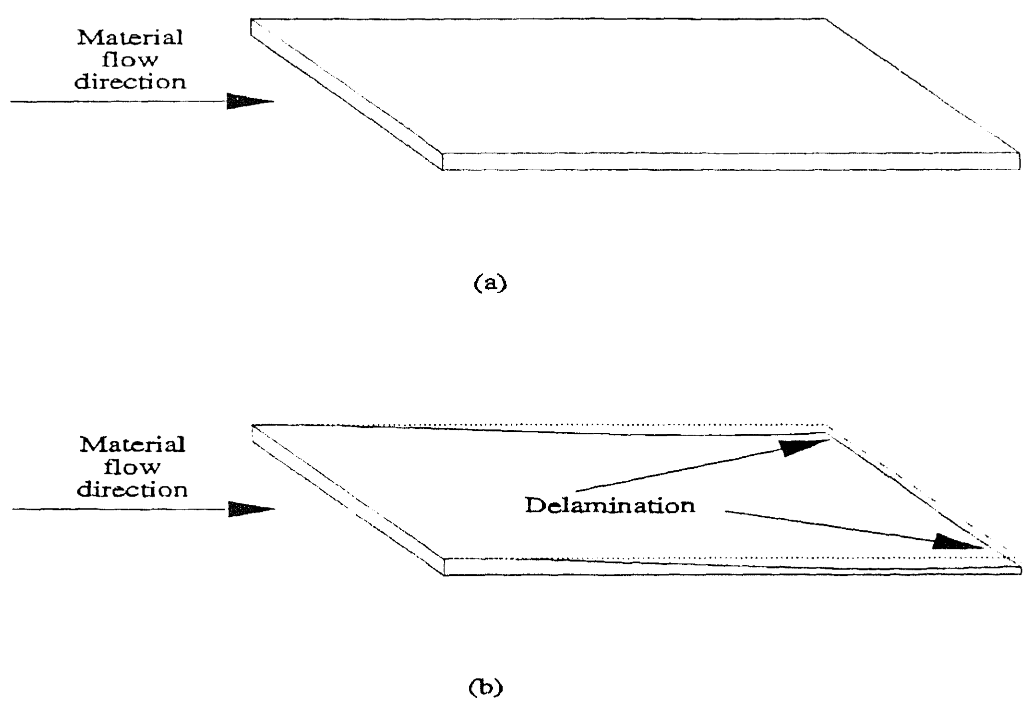


Figure 3.10 A schematic to illustrate the delamination deviation.

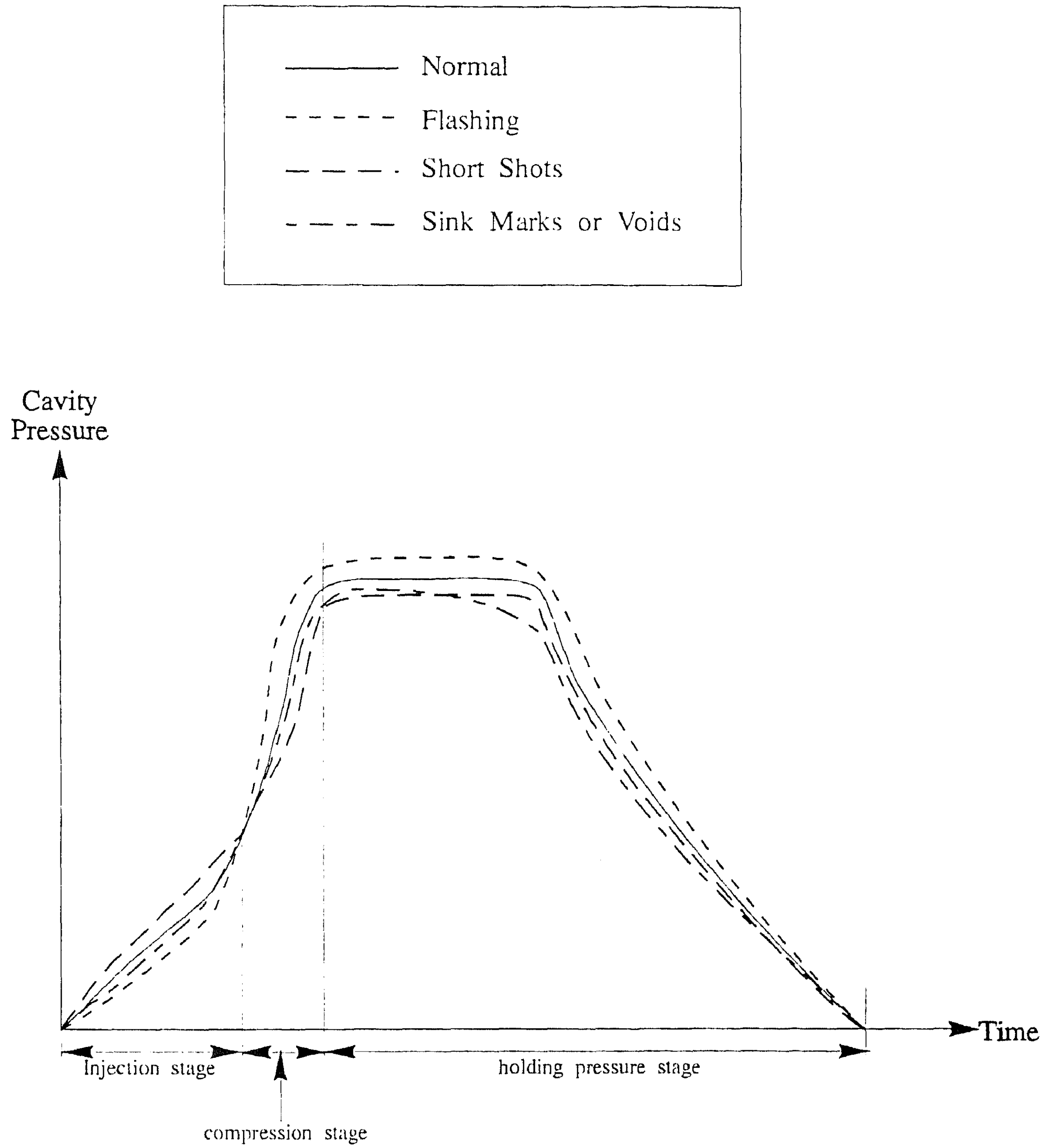


Figure 3.11 The deviations indicated from a cavity pressure-time relationship.

CHAPTER FOUR
PRINCIPLES OF EXPERT SYSTEM

4.1 Introduction

An expert system is a computer system that uses knowledge, fact, and reasoning techniques to solve complicated problems that would require extensive human expertise. To perform so, the expert system simulates the human reasoning process by applying specific knowledge and inference.

The expert system is a sub-branch computer technology of the artificial intelligence. The term of artificial intelligence appeared first in 1956, when a Summer Research Project held at Dartmouth College, that was stated by John McCarthy as the name of the processing [Parsaye and Chignell, 1988].

After 1956, the artificial intelligence became a specifically interested research field in the computer technology. Over the years, the artificial intelligence has been subtitled into several topics that includes,

- vision, its aim is to identify the objective as the human visual system performance,
- speech recognition, its aim is to analyze, recognize, and synthesize the way of the human speech,
- natural language processing, its aim is to understand and anatomize the way of the human writing,
- automatic reasoning and theorem proving, its aim is to automatically verify the theorem in the mathematics and logic as the human performance,
- automatic learning, its aim is to teach and generate the ability of the creation for the producing machines as the human learning procedures,
- robotics, its aim is to simulate the human motion, and,

- expert system, its aim is to solve the problem in specific domain as the human performance.

Currently, most of these topics are still in the developing or researching stage. However, the expert system is probably the most successful to employ into the real world.

The first successfully expert system, DENDRAL, was developed by Stanford University in the 1960's [Lindsay et al. 1980]. DENDRAL is an expert system to identify the structure of organic molecules from their mass spectrographic information and chemical formulas. Due to the components of organic molecules is usually very large, the number of possible structure for those is therefore tend to huge. In DENDRAL, it based on the knowledge of the expert organic chemist to construction specific rules, then allowed the program to use heuristic searching strategies and to solve the problem. DENDRAL's success demonstrated that the expert system can be used to solve the problem that requires the human expertise in specific domain.

After DENDRAL successfully developed, expert system became a useful computer technique for solving the problem that requires to use the knowledge of human expertise. There are several remarkable expert systems have been successfully developed. In the mid 1970's, MYCIN [Buchanan and Shortliffe, 1984] was successfully developed by Stanford University. MYCIN is an expert system to diagnose and recommend treatment for the meningitis and blood infections. The most important of MYCIN is that it introduces the rule-based representation concepts for the expert system. Furthermore, it introduces the uncertain and the reliable evaluation theorem into the expert system. Using these theorems, the expert system can be used in the incompleted information domain problems.

Other outstanding expert systems include the PROSPECTOR [Duda et al. 1979] for determining the probable location of and type of ore deposits based on geological information, the INTERNIST [Pople et al., 1975] for judging about relationship between diseases and symptoms in internal medicine, and the XCON for configuring VAX computers.

Today, expert systems have been widely use in the field of medicine, education, business, mathematics, computer science, engineering, and science. Waterman [Waterman, 1986] based on problem domain that is applied in expert system and classified expert system into several sub-groups. These are,

- interpretation, which forming high-level conclusions or descriptions from collections of raw date,
- prediction, which projecting probable consequence of given situations,
- diagnosis, which determining the cause of malfunctions of complex situations based on observable symptoms,
- design, which determining a configuration of system components that meets certain performance goals while satisfying a set of constraints,
- planning, which devising a sequence of actions that will achieve a set of goals given certain starting conditions,
- monitoring, which comparing the observed behavior of a system to its expected behavior,
- debugging and repair, which prescribing and implementing remedies for malfunctions,
- instruction, which detecting and correcting deficiencies in students understanding of a subject domain, and,
- control, which governing the behavior of a complex environment.

4.2 Architecture of the Expert Systems

As shown in Figure 4.1, an expert system consists of five fundamental components. These are,

- the user,
- the user interface,
- the knowledge base,
- the inference engine and,
- the explanation facility.

4.2.1 User

A user usually responds to the questions that the system asks and helps the system to define the domain of the problem. Once, the system domain has been developed, the system will, according the response of the user, gain its own problem solving skills. It will make the system perform more efficiently.

During developing an expert system, the potential users shall be involved into the development stages as early as possible. It is because that the potential users are well known about their problem domain requirements. Therefore, the system developers can base on their demands to ensure that the problem domain is well defined.

In addition, the potential users can provide their requirements about the interface styles of the user interface to the system developers. It allows the system developers to design a more conformable communication way between the user and the system. Furthermore, during the development testing, the potential users can test the prototype system in the previously stage. It can avoid a large modification in the finally system testing. Thus, the system development time can be saved.

There are several points that the potential users shall keep in their mind. First, an expert system is a tool to help them to resolve their problem. It is not a replacement for their jobs. If the potential users feel that their jobs are threatened by the expert system, then they may not like to find out the advantages of using the system. It results in that the system performance will be subtracted.

Second, there is no perfect expert system. An expert system likes a human will make some mistakes. The users shall remark these mistakes to the system developers. Therefore, the system developers can modify and improve the expert system performance.

Third, an expert system may not provide the best resolutions for the problems as the user expectation. An expert system is still a computer program. Its performance depends on their coding procedures. Sometimes, the system even cannot provide a resolution to the users. At this kind of conditions, the users could try another resolution procedures, or communicate with the system developers, and modify their coding procedures.

Fourth, the user is also a teacher of the system. Most of the expert systems have the self-learning capabilities. The system will base on the user respond, and improve their problem resolving skills.

4.2.2 User Interface

The user interface provides communication tools between the users and the system. It will offer a more comfortable environment to the user to use the system. As shown in Figure 4.1, the user interface can communicate with the inference engine, the knowledge base, and the explanation facility.

When the user interface communicates with the inference engine, the users tell the system which rules shall be employed to search the possible resolutions. As the user interface communicates to the knowledge base, the system on the basis of the inference engine search results, responds with the possible resolution to the user. When the user interface communicates with the explanation facility, the system provides the explanation for the user why chosen was suggested.

There are four common interface styles. These are,

- question and answer,

- menu driven,
- graphics and,
- natural language.

4.2.2.1 Question and Answer. The question and answer interface is the most popular way to communicate with the system. It is most by used in the problem can be elicited step by step. For instance, MYCIN uses the question and answer style to elicit the necessary information to define the problem from the user. Its performances are shown as below [Luger and Stubblefiled, 1989].

At first, a basic introduction of the MYCIN is shown in Figure 4.2. After the basic introduction, the system begins to elicit the necessary information from the user, and to define the problem as presented in Figure 4.3. In Figure 4.3, the user types "Why" to require the system to explain why take this action. The system will restate the question and then respond the answer as shown in Figure 4.4.

4.2.2.2 Menu Driven. The menu driven interface is similar to the question answer style. The only difference is the menu driven provides the possible answer to user in each question. It also satisfies to use in the problem domain that can be absolutely defined.

Usually, the menu driven interface uses a simple character or key in the keyboard to represent the answer. This provides several advantages. First, it can avoid the ambiguous answer from the users. Second, it provides a more comfortable user environment. Third, it can evade the typing mistake during responding the answer. Fourth, it can more accurately define the problem domain. Since the menu driven interface provides such advantages, in this program, it is selected as its user interface.

In here, this program's user interface is used to demonstrate the menu driven performance. First, the introduction screen is presented as shown in Figure 4.5.

Using this kind of the user interface, the questions domain usually are constrained in certain range, and have an absolute answer. To design this kind of user interface, the question statements shall be expressed as clearly as possible. It can avoid the user confusion to respond the answer.

After the introduction screen, the system begins to induce the necessary information from the user, the system then prompts to as presented in Figure 4.6. Once the system has been collected all the necessary information from the users, the system then begins to present the suggested actions to the users. The system then prompts the user to the suggested action as shown in Figure 4.7.

To design a good menu driven, there are several key factors. First, the answer listing shall contain all possible answer to the users. It then can avoid to define an incompleted problem domain. Second, the responded function keys shall be indicated as clearly and easy as possible. It then can avoid the mistake respond. Third, the system shall provide a explicit introduction for using their function. It then provides a more comfortable using environment.

4.2.2.3 Graphics. The graphics user interface is usually to be used in the applications, such as, design, control, monitor, and debug and repair proposals. Most of these applications can use a graphics to represent their objective or process procedures. The graphics user interface then provides a schematic that can help the users to a better understanding of their problem objective or their problem procedures.

The FMS Communication Diagnostic System (FMSCDS) [Martin and Oxman, 1988] is used to demonstrate that how is the performance of the graphics user interface. The FMSCDS was developed by Texas Instrument. It is an expert system to help identify and correct computer communication problems in the flexible manufacturing systems running at Texas Instrument facility.

After the FMSCDS was loaded, the introduction screen of the system then will be displayed as shown in Figure 4.8. In this introduction screen, an overall view of the part and the manufacturing cell is presented. In addition, the function keys of the operation are listed in the bottom of the screen.

By entering the "RETURN" key to continue the process, the system then requires the users to input their name, the date, and the time. After user's responded, the system then prompts the screen as presented in Figure 4.9. In Figure 4.10, the question which is asked by the system is presented on the top of the screen. The overall view of the part is present in the middle of the screen. And, the function keys of the operation are still listed in the bottom of the screen.

At this time, if the users hit the "Help" function key, then the screen appears as in Figure 4.11 In Figure 4.11, the system provides more information about question in the Figure 4.9. Once the users responded the answer to the system, the system then indicates an intermediate conclusion in the screen as shown in Figure 4.11. At this point, the users can either choice to stop the program, or to resolve the other problems.

There are two advantages to use the graphics interface. First, it uses the graphics to represent the problem objective or the problem procedures. It allows the users to see more clearly their problem domain. Second, the graphics interface usually is combined with the menu driven interface. Therefore, it includes the advantages of the menu driven interface. However, the major disadvantages of the graphics interface is the limitation of the graphics representation. Since the graphics representation occupies a lot of memory space in the computer system. To download the graphics representation into the expert system, it requires a lot of time to accomplish. It reduces the system performance speed. Therefore, to use graphics interface, the graphics representation of the problem domains is limited to the simple representation.

4.2.2.4 Natural Languages. The natural language allows the users directly communicate with the system by the way like human conversation. Doing so the system requires to add a translation process as shown in Figure 4.12. The translation process function is to interpret the natural language to an acceptable format for the system, and to translate the system responds by the voice like human speaking for the users. In addition, to analyze, recognize, and relate the concepts of the expressed by the user, the inference engine shall has the ability to identify and distinguish the structure of the natural language.

In here, an expert system termed ELIZA [Weizenbaum, 1966] is used to illustrate the performance of the natural language interface. ELIZA was an expert system that was designed to simulate appearances of the dialog of Rogerian psychotherapists, who reflect patient responses. ELIZA used echo to communicate with the users as presented in Figure 4.13.

The major advantage to use the natural language interface is that the user may not require to has any computer background. It can directly "talking" with the system. Nevertheless, using the natural language interface, the structure of the knowledge base and inference engine become more complicated to develop. In addition, it requires a translate processor that increasing the cost of the whole system. Therefore, the natural language interface is rarely to be used in the common expert system. Unless, the system requires to deal a lot communication between the system and the user.

4.2.3 Inference Engine

The inference engine uses the information provided by the knowledge base and the user's response to determine the facts. It allows the system to decide how, and in what order, the knowledge data in the knowledge base should be employed. To perform so, the inference engine is simulating the human proficiency to deduce the reasoning strategies used by the system.

Today, there are many searching strategies used in the expert system. The more popular and common searching strategies includes,

- backward chaining search,
- forward chaining search,
- depth-first search,
- breadth-first search and,
- rule-value search.

4.2.3.1 Backward-Chaining Search. Backward chaining sometimes is also called goal-driven or object-driven reason. Backward chaining focuses on the goal of the problem domain. It finds the rules that could produce the goal. It then chains backward through consequent rules and sub-goals to the given facts of the problem. Its searching direction is from the goal of the problem to the facts of the problem as present in Figure 4.14.

Backward chaining search is suggested to use in the problem, if,

- A goal or hypothesis is given in the problem statements or can easily be formulated. For example, in an expert system for proving a mathematics theorem, the goal is given in a clear problem statement that is to prove the theorem. Backward chaining search is therefore suggested to use in this kind of the expert system. Furthermore, in this program, the goal of this expert system can be clearly define as to reduce or eliminate the specific deviation during the injection molding process. In addition, this expert system considers potential diagnoses in a systematic fashion. It can easily be formulated in terms of the relationship between causes and effects. These are key factors to the choice of backward chaining search for this system.
- There are a large number of rules that match the facts of the problem and thus produce an increasing number of goals. Early selection of a goal can

eliminate most of these branches. It makes backward chaining search more effective. For instance, in this program, the number of rules to conclude a suggested action statement for resolving a specific deviation is much small than the number of rules to induce a deviation from a fact. It means that the rules to conclude a suggested action statement "increase the material temperature can be used to resolve short shot", are fewer than the rules to induce a fact "a too low material temperature is possibly cause for short shot".

- Problem data are not given but must be acquired by the users. In this case, backward chaining searching can help guide data acquisition. For example, in this expert system, a low material temperature is possibly a causes for the deviations, such as, short shots, surface ripples, or pit marks. If short shots are encountered, the user only needs to confirm the specific deviation, short shots, to the system. The system will then narrow its searching steps in the particular set of the rules that influence the occurrence of short shots. This allows a more efficient system performance.

4.2.3.2 Forward-Chaining Search. Forward chaining search is an alternative search of backward chaining. The user begins with the given facts of the problem and a set of the rules for changing its searching path. Searching procedures employ the rules to facts and generate new facts. These new facts then applies other rules to other facts and product more new facts. This procedure continues until a satisfied goal is found. Figure 4.15 presents is a schematic to illustrate the searching path of forward chaining search. Since the forward chaining search driven from the facts to the goal of the problem. Therefore, it is also named as data-driven search.

For the problems listed below, it is more appropriately to use forward chaining search. These are,

- In the initial problem statement, all or most of the data are given by the user. Interpretation problems are the best examples for the use forward chaining search. In most of interpretation problems, the user inputs their collected data to the system. Then, the system analyze these data to provide a reasonable result to the user. In the expert system, the PROSPECTOR, for instance, the system bases on the geological data that are input by the user to interpret their analyzed results, and to find what minerals are possibly contained at a site.
- There are a large number of potential goals, but only a few ways to use the facts and given information of a particular problem instance. For instance, the DENDRAL is an expert system based on formula, the mass spectrographic data and the knowledge of chemistry to find the molecular structure of organic compounds. For an unidentified organic compound, it has many possible molecular structures. However, in the DENDRAL, it bases on mass spectrographic data of organic compound to induce a few of possible structures, and then analyze it with other available information to identify its molecular structure.
- It is difficult to form a goal or a hypothesis. In the DENDRAL, for instance, due to the possible molecular structure of organic compounds is too huge to make a hypothesis or a goal, it is therefore no way to use backward chaining search.

4.2.3.3 Depth-First Search. In depth-first search, searching path is performed branch by branch as presented in Figure 4.16. Once a point of one branch has been examined. All of its children and descendants are examined first until reached at the end point of this branch. If it is still cannot find the goal, then the searching path will remove to another branch.

As shown in Figure 4.16, if the start point is in fact "A", and the goal point is in fact "L", and the searching path start from the "BDH" branch, then the point "B" will be examined first. Since the point "B" is not a goal point, the searching path then remove to the children point "D". Once the searching path reached the last descendant of the branch "BDH" point "H", and is still cannot find the goal point. The searching path then removes to other branch "BDI". The searching path will repeat this procedure until the goal point "L" is be found. Therefore, the searching path in Figure 4.16 is in the order "B", "D", "H", "I", "E", "J", "K", "C", "F", and "L".

Depth-first search will guarantee that the goal point can be found, if the goal point is in the searching space. However, if there is more than one goal points exist in the searching space, then depth-first search will not promise that it can reach to the optimal goal. It is due to depth-first search will stop when any goal point is reached first. Furthermore, depth-first search will search each point exist in the branch even the goal point is not exist in this branch. For a small search space, it possible is not a problem. However, if the search space is very large, and the goal point exist in the last branch of searching path, then depth-first search may require a lot time to find the goal point. It influences the system performance.

4.2.3.4 Breadth-First Search. Breadth-first search, in contrast with depth-first search, its searching path is level by level as presented in Figure 4.17. Breadth-first search examines from the points in the higher level down to the point in the lower level. Once, if one of the point in a certain level is examined, then the other point in the same level will examined sequentially until the searching path reached in the last point of this level. At this time, if the goal point is still not found, then, the searching path move down to the point in the next lower level.

Using Figure 4-17 illustrates breadth-first search. If the goal point is in fact "L", and the start point is in fact "A". Then breadth-first search will begin from the level

"BC", and start from point "B". Since the goal point is not in this level. Once, the searching path reached the last point, "C", of the level "BC". The searching path then move down to next lower level "DEFG", and repeat this sequence, until the goal point "L" is found. Thus, in this example, the order of breadth-first search is "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", and "L".

As depth-first search, breadth-first search can guarantee the goal point will be found. But, breadth-first search same as depth-first search, once a goal point is found, then the searching path will stop. It results in that breadth-first cannot promise the reached goal point is an optimal goal point.

Furthermore, breadth-first search has same disadvantage as depth-first search to be used in a large searching space. It needs to spend a lot time for searching the goal point. However, in a simple problem if there is a simple solution and the explored level is not too much, then breadth-first search maybe can be considered to employ.

4.2.3.5 Rule-Value Search. Rule-value search is developed to improve the aimless searching for both of depth-first and breadth-first search. Its searching path is a combination of depth-first and breadth-first search as presented in Figure 4.18.

In rule-value search, each facts will be assigned a value. It is termed as "weighting factor". The searching track of rule-value search is that examining the weighting factors of each facts in the same level first. It then selects the most likely fact which has the most important influence on the goal. Once the most likely fact has been selected, its path then goes through to the facts which are the children facts of selected fact. If there is more than one child, then the comparison of the children's weighting factors will be employ first and selected one of the most likely children fact. This performance will continue until the goal is found, or there is not any children facts for the last fact.

For instance, in Figure 4.18, the starting points is in the fact "A", and the goal point is in the fact "L". In rule-value search, first, the searching path is to compare the weighting factors, "BV" and "CV", of the facts "B" and "C", which are the children facts of the fact "A". If the weighting factor "CV" is more important than "BV" for goal fact "L", then the searching path go down to "C" children facts "F" and "G". Since, there have two children of "C". Therefore, the comparison of "FV" and "GV", which are the weighting factors of "F" and "G" respectively, is examined foremost. Here, if "FV" has more influence than "GV", then the searching path goes down to "F" children facts "L" and "M". It is obviously that the goal fact "L" appears in this step. Therefore, the searching step goes to "L", and accomplishes whole searching procedures. Thus, in this example, the searching path is "A", "C", "F", and "L".

In section 4.2.3.3 and 4.2.3.4, the searching space, which was used to present the searching procedures of depth-first and breadth-first search, is the same as the searching space, which was used in the rule-value search. And, the searching steps for depth-first and breadth-first search are 11 steps and 12 steps, respectively. But, in rule-value search, it only requires 4 steps to complete the searching. It is obviously that rule-value search has more efficiently searching performance than depth-first and breadth-first search.

However, the most difficult to accomplish rule-value search is how to evaluate the weighting factors of each facts. In a large problem domain, the related facts probably tends to be too huge to allow system developers to analyze their influences on the goal. It results in difficulty to judge the weighting factors.

Furthermore, in rule-value search, it sometimes cannot find the goal. It is due to the path of rule-value search is always an "one-way" searching. It means that when the facts, which has been passed by during the searching procedures, will not be re-examined again. For instance, in above example as shown in Figure 4.11, if the goal is in the fact "O" instead to in the fact "L", and the weighting factors of each facts is still the same as above described. Then the searching path is still "A", "C", "F", and "L". When the searching path reached to fact "L", since there is not any children fact for "L". Then the searching will be stopped. It results in an unsuccessful searching. At this time, the searching path will not return to "C" and re-check the fact "G". Therefore, using rule-value search, an accurate evaluation of the weighting factors becomes the most important key factors.

4.2.4 Knowledge Base

The knowledge base is the key component of the expert system. It consists of the knowledge of the expert on a particular subject. According to Luger and Stubblefield [Luger and Stubblefield, 1989], they defined the knowledge as:

> *Knowledge is information about the world that allows an expert to make decisions.*

In order to use the knowledge for reasoning in an expert system, the format of knowledge representation is required to be employed into the inference engine for firing searching strategy and can be represented the conclusion to the user. To perform so, the knowledge representation must be sorted and induced how apply in searching strategy.

Today, the most popular knowledge representation is the rule-base representation. The basic concept of rule base representation is to use the if-then rules. In the if-then rule, knowledge represents the relationship between the facts and goal, or the facts and their children facts. The basic structure of the if-then rules is "IF premise is true; THEN output its corresponded conclusion".

There are several advantages to use the rule-base representation. First, it is flexible. In the rule-base representation, the individual rules can be easily added, removed, and updated. Second, the rule represents only the relationship between the facts and the goal, or the facts and their children facts that are linked by the rule. It allows easily to interpret their relationship. Furthermore, the rule-base representation is always structured in a way similar to the way people think to resolve the problem. This allows an expert system to perform as a human expert. However, the primarily disadvantage of the rule-base representation is to design an more efficient inference engine to reduce the searching time. It is due to most of the rule-base representation requires a large of rules to represent the whole problem domain. An inappropriate searching strategy causes a lot of unnecessary searching steps. It reduces the system performance capability.

4.2.5 Explanation Facility

The explanation facility allows the system to explain the reason for the system conclusion. These explanations includes why the system requires a particular data or a respond from the user, and how to justify the conclusion provides from the system.

In a "why" query, the user usually has been asked to respond some information from the system, and does not understand why needs to respond the question or how to answer the question. For instance, in this program, the "why" query is provided in each interrupted step during the system execution. As shown in Figure 4.19, the system provides a function key "?" to allow the user asks why is this action shall be took. Once the user respond the function key "?" to ask the system why take this action. The system then shows the explanation statement for explaining why take this action.

In a "how" query, the system usually has been reached a conclusion for the problem. However, the user probably do not understand why is the reason to reach this conclusion or how to execute this conclusion. In this program, for example, as the conclusion is given by the system, the system provides a function key, "?", to allow the user asks the system why the conclusion is reached or how to execute the conclusion. It is shown in Figure 4.20. Once the user responded "?", then the system provides an explanation statement for interpreting the reason to take this action.

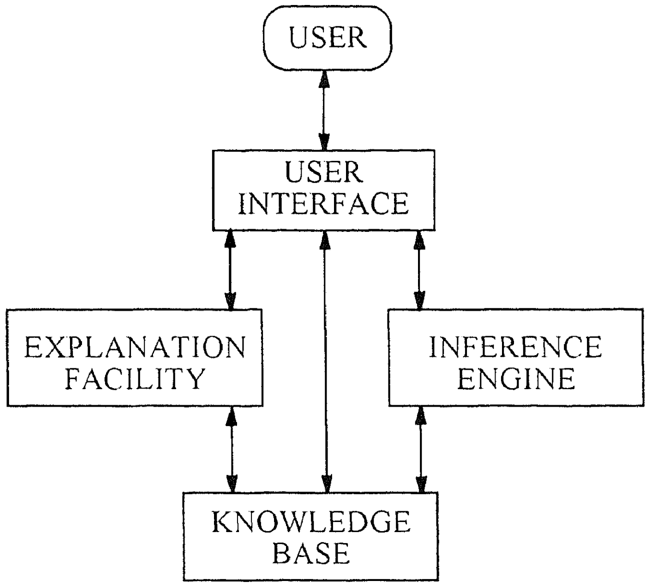


Figure 4.1 Basic expert system architecture.

```
SYSTEM:    @mycin
           MYCIN 26-March....
           Instructions?(Y or N)
USER:      N
SYSTEM:  26-March 10:42
           ----PATIENT----
           Mycin has created an object (PATIENT-1) in its dynamic
           knowledge base or working memory. Certain attributes are
           immediately requested and are necessary for every
           patient. These make up the first four question....
```

Figure 4.2 The introduction screen of the MYCIN.

```
SYSTEM:    (1) Patient's name:
USER:      Alan Apt
SYSTEM:    (2) Age?
USER:      34
SYSTEM:    (3) Sex
USER:      Why
```

Figure 4.3 The information induce screen of the MYCIN.

```
SYSTEM:     [i.e., why is it important to determine the sex of
            PATIENT-1]
            [1.0] The sex of the patient is one of the four standard
            parameters that identify a patient. This may be useful
            later....
USER:       Male
SYSTEM:     (4) Race?
USER:       Caucasian
SYSTEM:     (5) Are there any cultures for Alan Apt that may be
            related to the present illness, and from which organisms
            have been grown successful in the lab?
USER:       No
and so on ....
```

Figure 4.4 The explanation screen of the MYCIN.

```
************************************************************
         WELCOME TO THE N.J.I.T. EXPERT SYSTEM FOR
    THE INJECTION MOLDING OF ENGINEERING THERMOPLASTICS
       THIS SYSTEM IS USED TO ELIMINATE OR TO REDUCE
          DEVIATIONS IN THE INJECTION MOLDING OF
                 ENGINEERING THERMOPLASTICS
************************************************************
```

Figure 4.5 The introduction screen of the N.J.I.T. injection molding expert system.

```
**********************************************************
            IDENTIFICATION OF MOLDING MATERIAL
**********************************************************
IF YOU WANT TO KNOW WHY TAKE TO THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
PLEASE INDICATE YOUR MOLDING MATERIAL BY ENTERING THE CODE NUMBER
1. ACETAL_COPOLYMER
2. ACETAL_HOMOPOLYMER
3. NYLON_6
4. NYLON_66
5. POLYCARBONATE
6. PBT
7. PET
8. POLYSTYRENE
9. ABS
10. SAN
11. HIGH_IMPACT_POLYSTYRENE
12. EXIT
MATERIAL CODE NUMBER = 1
YOUR MATERIAL INDICATED NUMBER IS : 1
YOUR MATERIAL NAME IS : ACETAL_COPOLYMER
```

Figure 4.6 The material identification screen of the N.J.I.T. injection molding expert system.

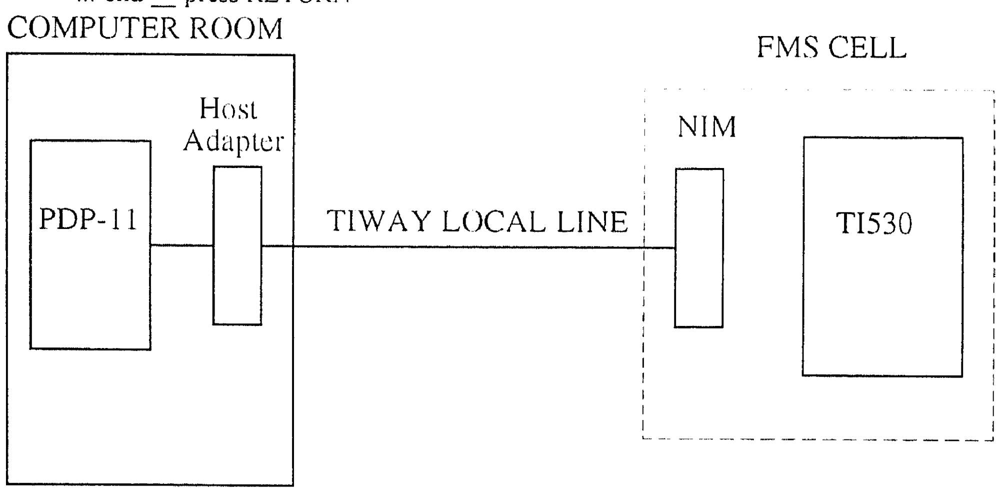
```
SYSTEM:
************************************************************
          BEGIN TO RESOLVE THE DEVIATION FLASHING
************************************************************
SUGGESTED ACTION:
DOES THE MOLD SURFACE STICK WITH MATERIAL
OR/AND FOREIGN CONTAMINATION
IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
USER:
C
```

Figure 4.7 The corrective action screen for the method corrective action "clean the mold surface" of the N.J.I.T. injection molding screen.

Knowledge Base: : Diagnostic for FMS Communications Link

The following is a diagnostic expert system for the Trinity Mills communications link between the PDP-11/24 host computer and the TI-530 Programmable Controller.

... end _ press RETURN



Up	Down	CF	Unknown	Done	---	Why	How	Help	Undo	Stop
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	ESC

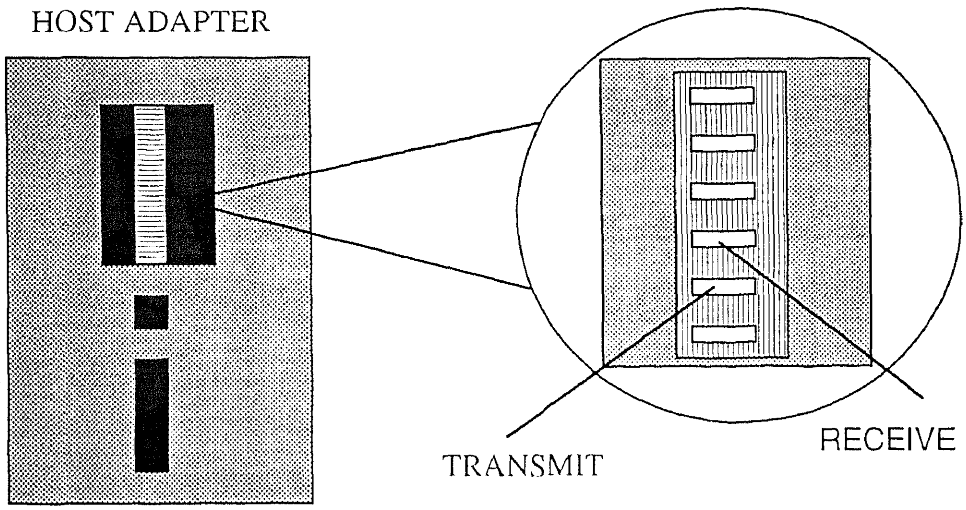
Figure 4.8 The introduction screen of the FMSCDS.

Knowledge Base: : Diagnostic for FMS Communications Link

Check the Transmit and Receive lights on the HA (host adapter).
Are they blinked or are they off?

Select one of the following: (Press 'F3' for selection help)

BLINKING
OFF



Up	Down	CF	Unknown	Done	---	Why	How	Help	Undo	Stop
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	ESC

Figure 4.9 The question screen of the FMSCDS.

Knowledge Base: : Diagnostic for FMS Communications Link

The Transmit and Receive lights are located on the front panel of the HA box, and are the fourth and fifth lights down on the box.

Select one of the following: (Press 'F3' for selection help)

BLINKING
OFF

Up	Down	CF	Unknown	Done	---	Why	How	Help	Undo	Stop
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	ESC

Figure 4.10 The help screen of the FMSCDS.

Knowledge Base: : Diagnostic for FMS Communications Link

The Host Adapter is not receiving any signal so the problem is probably in the computer room.
...end __ press RETURN

Up	Down	CF	Unknown	Done	---	Why	How	Help	Undo	Stop
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	ESC

Figure 4.11 The conclusion screen of the FMSCDS.

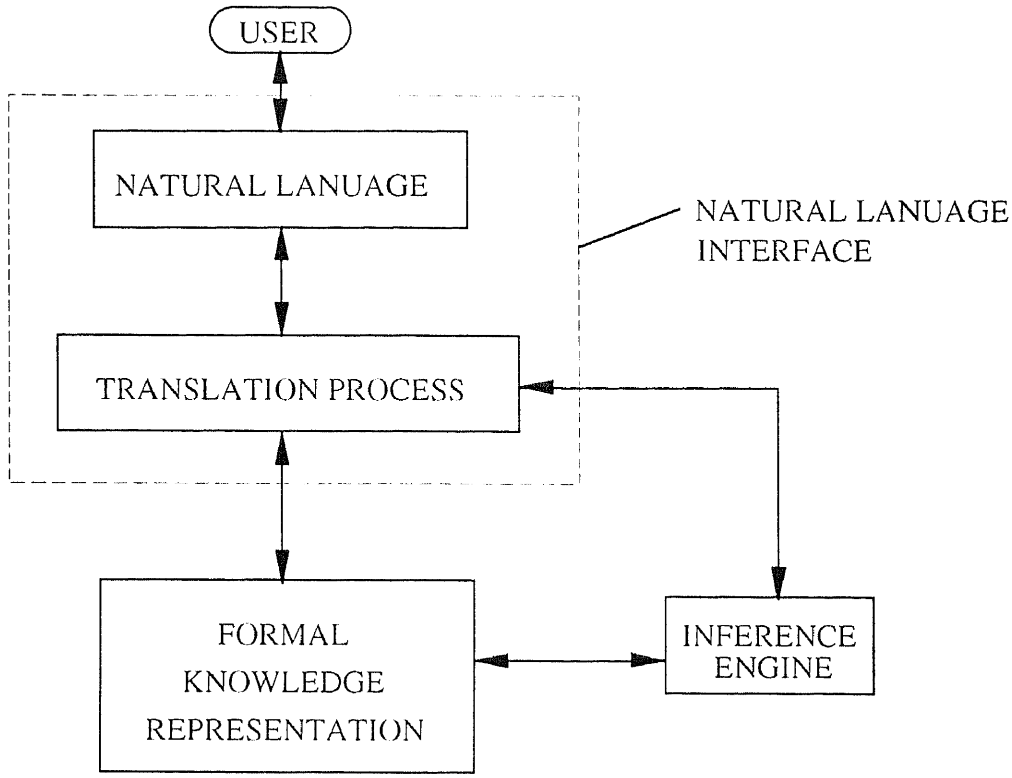


Figure 4.12 A schematic represents the natural language interface.

USER VOICE:	**They are always bugging us about something or other.**
SYSTEM VOICE:	Can you thick of a specific example?
USER VOICE:	**Well, my boyfriend made me come here.**
SYSTEM VOICE:	Your boyfriend made you come here?
USER VOICE:	**He say I'm depressed much of the time.**
SYSTEM VOICE:	I am sorry to hear that you are depressed.
USER VOICE:	**Its true, I am unhappy.**
SYSTEM VOICE:	Do you think coming here will help you not to be unhappy?
USER VOICE:	**I need some help, that much seems certain.**
SYSTEM VOICE:	What about it mean to you if you got some help?
USER VOICE:	**Perhaps I could learn to get along with my mother.**
SYSTEM VOICE:	Tell me more about your family.
and so on....	

Figure 4.13 An example of ELIZA for interpreting the natural language interface.

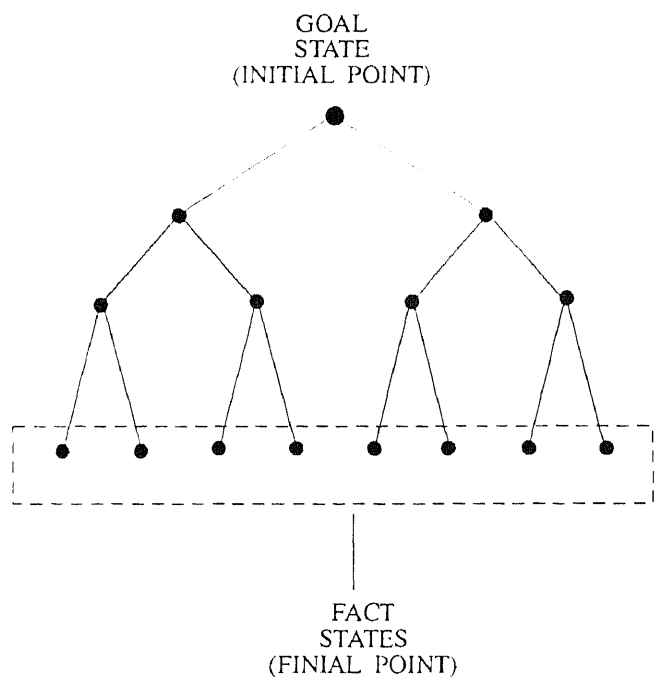


Figure 4.14 Graph for backward chaining search.

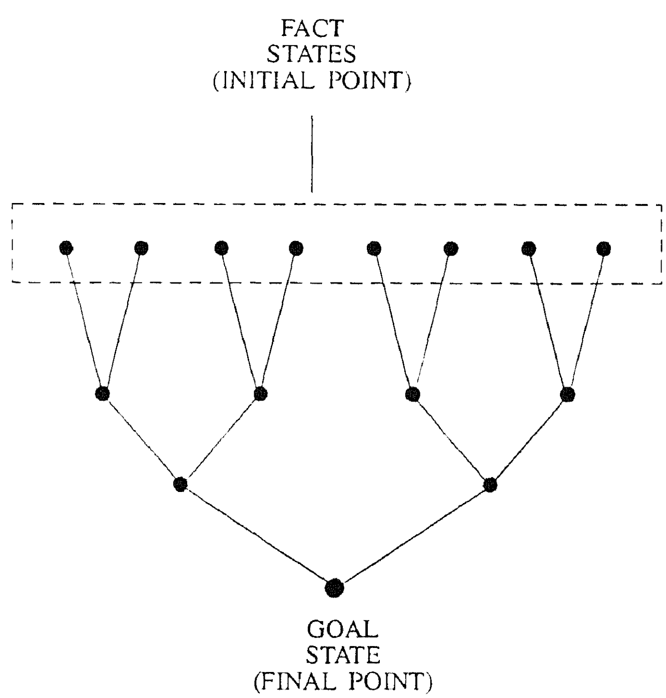


Figure 4.15 Graph for forward chaining search.

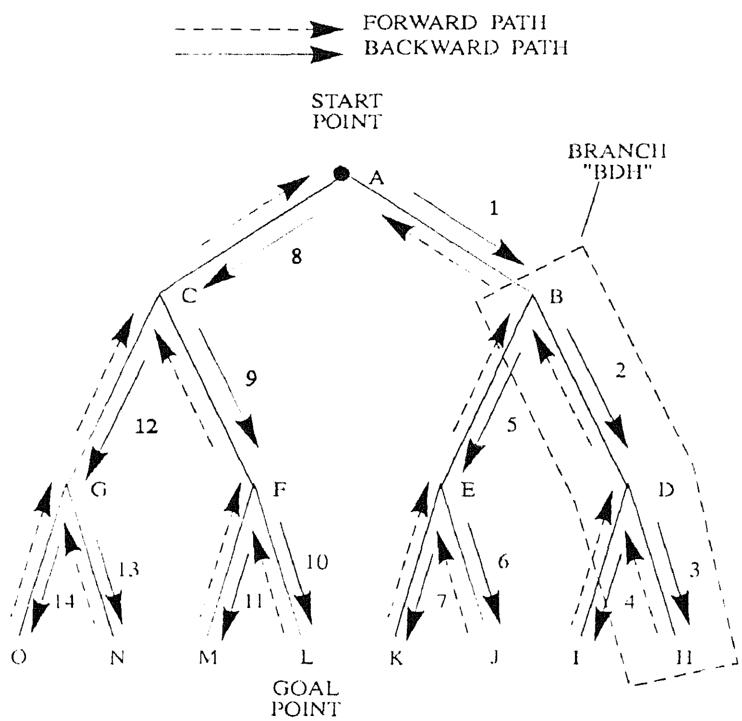


Figure 4.16 Graph for depth-first search.

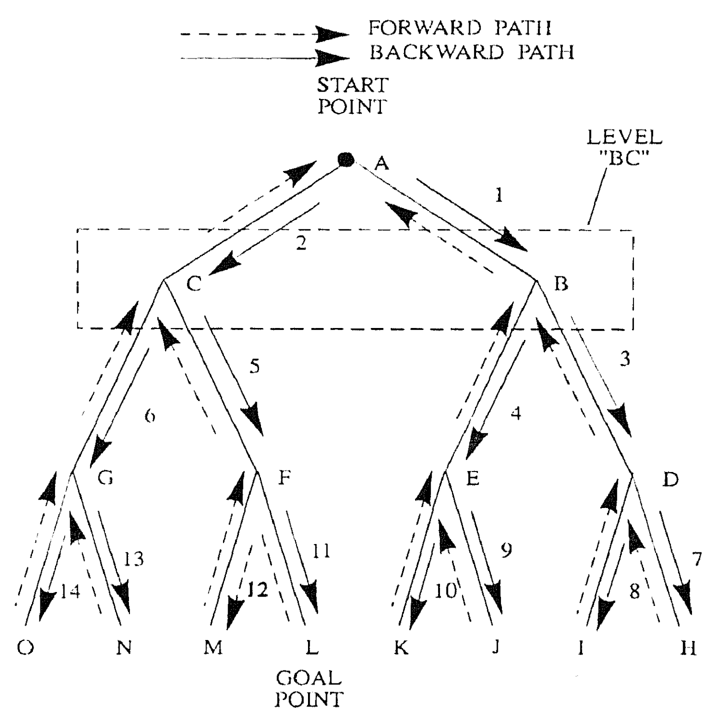


Figure 4.17 Graph for breadth-first search.

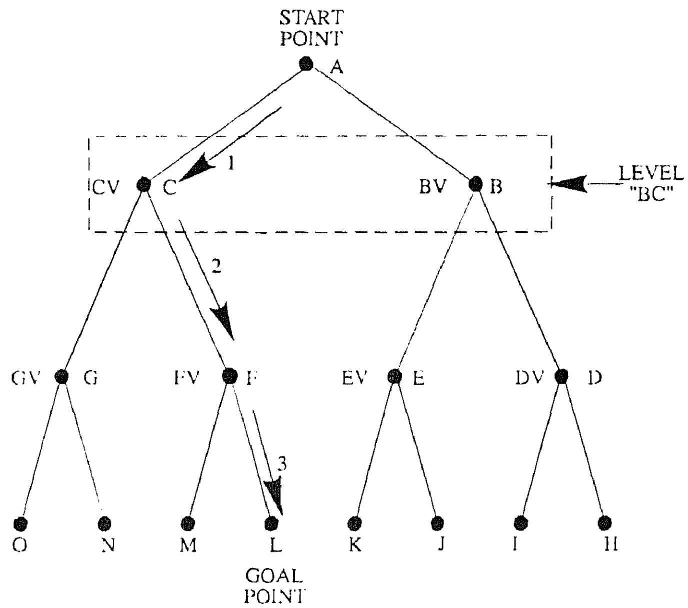


Figure 4.18 Graph for rule-value search.

```
**************************************************************
ACTION: MATERIAL IDENTIFICATION
REASON: SINCE DIFFERENT MATERIALS HAVE DIFFERENT PHYSICAL PROPERTIES
WHICH CAUSE DIFFERENT DEGREES OF INFLUENCE FOR THE SPECIFIC
DEVIATION
THE SYSTEM REQUIRES THE MATERIAL TO BE IDENTIFIED
**************************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 4.19 The explanation screen for identifying the molded material of the N.J.I.T. injection molding expert system.

```
SUGGESTED ACTION:
IS THE CLAMPING FORCE IN MAXIMUM
IF IT IS NOT, PLEASE USE THE MAXIMUM CLAMPING FORCE
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
REASON:
THE INJECTION PRESSURE GREAT THAN THE CLAMPING FORCE IS ONE OF THE
MAJOR CAUSES FOR FLASHING DEVIATION THEREFORE, TO USE THE MAXIMUM
CLAMPING FORCE CAN ENSURE THE INJECTION PRESSURE NOT EXCEED THE
CLAMPING FORCE
PLEASE ENTER ANY KEY TO CONTINUE
C
```

Figure 4.20 The explanation screen for using the corrective action "use the maximum clamping force". of the N.J.I.T. injection molding expert system.

CHAPTER FIVE
DESIGN PROCEDURES OF AN EXPERT SYSTEM

5.1 Introduction

In the design of any expert system, five major steps are required as presented in Figure 5.1. These are,

- identification of the problem domain,
- development of the knowledge engineering,
- development of the inference engine,
- selection of the tool and,
- system testing.

In this chapter, the design procedures of this program are used to illustrate how to design an appropriate expert system. In the first step, the problem domain requires identification. The major consideration is how to select an appropriate problem domain and evaluate its suitability for the expert system.

Once the problem domain has been defined, then the design procedures of knowledge engineering can be employed. In this stage, the major concerns are knowledge acquisition, knowledge analysis, and knowledge representation. In knowledge acquisition, knowledge for the selected problem domain is collected. After the necessary knowledge has been collected, it must then be analyzed. For the analysis the knowledge, it is necessary to understand the contribution of the resolution knowledge to the problem domain. This knowledge assists the inference engine in setting up an appropriate search strategy. The knowledge base then follows the knowledge attribution to select the appropriation knowledge representation model to obtain a readable and understandable format.

In the development stage of the inference engine, the major focus is the development of a suitable search strategy using the resolution procedures in the problem domain. This includes the search path and the search procedures with the knowledge information which is stored in the knowledge base. Furthermore, the inference engine also provides the knowledge acquisition tool for the user to collect the necessary information. This compilation information allows the system to reduce the search time and to execute operations more efficiently.

Once, the inference engine has been developed, the tools selection then can be enacted. The tools selection stage is based on the attribution of knowledge and on the characteristics of the inference engine, and an appropriate tool is selected to codify the resolution knowledge and the inference engine.

After the tool selection procedure has been completed, the system development is accomplished. For a complex problem domain only a partial domain is chosen for the development of a prototype system in the early stages. This prototype system is then evaluated based on whether the expert system is suitable for use in this problem domain. Once the prototype system has been developed, it requires examination, correction, and evaluation. The examination focuses on the reliability of the resolution procedures searching strategy. The correction focuses on the accuracy of the knowledge representation. The evaluation focuses on this prototype system structure and whether it is representative of the entire problem domain.

If the prototype system passes the test and correction, then exploration of the whole problem domain is made. Once the whole system has been completed, the system then allows the potential user to test, to correct, and to evaluate its performance. In the final test, it also includes the reliability of searching strategy, the accuracy of the knowledge representation, and the structure of the system. The user response amends the system. Once the system has passed the final test, system development is completed and can be supplied to the potential user. Even though, the whole system has been developed, the correction and evaluation of the system performance continues with the user responses. This ensures the system performance becomes more efficient. These development procedures will be discussed in detail in the following text.

5.2 Problem Domain

During the development of an expert system, the primary consideration is to select an appropriate problem domain. Since developing an expert system involves much money and manpower, a problem domain, which is inappropriate for the use of expert system techniques, results in failure and lost investment. Therefore, the most important procedure for developing an expert system is the selection of a suitable problem domain. Selecting the problem domain includes two major steps. These are,

- problem domain definition and,
- problem domain evaluation.

5.2.1 Problem Domain Definition

The definition of the problem domain defines the characteristics of the problem domain. The problem domain characteristics guide the system developers in evaluating the suitability of the problem, in collecting the necessary knowledge, and in developing an appropriate inference engine. Evaluating the suitability of the problem allows the system developers to determine whether the problem is suitable for the use of expert system techniques. Collecting the necessary knowledge allows the system developers to devise an accurate knowledge base that provides the necessary information to resolve the problem. Developing an appropriate inference engine allows the system developers to construct an optimal searching strategy and to reach a resolution of the problem.

In this program, the system is designed to resolve the deviations encountered during the injection molding of engineering thermoplastics. This particular domain was chosen because the traditional approach was to resolve the deviation using molding experts. Although, there are many molding experts for the injection molding process, it is difficult for the molding operators to keep up with advances in the deviation resolution methods. Furthermore, the number of molding experts has tended to decline in the injection molding process during recent years. Moreover, due to the complexity of the injection molding process, resolution of deviations is a time consuming process. Thus, in this program, an expert system technique has been used to overcome these limitations.

Here, the major task is resolution of the deviations in the injection molding process. However, there are thousands of combinations of deviations and thermoplastics which must be accounted for in the injection molding process. If all of these are included in the system then it will become excessively large. Searching the knowledge base in the expert system will then consume unrealistic times, and reduce the efficiency of the system. Furthermore, the capability of current computer systems does not allow the design of an expert system that has such an excessive knowledge base. Therefore, in this program, focus is made on the treatment of the engineering thermoplastics that are the most difficult and the most widely used. These materials include acetal copolymer, acetal homopolymer, nylon 6, nylon 6.6, polycarbonate (PC), polybutylene terephthalate (PBT), polyethylene terephthalate (PET), polystyrene (PS), acrylonitrile-butadiene-styrene (ABS), styrene-acrylonitrile (SAN), and high impact polystyrene. Furthermore, the system selects the most commonly encountered deviations of the injection molding process. These deviations are surface ripples, pit marks, splay mark, sink marks, voids, flashing, short shots, warpage, distortion, and delamination.

Therefore, the problem domain of this program may be finally defined as,

Using this expert system resolves the particular deviations encountered during the injection molding process of the specific molded materials. These particular deviations include surface ripples, pit marks, splay marks, sink marks, voids, flashing, short shots, warpage, distortion and, delamination. These specific thermoplastics include acetal copolymer, acetal homopolymer, nylon 6, nylon 6.6, polycarbonate (PC), polybutylene terephthalate (PBT), *polyethylene terephthalate (PET), polystyrene (PS), acrylonitrile-butadiene-styrene (ABS), styrene-acrylonitrile (SAN), and high impact polystyrene.*

5.2.2 Problem Domain Evaluation

Once, the problem domain has been defined, then the following guidelines can lead the expert system developers to determine whether a problem is suitable for the use of an expert system or not. These include,

1. The problem is often encountered in the real world. Prior studies have shown that products manufactured by the injection molding process have a 5% defect rate [Rosato and Rosato, 1987]. This means that if the total cycle time of an injection molding process is one minute, then every 20 minutes the operator adjusts the operating conditions of the machine to resolve aberrations. It is very clear therefore that quality control problems are often encountered during the injection molding process. Furthermore, in 1986 there was a total of 20,000,000 pounds of engineering thermoplastics consumed in the injection molding process [Rosato and Rosato, 1987]. Approximately 1,000,000 pounds were defective. Therefore, it is also clear that quality in the injection molding process is well matched to the use of expert system techniques.
2. Human experts are scarce in this particular field. There is a lack of human expert for the injection molding process. This is due to the sophistication of the process with computer controls and the wide range of material grades processed. The lead time to train an operator to the required level is often too long for the company to gain any benefits. There is also a major decline of human experts for the injection molding process. By employing an expert system for the injection molding process, in place of the well trained operator the manpower problem may be minimized or resolved.

3. A symbolic reasoning technique can be employed for resolving the problem domain. This means that the deviation is resolved using expert system techniques without the use of physical dexterity or perceptual skill. In the injection molding process, for example, most of the resolutions to deviations only require adjustment of the operating conditions. These resolutions can be easily formatted into symbolic reasoning characteristics.
4. The problem may not be resolved using computer simulation. Due to the heuristic searching strategies, an expert system simulation would be preferred if it was possible. However, due to the complexity of the injection molding process, numerical simulation cannot be applied to resolve the deviations. Therefore, the injection molding process becomes an excellent candidate for use of expert system techniques.
5. The knowledge of the problem can be reliably and conveniently obtained. The knowledge used in an expert system is usually provided by human experience and is not found in textbooks. It is important to ensure that the knowledge can be acquired in a convenient and reliable way. For the past 40 years, the reciprocating screw injection molding process has been widely used to manufacture plastics products. The resolution knowledge governing the deviations of the injection molding process have been accumulated and may be stated with a large degree of confidence. Obtaining this resolution knowledge is therefore not too difficult a task, only time consuming.
6. The problem must be of the proper size and scope. Developing an expert system requires a consideration of the capabilities of the expert system technique. If a problem exceeds the capabilities of expert system technique, the system tends to require a huge memory computer system for execution. For instance, if using an expert system develops a system to resolve all plastic processing problems, then the knowledge base requires an enormous

memory to store this knowledge. Furthermore, for a large knowledge demand, searching procedures to achieve problem resolution are more extensive. This influences the system performance. Therefore, in this program, the problem domain is narrowed to particular deviations encountered during the injection molding process of specific engineering thermoplastics.

From all of the above discussions, it can be concluded that the problem domain defined in section 5.2.1 is suitable for using expert system techniques.

5.3 Knowledge Engineering

In developing an expert system, the second procedure is the development of the knowledge engineering. The most important aspect of knowledge engineering is the establishment of an appropriate knowledge base. This knowledge base constitutes the necessary information required to resolve the particular problem domain. To design an appropriate knowledge base involves three major procedures. These are,

- knowledge acquisition,
- knowledge analysis and,
- knowledge representation.

The knowledge base must contain the essential information required to resolve the problem. Analysis of the knowledge allows one to understand the attribution of the knowledge for the problem domain. According to the knowledge attribution, an appropriate knowledge representation can be selected to represent the resolution information. Furthermore, an understanding of the knowledge attribution assists the inference engine in the selection of an appropriate searching strategy for the resolution procedures. Representation of the knowledge allows the knowledge to be easily read, understood, and updated.

5.3.1 Knowledge Acquisition

A completed knowledge base should provide all the necessary information required to resolve a problem. To accomplish this, the knowledge acquisition facility employs a key factor. To acquire the necessary knowledge requires an understanding of characteristics of the knowledge. The characteristics of the knowledge are often determined from the source of the knowledge to be acquired. For this particular case, resolving the deviations for the injection molding process, there are two kinds of knowledge sources. These are,

- from printed matter and,
- from molding experts.

All of these available sources of knowledge have been employed for this program to ensure the necessary information for building a knowledge base was collected.

5.3.1.1 Knowledge Source From Printed Matter. The knowledge sources from printed matter includes textbooks, raw material supplier bulletins, journal papers, and conference proceedings. All of these sources provide a primary advantage, that the knowledge from these sources can be conveniently and easily acquired. For instance, the basic concepts of polymer processing, of the injection molding process, and of mold design, and the computer aided engineering for injection molding can be found the textbooks as presented in Table 5.1. The journal papers and conference proceedings for discussion of the injection molding process include *Society of Plastics Engineering Annual Technical Papers, Polymer Engineering and Science*, *Polymer Process Engineering*, *Journal of Polymer Engineering*, *Polymer Science and Technology*, *Plastics Technology*, *Advances in Polymer Technology*, and *Kunststoffe-German Plastics*. The raw material bulletins can be obtained from the raw material companies. For instances, the acetal copolymer bulletins can obtain from Hoechst Celanese, DuPont, BASF, Mitsubishi, and Polyplastics.

Furthermore, these knowledge sources provide a theoretical understanding of the causes of the specific deviations. This means that knowledge provides an explanation for the causes of the deviation. This allows the system developers to trace the causes and the resolution of the deviation. For instance, for the flashing deviation, according to these knowledge sources, the causes can be divided into several categories as presented in Table 3.7. These causes include material over charge, mold wear, too low a material viscosity, and the instruments reading failure. Then, the system developers can trace the corrective actions to remedy the flashing deviation from among these causes. For instance, material over charge can be resolved by decreasing shot size, or increasing cushion as shown in Table 3.7.

Moreover, these knowledge sources provide the operating range for the particular parameters of the specific material. For instance, for the material acetal copolymer, Celanese M-90, several operating variable ranges can be found from the bulletin as presented in Table 5.2 [Hoechst Celanese Bulletin C3A, 1986]. Such information assists the system developers in determining the resolution sequences for the specific deviation. A detailed discussion for determining the resolution sequences is presented in section 5.4.

However, these knowledge sources have several disadvantages. The primary disadvantage is that these knowledge sources in general are constrained to a particular material. This means that the materials have many different sources to search which requires much time to accomplish. This is due to distinct materials having different material properties. The material properties result in different corrective actions. For instance, the thermal conductivity of acetal copolymer is 5.5×10^{-4} cal/sec cm °C; and the thermal conductivity of PET is $3.3\text{-}3.6 \times 10^{-4}$ cal/sec cm °C. It is clear that decreasing the barrel temperature has more effect for the acetal copolymer than the PET. For example, the flashing deviation may be resolved by decreasing the material temperature which in turn increases the material viscosity and remedies the deviation. Comparing the thermal conductivities of PET and acetal copolymer, shows that decreasing the material temperature of acetal copolymer is easier than decreasing material temperature of PET. This means that flashing deviation resolution for acetal copolymer by decreasing material temperature is probably the best corrective action. However, for PET decreasing material temperature is probably not the best corrective action. It is clear that the resolution knowledge for PET is not necessarily satisfactory for acetal copolymer. Furthermore, the knowledge sources from printed matter in general only provides resolutions for a specific material. Therefore, to complete the knowledge acquisition from printed materials, requires much time to accomplish.

Moreover, the printed matter sources only provide the common resolutions for remedying the problem. For instance, using the knowledge source from raw material supplier bulletins resolves the deviations. If the molded material is acetal copolymer, e.g. Celcon M-90, then the company provides a shooting trouble guide as presented in Table 5.3. This table indicates that these suggested resolutions are only a rough guideline. Furthermore, Table 5.3 does not indicate how to execute these resolutions and what the resolution sequence is. To correct the flashing deviation, for instance, the trouble shooting guide just indicates resolutions as follows,

- decrease material temperature,
- decrease mold temperature,
- decrease injection pressure,
- decrease feed,
- increase clamping force,
- increase cushion,
- decrease gate size and,
- decrease injection time.

From the above listings, it is very difficult to know the sequence in which to execute these corrective actions. This is because the resolution procedures depend on the operating conditions. For instance, if the flashing deviation exists, the material is acetal copolymer, Celcon M-90, and the operating conditions are,

material temperature = 420 °F,

mold temperature = 180 °F,

injection pressure = 5000 psi and,

injection speed = minimum.

From Table 5.3, flashing can be resolved by decreasing the material temperature, the mold temperature, the injection pressure, or the injection speed. However, according to the guide of recommended operating conditions from Hoechst Celanese as shown in Table 5.2, the injection pressure is the minimum recommended injection pressure; the injection speed is the minimum recommended injection speed; the material temperature is the maximum recommended material temperature; and the mold temperature is the average recommended mold temperature. It is very clear that to decrease the material temperature is the best resolution to remedy flashing. In this program, a decision algorithm for determining the sequence of corrective actions will be introduced in the section 5.4, explaining the thought processes involved.

Once, the decrease in material temperature has been recognized the first priority action to resolve this problem is set. The question is how much should the material temperature be decreased? In this program, the amount, that each operating variables require to be adjusted to remedy the deviation, has been input by the molding experts. This will be discussed in detail in the section 5.4.

In this program, the knowledge from printed matter has been acquired and formed into the table format as presented in Tables 3.2 to 3.11. In this acquisition, the major criterion is to understand all possible causes for each of the deviations. Then, for these possible causes, possible corrective actions are listed. The fishbone diagram is used to classify the category of the resolution actions. Classification of the resolution actions allows an understanding of the attributions of the knowledge for resolving the deviation. It also assists the system developer to determine the resolution sequences. A detailed discussion of the fishbone diagram is presented in the section 5.3.2.1 and detailed discussion of the decision algorithm for determining resolution sequences is presented in the section 5.4.2.

5.3.1.2 Knowledge Source From Molding Experts. The other primary knowledge source to resolve the deviations of the injection molding process, comes from the molding experts. The traditional approach would be to resolve the problem with the assistance of a molding expert. If the thought process is studied it can be seen that it is really a question and answer sequence, or an interview. This is the basis of an interactive expert system for resolution of the deviation. For instance, to resolve flashing the interview would be as shown in Figure 5.2.

From the above interview procedures, it can be found that, prior the resolution of the problem, the molding expert collects all the necessary information from the client. This information includes the material type, the material grade, and the operating conditions. In this program, this information is termed declarative knowledge. It allows the molding experts to judge the resolution procedures. Furthermore, prior to adjust the operating variables for resolving the deviation, the molding expert will recommend that the client conduct the corrective actions such as checking mold surface for sticking material or contamination exist, and checking instrument readings. These corrective actions, in this program, are termed method corrective actions. These method corrective actions in general are easily ignored.

After these method corrective actions have been exhausted, the molding expert, according to his experience, then recommends to the client adjusting the operating variables for resolving the deviation. These are termed operating variable corrective actions. It has been shown that the molding expert recommends that the client resolve the deviation step by step. In each resolution step, the molding expert expects to obtain a response from the client. From this reply, if the deviation is not completely eliminated, then the molding expert can base the next response on this negative to conclude the resolution. For the above case, for example, the molding expert first quotes his resolution for adjusting operating variables as "*use the maximum clamping force*". However, the reply from the client for this resolution is only that it improved the condition of flashing deviations. From this response, the molding expert recognized that a higher material pressure probably is the major cause for flashing deviations. However, at that time, the clamping force had already reached the maximum recommended. Therefore, the molding expert realized that the next most possible resolution is to decrease the injection pressure. This procedure was repeated until the problem was resolved.

According to the example listed in above, in this program, a "question-answer" form, as presented in Table 5.4, is prepared when the molding expert is interviewed. This "question-answer" form consists of three major elements. There is information on the molded material, information on the operating conditions, and the question procedures. The molded material information includes the type, the manufacturer, and the material grade. The operating condition information contains all the quantities of the operating variables and the mold dimensions. The question procedures are based on the experience of interviewing to conclude several significant statements that are the key factors to induce the resolution from the molding expert. These statements include the type of deviation and the response for consequence. The consequence is constrained into three types. These are "corrected", "improved", and "not improved".

After the knowledge has been acquired from the molding expert, this knowledge and the knowledge from the printed matter will be used in the *fishbone* diagrams to classify the category of resolution actions, and be used with the *Pareto principle* to assist in the determination of the correlative weighting factors between the operating variables and the deviation. In the section 5.3.2.1, the fishbone diagram is discussed in detail. In the section 5.3.2.2, the Pareto principle is discussed in detail.

5.3.2 Knowledge Analysis

Once all the necessary knowledge for resolving the deviations has been acquired, the next step for developing a knowledge base is to classify and to analyze this knowledge as corrective actions. Classification of these corrective actions allows an understanding of the characteristics of the corrective actions. Understanding the characteristics of the corrective actions assists in analyzing the degree of difficulty of the resolution actions. To analyze the degree of difficulty of the corrective actions assists in determining the resolution sequences. In this program, the fishbone diagram is introduced to classify the corrective actions. The Pareto principle is introduced to determine the correlative weighting factors between the operating variables and the deviations.

5.3.2.1 Fishbone Diagrams. The major function of the fishbone diagram is to classify the corrective actions and the effect of particular deviations. The technique of the fishbone diagram was devised by Kaoru Ishikawa [Owen, 1989], who first applied the fishbone diagram to analyze the causes and effects for forging steel. Here, the fishbone diagram is used to illustrate the analyzed procedures of the knowledge for resolving the deviation in the injection molding process.

The first stage in the development of a fishbone diagram is to write out the deviation as the effect at the end of a main spine. Once the effect is confirmed, the major branches are added to the main spine as presented in Figure 5.3. These branches correspond to the major categories of corrective actions. In this program, for example, according to the attribution of the corrective actions, five major categories are identified. These are the machines, the materials, the molds, the methods, and the measurements. Therefore, this fishbone diagram is also termed a 5M fishbone diagram.

The classification of the machines includes all the corrective actions associated with the machines. As discussed in the section 2.4.1, these operating variables are also classified into four different groups. These groups are temperatures, pressures, times, and displacements. The corrective actions from material properties are annotated into the category of the materials. According the discussion in the section 2.4.2, these material properties are categorized into the physical properties, the mechanical properties, the electric properties, the optical properties, the chemical properties, and the material operating variables. However, the mechanical properties, the electric properties, the optical properties, and the chemical properties do not to influence the deviations during the processing. Therefore, in the 5M fishbone diagram, the material category only lists two items that influence the deviation. These are the physical properties and the material operating variables.

The category of the molds includes the corrective actions concerning the gate systems, the runner systems, the cooling systems, and the venting systems. According to the discussion in the section 2.4.3, these corrective actions include both layout and dimensions. The group of the methods contains the corrective actions resulting from inappropriate operating procedures, such as material handling, mold operation, and machines operating. Inappropriate material handling could mean that hopper is empty. Improper mold operation means foreign components in the mold surface, unsuitable ejection pin layout, and mold wear. Incorrect machine operation includes inadequate nozzle size, and configuration. The measurements category involves all instrument readings from the machine. These instruments include the temperature indicators, the pressure transducers, the screw speed indicators, and the screw position indicators.

Based on the above discussion, a major constructure of the fishbone diagram for the flashing deviation is shown in Figure 5.4. After the major construction of the fishbone diagram is completed, detailed corrective actions are now added to each major category. For example, if the effect is flashing, and the major category is machines, then, the detailed corrective actions from temperatures are added into the sub-group of temperatures. According to Table 3.8, flashing may be caused by high material temperature, and high mold temperature. To remedy these causes, one can decrease barrel temperature, decrease nozzle temperature, or decrease mold temperature. Based on these procedures and the classification in Table 3.8, the other sub-groups in the machines and the other major categories can be entirely constructed. A completed construction of the machines category is presented in Figure 5.5, and, a completed structure of a fishbone diagram for short shots in Figure 5.6. From Figure 5.7 to Figure 5.15 the fishbone diagrams representing each deviation such as surface ripples, pit marks, splay marks, sink marks, voids, short shots, warpage, distortion, and delamination are represented.

Once the entire fishbone diagram is complete, the next step based on the characteristics of the corrective action, is to determine the degree of difficulty of the corrective action. In this program, the resolution procedures are dependent on the degree of difficulty of the corrective actions and the input of the experts. The first priority for resolving the deviation is to use method corrective actions which are simply applied. These corrective actions in general may be easy to observe and to remedy. These corrective actions include the resolved actions from the categories of measurements and methods. The sequence for resolving method correction actions are based on the input of molding experts and the degree of difficulty of resolution to determine the sequence. This sequence is rank ordered as material handling, mold operation, machine operation, and instrument readings respectively.

The second priority for resolving deviations is resolution by adjustment of the controllers, or the devices, and are termed operating variable corrective actions. The operating variable corrective actions include the actions from the category of machines and part of the category of materials. The actions from the category of the materials includes the material operating variables such as the moisture content and the regrind rate of the material. The order of the corrective actions is based on the priority weighting factors of these resolutions. A detailed discussion of these determinations for priority weighting factors is presented in section 5.4.2.

The third priority for resolving deviations is termed the mold corrective actions. This group includes the corrective actions for the category of the molds in the fishbone diagrams. In this program, these corrective actions will be employed only if the method corrective actions and the operation variables correction actions have failed, and the deviation still exists. To employ these corrective actions, modifications to the gate system, the runner system, the cooling system, and the venting system may be required. These modifications involve layouts and dimensions. However, to indicate how to modify the layout of the mold is difficult. Furthermore, in this program, these parameters are assumed to have been correctly designed into the mold. Therefore, in this program, the modification of the mold layout is placed low in priority, and only the actions for changing the dimensions are considered for resolving the deviation. The priority for changing these dimensions is dependent on the recommendations from the molding experts.

In this program, if the mold corrective actions still cannot resolve the problem, then the molded material must be modified. This is termed material corrective actions. However, due to changes in molded material, the material properties will be altered too. When the material properties are changed, the functional performance of the product is probably altered. To avoid this situation in this system when the material is changed, there is a caution statement that appears on the screen to warn the user. The caution statement is as follows,

> "*When you change the molded material, please carefully consider the altered material properties such as the mechanical properties, the electric properties, the optical properties, and the chemical properties. Checking these properties ensures that these properties are suitable for the functional performance of the product.*"

5.3.2.2 The Pareto Principle. The primary application of the Pareto principle in this program is help to determine the correlative weighting factors between the operating variables and a particular deviation. These correlative weighting factors are then employed in the decision algorithm, that will be discussed in detail in section 5.4.2. This concludes the priority weighting factors of the resolution sequences for the operating variable corrective actions.

The Pareto principle was developed by an Italian economist, Vilfredo Pareto [Owen, 1989], who studied the distribution of income and wealth in Italy at the turn of the century. When the Pareto principle is applied in the corrective actions - effects analysis, it can assist in determining the attribution of these actions to the corresponding effects. The effect then resolved is based on the contribution of corrective actions to tackle the causes of the deviation step by step. The application of the Pareto principle in this program is used to illustrate the Pareto principle development procedures.

The first development procedure of the Pareto principle is to list all the possible causes for the specific effect. In this program, the deviation for acetal copolymer from Hoechst Celanese, Celcon M-90, for all possible operating condition corrective actions for resolving the deviation include,

- increase or decrease barrel temperature,
- increase or decrease nozzle temperature,
- increase or decrease mold temperature,
- increase or decrease injection pressure,
- increase or decrease injection speed,
- increase or decrease injection time,
- increase or decrease total cycle time,
- increase or decrease cooling time,
- increase or decrease mold close time,
- increase or decrease mold open time,
- increase or decrease decompression,
- increase or decrease regrind rate,

- increase or decrease screw speed,
- increase or decrease shot size,
- increase or decrease cushion and,
- decrease material moisture.

Once the all possible operating variable corrective actions for resolving the deviation are listed, these possible correction actions are formed as a check sheet as presented in Table 5.5. At this time, for the flashing deviations, for instance, the data for each corrective action is now added. These data are the frequency of each corrective action employed successfully in resolving flashing deviations. These are the results obtained from the interview of the molding expert. When the molding experts were interviewed, they were asked for every 100 cases of flashing deviation per mold for three different types of mold. The frequency of each corrective actions successful to remedy the problem was required. The information from Table 5.5 can now be condensed and rearranged in order of frequency as presented in Table 5.6.

A Pareto diagram can also now be drawn. The Pareto diagram consists of a series of rectangles. The height of these rectangles indicates the frequencies of occurrence of each corrective actions. As presented in Figure 5.16, it shows a variety of corrective actions from operating variables to resolve flashing deviations. In this case, it indicates the resolution sequences of flashing deviations can be rank ordered as follows,

1. decrease injection pressure,
2. decrease injection speed,
3. decrease shot size,
4. increase cushion,
5. increase injection time,
6. decrease barrel temperature,
7. decrease nozzle temperature,
8. decrease mold temperature,

9. decrease screw speed and,

10. decrease regrind rate.

However, in Figure 5.16, it is not so obvious what the relative contribution for resolving the deviation by each corrective action is. To indicate these relative contributions, the cumulative percentage and the total frequencies percentage of each corrective actions as presented in Table 5.7 is calculated. A new Pareto diagram can now be drawn where the Y-axis is the percentage of the total frequency as presented in Figure 5.17. In addition, a cumulative line can be added into Figure 5.17 as presented in Figure 5.18. Now, in Figure 5.18, the Y-axis is interpreted as a linear representation for the Pareto diagram and a cumulative figure for the cumulative line. This cumulative line makes it easier to show the contributions made by the different features.

According to the above procedures, the cumulative percentage and the total frequency percentages of other deviations can now be calculated. Table 5.8 to 5.16 represent the cumulative percentage and the total frequency percentage of the deviation for surface ripples, pit marks, splay marks, sink marks, voids, short shots, warpage, distortion, and delamination respectively.

Once the cumulative percentage and the total frequency percentage of each deviation had been recognized, the correlative weighting factors between the operating variables and each of the deviations are now calculated. In this program, the corrective action, which has the highest total frequency percentage is assumed to have a correlative weighting factor for operating variables equal to 0.5. Comparing with this number, the other correlative weighting factor, $V_{i,j}$, between the operating variables, j, and the specific deviation i, are given as,

$$V_{i,j} = 0.5 \times \frac{F_{i,j}}{F_{i,h}} \tag{5.1}$$

where,

$V_{i,j}$ is the correction weighting factor of operating variable, j, for the specific deviation i,

$F_{i,h}$ is the highest total frequency percentage of correspond correction action for the specific deviation, i, and,

$F_{i,j}$ is the total frequency percentage of correspond correction action, j, for the specific deviation, i.

A completed matrix of correlative weighting factors for each operating variable and each deviation for Hoechst Celanese Celcon M90, acetal copolymer, is presented in Table 5.17. In the section 5.4.2, these correlative weighting factors will be employed into the decision algorithm to calculate the priority corrective weighting factors for each operating variable.

5.3.3 Knowledge Representation

The primary role of the knowledge representation is to prevent ambiguous resolution in the problem resolving procedures. Performing that depends on the characteristics of the problem domain and employing a suitable model of knowledge representation. Mylopoulos and Levesque [1984] classified the knowledge representation model into four categories. These are,

- networks representation model,
- predicate logic representation model,
- structure representation model and,
- procedural representation model.

In this program, a hybrid knowledge representation, that combines with structure representation and procedural representation, is employed into this system. However, in the following discussion, these representation models will be introduced individually.

5.3.3.1 Network Representation Model. The network representation model uses a scheme for representing abstract relations between each object in a problem domain. It consists of a set of nodes that represent the objects and a set of arcs, that represent the relations or associations among those objects. An example of network representation for defining the properties of snow is presented in Figure 5.19. In this example, the nodes represent the main objects, snow, and the sub-objects, snow properties, such as water, soft, slippery, white, and cold. The arcs describe the relationships between the main object and property objects. For instance, the arc, "temperature is", describes the relation between the main object, "snow", and sub-object, "cold". It represents the temperature property of snow as cold.

The primary advantage of network representation is that the nodes and the arcs can be easily added, deleted, or modified. However, to employ a network representation model, the problem domain requires a formal definitive structure. This is difficult to accomplish when a problem domain, such as the problem domain in this program, is very complex. Therefore, network representation is not considered in this program.

5.3.3.2 Predicate Logic Representation Model. Predicate logic representation uses expressions in formal logic to represent a knowledge base. These logical expressions use predicate calculus to induce inference rules or proof procedures by asserting the truthfulness or falseness of propositional statements. When a problem domain requires to compact and to unify theory, such as is the case in physics, chemistry, mathematics, and other scientific fields, it is suitable to use a predicate logic representation model. An animal identification system is a case that uses predicate logic representation. A set of animal attributions is presented in Table 5.18. These attributions include the animal characteristics such as diet, size, color, habitat, and species. According to these attributes, the inference rules can now be employed as follows,

Object "DIET" is "MEAT" $\rightarrow$ TRUE, and;

Object "SIZE" is "LARGE" → TRUE, and;

Object "COLOR" is "STRIPED" → TRUE, and;

Object "HABITAT" is "JUNGLE" → TRUE.

Then, it can concluded that the object is a species "TIGER".

The advantages of predicate logic representation include,

- these logical expressions can be readily understandable,
- these logical expressions can be added, deleted, or modified without affecting other logical expressions and,
- the new logical expression can be derived by the old logical expression.

However, the predicate logic representation is difficult to manage in a large knowledge base. This is due to the limitation of the organizational structure for the predicate logic representation. Furthermore, in the predicate logic representation it is difficult to include procedural and heuristic knowledge. In this program, since the problem domain requires the use of procedural knowledge and heuristic knowledge to resolve the deviation, the predicate logic representation is also eliminated from consideration.

5.3.3.3 Structure Representation Model. In the structure representation, knowledge information is grouped together into a slot. This slot describes the attributions of the objects, the class of object, the situation, the action, or the event. This knowledge representation, in general, is used for representing declarative knowledge. Declarative knowledge is knowledge that cannot be executed directly. However, it can be stored and retrieved as required for assisting the inference engine in deciding the resolution searching strategy for resolving the deviation.

This knowledge representation is organized into a tabular format associated with the object. This organization forces experts to organize their expert knowledge about the specific problem domain into chunks of data. Psychologists believe that when experts recall the knowledge about a specific object, all the typical attributes of the object are recalled at the same time as a group. These groups of object attributes are termed a frame or a slot. When experts resolve a particular problem, they seek the relevant frames that contain the information required to resolve the deviation. If the frame is not a resolution for a given problem, then another frame is signed. Structure representation is simulated the organization of the expertise knowledge of the experts to represent the knowledge.

The advantages of structure representation include,

- it is arranged in a hierarchical manner such that can inherit relationships from other structure representation,
- it facilitates faster searches of the knowledge base through the concise and compact representation information and,
- it permits the representation of inheritance relationships among objects.

In this program, the structure representation is used to represent the declarative knowledge such as,

- the material types, as presented in Table 5.19,
- the material manufacturers, as presented in Table 5.20,
- the material grade, as presented in Table 5.21,
- the recommended operating conditions of the specific material, as presented in Table 5.22,
- the correlative weighting factors between the inherent physical properties and the operating variables, as presented in Table 5.23,
- the correlative weighting factors between the influence physical properties and the operating variables, as presented in Table 5.24,
- the correlative weighting factors between the influencing physical properties and the specific deviation, as presented in Table 5.25,
- the correlative weighting factors between the operating variables and the specific deviation, as presented in Table 5.26,
- the deviations types for the problem domain, as presented in Table 5.27 and,

- the resolution knowledge for the particular deviation, as presented in Table 5.28.

These frames are stored individually in the program. They influence the firing of the inference engine. When the inference engine is fired, the rule knowledge is employed to connect relationships between each of the relevant frames. The rule knowledge is stored in the format of the procedural representation model. The procedural representation will be introduced in the next section.

5.3.3.4 Procedural Representation Model. Procedural representation sometimes is also termed as rule representation. It is widely used for problem domains such as to resolve the conclusion, or to resolve the premises in the order. It provides a formal way to represent recommendations, directives, or strategies. The knowledge, that uses procedural representation is set up in the rules, "IF CONDITIONS THEN ACTIONS", to link the relationship between the objects and the objects, or the objects and the attributions.

The advantages of procedural representation include,

- it can be easily be removed, added, or updated,
- it can be linked by the logical connective AND and OR,
- it provides a straight-forward representation of knowledge that is easy to interpret and,
- it is structured in a way similar to the way people rationalize resolution of the deviations.

However, the primary disadvantage of procedural representation is the requirement for a very efficient inference engine to determine appropriate rules during the expert system consultation.

In this program the rule knowledge that is used in a procedural representation model includes,

- determination of the material manufacturers, as presented in Table 5.29,

- determination of the material grades, as presented in Table 5.30,
- determination of the recommended molding conditions, as presented in Table 5.31,
- determination of the confirmation of the recommended operating conditions, as presented in Table 5.32,
- determination of the confirmation for the user operating conditions, as presented in Table 5.33,
- determination of the confirmation for the correlation weighting factors, as presented in Table 5.34,
- determination of the deviation resolution knowledge, as presented in Table 5.35, and
- determination of the remedied degree for resolution actions, as presented in Table 5.36.

To determine when the rules are employed depends on the searching strategy of the inference engine which is discussed in the following section.

5.4 Inference Engine

In an expert system, the inference engine is a facility which simulates the expert reasoning skills required for resolving a particular problem domain. It also controls the performance of the user interface between the user and the system. To perform this task, the inference engine drives the searching strategy, which is embedded in the system, and the searches for a suitable resolution from the knowledge base. In the following text, the discussion of the searching strategy, which applies in this program, and the decision algorithms for determining the priority weighting factors of the operating variable corrective actions will be presented.

5.4.1 Searching Strategy

As mentioned in the section 4.2.3, the type of searching strategy includes backward-chaining searches, forward-chaining searches, depth-first searches, breadth-first searches, and rule-value searches. A hybrid type searching strategy is employed in this program. This searching strategy is combined with the backward chaining search and the rule value search.

In this program, the searching strategy is classified into two levels as shown in Figure 5.20. The first level is the searching strategy for inducing the declarative knowledge. The backward chaining search is employed in this level. The second level is the searching strategy for determining the priority of the resolution actions. It uses a rule value search to determine the priority of the resolution actions.

The searching strategy of the declarative knowledge is ordered as presented in Figure 5.20. This sequence is based on the expertise of interviewing the molding experts to conclude its orders. This sequence is,

- declare the molded material,
- declare the manufacturer of molded material,
- declare the grade of molded material,
- declare the recommended operating conditions,
- declare the confirmation of recommended operating conditions,
- declare the operation conditions of the user,
- declare the confirmation of the operation conditions of the user,
- declare the confirmation of the correlation weighting factors between the inherent physical properties and the operating variables,
- declare the confirmation of the correlation weighting factors between the influencing physical properties and the operating variables,
- declare the type of deviation,

- declare the confirmation of the correlation weighting factors between the influencing physical properties and the deviation, and,
- declare the confirmation of the correlation weighting factors between the operating variables and the deviation.

When the declarative knowledge is induced, it is governed by the rules embedded in the rule knowledge, as presented from Tables 5.29 to 5.36, to determine which declaration knowledge will be employed to decide the appropriate knowledge.

Once the declarative knowledge has been completely induced, the inference engine then moves into the second level searching strategy, the rule value search, to determine the priority of the resolution actions. As the discussion in section 5.3.2.1 explains the resolution actions are classified into four different priority levels and are ordered as shown in Figure 5.20. These are,

1. the method corrective actions,
2. the operating variable corrective actions,
3. the mold corrective actions and,
4. the material corrective actions.

In this program, the method corrective actions always have higher priority than the other correction actions in resolving the deviation. These method corrective actions will be assigned to a rule value to determine the order of the corrective sequences. These rule values of the method corrective actions are based on the advice of the experts and their degree of difficulty for resolving the deviation as presented in Table 5.37. According to these rule values, ranking is made from the higher value to the lower value, then the search strategy induces the appropriate corrective resolution from the knowledge base to address the deviation.

When the method corrective actions are employed to resolve the deviation, the resolution result is constrained to two answers. These are "DEVIATION CORRECTED", or "DEVIATION NOT CORRECTED". This is because the method corrective actions are the "*absolute corrective actions*". The term "absolute corrective actions" means that when the action is employed, it is employed only once during the entire resolution procedures. For instance, in the flashing deviation resolution procedure, when the corrective action, "USE MAXIMUM CLAMPING FORCE," is employed to resolve the flashing deviation, and the deviation is still not to be resolved; then the system will prompt the resolution procedures to another method corrective action or other type of corrective actions; and the corrective action ,"USE MAXIMUM CLAMPING FORCE"; will never be employed again. Therefore, the response of the resolution result of the method corrective action is constrained in two ways such as "DEVIATION CORRECTED " and "DEVIATION NOT CORRECTED".

Once the method corrective actions have been exhausted, the search strategy then focuses on the operating variable corrective actions. These corrective actions will be assigned the rule value to determine their order of activation. The determination of these rule values is governed by the decision algorithm that will be discussed in section 5.4.2. The search strategy is then based on a comparison of the rule value to induce the resolved actions step by step.

During the resolution procedures of the operation corrective actions, the response of the resolution results is indicated in three ways. These are "DEVIATION CORRECTED", "DEVIATION IMPROVED", and "DEVIATION NOT IMPROVED". These corrective actions will be repeated until the operating variables have reached the extreme of their range and termed as *conditional corrective actions*. For example, in the flashing deviation resolution procedure, when the operating corrective action, "DECREASE BARREL TEMPERATURE", is employed to resolve the flashing deviation, the decrease will be indicated only up to a recommended limit. The calculation of this recommended limit is based on the decision algorithm in section 5.4.2.

When this corrective action has been applied to resolve the problem, and the problem still cannot be resolved, then the searching strategy will be based on the user response to update the correlative weighting factors for the operating variables and the operating conditions. The update procedures of the correlative weighting factors and the operating variables will be discussed in the next section. Based on these updated correlative weighting factors and the operating conditions, the decision algorithm re-calculates the priority weighting factors of each of the operating variables. The searching strategy is then based on the new calculation to search the new corrective action. This procedure will be ended when all the priority weighting factors of the operating variables reach zero. When these priority weighting factors have reached zero, then there are no further resolutions in the operating variable corrective actions. The searching strategy then jumps to the next priority level, the mold corrective actions level.

If the operating variable corrective actions still cannot remedy the deviation, then the search strategy will employ the mold corrective actions to resolve the deviation. The rule values of the mold corrective actions are the same as the method correction actions that are determined by the experts recommendation and their influence degree for resolving the deviation as presented in Table 5.38. In this program, as mentioned in section 5.3.2.1, the layouts of the mold system are assumed to have been correctly performed prior to the injection molding process. Therefore, the mold corrective actions induce only dimensional changes of the mold system.

Furthermore, since the dimensioned change of the mold system are very complicated procedures, in this program, the quantities changed are always recommended to be changed to the extreme recommended value. For instance, when the flashing deviation occurs, and the corrective action, "DECREASE GATE SIZE", is employed for resolving the deviation; the system will then suggest the corrective action as "DECREASE GATE SIZE TO 0.1 in". The number, "0.1 in", is the minimum quantity allowed by the recommended operating conditions, as presented in Table 5.2, for Hoechst Celanese acetal copolymer, Celcon M-90.

The mold corrective actions are the same as the method corrective actions in that they are the "absolute corrective actions". Therefore, in the level of corrective actions, the user response is constrained into two ways such as "DEVIATION CORRECTED ", and "DEVIATION NOT CORRECTED".

Once the mold corrective actions have been exhausted, the resolution searching strategy will then jump to the material corrective actions. In this stage, the suggested action is based on the knowledge analysis in section 5.3.2.1 to indicate the kind of material which should to be substituted. For instance, in the flashing deviation, when the material corrective action is employed, according to the fishbone diagram in Figure 5.6, for the material corrective action, for Celcon M-90, the prompts "CHANGE TO A LOWER THERMAL CONDUCTIVITY MATERIAL" is given.

Here, the question is "what is a higher thermal conductivity material?". To avoid this ambiguous answer, the system will indicate the physical properties of the original molded material to the user. For instance, in the flashing deviation, when the material corrective action, "CHANGE TO A LOWER CONDUCTIVITY MATERIAL", is employed, the corrective action will be presented as follows,

> *"PLEASE CHANGE MATERIAL TO ONE WITH A LOWER THERMAL CONDUCTIVITY. THE ORIGINAL CONDUCTIVITY OF THE MATERIAL, ACETAL COPOLYMER, IS 5.5 × 10^{-4} cal/sec cm °C"*

Based on this comparison the user can search for an appropriate substitute material. However, due to the change in molded material, the material properties will be altered too. When the material properties are changed, the functional performance of the product is also altered. To avoid this situation, in this program, when a material corrective action is employed, there is a caution statement that appears on the screen to warn the user. The caution statement is presented as follows,

> *"When you change the molded material, please carefully consider the altered material properties such as the mechanical properties, the electric properties,*

the optical properties, and the chemical properties. Checking these properties ensures that these properties are suitable for the functional performance of the product."

In the final searching strategy stage, if all above corrective actions cannot resolve the problem, the system will then present a recommendation to the user. This recommendation is presented as follows,

"There is no further correction action available. Please consult with the molding experts or the raw material supplier to resolve the problem".

5.4.2 Decision Algorithms

As discussed in section 2.4, the parameters creating deviations may be classified into two different types. These are operating variables of the molding machine and physical properties of the thermoplastic which is being molded. The operating variables are the parameters which can be directly controlled by adjustments to the machine. These include temperatures, pressures, times, and displacements. The physical properties can be classified into two categories. These are the parameters which vary with the changes of the operating variables and cannot be altered with adjustments to the machine. However, these parameters also control the degree of influence on the deviations. These physical properties include shear rate, shear stress, and viscosity. These physical properties are termed influencing physical properties. The another category of the physical properties is the parameters which are the constants of the material properties and not effect the degree of influence on the deviation. However, these properties influence the degree of difficulty for adjusting the operating variables. These include thermal conductivity, thermal diffusivity, specific volume, and specific heat. These physical properties are termed inherent physical properties. Complete listings of the operating variables and physical properties are presented in Table 2.3.

Since, each operating variables and each influencing physical property has a different degree of influence on the deviation; and each influencing physical properties and each inherent physical property have also a different degree of influence on the operating variables, therefore, the correlative weighting factors are assigned to indicate the degrees of influence. These correlative weighting factors can be classified as the influence of,

1. operating variables, j, on influencing physical properties, m, $A_{j,m}$, as shown in Table 5.39,
2. operating variables, j, on inherent physical properties, n, $B_{j,n}$ as shown in Table 5.40,
3. influencing physical properties, m, on deviations, i, $E_{i,m}$, as shown in Table 5.41 and,
4. operating variables, i, on deviations, i, $V_{i,j}$ as shown in Table 5.42.

Based on the above categorization, these correlative weighting factors are summarized in Table 5.39 to Table 5.42. These correlative weighting factors range between 0 to 1, and can also be positive or negative. The value of these correlative weighting factors is provided from the expert or from mathematical equations.

For the correlative weighting factors between influencing physical properties and operating variables, A, and inherent physical properties and operating variables, B, if the value is equal to 0, then, these two parameters do not have any influence on each other. If the value is equal to 1, then, these two parameters have a most significant influence on each other. If the correlative weighting is a positive, then, these parameters will change proportionally. If the correlative weighting is negative, these parameters will change inversely.

For the correlative weighting factors between the influencing physical properties and the deviations, E, the range is also from 1 to 0, and the definitions are the same as the above correlative weighting factors, A and B. But, the positive and negative signs for these correlative weighting factors have a totally different meaning from the above weighting factors. If the correlative weighting factor has a positive value, then, the influencing physical properties must be increased to resolve the deviation. Alternatively, if it is negative, the influencing physical properties must be decreased to resolve the deviation. The same is true for the correlative weighting factors between operating variables and deviations, V.

In this program, since the correlative weighting factors cannot be readily deduced from mathematical equations the values of these correlative weighting factors are initiated by the advice of the experts. To prevent the incorrect input data from the experts, there are two mechanisms embedded in the system. There are the confirmation mechanism of these corrective weighting factors, and the self-learning mechanism of the system. The confirmation mechanism allows the users based on their own expertise to modify these correlative weighting factors. Furthermore, the self-learning mechanism of this program, which is based on the response of the resolution actions from the user, modifies these correlative weighting factors. These modifications allow the correlative weighting factors to become more significant in their influence degree for resolving the problem. The self-learning mechanism will be discussed in the next section. The performance of confirmation mechanism and self-learning mechanism will be presented in the next chapter.

However, in this program, the primary influence correlative weighting factors, the weighting factors between the operating variables, j, and the deviations, i, $V_{i,j}$ is given by Eq. 5.1. As discussed in section 5.3.2.2, the Pareto principle was employed to analyze the input data of the expert. Based on this analysis, the correlative weighting factors $V_{i,j}$, can be more significant in expressing the relationships between the operating variables and the deviation.

According to the above definitions for correlative weighting factors, the priority weighting factors, $R_{i,j}$, for operating variables, j, on the particular deviation, i, are calculated in the following manner.

Since, the correlative weighting factors $A_{j,m}$ are affected by both the correlative weighting factors, $E_{i,m}$, and $V_{i,j}$, and, the correlative weighting factors $E_{i,m}$ and $V_{i,j}$ depend on the particular deviation, then, $A_{j,m}$ will be modified by $E_{i,m}$ and $V_{i,j}$ to satisfy the particular deviations, i, as follows:

Firstly, $A_{j,m}$, is modified by $E_{i,m}$, and is given as,

$$A'_{m,j,i} = E_{i,m} \times A_{m,j} \tag{5.2}$$

for $i = 1$ to $i = \mathrm{i}$, $j = 1$ to $j = j$, and $m = 1, 2, \ldots, m$.

After $A_{j,m}$ has been modified by $E_{i,m}$, as presented in Eq. (5.2), $A'_{m,j,i}$ is then modified by $V_{i,j}$, as follows,

$$A^*_{j,m,i} = V_{i,j} \times A'_{j,m,i} \tag{5.3}$$

for $i = 1$ to $i = \mathrm{i}$, $j = 1$ to $j = j$, and $m = 1, 2, \ldots, m$,

where,

$A^*_{j,m,i}$ is correlative weighting factors of influencing physical property, m, and, operating variable, j, after the modification of correlative weighting factor, $E_{i,m}$ and $V_{i,j}$, for particular deviation, i,

$A'_{j,m,i}$ is correlative weighting factor of influencing physical property, m, and operating variable, j, after the modification of correlative weighting factor, $E_{i,m}$,

$E_{i,m}$ is correlative weighting factor of influencing physical property, m, and the particular deviation, i,

$V_{i,j}$ is correlative weighting factor of the particular deviation, i, and operating variable, j,

i is the number of deviation, i.

j is the number of operating variable, j, and,

m is the number of influencing physical property,

Furthermore, since the correlative weighting factor, $B_{j,n}$, is only affected by correlative weighting factor, $V_{i,j}$, which depend on the particular deviation, then, $B_{j,n}$ will be modified by $V_{i,j}$ to satisfy the particular deviation, i, as follows,

$$B^*_{j,n,i} = V_{i,j} \times B_{j,n} \tag{5.4}$$

for $i = 1$ to $i = i$, $m = 1$ to $n = n$, $j = 1, 2, \ldots, j$,

where,

$B^*_{j,n,i}$ is correlative weighting factors of inherent physical property, n, and, operating variables, j, after the modification of correlative weighting factor, $V_{i,j}$, for particular deviation, i,

$B_{j,n}$ is correlative weighting factor of inherent physical property, n, and operating variables, j,

$V_{i,j}$ is correlative weighting factor of operating variable, j, and the particular deviation, i,

i is the number of deviation, i.

j is the number of operating variable, j, and,

n is the number of inherent physical properties,

After, the correlative weighting factors A, B are modified for a particular deviation, then the total correlative weighting factor, $r_{i,j}$, for each operating variable, j, on the particular deviation, i, is calculated as follows,

$$r_{i,j} = \frac{\sum_{m=1}^{m} A^*_{j,m,i} + \sum_{n=1}^{n} B^*_{j,n,i}}{m+n} \tag{5.5}$$

for $i = 1$ to $i = i$, and $j = 1$ to $j = j$,

where,

i is the number of deviation, i.

j is the number of operating variable, j, and,

m is the number of influencing physical properties,

n is the number of inherent physical properties,

However the deviation also depends on the operating range of the operating variables. To calculate the final priority weighting factors, $R_{i,j}$, for each operating variable, j, for on particular deviation, i, the value of the operating variable must be compared with the recommended operating conditions and accounted for. Furthermore, since the sign of the weighting factor, r, only indicates the adjustment direction of operating variable, and is independent on the resolution sequence. Therefore, to calculate the final priority weighting factor, R, the weighting factor, r, is always considered as an absolute value. Thus the weighting factor, R, is calculated as follows,

When $V_{i,j} > 0$,

$$R_{i,j} = \left|r_{i,j}\right| \times \frac{-VA_j + VA_j^+ + VA_j^-}{2 \times VA_j^-} \tag{5.6}$$

When $V_{i,j} < 0$,

$$R_{i,j} = \left|r_{i,j}\right| \times \frac{VA_j - VA_j^+ - VA_j^-}{2 \times VA_j^-} \tag{5.7}$$

where

VA_j = the operating condition of operating variable i,

$$VA_j^+ = \frac{VA_{(j)\max} + VA_{(j)\min}}{2}$$

$$VA_j^- = \frac{VA_{(j)\max} - VA_{(j)\min}}{2}$$

$VA_{(j)\max}$ = the maximum recommended condition of operating variable, j,

$VA_{(j)\min}$ = the minimum of recommended condition of operating variable, j.

Based on above equations, the resolution procedures of the operating variable corrective actions is determined by comparison of the priority weighting factors, $R_{i,j}$, for the particular deviation, i.

Once the priority weighting factor, $R_{i,j}$, was determined, the value, $VA_{(j)\text{adj}}$, is assigned to indicate the adjustment amount of the changed operating variable, V_j. This adjustment amount, $VA_{(j)\text{adj}}$, is based on the advise of the experts and defined in the following manner.

When the adjusted operating variable, V_j, is temperature, then the adjustment amount, $VA_{(j)\text{adj}}$, is given as,

$$VA_{(j)\text{adj}} = 10\ °\text{F} \tag{5.8}$$

when adjusted operating variable, V_j, is pressure, then the adjustment amount, $VA_{(j)\text{adj}}$, is given as,

$$VA_{(j)\text{adj}} = 500\ \text{psi} \tag{5.9}$$

When the adjusted operating variable, V_j., is the cycle time, mold open time, or mold closed time, then the adjustment amount, $VA_{(j)\text{adj}}$, is given as;

$$VA_{(j)\text{adj}} = 2\ \text{second} \tag{5.10}$$

when adjusted operating variable, V_j, is the injection time, then the adjustment amount, $VA_{(j)\text{adj}}$, is given as,

$$VA_{(j)\text{adj}} = 2\ \text{second} \tag{5.11}$$

When the adjusted operating variable, V_j, is position, and greater than or equal to 1 inch, then the adjustment amount, $VA_{(j)\text{adj}}$, is given as,

$$VA_{(j)\text{adj}} = 0.2 \text{ inch} \tag{5.12}$$

When the adjusted operating variables, V_j, is position, and less than 1 inch, then the adjustment amount, $VA_{(j)\text{adj}}$, is given as,

$$VA_{(j)\text{adj}} = 0.1 \text{ inch} \tag{5.13}$$

Based on the Eqs. (5.8) to (5.12), the new operating conditions, $VA_{(j)\text{new}}$, of the operating variable, V_j, can now be calculated in the following manner.

When operating conditions, VA_j, of operating variable, j, requires to be increased for remedying the deviation, then, the new operating conditions, $VA_{(j)\text{new}}$, are given as,

$$VA_{(j)\text{new}} = VA_j + VA_{(j)\text{adj}} \tag{5.14}$$

When operating conditions, VA_j, of operating variable, j, requires to be decreased for remedying the deviation, then, the new operating variable, $VA_{(j)\text{new}}$, are given as,

$$VA_{(j)\text{new}} = VA_j - VA_{(j)\text{adj}} \tag{5.15}$$

where,

VA_j = the original operating condition of the operating variable, j,

$VA_{(j)\text{adj}}$ = the adjusted amount of operating variable, j,

$VA_{(j)\text{new}}$ = the new operating condition of operating variable, j,

5.4.3 Self-Learning Mechanism

As mentioned in section 5.4.2, in this program, there has a self-learning mechanism which is based on the response of the resolution actions from the user to update the correlative weighting factors and recommended operating conditions. This self-learning mechanism simulates the molding experts learning capability and allows the system to provide a more significant resolution procedures. In the following text, the self-learning mechanism will be discussed in detail.

As discussed in section 5.3.2.1, the resolution actions for eliminating or reducing the deviations during the injection molding process are classified into four different priority levels. These are the method corrective actions, the operating variable corrective actions, the mold corrective actions, and the material corrective actions. Furthermore, in section 5.4.1 these resolution actions are also categorized into two different types. These are the *absolute corrective actions* and the *conditional corrective actions*. The absolute corrective actions include the method corrective actions, the mold corrective actions, and the material corrective actions. The conditional corrective actions include the operating variable corrective actions.

Nevertheless, as discussed in the section 5.4.1, the absolute corrective actions will be only employed once for resolving the deviation during the resolution procedure, and will never be retrieved again. Moreover, in this program, the resolution procedures of these absolute corrective action have been correctly examined by the molding experts. Therefore, in this program the self-learning mechanism will not be employed into the resolution procedures of the absolute corrective actions. Thus, in this self-learning mechanism the primary consideration focuses on the modification of the resolution procedures of the operating variable corrective actions.

According to the discussion in the section 5.4.2, the resolution procedures of the operating variable corrective actions are based on the comparison of the priority weighting factor, $R_{i,j}$. Furthermore, in the discussion of the section 5.4.1, the resolution results of operating variable corrective actions are indicated in three ways "DEVIATION CORRECTED", "DEVIATION IMPROVED", and "DEVIATION NOT IMPROVED". Therefore, based on these resolution responses, the system will modify the correlative weighting factors, $R_{i,j}$, and the recommended operating conditions in the following manner.

When the operating variable, j, was employed to resolve the deviation, i, and the response of resolution result is "DEVIATION NOT IMPROVED", this means that the operating variable, j, has less influence than anticipated on the deviation, i. In other word, the priority weighting factor, $R_{i,j}$, has a higher priority than anticipated. According to the Eqs. 5.5 to 5.9, the priority weighting factor, $R_{i,j}$, is a function of the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, $E_{i,m}$, and $V_{i,j}$, the user operating conditions, and the recommend operating conditions. However, the recommended operating conditions are considered as fixed parameters during each deviation resolution procedure. Furthermore, the user operating conditions are adjusted in advance with each resolution procedure which will be adjusted automatically. Moreover, the correlative weighting factors, $E_{i,m}$, will not be directly influence by the adjustment of operating variables. Therefore, in this level of the resolution, these parameters which are the user operating conditions, the recommended operating conditions, and the correlative weighting factor, $E_{i,m}$, are not modified.

Therefore, lowing the priority factor, $R_{i,j}$ can modify the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$. The modification of these correlative weighting factors is calculated in the following manner.

When the value of the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$,are greater than or equal to 0.1, then, the new correlative weighting factors are given as,

$$A^{*}_{j,m} = A_{j,m} - 0.05 \tag{5.16}$$

$$B^{*}_{j,n} = B_{j,n} - 0.05 \tag{5.17}$$

$$V^{*}_{i,j} = V_{i,j} - 0.05 \tag{5.18}$$

for $i = i$ and $j = j$.

When the value of the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, are less than or equal to -0.1, then, the new correlative weighting factors are given as,

$$A^*_{j,m} = A_{j,m} + 0.05 \tag{5.19}$$

$$B^*_{j,n} = B_{j,n} + 0.05 \tag{5.20}$$

$$V^*_{i,j} = V_{i,j} + 0.05 \tag{5.21}$$

for $i = i$, and $j = j$,

where,

$A^*_{j,m}$ is the new correlative weighting factor of operating variable, j, and influencing physical property, m,

$A_{j,m}$ is the old correlative weighting factor of operating variable, j, and influencing physical property, m,

$B^*_{j,n}$ is the new correlative weighting factor of operating variable, j, and inherent physical property, n,

$B_{j,n}$ is the old correlative weighting factor of operating variable, j, and inherent physical property, n,

$V^*_{i,j}$ is the new correlative weighting factor of deviation, i, and operating variable, j, and

$V_{i,j}$ is the old correlative weighting factor of deviation, i, and operating variable, j.

When the value of the correlative weighting factors are greater than -0.1, and less than 0.1, then, the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$ will not be changed, Once, the new correlative weighting factors, $A^*_{j,m}$, $B^*_{j,n}$, and $V^*_{i,j}$ have been modified, these new correlative weighting factors will be employed into Eqs. 5.5 to 5.9 to calculate the new priority weighting factors, $R^*_{i,j}$. Based on the comparison of the new priority weighting factors, $R^*_{i,j}$, the system rearranges the new resolution procedures of the operating variable corrective actions until the deviation is eliminated or the operating variable corrective actions are exhausted.

If the resolution result of the operating variable, j, on the deviation, i, is "DEVIATION IMPROVED", the operating variable, j, has a more meaningful influence on the deviation, i. To indicate this significant influence of the operating variable, j, on the deviation, i, increases the correlative weighting factor, $R_{i,j}$. As discussed above, modifying the correlative weighting factor, $R_{i,j}$., can be achieved by modifying the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$. The modification method of the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, which will increase the correlative weighting factor, $R_{i,j}$., is calculated in the following manner.

When the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, are greater than zero, then the new correlative weighting factors are given as,

$$A^*_{j,m} = A_{j,m} + 0.05 \tag{5.22}$$

$$B^*_{j,n} = B_{j,n} + 0.05 \tag{5.23}$$

$$V^*_{i,j} = V_{i,j} + 0.05 \tag{5.24}$$

for $i = i$ and $j = j$.

When the value of the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, less than zero, then, the new correlative weighting factors are given as,

$$A^*_{j,m} = A_{j,m} - 0.05 \tag{5.25}$$

$$B^*_{j,n} = B_{j,n} - 0.05 \tag{5.26}$$

$$V^*_{i,j} = V_{i,j} - 0.05 \tag{5.27}$$

for $i = i$ and $j = j$,

where,

$A^*_{j,m}$ is the new correlative weighting factor of operating variable, j, and influencing physical property, m,

$A_{j,m}$ is the old correlative weighting factor of operating variable, j, and influencing physical property, m,

$B^*_{j,n}$ is the new correlative weighting factor of operating variable, j, and inherent physical property, n,

$B_{j,n}$ is the old correlative weighting factor of operating variable, j, and inherent physical property, n,

$V^*_{i,j}$ is the new correlative weighting factor of deviation, i, operating variable, j, and

$V_{i,j}$ is the old correlative weighting factor of deviation, i, operating variable, j.

Similarly for the self-learning mechanism when the resolution response, "DEVIATION NOT IMPROVED", the correlative weighing factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, will be employed into Eqs 5.5 to 5.9, to calculate the new priority weighting factor, $R^*_{i,j}$. The system is then based on the new priority weighting factor, $R^*_{i,j}$, to provide a new resolution sequence until the deviation is eliminated or the operating variable corrective actions are exhausted.

When the resolution response, "DEVIATION CORRECTED", is input, the deviation has been resolved by the resolution procedure which is accommodated by the system. To provide a more significant performance of the system for use the next time at the same deviation, the system will update the correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, which are provided by the system or have been modified, into its knowledge base. Therefore, when the system is employed in resolving the same deviation, which has been encountered previously, the system can provide more significant correlative weighting factors, $A_{j,m}$, $B_{j,n}$, and $V_{i,j}$, to indicate the more meaningful resolution procedures.

Furthermore, according to Eqs. 5.5 to 5.9, the resolution procedures of the operating variable corrective actions, which are determined by the correlative weighting factors, $R^*_{i,j}$, also depends on the recommended operating conditions, and the user operating conditions. When the resolution response, "DEVIATION CORRECTED", is attained the final operating conditions are the optimal situation for the deviation. Therefore, the system based on these final operating conditions updates the recommended operating conditions in the following manner.

When the operating condition, VA_j, of operating variable, j, is required to be increased for resolving the deviation, i, then the new recommended conditions, VA*s, of operating variable, j, are given as,

$$VA^*_{(j)\min} = VA_j \tag{5.28}$$

for $j = 1$ to $j = j$,

and the $VA_{(j)\max}$ will not be changed,

When the operating condition, VA_j, of operating variable, j, is required to be decreased for resolving the deviation, i, then the new recommended conditions, VA*s, of operating variable, j, are given as,

$$VA^*_{(j)\max} = VA_j \tag{5.29}$$

for $j = 1$ to $j = j$,

and the $VA_{(j)\min}$ will not be changed,

where,

VA_j is the final operating condition of operating variable, j,

$VA^*_{(j)\max}$ is the new maximum recommended operating condition of operating variable, j,

$VA^*_{(j)\min}$ is the new minimum recommended operating condition of operating variable, j,

$VA_{(j)\max}$ is the old maximum recommended operating condition of operating variable, j,

$VA_{(j)\min}$ is the old minimum recommended operating condition of operating variable, j,

According to Eqs. 5.16 to 5.17, the system updates the recommended operating conditions when the resolution response, "DEVIATION CORRECTED", is attained. When the system is used to resolve a deviation, which has been resolved previously, these recommended operating conditions will then be employed into Eqs. 5.5 to 5.9 to calculate the correlative weighting factors, $R_{i,j}$, and provide a more significant resolution procedure for eliminating or reducing the deviation.

5.5 Tool Selection

Once the inference engine has been developed, the next development procedure is to select an appropriate tool to codify the knowledge base, the inference engine, and the user interface. In this program, several tools have been examined. These tools can be classified into two categories. These are programming languages and expert system shells. The programming languages include FORTRAN, LISP, PROLOG, and C programming languages. The expert system shells include VP-EXPERT and KEE.

In this program, these tools have been employed in the system. However, comparing their advantages and disadvantages, the final decision was to select the C programming language to codify for this system. This selection is based on the following comparisons with other tools.

1. The C programming language has the ability to represent the symbolic data the same as symbolic programming languages such as LISP and PROLOG. However, the price of the C programming compiler is cheaper than LISP and PROLOG.
2. The C programming language provides a compatibility of transformation between the personal computer and mainframe computer. When using C programming language in any computer systems, only a few change are required, sometimes none, to transfer the program between different computer systems. However, LISP and PROLOG require a specific machine to run their language. It is not easy to transfer the program between the mainframe and the personal computer.
3. The C programming language provides the ability to evaluate a complex mathematical equation. Although, the FORTRAN programming language has a powerful ability to handle the complex mathematical calculation, it has only a weak ability to handle the symbolic data. Furthermore, for LISP and PROLOG programming language, it is difficult to calculate a complex mathematical equation.
4. The C programming language is easy to transfer to a object oriented programming language. Currently, object oriented programming language is widely used in the field of artificial intelligence application.
5. Using C programming language can avoid the limitation of rule based set-up. For instance, to use the VP-EXPERT expert shell, this rule base is constrained into 1,000 rules. When an expert system requires over a

thousand rules to represent its knowledge, this expert system exceeds the capability for using the VP-EXPERT expert system shell.

6. The C programming language is easy to learning. LISP and PROLOG programming languages are not a very friendly programming languages.

5.6 System Testing

Based on the above development procedures, an expert system has been develoþed. Once the system has been developed, the system then requires examination, correction, and evaluation. For examination focus on the reliability of the resolution procedures searching strategy is made. For correction focus on the accuracy of the knowledge representation is made. For evaluation focus on this system structure and whether it can be represented to the whole problem domain is made. 5.6

For a complex problem domain, the development procedure, in general, is classified into two stages. These are the prototype system development and the final system development. In the prototype system development, a partial of the whole problem domain is selected for development. The system development procedure is then based on this prototype system for development of the final system. In this program, for instance, the development procedures are also divided into two stages such as prototype system development and final system development. In the following text, this system testing is presented.

5.6.1 Prototype Testing

In this program, the prototype system has been developed in the early stages of the development procedure. In this prototype system, the problem domain is constrained in some situations. These are,

molded material: acetal copolymer,

material manufacturer: Hoechst Celanese, material grade: M-90, and

deviation type: surface ripples and warpage.

The results of this prototype system have been sent to the experts of the field of injection molding for examination [Jan and O'Brien, 1991] [Jan and O'Brien, 1992]. The algorithm for determining the operating variable corrective actions has also been sent for journal publication [Jan and O'Brien, 1991]. In addition, the results of this prototype system have been simulated in the mold filling package, MOLDFLOW, to examine its resolution procedures. The simulation results will be discussed in chapter six.

From the responses of these examinations, the expert system technique is suitable to be employed to resolve the deviations in the injection molding process. Furthermore, the knowledge representation of this prototype system can accurately represent the resolution information. Moreover, the resolution strategy of this prototype system can reasonably simulate the reasoning mechanism of the molding expert to resolve the deviation. Therefore, it can be concluded that this is a successful prototype system.

5.6.2 Final Testing

Based on the prototype system, an expert system for the injection molding of engineering thermoplastics has been developed. In this final system, the resolution knowledge of a molding material, acetal copolymer, from Hoechst Celanese, has been developed. This final system has also been developed to resolve common deviations such as surface ripples, pit marks, splay marks, sink marks, voids, flashing, short shots, warpage, distortion, and delamination. Currently, this final system is under evaluation by potential users to evaluate its performance. However, this final system has also been examined using the mold filling simulation package, MOLDFLOW, to simulate its performance. From the simulation results, it appears that the performance of this system provides a reasonable resolution procedure for remedying the deviations. Furthermore, the resolution actions can actually influence the primary parameters for resolving the particular deviation. In the next chapter, the simulation results will be discussed in detail.

Table 5.1 Knowledge sources from text books.

DISCUSSION	NAME OF TEXTBOOKS
Basic Concept of Polymer Processing	1. Tadmor, Z. and C. Gogos., *Principles of Polymer Processing.* 2. Middleman, S., *Fundamentals of Polymer Processing.* 3. Rosato, D. and D. Rosato., *Plastics Processing Data Handbook.* 4. DuBois, J. and F. John., *Plastics.* 5. McKelvey, J., *Polymer Processing.*
Basic Concept of Injection Molding Process	1. Rubin, I., Injection Molding: *Theory and Practice.* 2. Rosato, D. and D. Rosato., *Injection Molding Handbook.* 3. Hohannaber, F., *Injection Molding Machines.* 4. Dym, J., *Injection Molds and Molding.* 5. DuBois, J. and W. Pribbles eds., *Plastics Mold Engineering Handbook.* 6. Whelan, A., *Injection Molding Machines.*
Injection Mold Design	1. Rubin, I., *Injection Molding: Theory and Practice.* 2. Rosato, D. and D. Rosato., *Injection Molding Handbook.* 3. Hohannaber, F., *Injection Molding Machines.* 4. Gastrow, H., and L. Stoeckhert eds., *Injection Molds.* 5. Stoeckhert, K. ed., *Mold-Making Handbook.* 6. Pye, R., *Injection Mold Design.*
Computer Aided Engineering for Polymer Processing	1. Bernhardt, E. ed., *Application of Computer Aided Engineering for Injection Molding.* 2. Manzione, L. ed., *Computer Aided Engineering for Injection Molding.* 3. Tucker, C. ed., *Fundamentals of Computer Modeling for Polymer Processing.* 4. O'Brien, K. ed., *Computer Modeling for Extrusion and Other Continuous Polymer Processes.*

Table 5.2 Range of operating variables [Hoechst Celanese Bulletin C3A, 1986].

Variable Name	Maximum	Minimum
Barrel Temperature	420 °F	380 °F
Nozzle Temperature	420 °F	380 °F
Mold Temperature	200 °F	160 °F
Injection Pressure	5,000 psi	7,000 psi
Injection Time	8 sec	4 sec
Mold Close Time*	20 sec	16 sec
Mold Open Time*	5 sec	10 sec
Cycle Time	20 sec	30 sec
Decompression	4 sec	1 sec
Shot Size*	3.2 in	2.0 in
Cushion*	0.5 in	0.1 in
Screw Speed*	60 rpm	40 rpm
Regrind Rate	20 %	0 %
Injection Speed	Maximum	Medium
Gate Size*	0.2 in	0.1 in
Cooling Channel Size*	0.5 in	0.3 in
Runner Size*	0.4 in	0.2 in
Venting Channel Size*	0.1 in	0.05 in
*Depend on part design		

Table 5.3 The troubleshooting table [Hoechst Celanese Bulletin C3A, 1986].

PROBLEM	SUGGESTED CORRECTIVE ACTION
Short Shot, Pit Marks, and Surface Ripples	1. Using maximum injection speed or booster. 2. Increase material temperature. 3. Increase mold temperature. 4. Check hopper for material supplier. 5. Increase injection pressure. 6. Increase feed 7. Decrease cushion 8. Preheat material in hopper 9. Increase gate size 10. Increase overall cycle
Splay Marks Due to jetting at gate	1. Reduce injection speed 2. Increase mold temperature 3. Increase gate size 4. Install electric heating cartridge in gate area

Table 5.3 (continued) The troubleshooting table [Hoechst Celanese Bulletin C3A, 1986]

Due to Excessive Moisture	1. Reduce material temperature 2. Dry material
Due to Drooling at Nozzle	1. Reduce material temperature 2. Reduce nozzle temperature 3. Reduce die-open time 4. Reduce injection time 5. Reduce cushion 6. Use nozzle with smaller orifice 7. Increase decompression time
Warpage, and Distortion	1. Set uniform temperature in both halves of mold 2. Observe mold for uniformity of part ejection 3. Check handling of parts after ejection from mold 4. Increase injection pressure 5. Increase injection time 6. Reduced mold temperature 7. Increase die closed time 8. Reduce melt temperature 9. Jig the part and cool uniformly
Voids	1. Increase injection pressure 2. Use booster and maximum injection speed 3. Increase injection time 4. Decrease cushion 5. Raise mold temperature 6. Increase gate size
Sink Marks	1. Increase injection pressure 2. Use booster and maximum injection pressure 3. Increase injection time 4. Decrease cushion 5. Reduce material temperature 6. Reduce mold temperature 7. Increase gate size 8. Relocate gating nearer heavy section
Flashing	1. Increase clamping force 2. Decrease material temperature. 3. Decrease mold temperature. 4. Decrease injection pressure. 5. Decrease feed 6. Increase cushion 7. Reduce gate size 8. Decease injection time
Delamination	1. Eliminate contamination from feed 2. Increase gate size

Table 5.4 A interview question table.

Molding Material Information	
Molded Material Type	Acetal Copolymer
Material Manufacturer	Celanese
Material Grade	M90
Operating Conditions	
Barrel Temperature (°F)	400 °F
Nozzle Temperature (°F)	400 °F
Mold Temperature (°F)	180 °F
Injection Speed (Max., Med., Min.)	Min.
Injection Pressure (psi)	5000 psi
Injection Time (second)	6 sec
Total Cycle Time (second)	25 sec
Part Maximum Thickness (inch)	0.125 in
Mold Close Time (second)	18 sec
Mold Open Time (second)	8 sec
Decompression (second)	3 sec
Shot Size (inch)	2.6 in
Cushion (inch)	0.3 in
Screw Speed (rpm)	50 rpm
Regrind Rate (%)	10 %
Gate Size (inch)	0.15 in
Cooling Channel Size (inch)	0.2 in
Runner Size (inch)	0.3 in
Venting Channel Size (inch)	0.1 in
Question Procedures	
(Interviewer: I1): Type of Deviation: **(Interviewee: I2):** Resolution Step 1: **(I1):** Response of Resolution Step 1: (Resolved, Improved, Not Improved) **(I2):** Resolution Step 2: **(I1):** Response of Resolution Step 2: (Resolved, Improved, Not Improved) . . . Until the Deviation is Eliminated.	

Table 5.5 The data correction form for application of the Pareto principle (deviation: flashing).

Deviation: Flashing				
	MOLD 1	MOLD 2	MOLD 3	TOTAL
Increase barrel temperature	×	×	×	×
Decrease barrel temperature	10	8	7	25
Increase nozzle temperature	×	×	×	×
Decrease nozzle temperature	8	9	8	25
Increase mold temperature	×	×	×	×
Decrease mold temperature	4	4	4	12
Increase injection pressure	×	×	×	×
Decrease injection pressure	18	20	18	56
Increase injection speed	×	×	×	×
Decrease injection speed	17	16	17	50
Increase injection time	×	×	×	×
Decrease injection time	11	10	9	30
Increase shot size	×	×	×	×
Decrease shot size	17	15	16	48
Increase cushion	14	15	15	44
Decrease cushion	×	×	×	×
Increase screw speed	2	2	2	6
Decrease screw speed	×	×	×	×
Increase regrind rate	1	2	1	4
Decrease regrind rate	×	×	×	×
Increase cycle time	×	×	×	×
Decrease cycle time	×	×	×	×
Increase cooling time	×	×	×	×
Decrease cooling time	×	×	×	×
Increase mold close time	×	×	×	×
Decrease mold close time	×	×	×	×
Increase mold open time	×	×	×	×
Decrease mold open time	×	×	×	×
Increase decompression	×	×	×	×
Decrease decompression	×	×	×	×
TOTAL	100	100	100	300

Table 5.6 The total percentage frequency of corrective actions for flashing deviation (acetal copolymer Celcon M-90).

Deviation: Flashing	
CORRECTION ACTIONS	FREQUENCY
Decrease injection pressure	56
Decrease injection speed	50
Decrease shot size	48
Increase cushion	44
Increase injection time	30
Decrease barrel temperature	25
Decrease nozzle temperature	25
Decrease mold temperature	12
Increase screw speed	6
Increase regrind rate	4
TOTAL	300

Table 5.7 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for flashing deviation (acetal copolymer Celcon M-90).

Deviation: Flashing			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Decrease injection pressure	56	18.6	18.6
Decrease injection speed	50	16.7	35.3
Decrease shot size	48	16.0	51.3
Increase cushion	44	14.7	66.0
Increase injection time	30	10.0	76.0
Decrease barrel temperature	25	8.3	84.3
Decrease nozzle temperature	25	8.3	92.6
Decrease mold temperature	12	4.0	96.6
Increase screw speed	6	2.0	98.6
Increase regrind rate	4	1.4	100
TOTAL	300	100	

Table 5.8 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for surface ripples deviation (acetal copolymer Celcon M-90).

Deviation: Surface Ripples			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Increase injection pressure	56	18.7	18.7
Increase injection speed	50	16.7	35.4
Increase shot size	39	13.0	48.4
Decrease injection time	37	12.3	60.7
Decrease cushion	36	12.0	72.7
Increase barrel temperature	28	9.3	82.0
Increase nozzle temperature	22	7.4	89.4
Increase mold temperature	12	4.0	93.4
Increase cycle time	10	3.3	96.7
Decrease screw speed	6	2.0	98.7
Decrease regrind rate	4	1.3	100
TOTAL	300	100.0	

Table 5.9 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for pit marks deviation (acetal copolymer Celcon M-90).

Deviation: Pit Marks			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Increase injection pressure	56	18.7	18.7
Increase injection speed	50	16.7	35.4
Increase shot size	39	13.0	48.4
Decrease injection time	37	12.3	60.7
Decrease cushion	36	12.0	72.7
Increase barrel temperature	26	8.7	81.4
Increase nozzle temperature	24	8.0	89.4
Increase mold temperature	12	4.0	93.4
Increase cycle time	10	3.3	96.7
Decrease screw speed	6	2.0	98.7
Decrease regrind rate	4	1.3	100.0
TOTAL	300	100	

Table 5.10 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for splay marks deviation (acetal copolymer Celcon M-90).

Deviation: Splay Marks			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Reduce injection speed	70	23.3	23.3
Reduce nozzle temperature	65	21.7	45.0
Reduce barrel temperature	45	15.0	60.0
Reduce injection time	43	14.3	74.3
Reduce mold open time	30	10.0	84.3
Reduce cushion	25	8.3	92.6
Reduce mold temperature	12	4.0	96.4
Reduce screw speed	10	3.4	100
TOTAL	300	100	

Table 5.11 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for sink marks deviation (acetal copolymer Celcon M-90).

Deviation: Sink Marks			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Increase injection pressure	70	23.3	23.3
Decrease mold temperature	55	18.3	41.6
Increase injection time	50	16.7	58.3
Increase injection speed	45	15.0	73.3
Decrease barrel temperature	35	11.7	85.0
Decrease cushion	25	8.4	93.4
Increase shot size	10	3.3	96.7
Decrease nozzle temperature	10	3.3	100
TOTAL	300	100	

Table 5.12 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for voids deviation (acetal copolymer Celcon M-90).

Deviation: Voids			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Increase injection pressure	70	23.3	23.3
Decrease mold temperature	60	20.0	43.3
Increase injection time	52	17.3	60.6
Increase injection speed	47	15.7	76.3
Decrease barrel temperature	33	11.0	87.3
Decrease nozzle temperature	20	6.7	94.0
Increase shot size	10	3.3	97.3
Decrease cushion	8	2.7	100
TOTAL	300	100	

Table 5.13 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for short shot deviation (acetal copolymer Celcon M-90).

Deviation: Short Shots			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Increase injection pressure	55	18.4	18.4
Increase injection speed	50	16.7	35.0
Increase shot size	37	12.3	47.4
Increase barrel temperature	34	11.3	58.7
Decrease cushion	33	11.0	69.7
Decrease injection time	30	10.0	79.7
Increase nozzle temperature	25	8.3	88.0
Increase mold temperature	18	6.0	94.0
Increase cycle time	10	3.3	97.3
Decrease regrind rate	5	1.7	99.0
Decrease screw speed	3	1.0	100.0
TOTAL	300	100	

Table 5.14 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for warpage deviation (acetal copolymer Celcon M-90).

Deviation: Warpage			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Decrease mold temperature	91	30.3	30.3
Increase mold close time	75	25.0	55.3
Increase injection pressure	48	16.0	71.3
Increase injection time	35	11.7	83.0
Decrease barrel temperature	22	7.3	90.3
Decrease nozzle temperature	17	5.7	96.0
Increase screw speed	12	4.0	100.0
TOTAL	300	100	

Table 5.15 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for distortion deviation (acetal copolymer Celcon M-90).

Deviation: Distortion			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Decrease mold temperature	96	32.0	32.0
Increase mold close time	78	26.0	58.0
Increase injection pressure	45	15.0	73.0
Increase injection time	30	10.0	83.0
Decrease barrel temperature	22	7.3	90.3
Decrease nozzle temperature	17	5.7	96.0
Increase screw speed	12	4.0	100
TOTAL	300	100	

Table 5.16 The frequency number, total percentage frequency, and cumulative percentage of corrective actions for delamination deviation (acetal copolymer Celcon M-90).

Deviation: Delamination			
CORRECTION ACTIONS	FREQUENCY	% of total	Cumulative %
Increase injection pressure	70	23.3	23.3
Increase injection speed	65	21.7	45.0
Increase mold temperature	52	17.3	62.3
Increase barrel temperature	42	14.0	76.3
Increase nozzle temperature	40	13.3	89.6
Increase cycle time	15	5.0	94.6
Decrease cushion	10	3.4	98.0
Decrease screw speed	6	2.0	100
TOTAL	300	100	

Table 5.17 The correlative weighting factors between operating variables and the deviations for the material acetal copolymer Celcon M90.

	Surface ripples	Pit marks	Splay marks	Sink marks	Voids	Flashing	Shot size	Warpage	Distortion	Delamination
Barrel temperature	0.25	0.25	-0.32	-0.25	-0.24	-0.22	0.31	-0.12	-0.11	0.30
Nozzle temperature	0.20	0.20	-0.47	-0.07	-0.14	0.25	0.25	0.25	0.25	0.25
Mold temperature	0.11	0.11	-0.09	-0.39	-0.43	-0.22	0.16	0.50	0.50	0.37
Injection pressure	0.50	0.50	-0.50	0.50	0.50	-0.50	0.50	0.26	0.23	0.50
Injection speed	0.45	0.25	0.00	0.32	0.34	-0.45	0.45	0.00	0.00	0.47
Injection time	0.33	0.25	-0.31	0.36	0.37	-0.27	0.27	-0.19	0.16	0.00
Shot size	0.35	0.35	0.00	0.07	0.07	-0.43	0.33	0.00	0.00	0.00
Cushion	-0.33	-0.33	-0.18	-0.18	-0.06	0.40	-0.30	0.00	0.00	-0.07
Screw speed	-0.03	-0.03	-0.07	0.00	0.00	0.05	-0.03	0.07	0.06	-0.04
Regrind rate	-0.05	-0.05	0.00	0.00	0.00	0.04	-0.05	0.00	0.06	0.00
Cycle time	0.09	0.09	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00
Mold close time	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41	0.41	0.00
Decompression	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.18 The attributions of an animal for an identification expert system.

Diet	Size	Color	Habitat	Species
meat	large	striped	jungle	tiger
meat	large	tawny	jungle	lion
meat	small	striped	house	tabby
meat	small	brown	jungle	weasel
grass	large	striped	plains	zebra
grass	small	gray	plains	rabbit
grass	large	tawny	plains	antelope

Table 5.19 The structure representation of the material type knowledge.

Frame Name: **MATERIAL_TYPE**
Object: MATERIAL_NAME **Attributions:** ACETAL_COPOLYMER ACETAL_HOMOPOLYMER NYLON_6 NYLON_66 POLYCARBONATE PBT PET POLYSTYRENE ABS SAN HIGH_IMPACT_POLYSTYRENE

Table 5.20 The structure representation of the material manufacturer knowledge for acetal copolymer material.

Frame Name: **ACETAL_COPOLYMER_MANUFACTURER**
Object: MANUFACTURER_NAME **Attributions:** CELANESE DuPONT BASF MITSUBISHI POLYPLASTICS

Table 5.21 The structure representation of the material grade knowledge for the Celcon acetal copolymer.

Frame Name: AC_CELANESE_GRADE			
Object:			
GRADE_NAME			
Attributions:			
U_10	EF_35	GB_25	GC_25A
M_25	UV_25	WR_25	M_50
EC_90+	EP_90	LW_90	LW_90_S2
LW_90_SC	M_90	MC_90	MC_90_HM
TX_90	TX_90+	UV_90	WR_90
M_140	AS_270	M_270	MC_270
MC_270_HM	AS_450	M_450	LW_GS_S2

Table 5.22 The structure representation of the recommended operating conditions knowledge (acetal copolymer Celcon M-90).

Frame Name: AC_C_M_90_ROC		
Object: RECOMMENDED_OPERATING_CONDITIONS **Attributions:**		
Variable Name	Maximum	Minimum
Barrel Temperature	420 °F	380 °F
Nozzle Temperature	420 °F	380 °F
Mold Temperature	200 °F	160 °F
Injection Pressure	5,000 psi	7,000 psi
Injection Time	8 sec	4 sec
Mold Close Time*	20 sec	16 sec
Mold Open Time*	5 sec	10 sec
Cycle Time	20 sec	30 sec
Decompression	4 sec	1 sec
Shot Size*	3.2 in	2.0 in
Cushion*	0.5 in	0.1 in
Screw Speed*	60 rpm	40 rpm
Regrind Rate	20 %	0 %
Injection Speed	Maximum	Medium
Gate Size*	0.2 in	0.1 in
Cooling Channel Size*	0.5 in	0.3 in
Runner Size*	0.4 in	0.2 in
Venting Channel Size*	0.1 in	0.05 in
*Depend on part design		

Table 5.23 The structure representation of the correlation weighting factors between the inherent physical properties and the operating variables (acetal copolymer Celcon M_90).

Frame Name: AC_C_M_90_CWF_HP_OV				
Object: C_W_FACTOR_HP_OV **Attributions:**				
	Thermal Conductivity	Thermal Diffusivity	Specific Volume	Specific Heat
Barrel Temperature	-0.45	-0.4	0.45	0.35
Nozzle Temperature	-0.5	-0.45	0.45	0.3
Mold Temperature	-0.45	-0.45	0.3	0.4
Injection Pressure	0.25	0.3	-0.35	0.45
Injection Speed	0.35	0.3	-0.25	0.35
Injection Time	0.25	0.3	-0.25	0.35
Shot Size	0.15	0.25	0.2	0.3
Cushion	0.20	0.25	0.3	0.35
Screw Speed	0.15	0.10	0.05	0.35
Regrind Rate	0.35	0.3	0.1	0.1
Cycle Time	-0.45	-0.45	0.2	-0.45
Mold Close Time	-0.5	-0.5	0.5	-0.4
Mold Open Time	0.1	0.05	0.05	0.25
Decompression	0.1	0.05	0.05	0.25

Table 5.24 The structure representation of the correction weighting factors between the influencing physical properties and the operating variables (acetal copolymer Celcon M_90).

Frame Name: AC_C_M_90_CWF_FP_OV			
Object: C_W_FACTOR_FP_OV **Attributions:**			
	Shear Rate	Shear Stress	Viscosity
Barrel Temperature	0.45	-0.5	0.4
Nozzle Temperature	0.5	-0.5	0.4
Mold Temperature	0.35	-0.45	0.4
Injection Pressure	-0.45	0.45	-0.4
Injection Speed	-0.45	0.40	-0.3
Injection Time	-0.35	0.25	-0.4
Shot Size	0.05	0.1	0.05
Cushion	0.05	0.01	0.05
Screw Speed	0.45	0.35	0.45
Regrind Rate	0.4	0.3	0.5
Cycle Time	0.3	-0.3	0.45
Mold Close Time	0.45	-0.4	0.45
Mold Open Time	0.2	0.2	0.05
Decompression	0.2	0.2	0.25

Table 5.25 The structure representation of the correction weighting factors between the influencing physical properties and the deviations (acetal copolymer Celcon M_90).

Frame Name: AC_C_M_90_CWF_FP_DEV										
Object: C-W_FACTORS_FP_DEV Attribution:										
	Surface ripples	Pit marks	Splay marks	Sink marks	Voids	Flashing	Shot size	Warpage	Distortion	Delamination
Shear rate	-0.45	-0.45	-0.45	0.50	0.50	0.45	-0.45	0.30	0.30	-0.50
Shear stress	-0.40	-0.40	-0.40	-0.45	-0.45	0.40	-0.45	-0.50	-0.50	-0.45
Viscosity	-0.50	-0.50	-0.50	-0.40	-0.40	0.50	-0.50	-0.40	-0.40	-0.40

Table 5.26 The structure representation of the correction weighting factors between the operating variables and the deviations (acetal copolymer Celcon M_90).

Frame Name: AC_C_M_90_CWF_OP_DEV										
Object: C-W_FACTORS_OP_DEV Attribution:										
	Surface ripples	Pit marks	Splay marks	Sink marks	Voids	Flashing	Shot size	Warpage	Distortion	Delamination
Barrel temperature	0.25	0.25	-0.32	-0.25	-0.24	-0.22	0.31	-0.12	-0.11	0.30
Nozzle temperature	0.20	0.20	-0.47	-0.07	-0.14	0.25	0.25	0.25	0.25	0.25
Mold temperature	0.11	0.11	-0.09	-0.39	-0.43	-0.22	0.16	0.50	0.50	0.37
Injection pressure	0.50	0.50	-0.50	0.50	0.50	-0.50	0.50	0.26	0.23	0.50
Injection speed	0.45	0.25	0.00	0.32	0.34	-0.45	0.45	0.00	0.00	0.47
Injection time	0.33	0.25	-0.31	0.36	0.37	-0.27	0.27	-0.19	0.16	0.00
Shot size	0.35	0.35	0.00	0.07	0.07	-0.43	0.33	0.00	0.00	0.00
Cushion	-0.33	-0.33	-0.18	-0.18	-0.06	0.40	-0.30	0.00	0.00	-0.07
Screw speed	-0.03	-0.03	-0.07	0.00	0.00	0.05	-0.03	0.07	0.06	-0.04
Regrind rate	-0.05	-0.05	0.00	0.00	0.00	0.04	-0.05	0.00	0.06	0.00
Cycle time	0.09	0.09	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.00
Mold close time	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41	0.41	0.00
Decompression	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.27 The structure representation of the deviation type knowledge.

Frame Name: DEVIATION_TYPE
Object: DEVIATION_NAME **Attributions:** SURFACE RIPPLES PIT MARKS SPLAY MARKS SINK MARKS VOIDS FLASHING SHORT SHOT WARPAGE DISTORTION DELAMINATION

Table 5.28 The structure representation of the resolution knowledge. (acetal copolymer Celcon M_90)

Frame Name: AC_C_M_90_SS_RES
Object: RESOLUTIONS **Attribution:** METHOD_CORRECTION_ACTIONS 1. CHECK HOPPER FOR MATERIAL SUPPLIED 2. CHECK TEMPERATURE INDICATOR, PRESSURE INDICATOR, TIME INDICATOR, DISPLACEMENT INDICATOR. OPERATING_VARIABLE_CORRECTION_ACTIONS 1. INCREASE BARREL TEMPERATURE 2. INCREASE MOLD TEMPERATURE 3. INCREASE NOZZLE TEMPERATURE 4. INCREASE INJECTION PRESSURE 5. INCREASE INJECTION SPEED 6. DECREASE INJECTION TIME 7. INCREASE CYCLE TIME 8. INCREASE SHOT SIZE 9. DECREASE CUSHION 10. DECREASE SCREW SPEED 11. DECREASE REGRIND RATE MOLD_CORRECTION_ACTIONS 1. INCREASE GATE SIZE 2. INCREASE RUNNER SIZE 3. INCREASE VENTING CHANNEL SIZE MATERIAL_CORRECTION_ACTIONS 1. CHANGE MATERIAL WITH A LOWER VISCOSITY FINAL_CORRECTION_ACTION 1. CONSULT WITH RAW MATERIAL SUPPLIER OR EXPERT MOLDING OPERATORS

Table 5.29 The rule knowledge for determining the material manufacturer (acetal copolymer Celcon M_90).

If MATERIAL_NAME = ACETAL_COPOLYMER
Then Go To ACETAL_COPOLYMER_MANUFACTURER Frame
If MATERIAL_NAME = ACETAL_HOMOPOLYMER
Then Go To ACETAL_HOMOPOLYMER_MANUFACTURER Frame
If MATERIAL_NAME = NYLON_6
Then Go To NYLON_6_MANUFACTURER Frame
If MATERIAL_NAME = NYLON_66
Then Go To NYLON_66_MANUFACTURER Frame
If MATERIAL_NAME = POLYCARBONATE
Then Go To POLYCABONATE_MANUFACTURER Frame
If MATERIAL_NAME = PBT
Then Go To PBT_MANUFACTURER Frame
If MATERIAL_NAME = PET
Then Go To PET_MANUFACTURER Frame
If MATERIAL_NAME = POLYSTYRENE
Then Go To POLYSTYRENE_MANUFACTURER Frame
If MATERIAL_NAME = ABS
Then Go To ABS_MANUFACTURER Frame
If MATERIAL_NAME = SAN
Then Go To SAN_MANUFACTURER Frame
If MATERIAL_NAME = HIGH_IMPACT_POLYSTYRENE
Then Go To HIGH_IMPACT_POLYSTYRENE_MANUFACTURER Frame

Table 5.30 The rule knowledge for determining the material grade (acetal copolymer Celcon M_90).

If MANUFACTURER_NAME = HOECHST_CELANESE
Then Go To AC_CELANESE_GRADE Frame
If MANUFACTURER_NAME = DuPONT
Then Go To AC_DuPOND_GRADE Frame
If MANUFACTURER_NAME = BASF
Then Go To AC_BASF_GRADE Frame
If MANUFACTURER_NAME = MITSUBISHI
Then Go To AC_MITSUBISHI_GRADE Frame
If MANUFACTURER_NAME = POLYPLASTICS
Then Go To AC_POLYPLASTICS_GRADE Frame

Table 5.31 The rule knowledge for determining of the recommended operating conditions (acetal copolymer Celcon M_90).

If GRADE_NAME = U_10
Then Go To AC_C_U_10_REC Frame
.
If GRADE_NAME = M_90
Then Go To AC_C_M_90_REC Frame
If GRADE_NAME = LW_GS_S2
Then Go To AC_C_LW_GS_S2_REC Frame

Table 5.32 The rule knowledge for confirmation of the recommended operating conditions (acetal copolymer Celcon M_90).

```
If CONFIRMATION = NO
Then  Go To AC_C_M90_ROC Frame and
   Find VARIABLE_NAME and
   Find MAXIMUM_VALUE and
   Find MINIMUM_VALUE and
   Input NEW_MAXIMUM_VALUE and
   Input NEW_MINIMUM_VALUE and
   Replace NEW_MAXIMUM_VALUE to MAXIMUM_VALUE and
   Replace NEW_MINIMUM_VALUE to MINIMUM_VALUE and
If CONFIRMATION = YES
Then  Go To Confirmation Rules of User Operating Conditions
```

Table 5.33 The rule knowledge for confirmation of the user operating conditions (acetal copolymer Celcon M_90).

```
If CONFIRMATION = NO
Then  Go To AC_C_USER_OC Frame and
   Find VARIABLE_NAME and
   Find VARIABLE_VALUE and
   Input NEW_VARIABLE_VALUE and
   Replace NEW_VARIABLE_VALUE to VARIABLE_VALUE
If CONFIRMATION = YES
Then  Go To Confirmation Rules of Correlative Weighting Factors
```

Table 5.34 The rule knowledge for confirmation of the correlation weighting factors between the inherent physical properties and the influencing physical properties (acetal copolymer Celcon M_90).

```
If CONFIRMATION = NO
Then  Go To AC_C_M_90_CWF_FP_HP and
   Find FP_NAME and
   Find HP_NAME and
   Find C_W_FACTORS_FP_HP and
   Input NEW_C_W_FACTORS_FP_HP and
   Replace NEW_C_W_FACTORS_FP_HP to C_W_FACTORS_FP_HP
If CONFIRMATION = YES
      Then Go To Confirmation Rule of Corrective Weighting Factor of Inherent Physical
      Properties and Operating Variables.
```

Table 5.35 The rule knowledge for determining the deviation resolution knowledge (acetal copolymer Celcon M_90).

If DEVIATION_TYPE = SURFACE_RIPPLES Then Go To AC_C_M_90_SR_RES Frame If DEVIATION_TYPE = PIT_MARKS Then Go To AC_C_M_90_PM_RES Frame If DEVIATION_TYPE = SPLAY_MARKS Then Go To AC_C_M_90_SPM_RES Frame If DEVIATION_TYPE = SINK_MARKS Then Go To AC_C_M_90_SM_RES Frame If DEVIATION_TYPE = VOIDS Then Go To AC_C_M_90_VO_RES Frame If DEVIATION_TYPE = FLASHING Then Go To AC_C_M_90_FL_RES Frame If DEVIATION_TYPE = SHORT_SHOTS Then Go To AC_C_M_90_SS_RES Frame If DEVIATION_TYPE = WARPAGE Then Go To AC_C_M_90_WA_RES Frame If DEVIATION_TYPE = DISTORTION Then Go To AC_C_M_90_DIS_RES Frame If DEVIATION_TYPE = DELAMINATION Then Go To AC_C_M_90_DEL_RES Frame

Table 5.36 The rule knowledge for determining the remedied degree of resolution actions (acetal copolymer Celcon M_90).

If REMEDIED_DEGREE = CORRECT Then Go To SELF_LEARNING_MECHANISM If REMEDIED_DEGREE = IMPROVED Then Go To SELF-LEARNING-MECHANISM and Return To Decision Algorithm and Return To RESOLUTION_BASE If REMEDIED_DEGREE = NOT_IMPROVED Then Return To Decision Algorithm and Return To RESOLUTION_BASE

Table 5.37 The rule values of the method correction actions.

Method Corrective Actions	Rule Value
Check hopper for material supplier	1.00
Clean mold surface	0.95
Increase clamping force	0.90
Jig the part and cool uniformly	0.85
Check ejector pins marks	0.80
Change a small nozzle orifice	0.75
Check temperature indicator	0.70
Check pressure indicator	0.65
Check screw speed indicator	0.60
Check screw position indicator	0.55
Set uniform temperature in both halves of mold	0.50
Relocate gate nearer heavy section	0.45

Table 5.38 The rule values of the mold correction actions.

Method Corrective Actions	**Rule Value**
Modify the gate size	0.9
Modify the cooling channel size	0.8
Modify the runner size	0.7
Modify the venting size	0.6

Table 5.39 The correlative weighting factors, $A_{j,m}$,between the influencing physical properties, m, and the operating variables, j,.

	FP_1	FP_2	......	......	FP_{m-1}	FP_m
VAR_1	$A_{1,1}$	$A_{1,2}$	......	....	$A_{1,m-1}$	$A_{1,m}$
VAR_2	$A_{2,1}$	$A_{2,2}$	......	......	$A_{2,m-1}$	$A_{2,m}$
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
VAR_{j-1}	$A_{j-1,1}$	$A_{j-1,2}$	......	......	$A_{j-1,m-1}$	$A_{j-1,m}$
VAR_j	$A_{j,1}$	$A_{j,2}$	......	......	$A_{j,m-1}$	$A_{j,m}$

where,
$A_{j,m}$ = correlative weighting factors,
FP = influencing physical properties,
VAR = operating variables,
j = number of variables,
m = number of influencing physical properties.

Table 5.40 The correlative weighting factors, $B_{j,n}$,between the inherent physical properties, n, and the operating variables, j.

	HP_1	HP_2	......	......	HP_{n-1}	HP_n
VAR_1	$B_{1,1}$	$B_{1,2}$	......	....	$B_{1,n-1}$	$B_{1,n}$
VAR_2	$B_{2,1}$	$B_{2,2}$	......	......	$B_{2,n-1}$	$B_{2,n}$
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
VAR_{j-1}	$B_{j-1,1}$	$B_{j-1,2}$	......	......	$B_{j-1,n-1}$	$B_{j-1,n}$
VAR_j	$B_{j,1}$	$B_{j,2}$	......	......	$B_{j,n-1}$	$B_{j,n}$

where,
$B_{j,n}$ = correlative weighting factors,
HP = inherent physical properties,
VAR = operating variables,
j = number of variables,
n = number of influencing physical properties.

Table 5.41 The correlative weighting factors, $E_{i,m}$, between the influencing physical properties, m, and the deviations, i.

	FP_1	FP_2	......	......	FP_{m-1}	FP_m
DEV_1	$E_{1,1}$	$E_{1,2}$	......	....	$E_{1,m-1}$	$E_{1,m}$
DEV_2	$E_{2,1}$	$E_{2,2}$	......	......	$E_{2,m-1}$	$E_{2,m}$
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
DEV_{i-1}	$E_{i-1,1}$	$E_{i-1,2}$	......	......	$E_{i-1,m-1}$	$E_{i-1,m}$
DEV_i	$E_{i,1}$	$E_{i,2}$	......	......	$E_{i,m-1}$	$E_{i,m}$

where,
DEV = Deviations,
$E_{i,m}$ = correlative weighting factors,
FP = influencing physical properties,
i = number of deviations,
m = number of influencing physical properties,

Table 5.42 The correlative weighting factors, $V_{i,j}$, between the operating variables, j, and the deviations, i.

	VAR_1	VAR_2	......	......	VAR_{j-1}	VAR_j
DEV_1	$V_{1,1}$	$V_{1,2}$	......	....	$V_{1,j-1}$	$V_{1,j}$
DEV_2	$V_{2,1}$	$V_{2,2}$	......	......	$V_{2,j-1}$	$V_{2,j}$
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
....	......	......	......	......	......	......
$DEV_{i-1,1}$	$V_{i-1,1}$	$V_{i-1,2}$	......	......	$V_{i-1,j-1}$	$V_{i-1,j}$
DEV_i	$V_{i,1}$	$V_{i,2}$	......	......	$V_{i,j-1}$	$V_{i,j}$

where,
DEV = Deviations,
$V_{i,j}$ = correlative weighting factors,
VAR = operating variables,
i = number of deviations,
j = number of variables,

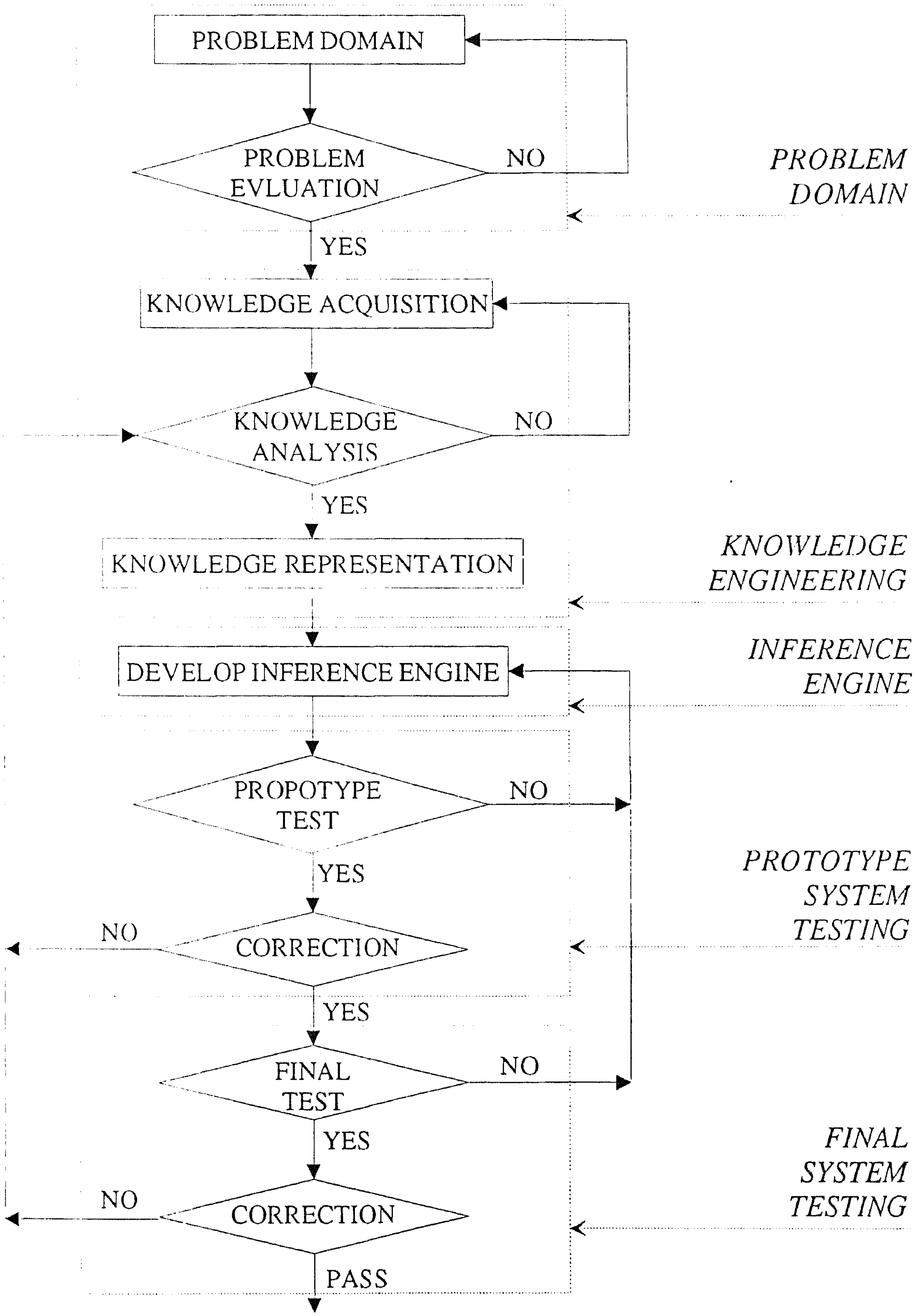


Figure 5.1 A schematic representation of the design procedures of an expert system.

Client (C): Right now, I have flashing (deviation) during processing. Could you give me assistance to eliminate the short shots.
Expert Molding Operator (E): OK, first, I need to know what your resin is?
C: The resin is acetal copolymer from Hoechst Celanese.
E: What is the material grade?
C: The material grade is Celcon M90.
E: Could you describe your operating conditions?
C: The operating conditions are barrel temperature = 400 °F; nozzle temperature = 400 °F; injection pressure = 6,000 psi; mold temperature = 180 °F; injection speed is medium; cushion = 0.3 in; shot size = 2.6 in; and screw speed = 60 rpm.
E: Do you use maximum clamping force?
C: Yes, the clamping force is maximum.
E: Are your instruments reading correctly?
C: I am not sure about that!
E: You better check your instrument readings.
C: Why?
E: According to your operating conditions, flashing (deviation) should not occur. The instrument reading failure is possibly the cause of your deviation.
C: OK, I have checked all the instruments, and they are reading correctly. But, flashing are still occurring. What is my next step?
E: You maybe need to use the minimum injection speed.
C: Why?
E: Comparing your operating conditions to the recommended the higher injection speed is the most likely cause for your flashing.
C: I have tried the maximum injection speed. The flashing have been reduced but not completely eliminated.
E: Well, you can now decrease your injection pressure.
C: How much should I decrease injection pressure.
E: Decrease injection pressure to 5,500 psi.
C: All right, now, the flashing have completely been eliminated.

Figure 5.2 The interview procedures for resolving the flashing deviation.

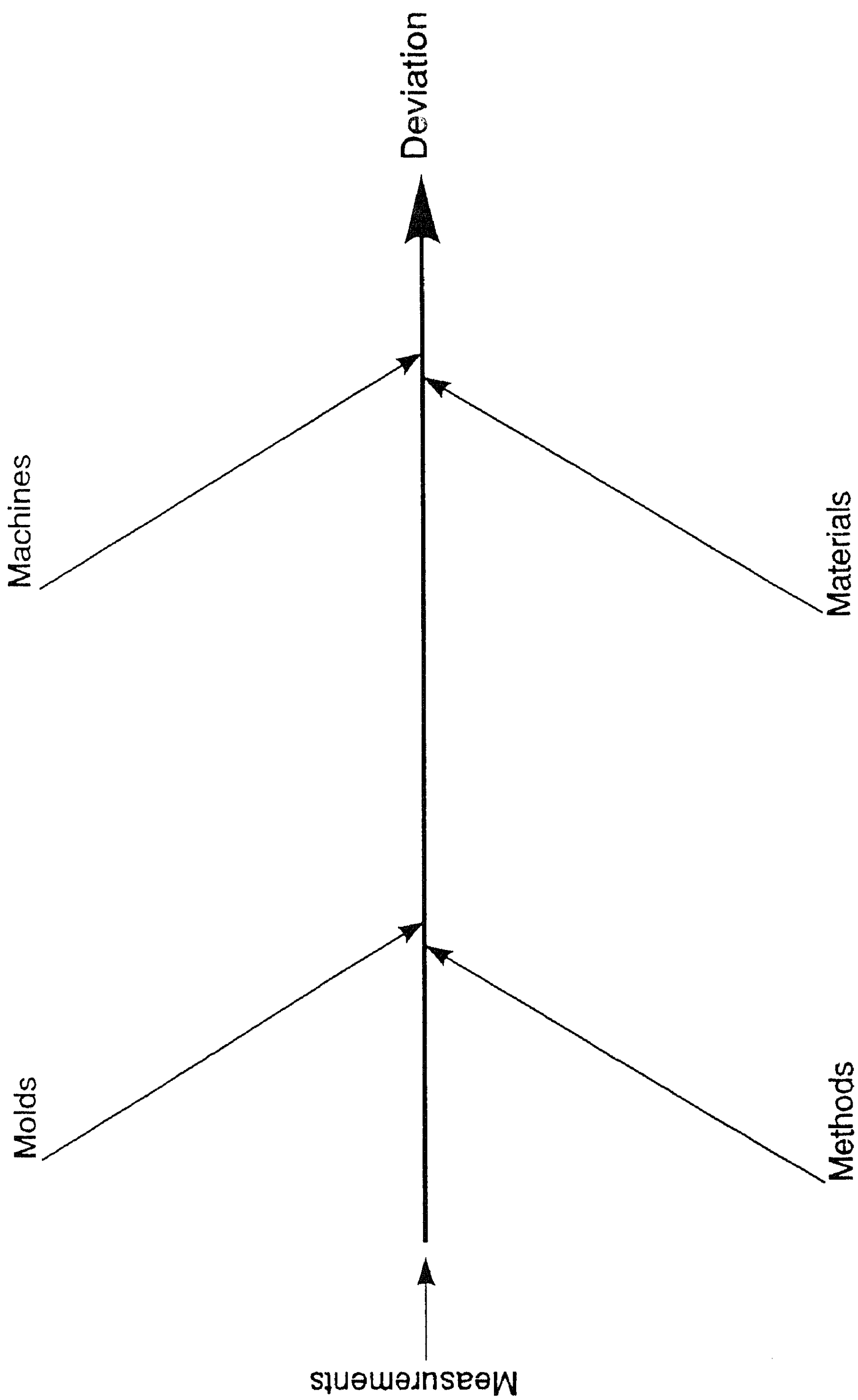


Figure 5.3 The basic structure of 5-M fishbone diagram for the injection molding deviations.

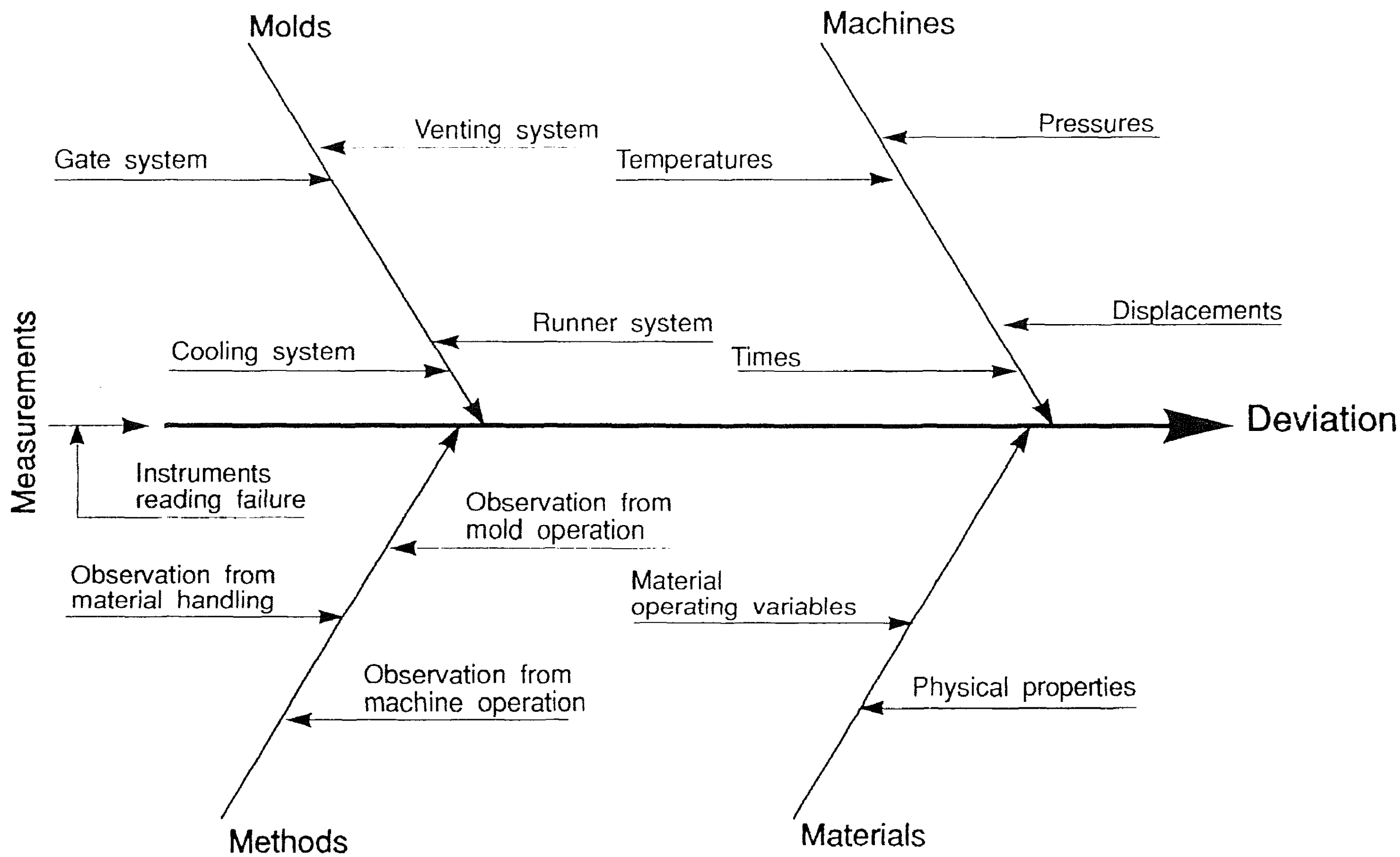


Figure 5.4 The detail structure of 5-M fishbone diagram for flashing deviation.

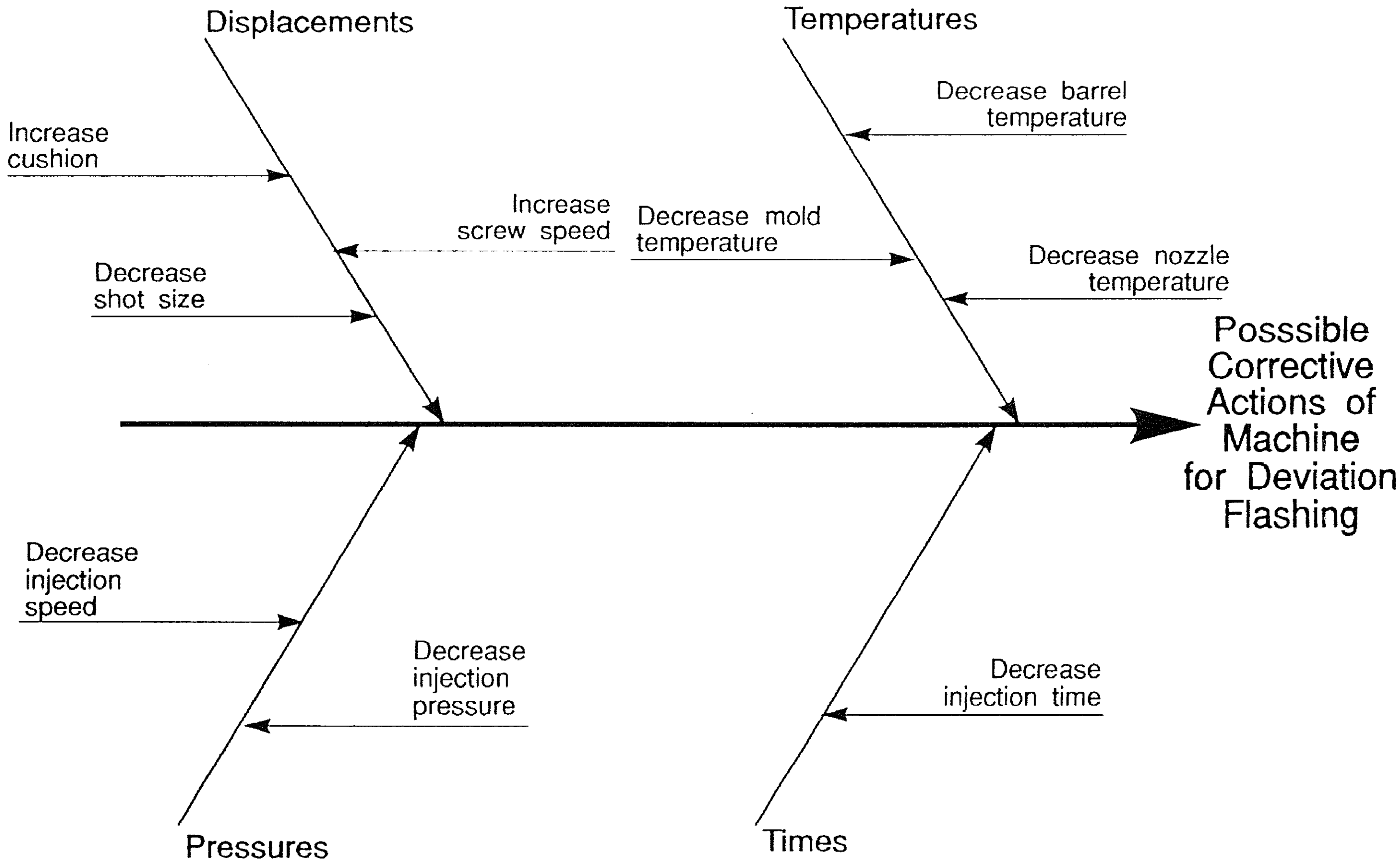


Figure 5.5 The "analyzing" results of the machine branch of fishbone diagram for flashing deviation.

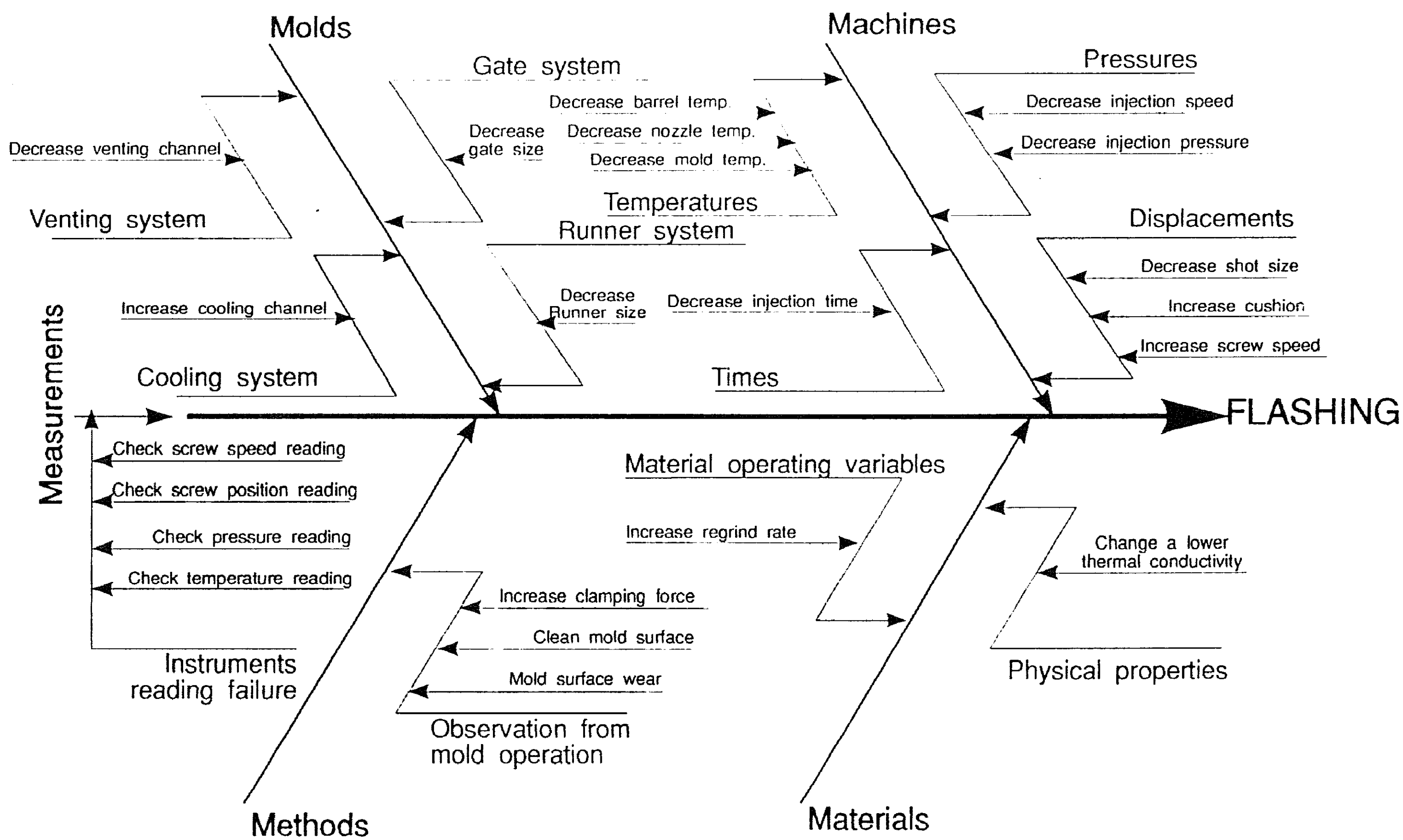


Figure 5.6 The total "analyzing" results of fishbone diagram for flashing deviation.

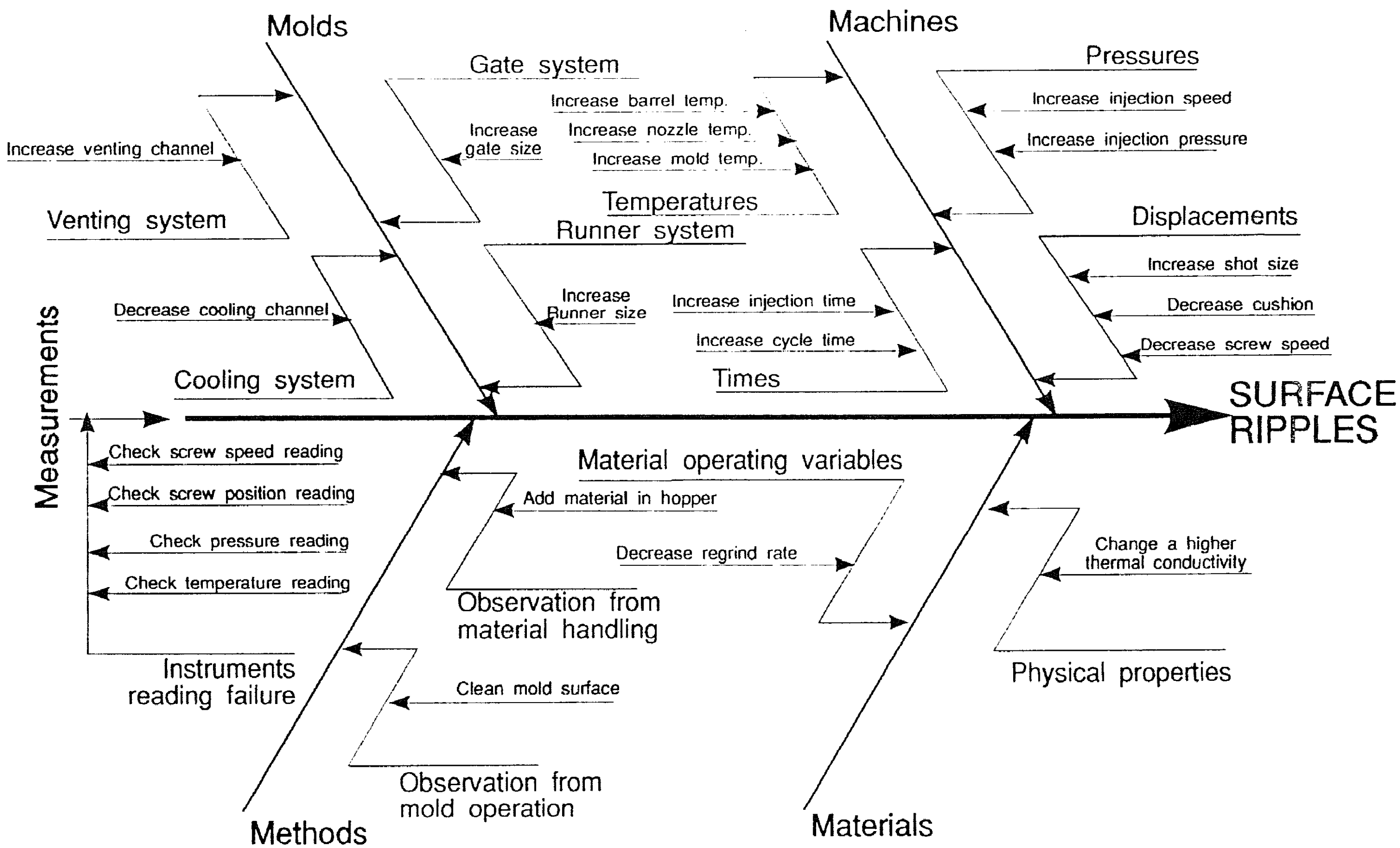


Figure 5.7 The total "analyzing" results of fishbone diagram for surface ripples deviation.

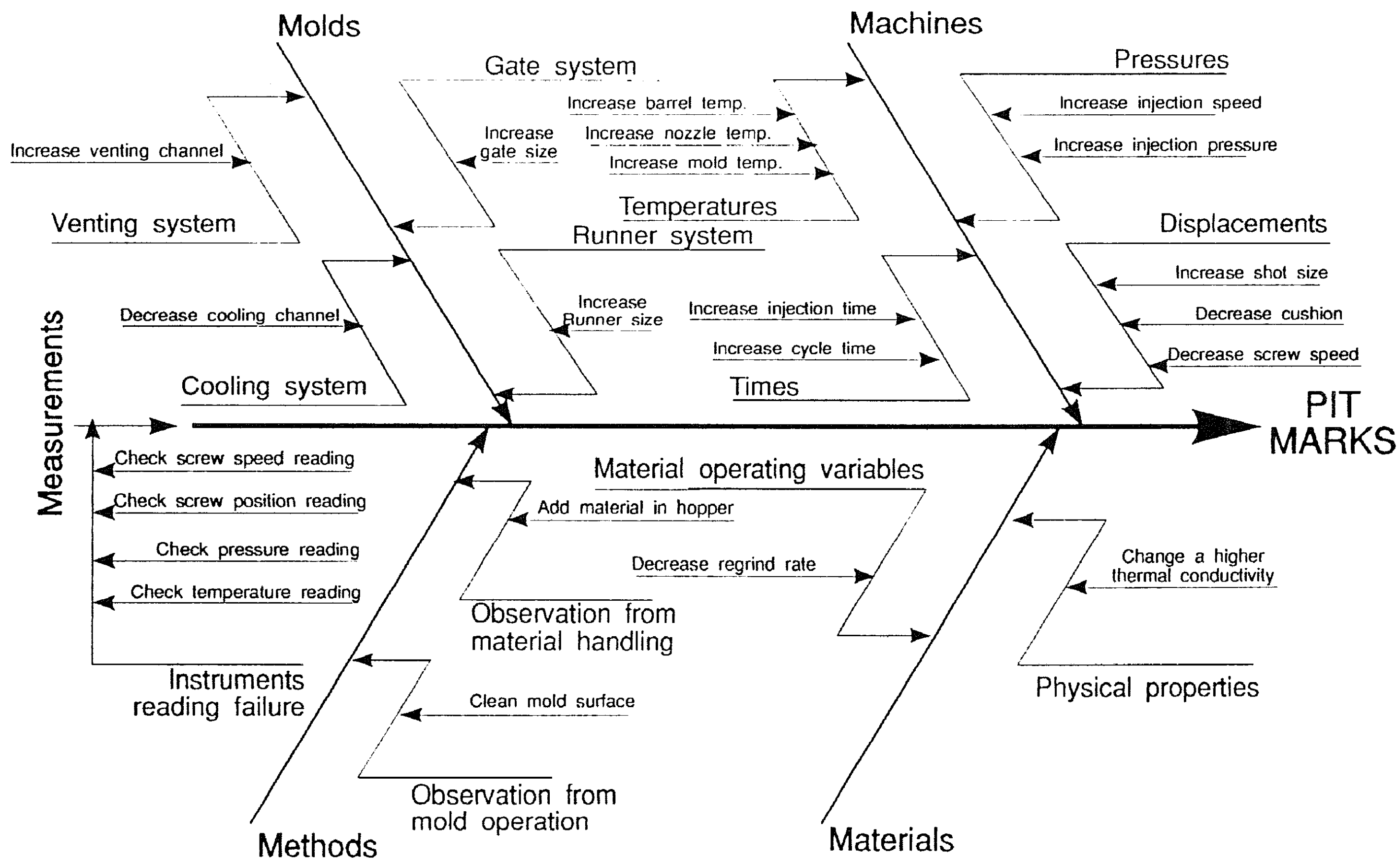


Figure 5.8 The total "analyzing" results of fishbone diagram for pit marks deviation.

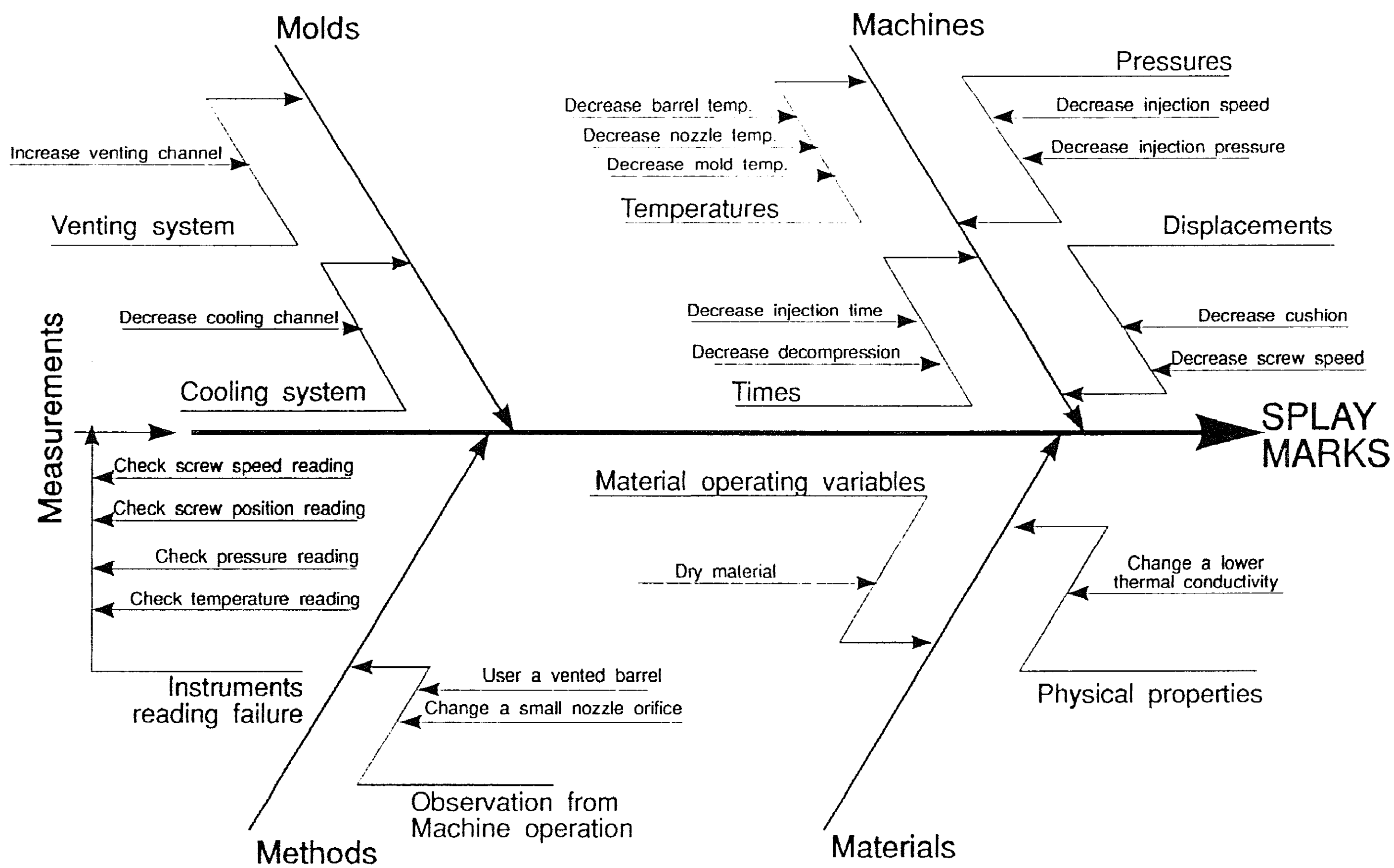


Figure 5.9 The total "analyzing" results of fishbone diagram for splay marks deviation.

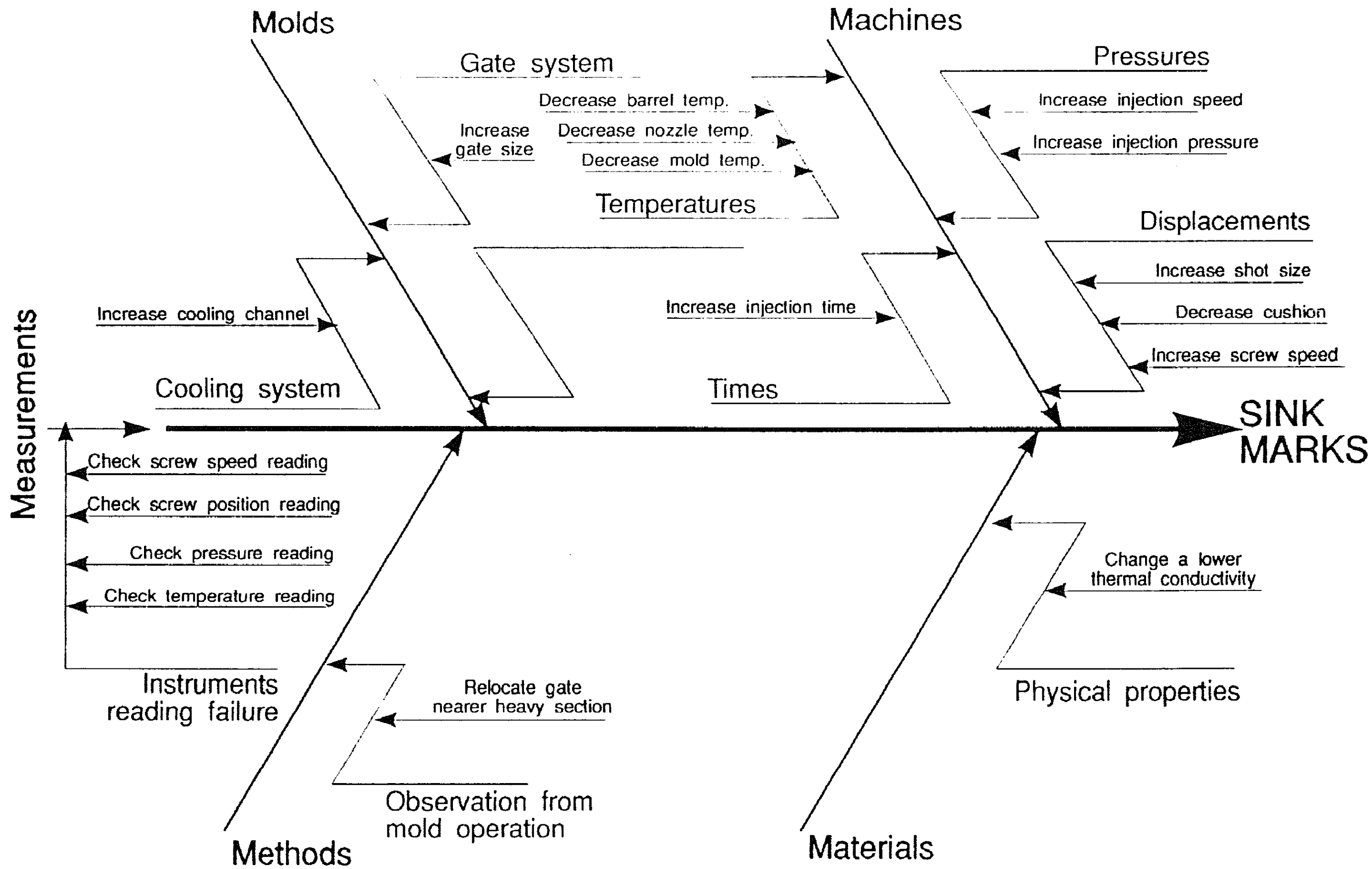


Figure 5.10 The total "analyzing" results of fishbone diagram for sink marks deviation.

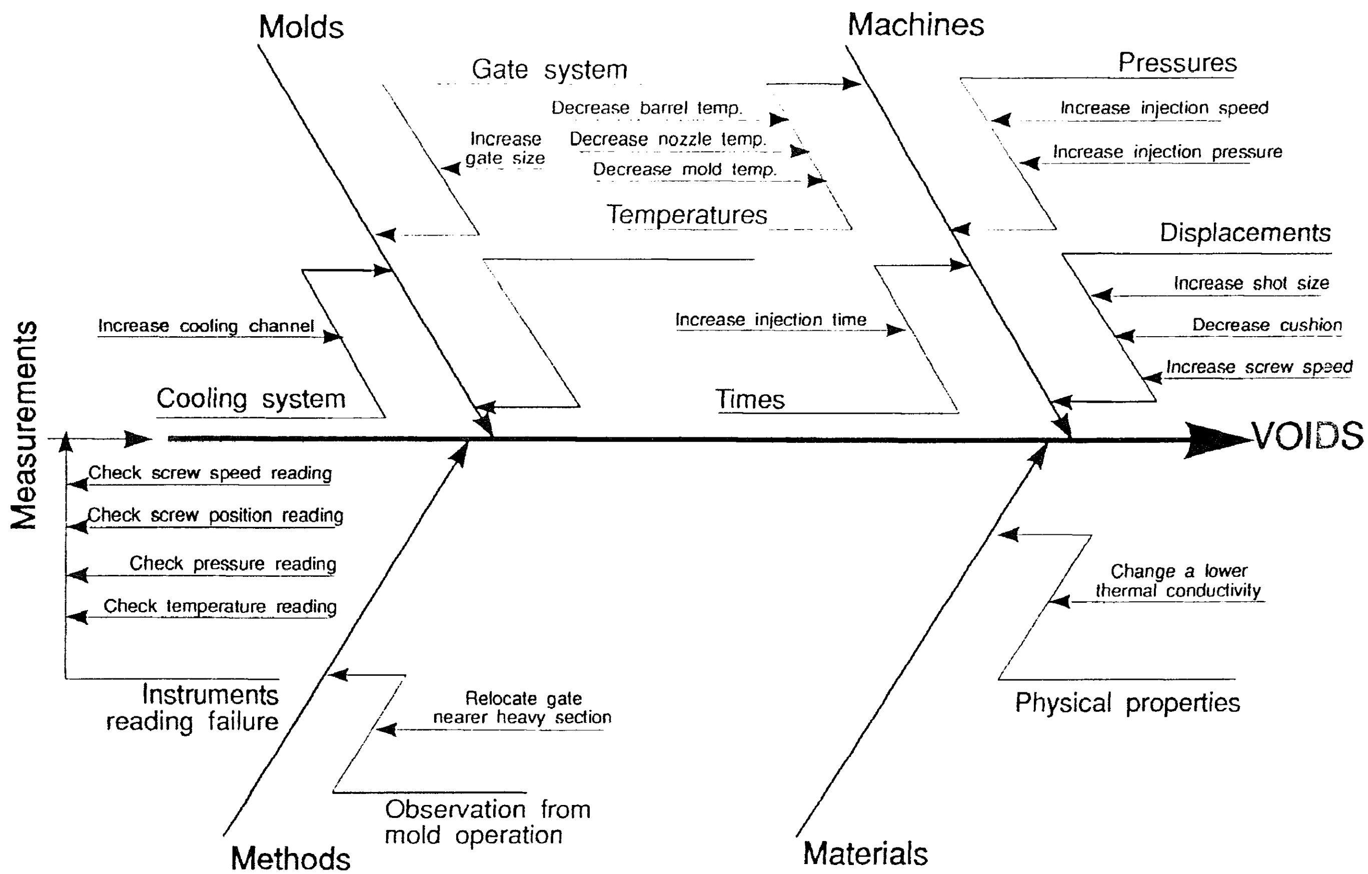


Figure 5.11 The total "analyzing" results of fishbone diagram for voids deviation.

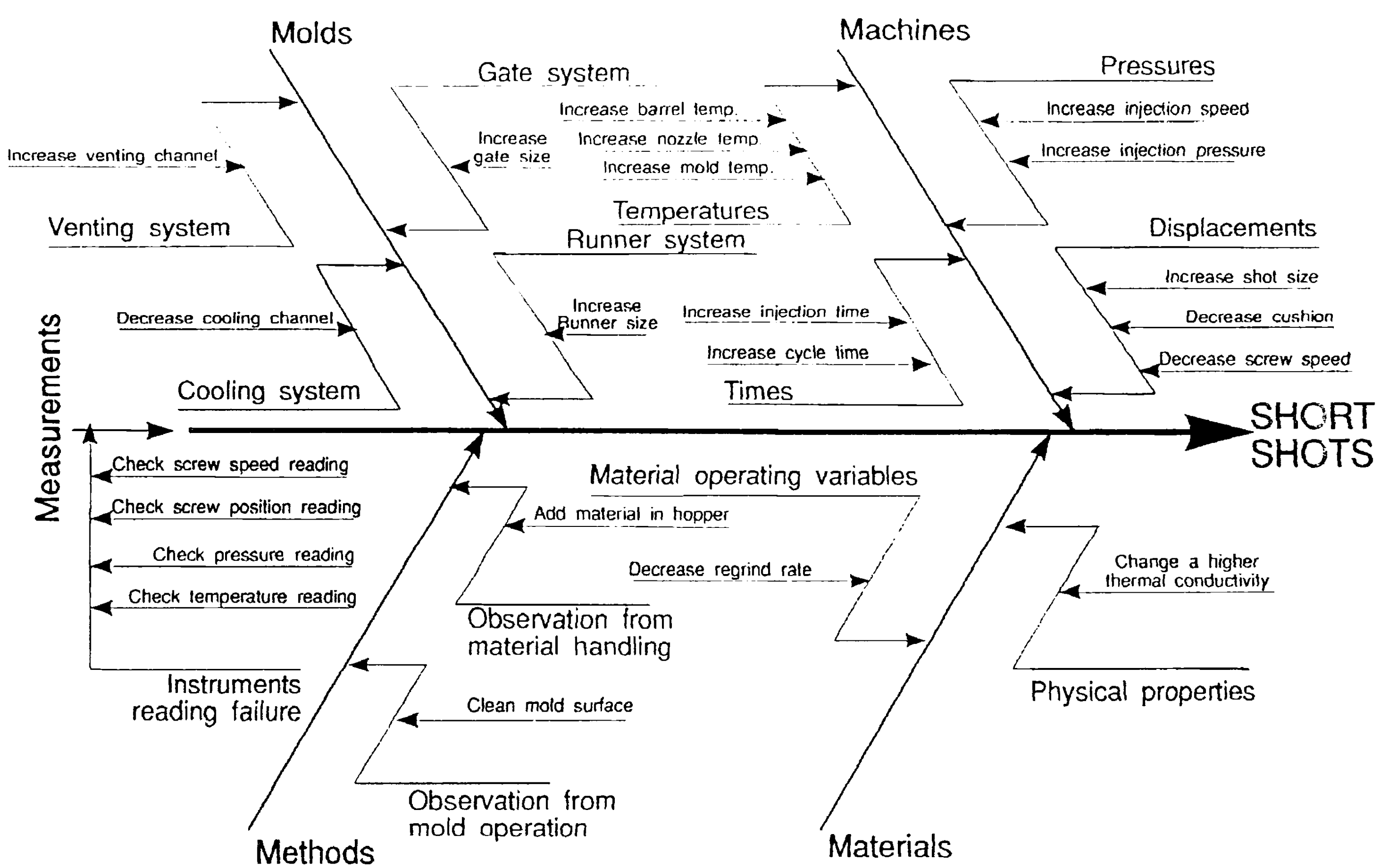


Figure 5.12 The total "analyzing" results of fishbone diagram for short shot deviation.

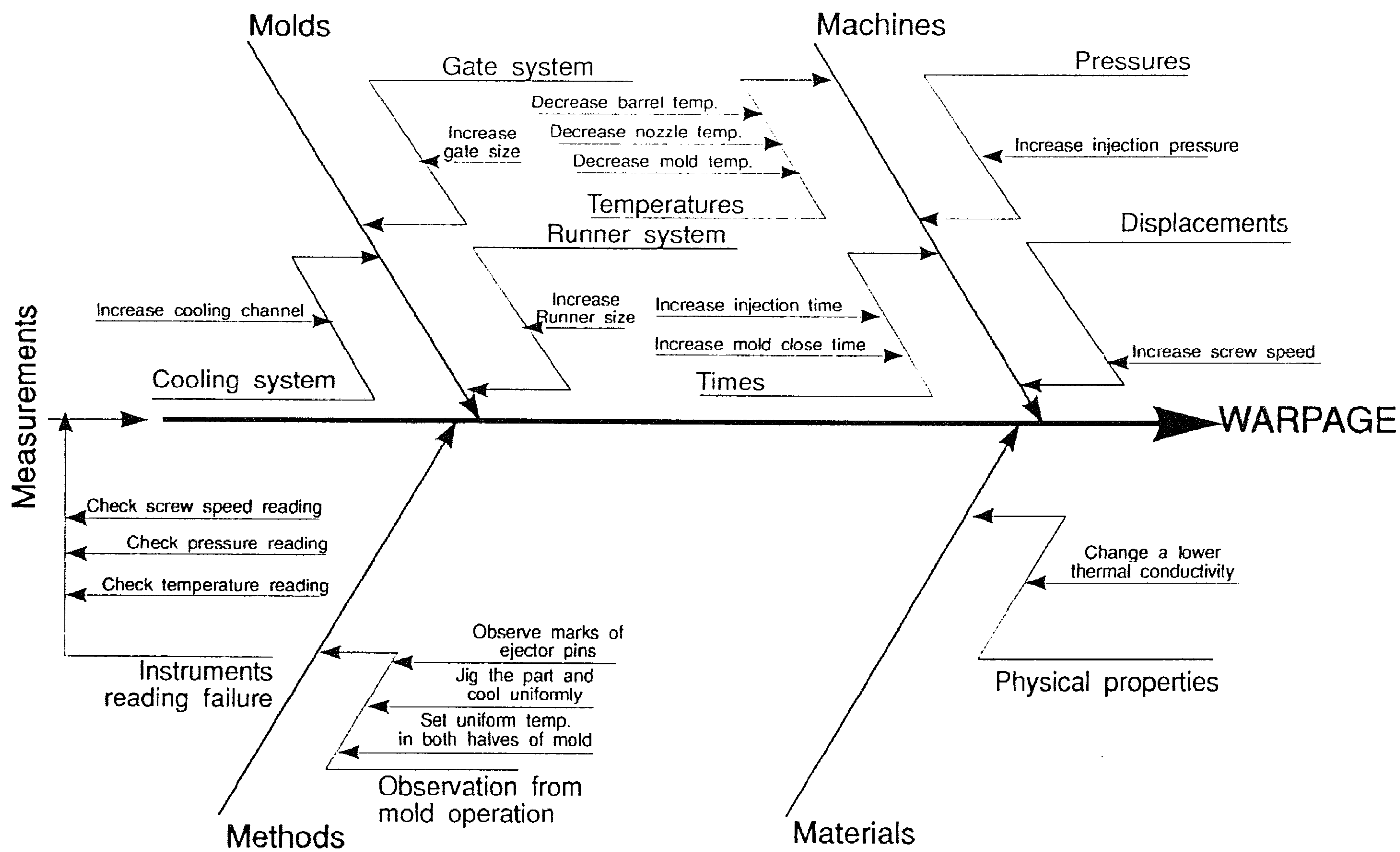


Figure 5.13 The total "analyzing" results of fishbone diagram for warpage deviation.

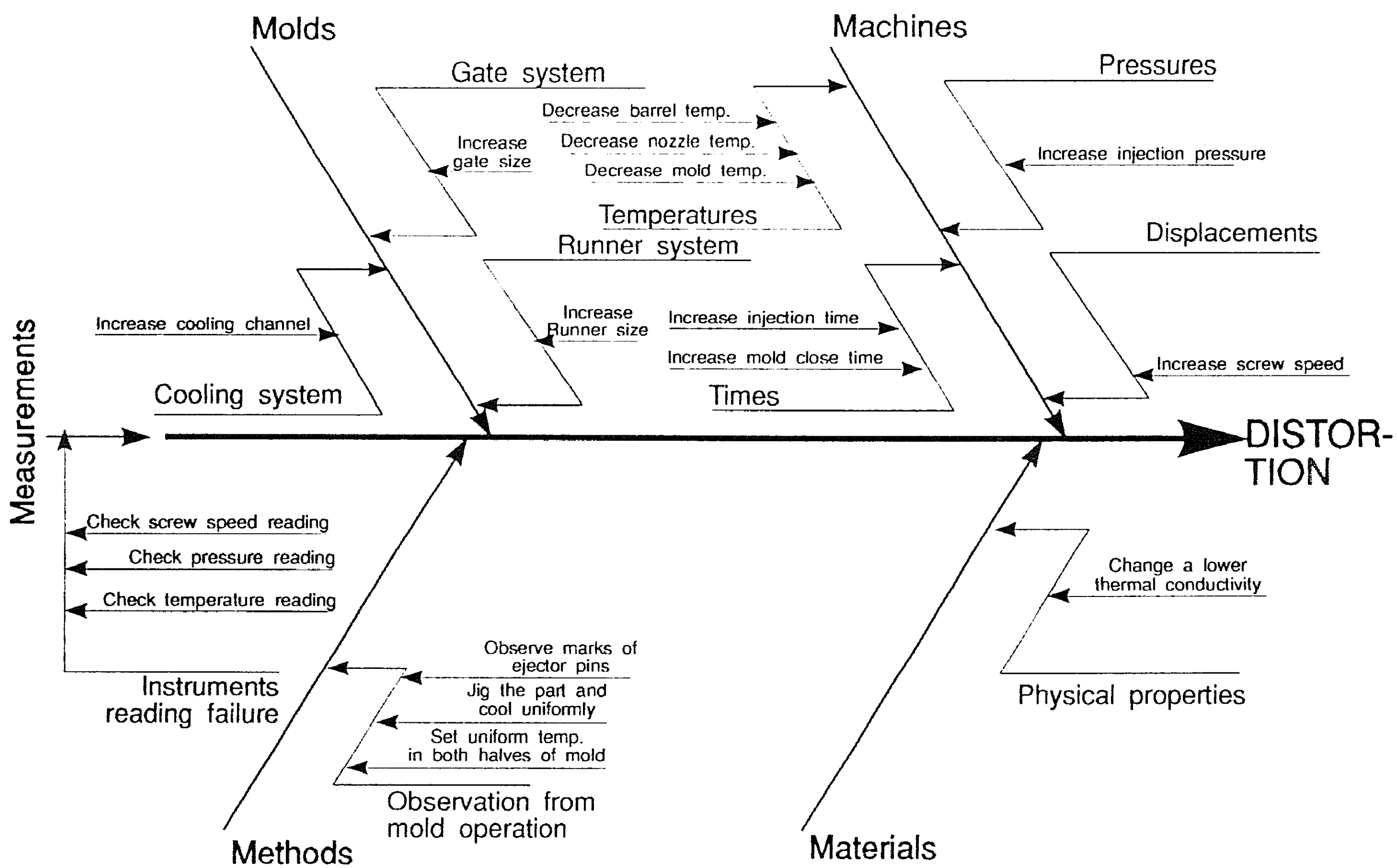


Figure 5.14 The total "analyzing" results of fishbone diagram for distortion deviation.

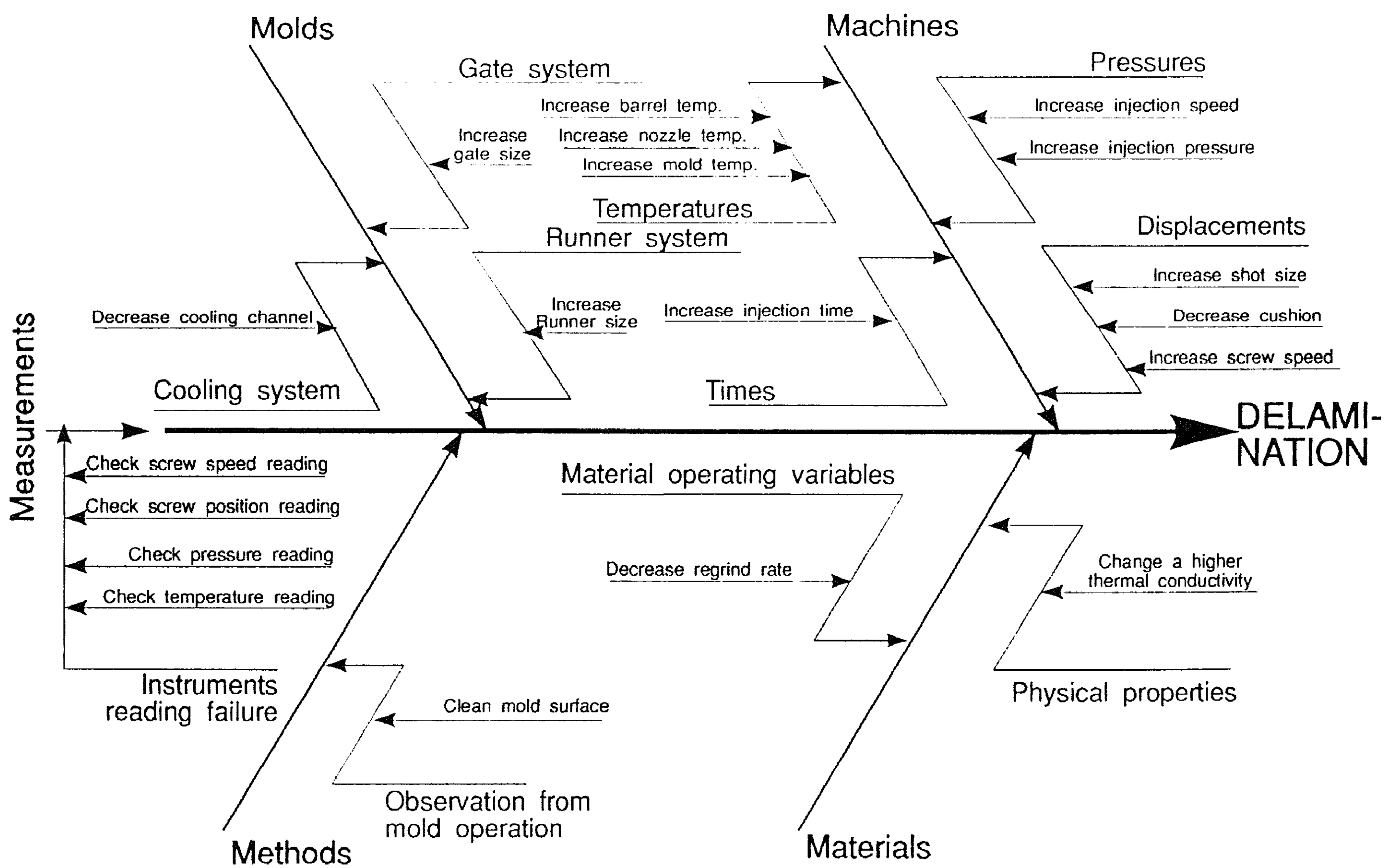


Figure 5.15 The total "analyzing" results of fishbone diagram for delamination deviation.

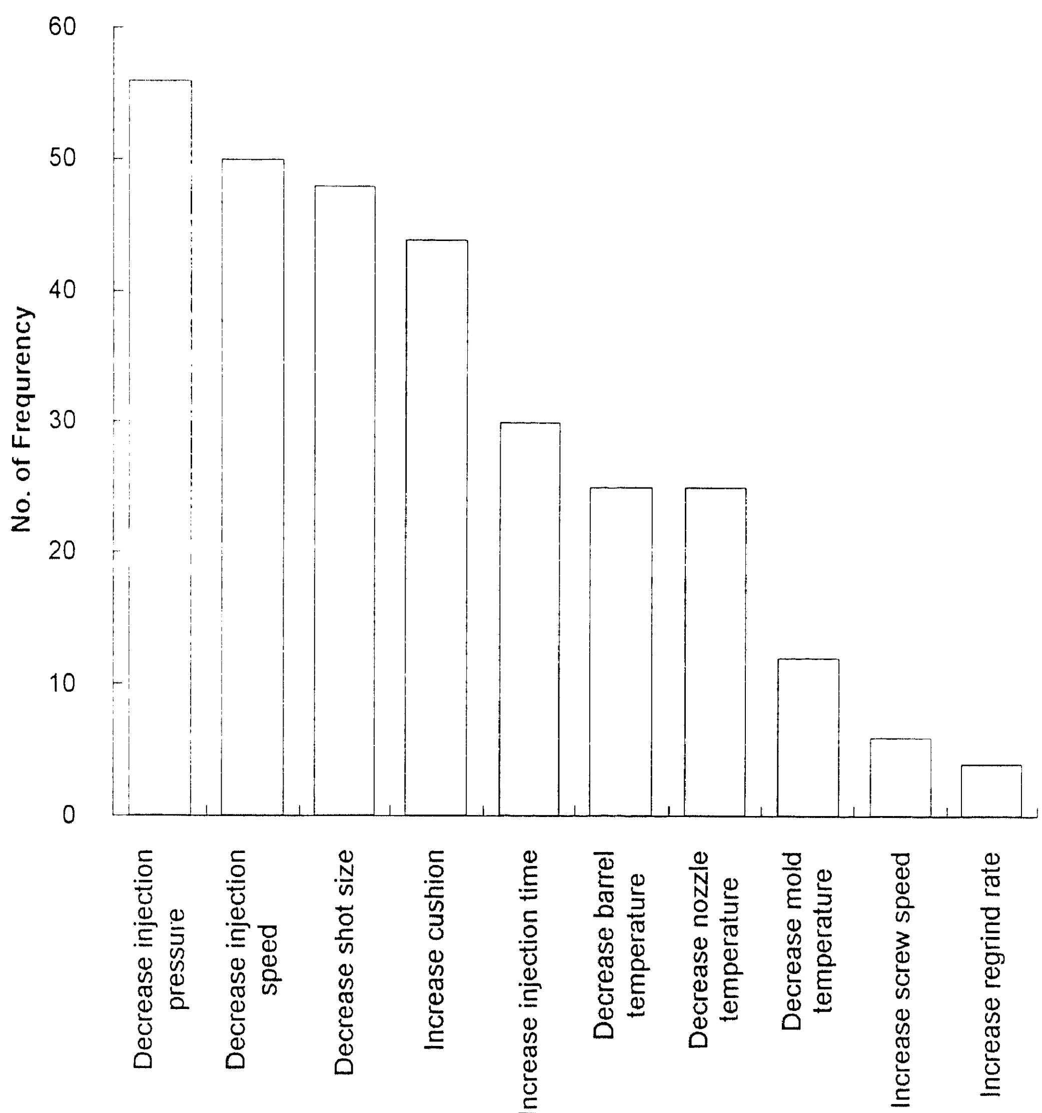


Figure 5.16 The Pareto diagram for flashing deviation.

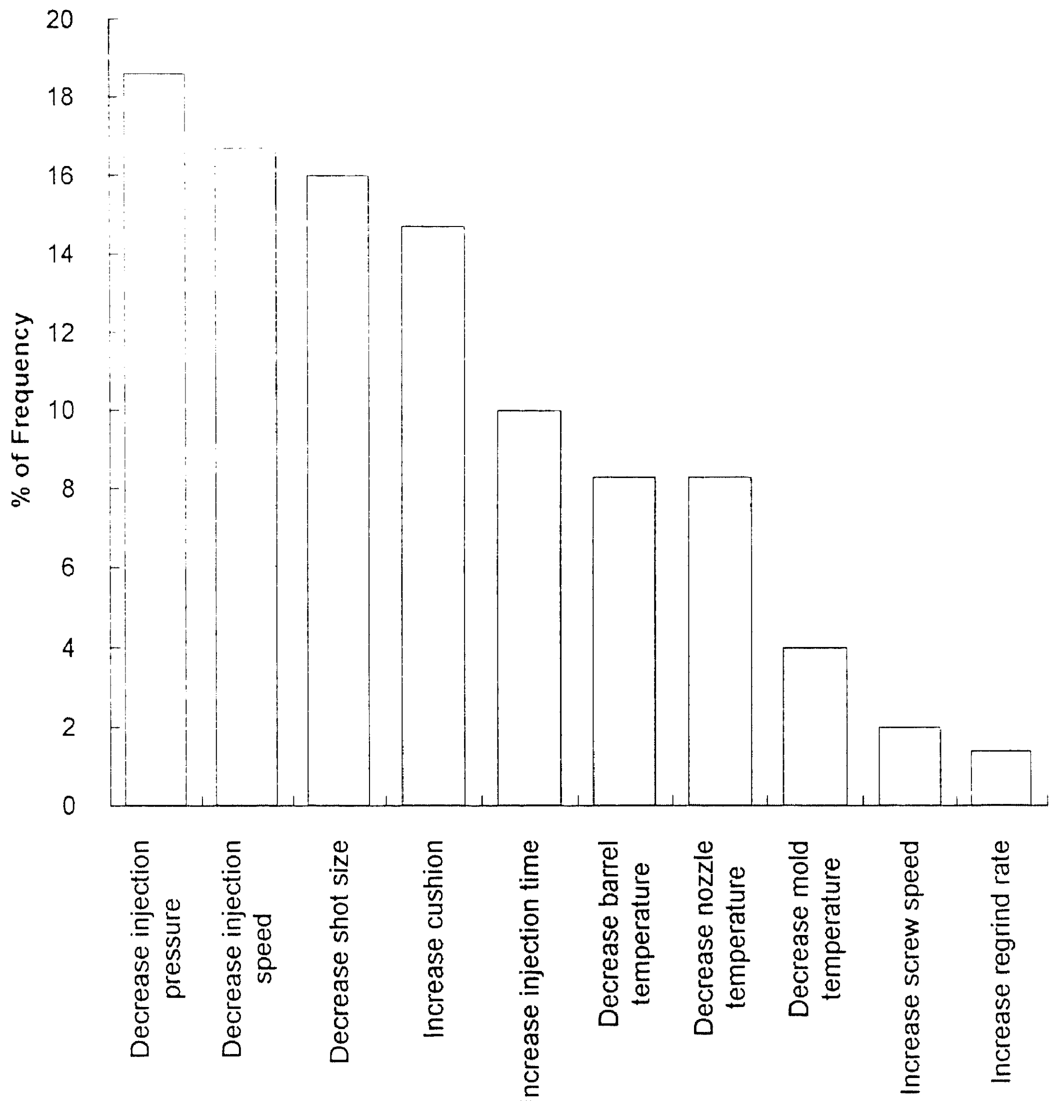


Figure 5.17 The total percentage frequency of the Pareto diagram for flashing deviation.

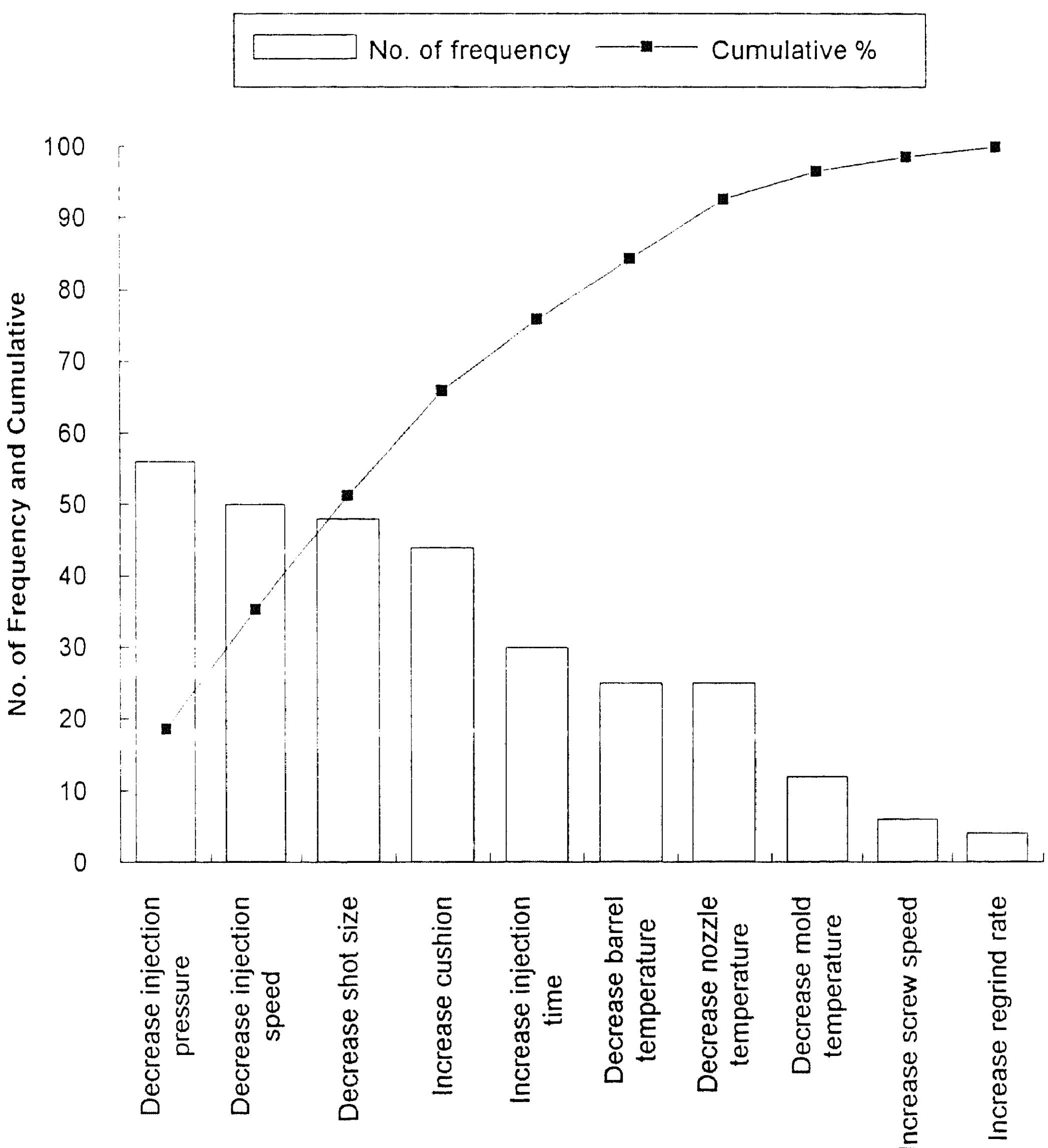


Figure 5.18 The number of frequency and cumulative percentage of the Pareto diagram for flashing deviation.

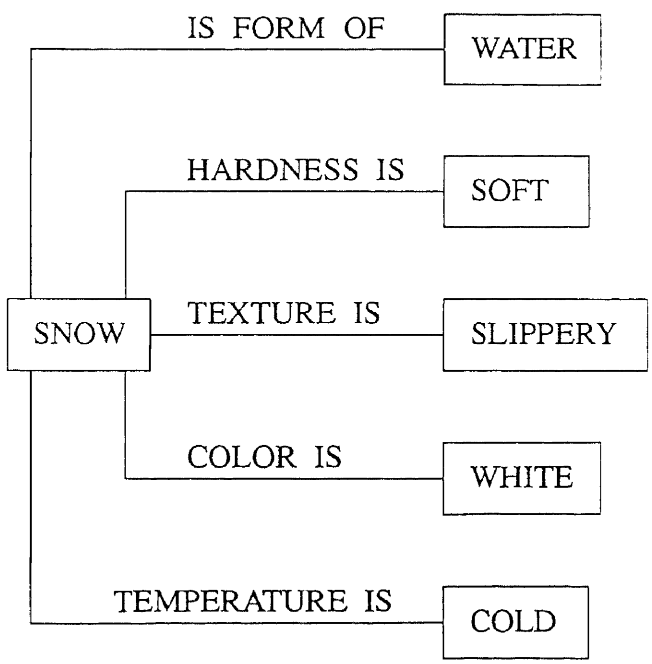


Figure 5.19 The network representation for defining the properties of snow.

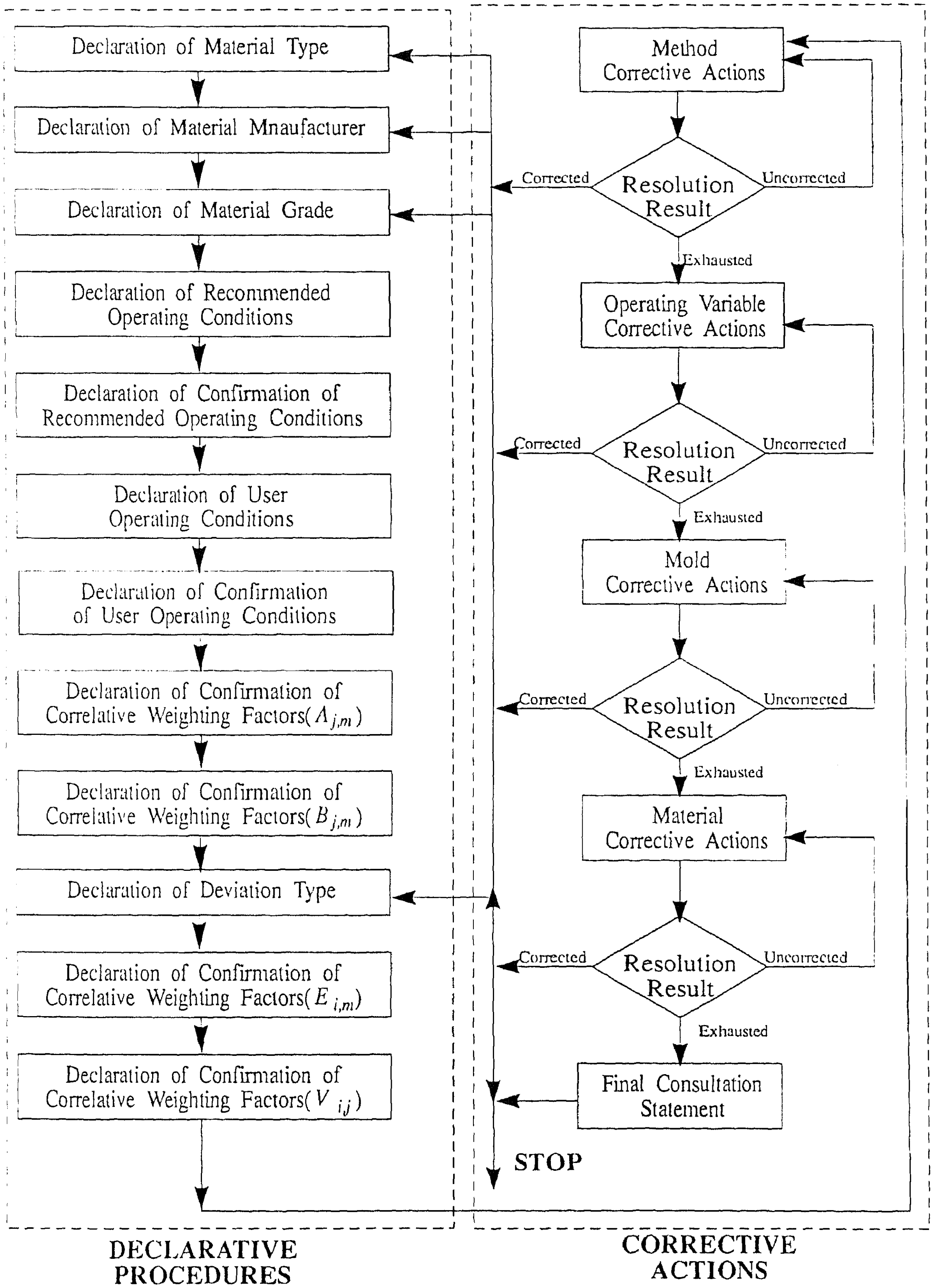


Figure 5.20 The searching strategy for the resolution procedures.

CHAPTER SIX
SYSTEM PERFORMANCE

6.1 Introduction

Based on the procedures in chapter five, an expert system for the injection molding of engineering thermoplastics has been developed. In this chapter, a molded part, as presented in Figure 6.1, is used to illustrate the expert system performance. The deviation, flashing, of acetal copolymer Celcon M-90, is used to interpret the resolution procedures of the expert system. The completed resolution procedures for the flashing deviation are listed in Appendix C-1. The remaining resolution procedures for deviations such as surface ripples, pit marks, splay marks, sink marks, voids, short shots, warpage, distortion, and delamination are then presented in Appendices C-2 to C-10 respectively.

As discussed in section 5.4.1, before the system can employ the resolution knowledge, there are several items of declarative knowledge required. Furthermore, in section 5.3.3.3, these declarative knowledge items are classified into global declarative knowledge and local declarative knowledge.

The global declarative knowledge includes the molded material, the manufacturer of the molded material, the grade of the molded material, the recommended operating conditions, the operating conditions, the correlative weighting factors of the operating variables and the inherent physical properties, and of the operating variables and the influencing physical properties. The local declarative knowledge includes the deviation type, the correlative weighting factors of the particular deviation and the operating variables, and of the particular deviation and the influencing physical properties. The global declarative knowledge is only dependent on the molded material, and is not dependent on the deviation. However, the local declarative knowledge is not only dependent on the molded material, but also is dependent on the deviation. Once, the declarative knowledge has been declared, the system then employs the resolution actions to eliminate the deviation.

In the next section the definition procedures for the global declarative knowledge will be introduced in detail. In section 6.3, the definition procedures for the local declarative knowledge are presented, and in section 6.4, the procedures of the resolution actions are presented.

6.2 Definition Procedures for Global Declarative Knowledge

As discussion in section 6.1, the first procedure is to define the global declarative knowledge. This includes the molded material, the manufacturer of the molded material, the grade of the molded material, the recommended operating conditions, the operating conditions, the correlative weighting factors of the operating variables and the inherent physical properties, and of the operating variables and the influencing physical properties. However, an introduction screen of this system is presented first, as shown in Figure 6.2.

Following the introduction screen, the molded material identification is then displayed by the system as shown in Figure 6.3. In Figure 6.3, the system provides an explanation facility to interpret why these procedures are taken by entering the function key "?". For instance, if the user wants to know the reasons to take this action and enters the function key "?", then the system displays an explanation statement as shown in Figure 6.4.

After the explanation statement, the selection menu of the molded material is then displayed as shown in Figure 6.5. In Figure 6.5, for example, the molded material was defined as acetal copolymer by entering the code number of the material, "CODE NUMBER = 1".

Once the molded material has been declared, the system then prompts the user to identify the molded material manufacturer as shown in Figure 6.6. In Figure 6.6, an explanation statement for selecting the molded material manufacturer is displayed since the function key "?" was entered. This explanation statement provides the reasons for this action. After the explanation statement of the molded material manufacturer, the system then prompts the user to select the material manufacturer as shown in Figure 6.7. The molded material manufacturer is defined by entering the code number of the manufacturer. For instance, the manufacturer code of HOECHST CELANESE, "CODE NUMBER = 1", was entered by the user. Now the molded material manufacturer is identified as acetal copolymer from HOECHST CELANESE.

The explanation facility performance and the identification procedures of the molded material grade are also the same as those for the molded material manufacturer as presented in Figures 6.8 and 6.9. In Figure 6.8, an explanation statement for interpreting the reasons for the identification of the molded material grade was displayed, and, in Figure 6.9, the selection procedure of the molded material grade was presented.

Once the definition of the molded material has been accomplished, the system then prompts the user to check whether the molded part has a prior history in the system. If it is the first time that the molded part has been entered into the system, then, the system will ask the user to input the name of the molded part as presented in Figure 6.10.

In Figure 6.10, the molded part was named "TEST1". The system then creates a sub-directory that is also named "TEST1" to store the data which is then modified or updated using the resolution procedures for the particular molded part. This data includes the recommended operating conditions, the user operating conditions, the correlative weighting factors of the operating variables and the inherent physical properties, of the operating variables and the influencing physical properties, of the operating variables and the particulate deviation, and of the influencing physical properties and particular deviation. When the molded part, "TEST1", is next used in the system for resolving the deviation, the system retrieve this data, which has been stored in a sub-directory, "TEST1", to help resolve deviations for the same molded part. This retrieval procedure is presented in Figure 6.11.

Once the name of the molded part has been declared, the system then prompts the user to define additional global declarative knowledge such as the recommended operating conditions, the user operating conditions, and the correlative weighting factors of the operating variables and the inherent physical properties, and of the operating variables and the influencing physical properties. First, the declaration procedures for the recommended operating conditions are employed as presented in Figures 6.12 and 6.13.

In Figure 6.12, an explanation screen for interpreting the reasons to confirm the recommended operating conditions is presented if the function key "?" is entered. In Figure 6.13, the system provides a confirmation mechanism for the user to allow modification of modifies these recommended operating conditions. The modification is based upon user experience and confidence in the data to perform the modifications. For instance, according to user experience, if the recommended conditions of the screw speed are likely to be incorrect, then the user can modify the recommended conditions by entering the code number of the function code, "YOUR ANSWER = 1", as shown in Figure 6.13.

Once the code number of the function code, "YOUR ANSWER = 1", has been entered, the system then prompts the user to the recommended operating conditions as shown in Figure 6.14. In Figure 6.14, based on user experience, the user considers the minimum recommended conditions of screw speed to be too high. So, the user wants to change the minimum recommended conditions of screw speed from 45 rpm to 40 rpm. At first, the system prompts the user to enter the code number of the modified operating variable, screw speed. In this case, for instance, the code number of screw speed is "12". Once the user has entered the code number of 12, i.e., "OPERATING VARIABLE CODE NUMBER =12", the system then prints out the old recommended operating conditions for the screw speed, and prompts the user to input the new recommended conditions for the screw speed.

Once the recommended conditions of the screw speed have been modified, the system will then print out the recommended operating conditions and again allow the user to confirm them as presented in Figure 6.15. At this time, if the user wishes to modify the recommended operating conditions, then the confirmation procedures discussed above will be repeated again. Otherwise, when the function code, "YOUR ANSWER = 2", is entered by the user, the system prompts the user to confirm the next global declarative knowledge, the operating conditions.

As presented in Figure 6.16, before the system prompts the user to input the operating conditions, the explanation statements for the input and to confirm the operating conditions are provided by the system when the function key, "?", was entered. After that, the user then be asked to input the operating conditions is sequence as presented in Figure 6.17. Once all the operating conditions have been input, the system then outputs the confirmation procedures for these inputs as presented in Figure 6.18.

This confirmation procedures are the same as for the recommended operating conditions. First, to determine if the confirmation procedures will be executed by entering control number of the function code, "YOUR ANSWER = ", as shown in Figure 6.18. If the confirmation procedures must be executed, then the same procedures for the confirmation of the recommended operating variables will be presented. Otherwise, the system will prompt the user for the confirmation procedures of the correlative weighting factors of the operating variables and the inherent physical properties, and of the operating variables and the influencing physical properties.

At this stage, the definition of the correlative weighting factors of the operating variables and the inherent physical properties will be employed first. In the discussion of section 5.4.2, these correlative weighting factors are defined as the influence degree between the operating variables and the inherent physical properties. As with the confirmation procedures for the recommended operating conditions, an explanation facility and a confirmation mechanism are also embedded in the declaration procedures. The procedure for activating the explanation facility of this correlative weighting factor is presented in Figure 6.19. The procedures for confirming the correlative weighting factor is displayed in Figure 6.20.

Once the confirmation procedures of the correlative weighting factors for the operating variables and the inherent physical properties have been completed, the system then prompts the user to confirm the correlative weighting factors of the operating variables and the influencing physical properties as displayed in Figures 6.21 and 6.22. In Figure 6.21, the procedures for the explanation facility for this correlative weighting factor are presented. In Figure 6.22, the confirmation procedures for this correlative weighting factor are presented.

Now that, all of the global declarative knowledge has been declared, the system can begin to define the local declarative knowledge including the deviation type, the correlative weighting factors of the operating variables and the particular deviation, and of the influencing physical properties. These declarative procedures are introduced in the following section.

6.3 Definition Procedures for Local Declarative Knowledge

As discussion in section 5.4.1, following the definition procedures of global declarative knowledge comes the definition of the local declarative knowledge. The local declarative knowledge includes the deviation type, the correlative weighting factors of the operating variable and the particular deviation, and of the influencing physical properties and the particular deviation.

In this section, the flashing deviation is used to illustrate the definition procedures. As discussed in section 5.4.1, the first definition procedures of local declarative knowledge is the type of deviation. Following the definition of the correlative weighting factors between the operating variables and the influencing physical properties, as presented in Figure 6.22, the system prompts the user to identify the deviation type as presented in Figures 6.23 and 6.24. In Figure 6.23, the procedures for inducing the explanation facility of the deviation type are presented. In Figure 6.24, the procedures for identifying the deviation type are presented. As with the declaration of the molded material, the deviation is defined as flashing by entering the code number of the function code, "DEVIATION CODE NUMBER = 9".

Once the deviation type has been defined, the system then selects the procedures to define the correlative weighting factors between the operating variables and the particular deviation, flashing. This performance is presented in Figures 6.25 and 6.26.

As with the definition procedures for the correlative weighting factors between the operating variables and the influencing physical properties, the system provides an explanation facility and a confirmation mechanism as presented in Figures 6.25 and 6.26. In Figure 6.25, the explanation statement was displayed when the function key "?" was entered. In Figure 6.26, the confirmation procedure of the correlative weighting factors for the operating variable and the flashing deviation is prompted. These procedures are essentially the same as those discussed in the previous section.

Based on the same definition procedures for the correlative weighting factors between the operating variables and the flashing deviation, the correlative weighting factors between the influencing physical properties and the particular deviation are confirmed as presented in Figures 6.27 and 6.28. In Figure 6.27, the explanation statement was display when the function key "?" was entered, and, in Figure 6.28, the confirmation procedures of this correlative weighting factors is prompted. These procedures are the same as the procedures for the correlative weighting factors between the operating variables and the flashing deviation.

Now, all of the necessary declarative knowledge has been defined. The deviation, flashing, of acetal copolymer, Celcon M-90 has been defined and all the parameters for inducing the resolution knowledge have similarly been defined. The system then employs its inference engine to search out the best resolution actions for the flashing deviation. The resolution action procedures are introduced in the following section.

6.4 PERFORMANCE OF THE RESOLUTION ACTIONS

Once all of the necessary declarative knowledge has been defined, the system can then employ its inference engine to search out the best resolution action for the user. As discussed in section 5.4.1, the resolution actions are employed by level as method corrective actions, operating variable corrective actions, mold corrective actions, and material corrective actions. In this section, all of the different level corrective actions will be introduced as detailed in the following discussion.

6.4.1 The Method Corrective Actions

Following the above definitions for correlative weighting factors between flashing and the influencing physical properties, as presented in Figure 6.28, the method corrective actions will be employed for eliminating or reducing the flashing first. Furthermore, as discussed in sections 3.3.7, 5.3.2.1, and 5.4.1, the method corrective actions for eliminating or reducing flashing include,

- cleaning mold surface, rule value = 0.95,
- using maximum clamping force, rule value = 0.90
- checking the temperature reading, rule value = 0.70
- checking the pressure reading, rule value = 0.65
- checking the screw speed reading, rule value = 0.60, and
- checking the screw position reading. rule value = 0.55.

Based on the sequence of the above rule values for the method corrective actions, the first method corrective actions for resolving flashing are presented in Figure 6.28.

As shown in Figure 6.29, the corrective action "CLEAN THE MOLD SURFACE" was output first. Following the corrective action, an explanation facility was provided by the system after the key, "?", was entered. After the corrective action, "CLEAN THE MOLD SURFACE", was employed to resolve the deviation, the system then required that the user input the resolution result as shown in Figure 6.29.

As mentioned in section 5.4.1, method corrective actions are *absolute corrective actions*, therefore, the response is constrained in two ways *CORRECTED* and *UNCHANGED*. As shown in Figure 6.30, the user response was *CORRECTED*, so the system prompted to reselect the menu as shown in Figure 6.31.

At this juncture, the user can re-run the program from the definition of the molded material type, from the molded material manufacturer, from the molded material grade, or from the deviation type. For the case shown in Figure 6.31, the user wants to re-define the molded material type, and inputs the response code key "1". The system then prompts the user for the material identification menu as presented in Figure 6.3, and all of the resolution procedures are repeated from the material identification onwards as discussed above. The same is true for the response key "2", "3", and "4" which prompt the user from the definition level of the material manufacturer, from the material grade, and from the deviation respectively. Furthermore, if the user wants to stop the process, then the response code key "0" is simply entered by the user.

However, when the response code key, "YOUR ANSWER", in Figure 6.30, is input as "2", the flashing deviation has not been eliminated by the corrective action, "CLEAN THE MOLD SURFACE", so the system prompts the user with the next method corrective action, "USE THE MAXIMUM CLAMPING FORCE", as presented in Figure 6.32, and then provides a resolution screen as presented in Figure 6.30. In Figure 6.32, the explanation statement for using this corrective action is also displayed.

Based on the above procedures, the method correctives action are employed one by one based on their rule values, until these corrective actions are exhausted. Once the method corrective actions have been exhausted, and if the deviation has still not been eliminated, the system then prompts the user with the next corrective actions level, the operating variable corrective actions, which are discussed in detail in the next section.

6.4.2 The Operating Variable Corrective Actions

As discussed in section 5.4.1, once the method corrective actions have been exhausted in the search to eliminate the deviation, and if the deviation has still not been eliminated, the operating variable corrective actions are then be employed by the system to eliminate the deviation. Furthermore, as discussed in section 5.4.2, when the operating variable correction actions are employed, the decision algorithm is used calculate the priority weighting factor of the operating variables and to determine the sequence of the resolution actions. The calculation of the priority weighting factors of the operating variables are given in Eqs. (5.2) to (5.7).

Based on Eqs (5.2) to (5.7), the first operating variable corrective action, "DECREASE SHOT SIZE (in) TO 2.4" is first employed as presented in Figure 6.33. In Figure 6.33, an explanation statement for using the corrective action, "DECREASE SHOT SIZE (in) TO 2.4", is also presented if the explanation facility control key, "?", is input. Once this corrective action has been employed, the system requires that the user respond with the result as shown in Figure 6.34.

As discussed in section 5.4.1, the operating variable corrective actions are *conditional corrective actions*, and therefore, the system provides three ways for the user to respond. These are "CORRECTED", "IMPROVED", and "NOT IMPROVED" as presented in Figure 6.34. In Figure 6.34, for instance, if the flashing deviation is eliminated using the corrective action, "DECREASE SHOT SIZE(in) TO 2.4", then the system will employ the self-learning mechanism to update the parameters. The performance of the self-learning mechanism is discussed detailed in section 7.4. At this stage, the system prompts the user as shown in Figure 6.35.

Following Figure 6.35, the system then prompts the user to re-select from the menu as shown in Figure 6.31. The performance of this re-select menu is the same as discussed in section 6.4.2.

If the resolution result in Figure 6.34 is 'IMPROVED", which means that the operating variable, "SHOT SIZE", has reduced the flashing deviation, then the system will employ the self-learning mechanism, in section 7.4, to update the correlative weighting factors. These correlative weighting factors are those for the operating variable on flashing deviation, for the operating variables on the inherent physical properties, and for the operating variables on the influencing physical properties. This self-learning mechanism is introduced in section 7.4.

Once these correlative weighting factors have been updated, the system then employs the decision algorithm to calculate the new priority weighting factors for the operating variables and to re-arrange the resolution sequence. Based on the above procedures, the second operating variable corrective action, "DECREASE SHOT SIZE(in) TO 2.2", will then be employed as shown in Figure 6.36.

After the corrective action outputs and user inputs shown in Figure 6.36 have been exchanged, the resolution response shown in Figure 6.34 is then employed to determine the corrective action. At this time, if the response of the resolution result is "UNCHANGED", which means that the operating condition, "CUSHION = 0.5 in", has no influence on the flashing deviation, then the system employs the self-learning mechanism, which will be discussed detailed in section 7.4, to update the correlative weighting factors. Then it employs these correlative weighting factors into the decision algorithm to calculate the new priority weighting factors for the operating variables. Once the new priority weighting factors have been calculated, then the next operating variable corrective actions will be employed as shown in Figure 6.37.

These resolution procedures for the operating variables will be explored until the flashing deviation is eliminated or all of the operating variable corrective action have been exhausted. Once all of the operating variables have been modified, and if the flashing deviation has still not been eliminated, then the next level of corrective actions, the mold corrective actions, will be employed as the discussion in the following section.

6.4.3 The Mold Corrective Actions

Once the system has exhausted all of the operating variable corrective actions, and has failed to eliminate the flashing deviation, the mold corrective actions are then employed in eliminate the flashing.

As discussed in sections 3.3.7, 5.3.2.1, 5.4.1, and Table 5.39 the mold corrective action and their rule-values are,

1. decrease gate size, rule-value = 0.9,
2. increase cooling channel size, rule-value = 0.8,
3. decrease runner size, rule-value = 0.7, and,
4. increase vent size, rule-value = 0.6.

Furthermore, at the mold corrective action level, the system provides dimensional modifications for the mold corrective actions. Moreover, the modified values of these corrective actions are always recommended as extreme values.

Based on the above definition of mold corrective actions, the first mold corrective action, "DECREASE GATE SIZE(in) TO 0.1" is output as shown in Figure 6.38, where an explanation statement is also presented when the function key code "?" is entered. Furthermore, comparing with the recommended the gate size of Figure 6.13, it shows that the recommended modification value for this correction action is equal to the minimum recommended condition of the gate size. The reason for this action has been discussed previously in section 5.4.1.

Following this corrective action, the system then requires the user to respond with the resolution result. The procedure is essentially the same as the method corrective actions, since the mold corrective actions are also *absolute corrective actions*. Therefore, the system provides a resolution result screen as presented in Figure 6.30 for the user.

As with the procedures for the resolution result of the method corrective actions , if the resolution result is "CORRECT", then the re-select menu in Figure 6.31 will be output. If the resolution result is "UNCHANGED", then the next mold corrective action, "INCREASE COOLING CHANNEL SIZE (in) to 0.5", is output as presented in Figure 6.39. The procedures will be modified until the flashing deviation is eliminated or all of the mold corrective actions have been employed, and the flashing deviation has still not been eliminated.

Whether the deviation is eliminated or not, the system will employ the self-learning mechanism to update the parameters, which have been modified in the above procedures, and to stored in the sub-directory, "TEST1", as discussed in section 6.2. The updated parameters include the recommended operating conditions, the actual operating conditions, and the correlative weighting factors of the operating variables and the inherent physical properties, of the operating variables and the influencing physical properties, of the operating variables and the particular deviation, and of the influencing physical properties and the particular deviation. At this time a update message is presented to the user as shown in Figure 6.35.

Once all of the mold corrective actions have been employed, and if the flashing deviation has still not been eliminated, then the system advances to the next corrective action level, the material corrective actions as discussed in the next section.

6.4.4 The Material Corrective Actions

When the mold corrective actions have been exhausted, and if the deviation is still exist, the system then employs the material corrective actions to eliminate the deviation. As discussed in sections 3.3.7, section 5.3.2.1, the material corrective action for eliminating the flashing deviation is "CHANGE TO A HIGHER THERMAL CONDUCTIVITY MATERIAL". At this time, the system then displays this corrective action screen as shown in Figure 6.40.

In Figure 6.40, not only is the physical property, which requires to be changed, indicated, but the original physical property is also displayed. This allows the user to search for an alter material based on the original values. Furthermore, in the discussion of section 5.4.1, when the material corrective action is employed, the material properties such as the mechanical properties, the electrical properties, the optical properties, and the chemical properties are also altered. These material properties influence the functional performance of the product when altered. To avoid this situation, a caution statement to warm the user is displayed as shown in Figure 6.40.

Since material corrective actions are *absolute corrective action*, then, as for with method corrective actions of the absolute corrective action as shown in Figure 6.30 is output. If the resolution response is "CORRECTED", then the re-select menu, as shown in Figure 6.31, will be presented. However, when the resolution response is "UNCHANGED', then no further corrective actions are available in the system. Therefore, the final statement is employed as shown in Figure 6.41. Following this final statement, the re-select menu as shown in Figure 6.31 is output. Now, the consultation procedures for resolving the flashing deviation have been completed. And at this juncture or before the deviation has been eliminated or reduced in nearly all cases. In Appendix C-1, the resolution procedure for the flashing deviation that has been declared in above text is presented in its entirely.

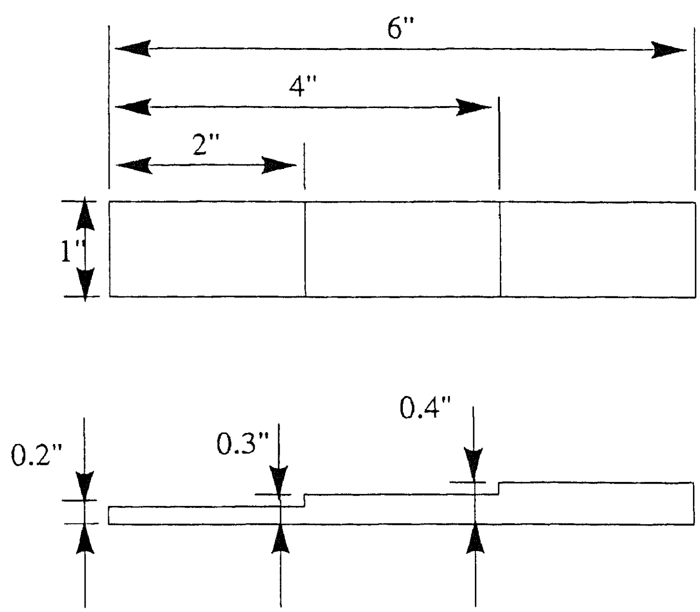


Figure 6.1 The molded part uses for interpreting the flashing deviation.

```
************************************************************
         WELCOME TO THE N.J.I.T. EXPERT SYSTEM FOR
    THE INJECTION MOLDING OF ENGINEERING THERMOPLASTICS
       THIS SYSTEM IS USED TO ELIMINATE OR TO REDUCE
          DEVIATIONS IN THE INJECTION MOLDING OF
                 ENGINEERING THERMOPLASTICS
*************************************************************
```

Figure 6.2 The introduction screen of the system

```
************************************************************
             IDENTIFICATION OF MOLDING MATERIAL
************************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
```

Figure 6.3 The molded material identification screen.

```
*************************************************************
ACTION: MATERIAL IDENTIFICATION
REASON: SINCE DIFFERENT MATERIALS HAVE DIFFERENT PHYSICAL PROPERTIES
WHICH CAUSE DIFFERENT DEGREES OF INFLUENCE FOR THE SPECIFIC DEVIATION
THE SYSTEM REQUIRES THE MATERIAL TO BE IDENTIFIED
*************************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.4 The explanation screen for the material type identification.

```
PLEASE INDICATE YOUR MOLDING MATERIAL BY ENTERING THE CODE NUMBER
1. ACETAL_COPOLYMER
2. ACETAL_HOMOPOLYMER
3. NYLON_6
4. NYLON_66
5. POLYCARBONATE
6. PBT
7. PET
8. POLYSTYRENE
9. ABS
10. SAN
11. HIGH_IMPACT_POLYSTYRENE
12. EXIT
MATERIAL CODE NUMBER = 1
YOUR MATERIAL INDICATE NUMBER IS : 1
YOUR MATERIAL NAME IS : ACETAL_COPOLYMER
```

Figure 6.5 The material type selection screen.

```
*************************************************************
      IDENTIFICATION OF MOLDING MATERIAL MANUFACTURER
*************************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
***********************************************************
ACTION: MATERIAL MANUFACTURER IDENTIFICATION
REASON: SINCE DIFFERENT MATERIAL MANUFACTURERS PRODUCTS HAVE
DIFFERENT PHYSICAL PROPERTIES WHICH CAUSE DIFFERENT DEGREES OF
INFLUENCE FOR THE SPECIFIC DEVIATION THE SYSTEM REQUIRES THE
MANUFACTURER TO BE IDENTIFIED
***********************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.6 The explanation screen for the molded material manufacturer identification.

```
PLEASE INDICATE YOUR MOLDING MATERIAL MANUFACTURER
BY ENTERING THE MANUFACTURER CODE NUMBER
1. HOECHST_CELANESE
2. DuPONT
3. BASF
4. MITSUBISHI
5. POLYPLASTICS
MANUFACTURER CODE NUMBER = 1
YOUR MANUFACTURER CODE NUMBER IS : 1
YOUR MANUFACTURER IS HOECHST_CELANESE
```

Figure 6.7 The molded material manufacturer selection screen.

```
***********************************************************
    IDENTIFICATION OF MOLDING MATERIAL GRADE FOR CELCON
***********************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
*********************************************************
ACTION: MATERIAL GRADE IDENTIFICATION
REASON: SINCE DIFFERENT MATERIAL GRADES HAVE DIFFERENT
PHYSICAL PROPERTIES WHICH CAUSE DIFFERENT DEGREES OF INFLUENCE FOR
THE SPECIFIC DEVIATION AND BECAUSE DIFFERENT MATERIAL GRADES HAVE
DIFFERENT RECOMMENDED OPERATING CONDITIONS WHICH WILL BE EMPLOYED
INTO DECISION ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY
WEIGHTING FACTORS WHICH INFLUENCES THE DEVIATION RESOLUTION
PROCEDURES THE SYSTEM REQUIRES THE MATERIAL GRADE TO BE IDENTIFIED
***********************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.8 The explanation screen for the molded material grade.

```
PLEASE INDICATE YOUR MOLDING MATERIAL GRADE
BY ENTERING THE GRADE CODE NUMBER
 1. U-10        2. EF-25       3. GB-25        4. GC-25A
 5. M-25        6. UV-25       7. WR-25_BLACK 8. M-50
 9. EC-90+     10. EP-90      11. LW-90       12. LW-90-S2
13. LW-90-SC  14. M-90       15. MC-90       16. MC-90-HM
17. TX-90     18. TX-90+     19. UV-90       20. WR-90_BLACK
21. M-140     22. AS-270     23. M-270       24. MC-270
25. MC-270-HM 26. AS-450     27. M-450       28. LW-GS-S2
GRADE CODE NUMBER = 14
YOUR CELCON GRADE CODE IS : 14
YOUR CELCON GRADE NAME IS : M-90
```

Figure 6.9 The molded material grade identification screen.

```
DOES THE MOLDED PART HAVE A PRIOR HISTORY IN THE SYSTEM?
PLEASE INDICATE YOUR ANSWER BY ENTERING CODE NUMBER
1. YES
2. NO
YOUR ANSWER = 2
YOUR ANSWER IS THAT THE MOLDED PART DOES NOT HAVE PRIOR HISTORY IN
THIS SYSTEM
PLEASE ENTER THE 'NEW' MOLDED PART NAME WITH 8 CHARACTERS
'NEW' MOLDED PART NAME = TEST1
```

Figure 6.10 The identification screen of the molded part name.

```
DOES THE MOLDED PART HAVE A PRIOR HISTORY IN THIS SYSTEM?
PLEASE INDICATE YOUR ANSWER BY ENTERING CODE NUMBER
1. YES
2. NO
YOUR ANSWER = 1
YOUR ANSWER IS THAT THE MOLDED PART DOES HAVE A PRIOR HISTORY IN THIS
SYSTEM
PLEASE ENTER THE 'OLD' MOLDED PART NAME WITH 8 CHARACTERS
'OLD' MOLDED PART NAME = TEST1
```

Figure 6.11 The retrieve procedures for the name of molded part.

```
***********************************************************
          CONFIRMATION OF RECOMMENDED OPERATING
                CONDITIONS FOR CELCON M90
***********************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
***********************************************************
ACTION: CONFIRMATION OF THE RECOMMENDED OPERATING CONDITIONS
REASON: SINCE THE RECOMMENDED OPERATING CONDITIONS ARE EMPLOYED IN
THE DECISION ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY
WEIGHTING FACTORS WHICH INFLUENCE THE DEVIATION RESOLUTION PROCEDURES
THE SYSTEM REQUIRES THE ACCURACY OF THE RECOMMENDED OPERATING
CONDITIONS
PLEASE PRESS ANY KEY TO CONTINUE
```

Figure 6.12 The explanation screen for the confirmation of the recommended operating conditions.

```
*************************************************************
    THE RECOMMENDED OPERATING CONDITIONS FOR CELCON M90
*************************************************************
                                NAME    MINIMUM    MAXIMUM
           BARREL_TEMPERATURE(F)     380.00     420.00
           NOZZLE_TEMPERATURE(F)     390.00     430.00
             MOLD_TEMPERATURE(F)     170.00     200.00
         INJECTION_PRESSURE(psi)   15000.00   20000.00
             INJECTION_TIME(sec)       5.00      10.00
            MOLD_CLOSE_TIME(sec)      15.00      30.00
             MOLD_OPEN_TIME(sec)      15.00      30.00
                 CYCLE_TIME(sec)      30.00      60.00
              DECOMPRESSION(sec)       1.00       4.00
                   SHOT_SIZE(in)       2.00       3.20
                     CUSHION(in)       0.10       0.50
                 SCREW_SPEED(rpm)      45.00      60.00
                  REGRIND_RATE(%)       0.00      30.00
   INJECTION_SPEED(1MAX2MED3MIN)       2.00       1.00
                   GATE_SIZE(in)       0.10       0.20
        COOLING_CHANNEL_SIZE(in)       0.30       0.50
                 RUNNER_SIZE(in)       0.20       0.40
        VENTING_CHANNEL_SIZE(in)       0.05       0.10
DO ANY RECOMMENDED CONDITIONS NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 1
```

Figure 6.13 The confirmation menu for the recommended operating conditions.

```
WHICH RECOMMENDED CONDITIONS NEED TO BE CHANGED
PLEASE INDICATE BY ENTERING THE CODE NUMBER
1 .BARREL_TEMPERATURE(F)
2 .NOZZLE_TEMPERATURE(F)
3 .MOLD_TEMPERATURE(F)
4 .INJECTION_PRESSURE(psi)
5 .INJECTION_TIME(sec)
6 .MOLD_CLOSE_TIME(sec)
7 .MOLD_OPEN_TIME(sec)
8 .CYCLE_TIME(sec)
9 .DECOMPRESSION(sec)
10 .SHOT_SIZE(in)
11 .CUSHION(in)
12 .SCREW_SPEED(rpm)
13 .REGRIND_RATE(%)
14 .INJECTION_SPEED(1MAX2MED3MIN)
15 .GATE_SIZE(in)
16 .COOLING_CHANNEL_SIZE(in)
17 .RUNNER_SIZE(in)
18 .VENTING_CHANNEL_SIZE(in)
OPERATING VARIABLE CODE NUMBER = 12
OLD RECOMMENDED CONDITIONS OF SCREW_SPEED(rpm) ARE
   MINIMUM   MAXIMUM
     45.00     60.00
PLEASE INPUT YOUR NEW RECOMMENDED CONDITIONS FOR
SCREW_SPEED(rpm)
MINIMUM = 40.000000
MAXIMUM = 60.000000
```

Figure 6.14 The modification screen for the recommended operating variable.

```
***********************************************************
   THE RECOMMENDED OPERATING CONDITIONS FOR CELCON M90
***********************************************************
                              NAME    MINIMUM    MAXIMUM
         BARREL_TEMPERATURE(F)     380.00     420.00
         NOZZLE_TEMPERATURE(F)     390.00     430.00
           MOLD_TEMPERATURE(F)     170.00     200.00
       INJECTION_PRESSURE(psi)   15000.00   20000.00
           INJECTION_TIME(sec)       5.00      10.00
          MOLD_CLOSE_TIME(sec)      15.00      30.00
           MOLD_OPEN_TIME(sec)      15.00      30.00
               CYCLE_TIME(sec)      30.00      60.00
            DECOMPRESSION(sec)       1.00       4.00
                 SHOT_SIZE(in)       2.00       3.20
                   CUSHION(in)       0.10       0.50
             SCREW_SPEED(rpm)       40.00      60.00
               REGRIND_RATE(%)       0.00      30.00
  INJECTION_SPEED(1MAX2MED3MIN)      2.00       1.00
                 GATE_SIZE(in)       0.10       0.20
       COOLING_CHANNEL_SIZE(in)      0.30       0.50
               RUNNER_SIZE(in)       0.20       0.40
       VENTING_CHANNEL_SIZE(in)      0.05       0.10
DO ANY RECOMMENDED CONDITIONS NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 2
```

Figure 6.15 The updated data of the recommended operating conditions.

```
***********************************************************
            INPUTTING THE OPERATING CONDITIONS
***********************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
***********************************************************
ACTION: INPUTTING OPERATING CONDITIONS
REASON: SINCE THE OPERATING CONDITIONS ARE EMPLOYED IN THE DECISION
ALGORITHM TO CALCULATE THE OPERATING VARIABLES PRIORITY WEIGHTING
FACTOR WHICH INFLUENCES THE DEVIATION RESOLUTION PROCEDURES THE
SYSTEM REQUIRES THE OPERATING CONDITIONS TO BE IDENTIFIED
***********************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.16 The explanation screen for the user input operating conditions.

```
************************************************************
           PLEASE INPUT YOUR OPERATING CONDITION
************************************************************
           BARREL_TEMPERATURE(F)  =     400.00
           NOZZLE_TEMPERATURE(F)  =     410.00
             MOLD_TEMPERATURE(F)  =     185.00
         INJECTION_PRESSURE(psi)  =   17500.00
             INJECTION_TIME(sec)  =       7.00
            MOLD_CLOSE_TIME(sec)  =      22.00
             MOLD_OPEN_TIME(sec)  =      22.00
                 CYCLE_TIME(sec)  =      44.00
              DECOMPRESSION(sec)  =       3.00
                   SHOT_SIZE(in)  =       2.40
                     CUSHION(in)  =       0.30
                SCREW_SPEED(rpm)  =      50.00
                 REGRIND_RATE(%)  =      20.00
   INJECTION_SPEED(1MAX2MED3MIN)  =       2.00
                   GATE_SIZE(in)  =       0.20
        COOLING_CHANNEL_SIZE(in)  =       0.30
                 RUNNER_SIZE(in)  =       0.20
        VENTING_CHANNEL_SIZE(in)  =       0.05
```

Figure 6.17 The user input operating conditions screen.

```
************************************************************
                THE USER OPERATING VARIABLE
************************************************************
           BARREL_TEMPERATURE(F)  =     400.00
           NOZZLE_TEMPERATURE(F)  =     410.00
             MOLD_TEMPERATURE(F)  =     185.00
         INJECTION_PRESSURE(psi)  =   17500.00
             INJECTION_TIME(sec)  =       7.00
            MOLD_CLOSE_TIME(sec)  =      22.00
             MOLD_OPEN_TIME(sec)  =      22.00
                 CYCLE_TIME(sec)  =      44.00
              DECOMPRESSION(sec)  =       3.00
                   SHOT_SIZE(in)  =       2.40
                     CUSHION(in)  =       0.30
                SCREW_SPEED(rpm)  =      50.00
                 REGRIND_RATE(%)  =      20.00
   INJECTION_SPEED(1MAX2MED3MIN)  =       2.00
                   GATE_SIZE(in)  =       0.20
        COOLING_CHANNEL_SIZE(in)  =       0.30
                 RUNNER_SIZE(in)  =       0.20
        VENTING_CHANNEL_SIZE(in)  =       0.05
DO ANY OPERATING CONDITIONS NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 2
```

Figure 6.18 The confirmation screen for the user operating conditions.

```
*************************************************************
      CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS
BETWEEN OPERATING VARIABLES AND INHERENT PHYSICAL PROPERTIES
*************************************************************
?
*************************************************************
ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLES AND INHERENT PHYSICAL PROPERTIES
REASON: SINCE THE CORRELATIVE WEIGHTING FACTORS BETWEEN OPERATING
VARIABLES AND INHERENT PHYSICAL PROPERTIES ARE EMPLOYED IN THE
DECISION ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY
WEIGHTING FACTORS WHICH INFLUENCES THE DEVIATION RESOLUTION
PROCEDURES THE SYSTEM REQUIRES THE CORRELATIVE WEIGHTING FACTORS
BETWEEN OPERATING VARIABLES AND INHERENT PHYSICAL PROPERTIES
***************************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.19 The explanation screen for the confirmation of the correlative weighting between the operating variables and the inherent physical properties.

```
*********************************************************
   THE CORRELATIVE WEIGHTING FACTORS BETWEEN OPERATING
  VARIABLES AND INHERENT PHYSICAL PROPERTIES FOR CELCON M90
*********************************************************
```

NAME	THER_COND	THER_DIFF	SPEC_VOL	SPEC_HEAT
BARREL_TEMPERATURE(F)	-0.45	-0.40	0.45	-0.35
NOZZLE_TEMPERATURE(F)	-0.50	-0.45	0.45	-0.30
MOLD_TEMPERATURE(F)	-0.45	-0.45	0.30	-0.40
INJECTION_PRESSURE(psi)	-0.25	-0.30	0.35	-0.45
INJECTION_TIME(sec)	0.25	0.30	-0.25	0.35
MOLD_CLOSE_TIME(sec)	-0.50	-0.50	0.50	-0.40
MOLD_OPEN_TIME(sec)	0.10	0.05	0.05	0.25
CYCLE_TIME(sec)	-0.45	-0.45	0.20	-0.45
DECOMPRESSION(sec)	0.10	0.05	0.05	0.25
SHOT_SIZE(in)	0.15	0.25	0.20	0.30
CUSHION(in)	0.20	0.25	0.30	0.35
SCREW_SPEED(rpm)	0.15	0.10	0.05	0.35
REGRIND_RATE(%)	0.35	0.30	0.10	0.10
INJECTION_SPEED(1MAX2MED3MIN)	-0.25	-0.30	0.35	-0.45

```
DO ANY CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLES AND INHERENT PHYSICAL PROPERTIES NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 2
```

Figure 6.20 The confirmation screen for the correlative weighting of the operating variables and the inherent physical properties.

```
************************************************************
  CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN
   OPERATING VARIABLES AND INFLUENCING PHYSICAL PROPERTIES
************************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
************************************************************
ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLES AND INFLUENCING PHYSICAL PROPERTIES
REASON: SINCE THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE OPERATING
VARIABLES AND INFLUENCING PHYSICAL PROPERTIES ARE EMPLOYED IN THE
DECISION ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY
WEIGHTING FACTORS WHICH INFLUENCES THE DEVIATION RESOLVED PROCEDURES
THE SYSTEM REQUIRES THE CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLES AND INFLUENCING PHYSICAL PROPERTIES
**************************************************************
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.21 The explanation screen for the confirmation of the correlative weighting between the operating variables and the influencing physical properties.

```
*******************************************************
    THE CORRELATIVE WEIGHTING FACTORS BETWEEN OPERATING
VARIABLES AND INFLUENCING PHYSICAL PROPERTIES FOR CELCON M90
*******************************************************
```

NAME	SHEAR_RATE	SHEAR_STRESS	VISCOSITY
BARREL_TEMPERATURE(F)	-0.45	0.50	0.40
NOZZLE_TEMPERATURE(F)	-0.50	0.50	0.40
MOLD_TEMPERATURE(F)	-0.35	0.45	0.40
INJECTION_PRESSURE(psi)	-0.45	0.45	0.40
INJECTION_TIME(sec)	0.35	-0.25	-0.30
MOLD_CLOSE_TIME(sec)	0.45	-0.40	-0.45
MOLD_OPEN_TIME(sec)	0.20	0.20	0.05
CYCLE_TIME(sec)	0.30	-0.30	0.45
DECOMPRESSION(sec)	0.20	0.20	0.05
SHOT_SIZE(in)	0.05	0.10	0.05
CUSHION(in)	0.05	0.10	0.05
SCREW_SPEED(rpm)	0.45	-0.35	-0.45
REGRIND_RATE(%)	-0.40	0.30	0.50
INJECTION_SPEED(1MAX2MED3MIN)	-0.45	0.45	0.40

```
DO ANY CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLES AND INFLUENCING PHYSICAL PROPERTIES NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 2
```

Figure 6.22 The confirmation screen for the correlative weighting factors of the operating variables and the influencing physical properties.

```
**************************************************************
              IDENTIFICATION OF THE DEVIATION
**************************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
**************************************************************
ACTION: DEVIATION IDENTIFICATION
REASON: SINCE DIFFERENT DEVIATIONS HAVE DIFFERENT DEVIATION
PROCEDURES. THE SYSTEM REQUIRES THE DEVIATION TO BE IDENTIFIED
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.23 The explanation screen for the deviation identification.

```
PLEASE ENTER THE CODE NUMBER OF THE DEVIATION
1. SHORT_SHOT
2. PITMARKS
3. SURFACE_RIPPLES
4. SPLAY_MARKS
5. WARPAGE
6. SINK_MARKS
7. PART_DISTORTION
8. VOIDS
9. FLASHING
10. DELAMINATION
DEVIATION CODE NUMBER = 9
YOUR DEVIATION CODE NUMBER = 9
YOUR ARE EXPERIENCING FLASHING
```

Figure 6.24 The deviation type identification screen.

```
**************************************************************
     CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS
 BETWEEN THE DEVIATION FLASHING AND THE OPERATING VARIABLES
**************************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE
DEVIATION AND THE OPERATING VARIABLES
REASON: SINCE THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE
DEVIATIONS AND THE OPERATING VARIABLES ARE EMPLOYED IN THE DECISION
ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY WEIGHTING
FACTORS WHICH INFLUENCE THE DEVIATION RESOLUTION PROCEDURE THE SYSTEM
REQUIRES THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE DEVIATIONS AND
THE OPERATING VARIABLES TO BE IDENTIFIED
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.25 The explanation screen for the confirmation of the correlative weighting between the operating variables and the flashing deviation.

```
**********************************************************
          CORRELATIVE WEIGHTING FACTOR BETWEEN
      OPERATING VARIABLE AND THE DEVIATION FLASHING
**********************************************************
          BARREL_TEMPERATURE(F) = -0.22
          NOZZLE_TEMPERATURE(F) = -0.11
            MOLD_TEMPERATURE(F) = -0.22
        INJECTION_PRESSURE(psi) = -0.50
           INJECTION_TIME(sec) = -0.27
          MOLD_CLOSE_TIME(sec) =  0.00
           MOLD_OPEN_TIME(sec) =  0.00
               CYCLE_TIME(sec) =  0.00
            DECOMPRESSION(sec) =  0.00
                 SHOT_SIZE(in) = -0.43
                   CUSHION(in) =  0.40
              SCREW_SPEED(rpm) =  0.05
               REGRIND_RATE(%) =  0.04
   INJECTION_SPEED(1MAX2MED3MIN) = -0.45
DO ANY CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLES AND DEVIATIONS NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 2
```

Figure 6.26 The confirmation screen for the correlative weighting factors of the operating variables and the flashing deviation.

```
**********************************************************
 CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN
THE DEVIATION FLASHING AND THE INFLUENCING PHYSICAL PROPERTIES
**********************************************************
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
**********************************************************
ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE
DEVIATION FLASHING AND THE INFLUENCING PHYSICAL PROPERTIES
REASON: SINCE THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE DEVIATION
AND INFLUENCING PHYSICAL PROPERTIES ARE EMPLOYED IN THE DECISION
ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY WEIGHTING
FACTORS WHICH INFLUENCES THE DEVIATION RESOLUTION PROCEDURE THE
SYSTEM REQUIRES TO THE CORRELATIVE WEIGHTING FACTORS BETWEEN THE
DEVIATIONS AND INFLUENCING PHYSICAL PROPERTIES TO BE IDENTIFIED
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.27 The explanation screen for the confirmation of the correlative weighting between the influencing physical properties and the flashing deviation.

```
*************************************************************
          CORRELATIVE WEIGHTING FACTORS BETWEEN
   INFLUENCING PHYSICAL PROPERTIES AND DEVIATION FLASHING
*************************************************************
          SHEAR_RATE = -0.45
        SHEAR_STRESS = -0.40
           VISCOSITY =  0.50
DO ANY CORRELATIVE WEIGHTING FACTORS BETWEEN
INFLUENCING PHYSICAL PROPERTIES AND DEVIATIONS NEED TO BE CHANGED
YES = 1, OR NO = 2
YOUR ANSWER = 2
```

Figure 6.28 The confirmation screen for the correlative weighting factors of the influencing physical properties and the flashing deviation.

```
*************************************************************
          BEGIN TO RESOLVE THE DEVIATION FLASHING
*************************************************************
SUGGESTED ACTION:
DOES THE MOLD STICKING OCCUR OR DOES FOREIGN CONTAMINATION EXIST
IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
REASON: WHEN MOLD STICKING OCCURS OR FOREIGN CONTAMINATION EXIST, THE MOLD
SURFACES CANNOT CLOSE TIGHTLY CAUSES FLASHING OCCURS.
THEREFORE, THE MATERIAL MUST BE REMOVED
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.29 The corrective action screen for the method corrective action "clean the mold surface".

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATE YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 1
YOUR ANSWER IS THAT THE DEVIATION WAS CORRECTED
```

Figure 6.30 The resolution result screen for the absolute corrective actions.

```
IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 1
```

Figure 6.31 The reselection menu.

```
SUGGESTED ACTION:
IS THE CLAMPING FORCE MAXIMUM
IF NOT, PLEASE USE THE MAXIMUM CLAMPING FORCE
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
REASON:
THE INJECTION PRESSURE IS GREATER THAN THE CLAMPING FORCE AND CAUSES
FLASHING. THEREFORE, USING THE MAXIMUM CLAMPING FORCE CAN ENSURE THE
INJECTION PRESSURE NOT EXCEED THE CLAMPING FORCE
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.32 The corrective action screen for the method corrective action "use the maximum clamping force".

```
************************************************************
BEGIN THE OPERATING VARIABLE CORRECTIVE ACTIONS
************************************************************
SUGGESTED ACTION:
DECREASE CUSHION (in) TO      2.40
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
SINCE THE MATERIAL OVER-CHARGE IS ONE OF THE MAJOR CAUSES FOR THE
FLASHING. DECREASING CUSHION WILL DECREASE THE QUANTITY. OF THE
MATERIAL FILLING THE MOLD
THEREFORE, THE SYSTEM PROVIDE THIS SUGGESTED ACTION
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.33 The corrective action screen for the flashing deviation of the operating variable correction action "decrease shot size to 2.4 in".

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATE YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 1
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN CORRECTED
```

Figure 6.34 The resolution result screen for the conditional corrective action.

```
************************************************************
CURRENTLY, THE SYSTEM IS UPDATING YOUR DATA, PLEASE WAIT.
************************************************************
```

Figure 6.35 The update message screen.

```
*************************************************************
SUGGESTED ACTION:
DECREASE CUSHION (in) TO     2.20
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATE YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 3
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
```

Figure 6.36 The corrective action screen for the flashing deviation of the operating variable correction action "decrease shot size to 2.2 in".

```
*************************************************************
SUGGESTED ACTION:
DECREASE INJECTION PRESSURE (PSI) TO     5500
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
*************************************************************
THE INJECTION PRESSURE IS GREATER THAN THE CLAMPING FORCE AND CAUSES
FLASHING.
THEREFORE, THE SYSTEM PROVIDE THIS SUGGESTED ACTION AS SEQUENCE.
PLEASE ENTER ANY KEY TO CONTINUE
```

Figure 6.37 The corrective action screen for the flashing deviation of the operating variable correction action "decrease injection pressure to 5500 psi".

```
*************************************************************
BEGIN THE MOLD CORRECTIVE ACTIONS
*************************************************************
SUGGESTED ACTION:
DECREASE GATE SIZE (in) TO  0.10
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
*************************************************************
SINCE THE OVERFLOW IN THE MOLD CAVITY IS ONE OF THE MAJOR CAUSES FOR
FLASHING. DECREASING GATE SIZE WILL RESOLVE THE DEVIATION
THEREFORE, THE SYSTEM PROVIDES THIS ACTION
PLEASE PRESS ANY KEY TO CONTINUE
C
```

Figure 6.38 The corrective action screen for the flashing deviation of the mold correction action "decrease gate size to 0.1 in".

```
SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
?
**************************************************************
INCREASING COOLING CHANNEL SIZE WILL DECREASE THE MATERIAL
TEMPERATURE AND DECREASE THE MATERIAL VISCOSITY IN THE MOLD CAVITY.
THEREFORE, THE SYSTEM PROVIDES THIS ACTION
PLEASE PRESS ANY KEY TO CONTINUE
```

Figure 6.39 The corrective action screen for the flashing deviation of the mold correction action "increase cooling channel size to 0.5 in".

```
**************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
**************************************************************
PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY
CONSIDER THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES, THE ELECTRICAL PROPERTIES, THE OPTICAL PROPERTIES, AND
THE
CHEMICAL PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT
```

Figure 6.40 The corrective action screen for the flashing deviation of the material correction action.

```
**************************************************************
THERE IS NO FURTHER CORRECTIVE ACTION AVAILABLE.
PLEASE CONSULT WITH MOLDING EXPERT
OR RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
**************************************************************
```

Figure 6.41 The final consultation of the system.

CHAPTER SEVEN
DISCUSSION

7.1 Resolution Procedure

Based on the definition of the declarative knowledge in sections 6.2 to 6.3 which presented in Table 7.1, the complete resolution procedures for the deviation, flashing, of acetal copolymer, Celcon M-90, is listed in Appendix C-1. In the resolution procedures listing for the flashing deviation, the resolution procedures are developed accordingly to the following assumptions.

Firstly, the flashing deviation is not necessarily corrected at each level of the corrective actions. This includes in the *absolute corrective actions* such as method corrective actions, mold corrective actions, and material corrective actions, and in the *conditional corrective actions* such as operating variable corrective actions. Second, in the level of the operating variable corrective actions, the resolution result is always "DEVIATION HAS BEEN IMPROVED". Therefore, according to the above assumptions, the final consultation statement, *THERE IS NO FURTHER CORRECTIVE ACTION AVAILABLE. PLEASE CONSULT WITH A MOLDING EXPERT OR A RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM.* Furthermore, all of the corrective actions in each corrective action level can be employed completely and be examined.

Following the above assumptions, and the definitions in sections 6.2 and 6.3, the level of the method corrective actions is employed by the system first. Furthermore, according the discussion in sections 3.3.7, 5.3.2.1, and 5.4.1, the method corrective actions for eliminating the flashing deviation are ranked as follows,

1. clean mold surface,
2. use maximum clamping force,
3. check the temperature reading,

4. check the pressure reading,
5. check the screw speed reading, and,
6. check the screw position reading.

Comparing the listing of the method corrective actions above with Appendix C-1, the system can employ these method corrective actions exactly. This means that the performance of the inference engine in the method corrective action level can exactly simulate the molding experts resolution skills to induce the correct method corrective actions.

Furthermore, the sequence of the method corrective actions employed, which has been discussed in section 5.4.1, is based on the degree of difficulty for resolving the deviations and the advice of the molding experts. In this case, for example, according to prior study and the advice of the molding experts, the primary cause for the flashing deviation is inappropriately molding closing. The mold surface sticking to the material or contamination are the most possible causes for the inappropriate closing. In addition, the unclean mold surface is easily observed and removed. Therefore, the "clean mold surface" is employed as the first ranked for eliminating the flashing deviation.

The corrective action, "use the maximum clamping force", is employed as the second ranked for the following reasons. The other primary cause for flashing is that the material pressure against the mold surface is greater than the clamping force. This causes mold slip and results in the material exuding from the parting line of the mold. Also, using the maximum clamping force is the most efficient way to overcome this cause. Therefore, the system provides the corrective action, "use the maximum clamping force", as the second ranked in the level of the method corrective actions.

Following the corrective action, "use the maximum clamping force", are corrective actions such as checking the temperature reading, the pressure reading, the screw speed reading, and the screw position reading. These corrective actions are employed to ensure that there are no misleading operating conditions which will influence the deviation.

However, to accurate by read these instruments, is time consumingly and reduces the productivity during the process. Therefore, the rank of these corrective actions is less than the above corrective actions and they are employed late. Furthermore, the degree of the difficulty for accurate reading of these instruments has been input by the molding experts, and ranked in sequence.

As discussed in section 5.4.1, following the method corrective actions for eliminating the deviation are the operating variable corrective actions. In this level of corrective action, the employment procedures are based on the priority weighting factors of the operating variables. These priority weighting factors are determined by the decision algorithm which was discussed in section 5.4.2. Furthermore, section 3.3.7 and section 5.3.2.1 explain that the operating variable corrective actions for eliminating the flashing deviation includes,

1. decrease barrel temperature,
2. decrease nozzle temperature,
3. decrease mold temperature,
4. decrease injection pressure,
5. decrease injection speed,
6. decrease shot size,
7. increase cushion,
8. increase screw speed,
9. increase regrind rate and,
10. decease injection time.

Based on the calculation results of the priority weighting factors, the operating variable corrective actions are ranked as presented in Table 7.2.

In Table 7.2, there are several interesting results. First, when the operating condition has been reached, or the initial operating condition is, the extreme recommended condition, which is the anticipation to be reached for resolving the problem, then, this operating variable corrective action will never be employed in the resolution procedures. This is a reasonable occurrence for the following reasons. First, for the some of the operating variables, once the operating conditions reach the extreme recommended condition, there is no further adjustment be made for the operating condition. For instance, reducing the injection speed to resolve the flashing deviation, the operating condition of the injection speed reaches the "minimum" level. There is no further operating condition of the injection speed less than the "minimum" level.

Second, once the operating conditions reach the extreme recommended conditions, in general, further adjustments of these operating variables will cause other deviations. For instance, reducing the injection pressure to resolve the flashing deviation, the system provides an adjustment quantity less than the minimum recommended condition of the injection pressure. Therefore, there is the possibility that other deviations such as short shots, surface ripples, pin marks, sink marks, voids, warpage, distortion, and delamination will be generated. To prevent this situation, the system is not therefore allowed to adjust operating conditions such that the extreme recommended conditions are reached.

Third, the correlative weighting factors between the operating variables and the deviation are the primary influence for the priority weighting factors of the operating variables. In this case, for example, before the first operating variable corrective action, "decrease shot size to 2.4 in" is provided for resolving the problem, the priority weighting factors of the operating variables, and the comparison of the priority weighting factors and the correlative weighting factors between operating variables and the deviation are presented in Table 7.3.

In Table 7.3, the rank of the priority weighting factors are provided in rank order of the correlative weighting factors between the operating variables and the deviation. The only exceptions are the rank of "shot size", and "injection speed". For the operating variable, "injection speed", this is because its initial condition is already at the extreme recommended condition. When the operating condition reaches the extreme recommended condition, the system will not employ the operating variable corrective action anymore. Therefore, in this case, the priority weighting factors of the injection speed are calculated as equal to zero. For the operating variable, "shot size", it is due to the shot size operating variable is less influenced by the inherent and the influencing physical properties which will reduce the value of the priority weighting factors. This results in the correlative weighting factors of the shot size and the flashing being the primary influence on the priority weighting factors of the shot size and so it has the greatest rank.

Furthermore, in Table 7.2, it can be seen that each time the operating variable reaches the extreme recommended condition, the other operating variable corrective actions will employed first. For instance, as presented in Table 7.2, before the corrective action, "decrease shot size to 2.0 in", which is the extreme corrective action of the operating variable shot size, is induced, the corrective action, "decrease injection pressure to 5500 psi", is first employed. This is because when using the extreme recommended conditions in resolving the deviation, it is possible to generate other deviations. For instance, too low a shot size is a possible causes of the short shot deviation. Therefore, before the extreme operating variable corrective action is induced, the other corrective actions will be induced first, and only then the extreme operating variable corrective action. This performance can prevent generation of other deviations.

From the above discussion, it is clear that the performance of the decision algorithm in the system can simulate the molding experts resolution decision for inducing the resolution procedures of the operating variable corrective actions. Furthermore, in section 7.2, the simulation results of these operating variable corrective action will be presented. From the simulation results, it is clear that the operating variable corrective actions provided by the system can meaningful by influence the primary parameters of the deviation.

Following the operating variable corrective actions, the mold corrective actions and the material corrective actions as presented in Appendix C-1 are activated. In the discussion of section 5.4.1, it is explained that the mold corrective actions are employed by the rank of their rule value. Furthermore, these rule values are dependent on the degree of difficulty for resolving the deviation and the input of the molding experts as shown in Table 5.38. In Appendix C-1, it is show that the induction procedures of the mold corrective actions are followed their rule value order. Moreover, in Appendix C-1, it is show that the adjustment quantity of these mold corrective actions is always at the extreme recommended conditions of these mold corrective variable. The reason for this performance has been discussed in section 5.4.1. It is because the modification of these parameters is a complicated procedure which is costly and time consumingly. Therefore, the mold corrective action is always suggested at the extreme recommended conditions.

The last level of corrective actions provided by the system are the material corrective action. Due to the performance of products which alter with materials are changed, it is frequently necessary to consider if the part performance is absorbed. It is costly and time consumingly. When the material is altered, the functional properties such as the physical properties, the mechanical properties, the electrical properties, the optical properties, and the chemical properties are also altered. Therefore, in this program, the material corrective action is placed to the lowest priority in the resolution procedures. However, once the other corrective actions have been exhausted, and the deviation can still be not eliminated, then the material corrective actions will be induced to eliminate the deviation. When the material corrective action is induced, there is a caution statement provided by the system to warn the user that altered material results in the altered product performance. The caution statement is *"CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES, THE ELECTRICAL PROPERTIES, THE OPTICAL PROPERTIES, AND THE CHEMICAL PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"*.

However, if the material corrective actions can still not eliminate the deviation, then the system has exhausted all corrective actions for the user. In this situation, the final consultation statement will displayed for the user to indicate the other way of resolving the problem. This last consultation statement is presented as " *THERE IS NO FURTHER CORRECTIVE ACTION AVAILABLE. PLEASE CONSULT WITH A MOLDING EXPERT OR A RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM*".

From all of the above discussions, it is clear that the inference engine of the system simulate the molding experts resolution skills and provide the optimum resolution procedures for the user. Furthermore, in the next section, the mold filling simulation package, MOLDFLOW, is used to confirm the operating variable corrective actions influence on the control parameters of the deviation.

7.2 Simulation Results

The purpose of using the MOLDFLOW mold filling simulation package is to confirm the operating variable corrective actions provided by the system can meaningful by influence the control parameters for the deviation.

MOLDFLOW is a software package which uses the finite element method and the finite difference method to simulate the injection molding process. It includes a mold filling simulation, a mold packing simulation, and a mold cooling simulation. However only the mold filling simulation was available for this work. Therefore, the influence of the operating variable corrective actions on the deviation could be confirmed only during the mold filling stage.

To use MOLDFLOW, the molded part must be meshed. In this study, a molded part, as presented in Figure 6.1, is meshed as shown in Figure 7.1. Once the model of the molded parted has been meshed, the system then requires the user to input material information such as the material type, the material manufacturer, and the material grade, and the operating conditions such as, the material temperature, the mold temperature, the injection time, and the injection gate node. Based on the input data, MOLDFLOW then outputs the simulation results such as shear rate, and shear stress of each element, temperature, and pressure of each node, and material filling distribution over time. The complete simulation results are presented in Appendix D.

In this study, all of the output is used to demonstrate that the operating variable correlative actions can influence the deviation. Furthermore, due to the limitation of the package, the operating variables that confirm their influence on the deviation only include the material temperature, the mold temperature, and the injection time.

However, in Table 7.2, the first six ranked corrective actions including corrective actions such as "decrease shot size", "decrease injection pressure", and "increase cushion" are presented. These corrective actions will directly influence the primary cause of the flashing deviation. For corrective actions, "decrease shot size" and "increase cushion", the shot size is reduce and the material pressure against to the mold surface is therefore reduced. As discussed in section 3.3.6, when the material pressure is larger than the clamping force, mold slip results and causes the material flow through from the parting line. This explains why the corrective actions, "decrease shot size" and "increase cushion" have a significant resolution ability for flashing. Therefore, the system provides a high rank for these corrective actions. Similarly the corrective actions, "decrease shot size", and "decrease injection pressure" receive high rank.

In Table 7.2, the seven ranked operating variable corrective action is "decrease injection time to 5 second". Prior to the introduction of this corrective action, the operating conditions (barrel temperature = 400°F, mold temperature = 180°F, and injection time = 6 sec) were simulated in MOLDFLOW. For purposes of clearly, this operating condition is termed *operating condition A*. The simulation results of *operating condition A* is presented in Tables 7.4, 7.5 and 7.6. In Table 7.4, the maximum shear rate and the maximum shear stress of each element is presented. In Table 7.5, the pressure history is presented and in Table 7.6 the temperature and pressure at each node is presented.

Once "decrease injection time to 5 sec" has been employed, the operating conditions (barrel temperature = 400°F, mold temperature = 180°F, and injection time = 6 sec) are simulated again in MOLDFLOW. This operating condition is termed *operating condition B*. In Tables 7.7, 7.8, and 7.9, the simulation results for *operating conditions B* are presented. The maximum shear rate and the maximum shear stress of each element are shown in Table 7.7. The pressure history is shown in Table 7.8, and the temperature and the pressure of each node are shown in Table 7.9.

Based on the output data of MOLDFLOW in Tables 7.4-7.9, comparisons of *operating condition A* and *operating condition B* are shown in Figures 7.2-7.5. In Figure 7.2, the comparison of the maximum shear rate of each element is presented. In Figure 7.3, the comparison of the maximum shear stress of each element is presented. The comparison of the temperature and the pressure for each node is presented in Figure 7.4 and 7.5 respectively. Furthermore, the percentage variations of each parameter between *operating condition A* and *operating condition B* are summarized in Table 7.10. These figures and tables are used to illustrate that the operating variables can exactly influence the controlled parameters of the deviation in the following discussion.

As discussed in section 3.3.6, the causes of flashing include the overcharged material, mold wearing, too a high cavity pressure, and too a polymer low viscosity. For the operating variables such as barrel temperature, mold temperature, and injection time, the influencing parameters can control through the cavity pressure and the viscosity. A higher material viscosity can be accomplished by increasing shear stress, and by decreasing shear rate and temperature. Lowering the cavity pressure can be accomplished by decreasing the material pressure. Furthermore, as shown in Figure 3.11, too small the pressure gradient between the injection stage and the compression stage can also cause flashing. This is caused by a lower pressure gradient between the injection stage and the compression stage allows a higher internal stress development in the compression stage, and results in holding pressure development during the packing stage. To increase this pressure gradient, the material pressure at end of the injection stage can be increased.

From Table 7.10, the percentage difference of the material temperature between *operating conditions A* and *operating conditions B* is only changed slightly, 1.0%, so it can be concluded that if the injection time is decreased, the material temperature will change only slightly and not can provide a enough significant improvement of the influent parameter, viscosity, for resolving the flashing.

However, for the maximum shear rate and the maximum shear stress of each element, the percentage difference increases to a significant value. As presented in Table 7.10, the average percentage change of the maximum shear rate is equal to 20.4%, and the maximum shear stress is equal to 12.5%. The increased maximum shear rate and shear stress increase the flow velocity which results in a high material pressure. In this case, the average material pressure increases to 12.2% as shown in Table 7.10. This results in a higher material pressure at the end of the injection stage. As shown in Table 7.10, the final material pressure increases 12.2% from *operating condition B* over *operating condition A*. As discussed above, the final material pressure in the injection stage results from the pressure gradient increase, and can significantly improve flashing. Therefore, it can be concluded that the corrective action "decrease injection time" can significantly influence the controlling parameter for eliminating flashing.

Following the procedures employed by the operating variable corrective actions as shown in Table 7.2, the next corrective action after "decrease injection time to 5 sec" is "decrease barrel temperature to 390°F". At this stage, the operating conditions then become barrel temperature = 390°F, mold temperature = 180°F, and injection time = 5 sec. In this discussion, these operating conditions are termed *operating condition C*. The simulation results for operating condition C are presented in Tables 7.11-7.13. The maximum shear rate and the maximum shear stress of each element are presented in Table

7.11. The pressure history is presented in Table 7.12. The temperature and the pressure of each node are presented in Table 7.13. Furthermore, the comparison of these parameters between *operating condition B* and *operating condition C* is presented in Figure 7.6, Figures 7.7-7.9, and Table 14.

Comparison of the maximum shear rate and the maximum shear stress of each element is presented in Figures 7.6 and 7.7 respectively. Figure 7.8 and Figure 7.9 present the comparison of the temperature of and the pressure each node respectively. In Table 7.14, the percentage difference of each parameter is presented for the different sets of operating conditions.

Table 7.14 indicates that the percentage difference of the average maximum shear rate for each element between *operating condition B* and *operating condition C* is changed only slightly value. This means that the barrel temperature has an insignificant influence for the maximum shear rate. However, Table 7.14 indicates that the average percentage difference of the maximum shear stress in this comparison is equal to 4.2%. The increased maximum shear stress results in an increased viscosity. Furthermore, the percentage change in the material temperature in *operating condition C* decreases by -2.4% as shown in Table 7.14. This decreased material temperature results in an increased viscosity of the material. To resolve the flashing, as discussed in section 3.3.6, increasing the viscosity is one of the primary resolutions. Therefore, it can be concluded that to decrease barrel temperature can increase material viscosity and meaningful by improve the flashing deviation.

Moreover, the average percentage of the pressure for each node is 4.1% and the final pressure is increased to 4.3% as shown in Table 7.14. As discussed above, the pressure gradient during the compression stage will be decreased, and reduces the internal stress. This results in a holding pressure decreases and reduced the flashing. From all of the above discussions, it can concluded that the corrective action "decrease barrel temperature to 390 °F" significantly reduce flashing.

Once the corrective action, "decrease barrel temperature to 390°F" has been implemented, and the flashing deviation can still not be eliminated, then the corrective action, "decrease injection time to 4 sec", will then be employed as sequentially. When the corrective action, "decrease injection time to 4 sec", is employed, the operating conditions become as barrel temperature = 390°F, mold temperature = 180°F, and injection time = 4 sec. In this study, this operating condition is termed *operating condition D*.

From Tables 7.15-7.17 the simulation results of *operating condition D* in MOLDFLOW are presented. The maximum shear rate and the maximum shear stress of each element are presented in Table 7.15. Table 7.16 represents the pressure-time distribution. In Table 7.17, the temperature and the pressure of each node are presented. Furthermore, comparisons of each parameter between *operating condition C* and *operating condition D* are presented in Figures 7.10-7.14, and Table 7.18. In Figures 7.10 and 7.11, comparisons of the maximum shear rate and of the maximum shear stress are presented respectively. The comparisons of the temperature and the pressure are presented in Figures 7.12-Figure 7.13 respectively. In Table 7.18, the percentage difference of these parameters is presented.

Comparing Table 7.10 and Table 7.18, the percentage difference in parameters between *operating condition B* and *operating condition D* show very similar results. This is since both the operating conditions change the injection time to resolve the flashing deviation and the only difference is the barrel temperature changing from 390°F to 380°F. Therefore, for the same reasons as *operating condition B*, it can concluded that with *operating condition D* flashing can be reduced by decreasing the pressure gradient in the compression stage.

Moreover, from the above comparison, it is clear that *operating condition D* has a more significant change than *operating condition C*. This is because when the injection time reaches 4 second, it is at the extreme recommended condition. As discussed in section 7.2, in this system, when the operating variable corrective action reaches the extreme recommended condition, the system will provide alternative operating variable corrective action first. This is a deviation prevention mechanism within the system. This mechanism prevents another deviation from is being created when the extreme recommended operating conditions are employed to resolve the deviation. As discussed in section 7.2, when the extreme operating conditions are employed, there is a chance that another deviation will be created. This can be demonstrated in the following discussion.

As the comparison between Table 7.10 and Table 7.18, shown the percentage difference of the final pressure for *operating condition B* and *operating condition D* are 12.2% and 13.4% respectively. Obviously, *operating condition D* has a more significant influence the final pressure of the injection stage and results in a significant decrease of the pressure gradient in the compression stage. However, for short shots, it an increase in the pressure gradient in the compression stage is required. It is clear that when too high decrease in pressure gradient during the compression stage, there is in a more possibility for creating the short shot deviation. Therefore, it can be concluded that the prevention mechanism control the solution sequence is both valuable and efficient.

Following the corrective action "decrease injection time to 4 sec", is the corrective action "decrease mold temperature to 170°F" as shown in Table 7.2. This operating condition, which is barrel temperature = 390°F, mold temperature = 170°F, and injection time = 4 second, is termed *operating condition E*. The simulation results of *operating condition E* are summered in Tables 7.19 to 7.21.

In Table 7.19, the simulation results for the maximum shear rate and the maximum shear stress of each element is presented. The simulation results of the pressure-time distribution, and the temperature and the pressure of each node are displayed in Tables 7.20 and 7.21 respectively. Furthermore, the comparison results between *operating condition D* and *operating condition E* are shown in Figures 7.14-7-17, and Table 7.22. Figure 7.14 and Figure 7.15 show the comparison of the maximum shear rate and the maximum shear stress respectively. The temperature and the pressure comparisons are shown in Figure 7.16 and Figure 7.17 respectively. Table 7.22 displays the percentage difference of these comparisons.

From Table 7.22, it is observed that the percentage change of each of these parameters is almost insignificant. This is because the mold temperature during the injection stage is not a primary parameter for resolving the deviation. However, during the packing and cooling stages, the mold temperature is then a primary parameter in eliminating the deviation. Furthermore, in Table 7.22, can be seen that the viscosity can still be increased by increasing the maximum shear stress or decreasing the material temperature. Moreover, the final pressure increase. All of these shows that decreasing the mold temperature can still influence the flashing deviation.

For the operating variable corrective actions which are ranked from eleventh to fourteenth in Table 7.12, it can be similarly shown the these corrective action can reduce or eliminate the deviation. For other lower ranked corrective actions shown in Table 7.12 such as those ranked fifteenth to seventeenth, confirmation by the molding expert has shown that these corrective actions can also exactly reduce or eliminate the deviation.

From all of the above discussions, it can be concluded that the operating corrective actions provided by the system can significantly influence the parameters which control the occurrence of the deviation and changes can reduce or eliminate the deviation.

7.3 Performance of Decision Mechanism

As discussed in section 7.2, the operating variable corrective actions can be employed to resolve the flashing deviation. This shows that the priority weighting factors calculation for the operating variables, as determined by the decision mechanism, can accurately simulate the resolution skills of molding experts to arrange the priority of the operating variable corrective actions.

In this section, the influence factors for the priority weighting factors which are determined by the decision mechanism will be presented. As discussed in section 5.4.2, the priority weighting factors are dependent on factors such as the recommended operating conditions, the operating conditions, and the correlative weighting factors between the operating variables and the inherent physical properties, the operating variables and the influencing physical properties, the operating variables and the particular deviation, and the influencing physical properties and the particular deviation. However, among these factors, the recommended operating conditions are considered as constant variables. Therefore, the change in the priority weighting factors is independent of the recommended operating conditions.

The influence of the operating conditions on the priority weighting factors is shown in Figures 7.18 and 7.19. Figure 7.18 shows an operating variable which must be decreased to resolve the deviation. Figure 7.19 shows an operating variable which requires to be increased to resolve the deviation. In Figure 7.18, the operating variable, shot size, is used to illustrate the influence of the modified operating condition on the priority weighting factors. In Figure 7.19, the operating variable, cushion, is used to illustrate the influence of the increased operating conditions on the priority weighting factors.

In Figure 7.18, the maximum priority weighting factors for the shot size can be seen to occurs at the maximum shot size. This is a reasonable result, since for resolution of the flashing deviation, a shot size decreased is suggested. Therefore, for the greatest shot size, a higher priority should be assigned compared to a smaller shot size. In contrast, when the operating condition must be increased to resolve the deviation, the highest priority weighting factor should occur at the minimum conditions. As shown in Figure 7.19, the operating variable, cushion, has the highest priority weighting factor when its operating condition is minimum. This is so that the flashing deviation may be resolved by decreasing the cushion. However, due to different operating variables, there are a different operating ranges and different units. Therefore, it become difficult to compare the influence of the different operating variables on the priority weighting factors.

Another which influences the priority weighting factor is the correlative weighting factor. As discussed in section 5.4.2, the correlative weighting is used to indicate the degree of influence between parameters. Figure 7.20 shows the degree of influence of each correlative weighting factors on the priority weighting factor. In Figure 7.20, the operating variable is shot size, and the operating conditions are declared in Table 7.1. Furthermore, in Figure 7.20, the correlative weighting factors are between the shot size and the flashing, the shear rate and the flashing, the shot size and the shear rate, and the shot size and the thermal conductivity. Moreover, in Figure 7.20, it is assumed that when one correlative weighting factor is changed then other correlative weighting factors are the constants as declared in Table 7.1.

Clearly from Figure 7.20, the degree of influence for changing the gradient of the priority weighting factors is ranked as follows,

1. the correlative weighting factor of the operating variable and the deviation,
2. the correlative weighting factor of the influencing physical property and the deviation,
3. the correlative weighting factor of the operating variable and the influencing physical property and,
4. the correlative weighting factor of the operating variable and the inherent physical property.

Since the operating variables are directly employed to resolve the deviation, therefore, the correlative weighting factor of the operating variables and the deviation should have a more significant influence on the priority weighting factors than the other correlative weighting factors do. For the correlative weighting factor of the influencing physical properties and the deviation, due to the direct influence of the physical properties on the deviation, it becomes controlled by the operating variables. Therefore, this correlative weighting factors has second higher for changing gradient of the priority weighting factor. The remaining correlative weighting factors such as the operating variables and the influencing physical properties, and the operating variables and the inherent physical properties have a less significant influence on the priority weighting factor. This is since these correlative weighting factors do not directly influence the deviation.

7.4 Performance of Self-Learning Mechanism

As discussed in sections 6.2 to 6.4, when the resolution procedures were employed by the system, parameters may be changed by the self-learning mechanism, or the confirmation mechanism. These parameters include the recommended operating conditions, the user operating conditions, the correlative weighting factors of the operating variables and the inherent physical properties, of the operating variables and the influencing physical properties, of the deviation and the operating variables, and of the deviation and the influencing physical properties. Furthermore, following the discussion in section 6.2, the molded part "TEST1" will be used in the following illustration.

When a deviation is eliminated at the level of method corrective actions, according to section 5.4.1, section 5.4.3, and section 6.4, these parameters are not employed in the decision mechanism. Therefore, there are not any modifications of the parameters through the self-learning mechanism. However, since the parameters may change during the confirmation procedures, the self learning mechanism will not be utilized at this stage. When the deviation is eliminated during the method corrective actions, however, the system will include this in the self-learning mechanism which was discussed in sections 6.2 and 6.3.

In section 6.4.1, for instance, the method corrective action, "CLEAN MOLD SURFACE" was employed and the flashing deviation was eliminated. The system then used the confirmation results which were discussed in sections 6.2 and 6.3 to update these parameters. In this case, the parameters which were modified include the recommended operating conditions. Therefore, the recommended operating conditions were updated as shown in Figure 6.15 and were stored into the sub-directory, "TEST1", and named "VARREC.DAT". In this case, the parameters which were not modified are still stored in the sub-directory, "TEST1", and named as presented in Table 7.23. These data files are retrieved for use the next time, when the molded part, "TEST1" is retrieved. This is presented in Figure 6.11. The performance improves as the history of the molded part is developed. When the molded part is retrieved by the system, the system can then provide updated data for the user.

If the resolution procedures prompted the user to the operating variable corrective action level, then these parameters will be employed into the decision and self-learning mechanisms during the resolution procedures as discussed in sections 5.4.1, 5.4.2 and 5.4.3. Therefore, these parameters may change through both the self-learning and the confirmation mechanisms. However, the self-learning mechanism will write over the updated data which was modified in the confirmation procedures. Therefore, at the operating variable corrective action level, the update procedures is through the self-learning mechanism.

The performance of the self-learning mechanism includes two stages as discussed in section 5.4.3. The first stage is the update procedure for the correlative weighting factors of the operating variables with the inherent physical properties, of the operating variables with the influencing physical properties, and of the operating variables with the deviations. The second stage is the final update procedure of all the parameters when the deviation has been eliminated in this corrective action level, or this corrective action level has failed to provide a solution.

In the first update procedure stage, the self-learning mechanism is based on the resolution result and the corrective action to update the correlative weighting factors and the user operating conditions. In the case of section 6.4.2, for instance, when the corrective action, "DECREASE SHOT SIZE (in) TO 2.4" was employed, and the flashing deviation was improved, the operating variable, shot size, was the most significant influence on the flashing deviation. Therefore, according to this resolution response, "DEVIATION HAS BEEN IMPROVED", the self-learning mechanism employs Eqs. (5.21) to (5.26) to update the correlative weighting factors of the operating variable, shot size, and the inherent physical properties of the operating variable, shot size, and the influencing physical properties, and of the operating variable, shot size, and the flashing deviation, and replaces the old cushion operating condition from 2.6 in to 2.4 in. These modified parameters are presented in Table 7.23. Once these parameters have been modified by the self-learning mechanism, these modified parameters then employ Eqs (5.2) to (5.7) to calculate the priority weighting factors of the operating variables. After the new priority weighting factors of the operating variables have been calculated, the system then provides the next operating variable corrective action to the user. In this case, according to the resolution procedures for the flashing deviation, which is presented in Appendix C-1, the operating variable corrective action, "DECREASE SHOT SIZE (in) TO 2.2 in", will be employed.

In the above case, if the correction action, "DECREASE CUSHION (in) TO 2.4 in", is employed, and the resolution result was, "UNCHANGED", then the operating variable, shot size, has less influence than anticipated on the flashing deviation. Therefore, according to the resolution response, "DEVIATION WAS UNCHANGED", the system employs Eqs (5.15) to (5.20), to update the parameters which are the same as for updating in resolution result, "IMPROVED". These updated parameters are presented in Table 7.25. Then, as with the procedures for calculating the priority weighing factors of the operating variables, the system calculates the new priority weighting factors and employs the next corrective action suggested.

If the flashing deviation has been corrected by the final operating variable corrective action, "INCREASE REGRIND RATE TO 30%", as in Table 7.2, then, the deviation has been corrected by the resolution procedures which are accommodated by the system as shown in Table 7.2. The system is then updated accordingly to the latest results for the parameters to stored in the file as shown Table 7.23. In this case, the updated results are summarized in Table 7.26.

From Table 7.26, it can be seen that the maximum and the minimum recommended operating conditions of the operating variables, which have been employed in resolving the deviation, are the same values. This indicates that if the extreme recommended operating conditions have the same value, then, this value must be the optimum operating condition for the molded part. Therefore, the optimum operating conditions for the molded part "TEST1", are the operating conditions as shown in the Table 7.26.

If the deviation is not eliminated by the operating variable corrective actions, the system will still record the modifications of the parameters as shown in Table 7.26. However, in this situation, there is either some incorrect input for the parameters, or inappropriate design of the molded part. Therefore, the mold corrective action must then be employed to resolve the deviation.

As with the method corrective actions, the self-learning mechanism will be employed in the mold corrective actions and the material corrective actions.

Table 7.1 The summation of the definition for the declarative knowledge.

Material Type	Acetal copolymer, Celcon M-90		
Deviation Type	Flashing		
Recommended Operating Conditions	NAME	MINIMUM	MAXIMUM
	BARREL_TEMPERATURE(F)	380.00	420.00
	NOZZLE_TEMPERATURE(F)	380.00	420.00
	MOLD_TEMPERATURE(F)	160.00	200.00
	INJECTION_PRESSURE(psi)	5000.00	7000.00
	INJECTION_TIME(sec)	4.00	8.00
	MOLD_CLOSED_TIME(sec)	15.00	30.00
	MOLD_OPEN_TIME(sec)	15.00	30.00
	CYCLE_TIME(sec)	30.00	60.00
	DECOMPRESSION(sec)	1.00	4.00
	SHOT_SIZE(in)	2.00	3.20
	CUSHION(in)	0.10	0.50
	SCREW_SPEED(rpm)	40.00	60.00
	REGRIND_RATE(%)	0.00	30.00
	INJECTION_SPEED(1MAX2MED3MIN)	2.00	1.00
	GATE_SIZE(in)	0.10	0.20
	COOLING_CHANNEL_SIZE(in)	0.30	0.50
	RUNNER_SIZE(in)	0.20	0.40
	VENTING_CHANNEL_SIZE(in)	0.05	0.10
Operating Conditions	BARREL_TEMPERATURE(F) =	400.00	
	NOZZLE_TEMPERATURE(F) =	400.00	
	MOLD_TEMPERATURE(F) =	180.00	
	INJECTION_PRESSURE(psi) =	6000.00	
	INJECTION_TIME(sec) =	6.00	
	MOLD_CLOSED_TIME(sec) =	22.00	
	MOLD_OPEN_TIME(sec) =	22.00	
	CYCLE_TIME(sec) =	44.00	
	DECOMPRESSION(sec) =	3.00	
	SHOT_SIZE(in) =	2.60	
	CUSHION(in) =	0.40	
	SCREW_SPEED(rpm) =	50.00	
	REGRIND_RATE(%) =	25.00	
	INJECTION_SPEED(1MAX2MED3MIN) =	3.00	
	GATE_SIZE(in) =	0.15	
	COOLING_CHANNEL_SIZE(in) =	0.20	
	RUNNER_SIZE(in) =	0.30	
	VENTING_CHANNEL_SIZE(in) =	0.10	

Table 7.1 The summation of the definition for the declarative knowledge (continued).

Correlative Weighting Factors of the Operating Variables and the Inherent Physical Properties	NAME	THER_COND	THER_DIFF	SPEC_VOL	SPEC_HEAT
	BARREL_TEMPERATURE(F)	0.45	0.40	0.45	-0.35
	NOZZLE_TEMPERATURE(F)	0.50	0.45	0.45	-0.30
	MOLD_TEMPERATURE(F)	0.45	0.45	0.30	-0.40
	INJECTION_PRESSURE(psi)	-0.50	-0.50	0.45	-0.45
	INJECTION_TIME(sec)	0.45	0.50	-0.45	0.45
	MOLD_CLOSED_TIME(sec)	0.50	0.50	0.50	-0.40
	MOLD_OPEN_TIME(sec)	0.10	0.05	0.05	0.25
	CYCLE_TIME(sec)	0.45	0.45	0.20	-0.45
	DECOMPRESSION(sec)	0.10	0.05	0.05	0.25
	SHOT_SIZE(in)	0.15	0.25	0.20	0.30
	CUSHION(in)	0.20	0.25	0.30	0.35
	SCREW_SPEED(rpm)	0.15	0.10	0.05	0.35
	REGRIND_RATE(%)	0.35	0.30	0.10	0.10
	INJECTION SPEED(1MAX2MED3MIN)	-0.25	-0.30	0.35	-0.45

Correlative Weighting Factors of the Operating Variables and the Influencing Physical Properties	NAME	SHEAR_RATE	SHEAR_STRESS	VISCOSITY
	BARREL_TEMPERATURE(F)	-0.15	-0.10	-0.50
	NOZZLE_TEMPERATURE(F)	-0.15	-0.10	-0.45
	MOLD_TEMPERATURE(F)	-0.10	-0.05	-0.40
	INJECTION_PRESSURE(psi)	-0.45	0.50	0.50
	INJECTION_TIME(sec)	0.45	-0.50	-0.50
	MOLD_CLOSED_TIME(sec)	0.45	-0.40	-0.45
	MOLD_OPEN_TIME(sec)	0.20	0.20	0.05
	CYCLE_TIME(sec)	0.30	-0.30	0.45
	DECOMPRESSION(sec)	0.20	0.20	0.05
	SHOT_SIZE(in)	0.05	0.10	0.05
	CUSHION(in)	0.05	0.10	0.05
	SCREW_SPEED(rpm)	0.45	-0.35	-0.45
	REGRIND_RATE(%)	-0.40	0.30	0.50
	INJECTION SPEED(1MAX2MED3MIN)	-0.45	0.45	0.40

Correlative Weighting Factors of the Flashing and the Operating Variables		
	BARREL_TEMPERATURE(F) =	-0.22
	NOZZLE_TEMPERATURE(F) =	-0.11
	MOLD_TEMPERATURE(F) =	-0.22
	INJECTION_PRESSURE(psi) =	-0.50
	INJECTION_TIME(sec) =	-0.27
	MOLD_CLOSED_TIME(sec) =	0.00
	MOLD_OPEN_TIME(sec) =	0.00
	CYCLE_TIME(sec) =	0.00
	DECOMPRESSION(sec) =	0.00
	SHOT_SIZE(in) =	-0.43
	CUSHION(in) =	0.40
	SCREW_SPEED(rpm) =	0.05
	REGRIND_RATE(%) =	0.04
	INJECTION SPEED(1MAX2MED3MIN) =	-0.45

Correlative Weighting Factors of the Flashing and Influencing Physical Properties		
	SHEAR_RATE =	0.45
	SHEAR_STRESS =	0.40
	VISCOSITY =	0.50

Table 7.2 The employed sequences of the operating variable corrective actions for eliminating the flashing deviation.

1. Decrease shot size to 2.4 in.
2. Decrease shot size to 2.2 in.
3. Decrease injection pressure to 5500 psi.
4. Decrease shot size to 2.0 in.
5. Increase cushion to 0.5 in.
6. Decrease injection pressure to 5000 psi.
7. Decrease injection time to 5 sec.
8. Decrease barrel temperature to 390 F.
9. Decrease injection time to 4 sec.
10. Decrease mold temperature to 170 F.
11. Decrease nozzle temperature to 390 F.
12. Decrease barrel temperature to 380 F.
13. Decrease mold temperature to 160 F.
14. Decrease nozzle temperature to 380 F.
15. Increase screw speed to 55 rpm.
16. Increase screw speed to 60 rpm.
17. Increase regrind rate to 30 %.

Table 7.3 Comparison of the priority weighting factors and the correlative weighting factors between the operating variables and flashing.

OPERATING VARIABLE	PRIORITY WEIGHTING FACTORS OF THE OPERATING VARIABLES	THE CORRECTIVE WEIGHTING FACTORS OF THE OPERATING VRIABLE AND THE FLASHING
SHOT SIZE	0.0303	-0.43
INJECTION PRESSURE	0.2690	-0.50
CUSHION	0.0170	0.40
INJECTION TIME	0.0135	-0.27
BARREL TEMPERATURE	0.0093	-0.22
MOLD TEMPERATURE	0.0084	-0.22
NOZZLE TEMPERATURE	0.0060	-0.11
SCREW SPEED	0.0017	0.05
REGRIND RATE	0.0010	0.04
MOLD CLOSE TIME	0.0000	0.00
MOLD OPEN TIME	0.0000	0.00
CYCLE TIME	0.0000	0.00
DECOMPRESSURE	0.0000	0.00
INJECTION SPEED	0.0000	-0.45

Table 7.4 Simulation results of the maximum shear rate and the maximum shear stress for each element under *operating condition A*.

Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)	Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)
1	53.224	10.719	25	20.059	5.455
2	51.559	10.542	26	22.017	5.870
3	50.719	10.428	27	31.025	7.415
4	68.119	12.704	28	20.133	5.431
5	46.751	9.980	29	23.941	6.140
6	53.937	11.039	30	23.712	6.076
7	45.446	9.903	31	20.571	5.460
8	54.372	11.240	32	20.145	5.357
9	45.446	9.903	33	16.294	4.619
10	49.109	10.505	34	11.454	3.557
11	71.483	13.257	35	11.455	3.558
12	46.129	9.901	36	16.711	4.723
13	52.001	10.652	37	11.312	3.543
14	40.581	8.945	38	15.218	4.434
15	79.305	13.828	39	11.176	3.527
16	74.046	13.234	40	30.681	7.339
17	29.711	7.119	41	11.176	3.527
18	20.520	5.468	42	16.501	4.720
19	20.377	5.441	43	16.796	4.767
20	30.832	7.330	44	11.312	3.543
21	20.263	5.456	45	13.241	3.979
22	23.983	6.192	46	13.139	3.941
23	20.059	5.455	47	11.439	3.535
24	25.112	6.444	48	11.441	3.535

Shear Rate Maximum	79.305	Shear Rate Minimum	11.176	Shear Rate Average	30.917
Shear Stress Maximum	13.828	Shear Stress Maximum	3.527	Shear Stress Maximum	7.078

Table 7.5 The simulation results of pressure history for *operating condition A*.

Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)
0.000	0.000	1.570	210.030	3.090	262.860	4.950	295.670
0.350	56.200	2.040	223.510	3.650	278.260	5.000	297.250
0.760	106.100	2.540	241.610	3.660	279.950	5.650	304.600
1.140	155.680	2.590	244.790	4.280	286.810	5.680	306.080
1.570	205.730	3.060	259.830	4.330	288.440	6.010	313.590

Table 7.6 Simulation result of the temperature and the pressure for each node under *operating condition A*.

Node No.	Pressure (psi)	Temperature (°F)	Node No.	Pressure (psi)	Temperature (°F)
1	0.000	378.559	21	90.957	387.678
2	0.000	378.581	22	74.043	388.661
3	10.464	378.739	23	90.957	387.765
4	5.435	379.328	24	108.623	389.325
5	10.464	378.774	25	91.855	387.608
6	19.652	380.109	26	108.623	389.187
7	11.768	380.160	27	157.522	392.526
8	19.652	380.071	28	108.667	392.067
9	28.464	381.385	29	157.522	392.522
10	20.319	381.414	30	205.493	395.586
11	28.464	381.432	31	159.000	394.569
12	37.145	382.494	32	205.493	395.584
13	28.942	381.310	33	250.913	397.582
14	37.145	382.428	34	209.406	396.948
15	55.203	384.265	35	250.913	397.584
16	37.159	385.439	36	285.899	398.975
17	55.203	384.282	37	260.072	398.876
18	73.145	386.017	38	285.899	398.975
19	55.638	387.082	39	313.362	399.999
20	73.145	386.001			
Temperature Maximum	399.999	Temperature Minimum	378.559	Temperature Average	387.689
Pressure Maximum	313.362	Pressure Minimum	0	Pressure Average	103.144

Table 7.7 Simulation results of the maximum shear rate and the maximum shear stress for each element under *operating condition B*.

Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)	Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)
1	64.062	12.008	25	24.099	6.118
2	62.034	11.799	26	26.543	6.585
3	60.906	11.660	27	37.757	8.352
4	82.121	14.182	28	24.118	6.085
5	56.155	11.149	29	28.892	6.916
6	65.775	12.420	30	28.576	6.831
7	54.571	11.040	31	24.686	6.135
8	65.548	12.522	32	24.161	6.022
9	54.571	11.040	33	19.854	5.238
10	59.082	11.701	34	13.740	4.004
11	86.295	14.772	35	13.741	4.005
12	55.405	11.061	36	20.373	5.353
13	62.499	11.904	37	13.568	3.984
14	48.741	10.036	38	18.486	5.014
15	95.457	15.432	39	13.421	3.968
16	89.038	14.772	40	36.875	8.166
17	36.027	8.017	41	13.420	3.968
18	24.637	6.164	42	19.887	5.299
19	24.425	6.126	43	20.472	5.396
20	37.513	8.278	44	13.568	3.985
21	24.309	6.121	45	15.981	4.489
22	28.942	6.955	46	15.859	4.450
23	24.100	6.118	47	13.769	3.992
24	30.444	7.247	48	13.665	3.970

Shear Rate Maximum	95.457	Shear Rate Minimum	13.420	Shear Rate Average	37.253
Shear Stress Maximum	15.432	Shear Stress Maximum	3.968	Shear Stress Maximum	7.934

Table 7.8 The simulation results of pressure history for *operating condition B*.

Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)
0	0	1.31	234.49	2.57	293.85	4.13	331.07
0.29	62.8	1.7	249.68	3.05	311.08	4.17	332.43
0.63	118.66	2.12	269.97	3.05	312.98	4.71	341.15
0.95	174.05	2.15	273.67	3.57	320.86	4.73	342.35
1.31	229.78	2.56	290.54	3.61	322.52	5.01	351.27

Table 7.9 Simulation results of the temperature and the pressure for each node under *operating conditions B*.

Node No.	Pressure (psi)	Temperature (°F)	Node No.	Pressure (psi)	Temperature (°F)
1	0.000	382.438	21	102.319	389.869
2	0.000	382.462	22	83.304	391.032
3	11.812	382.572	23	102.319	389.871
4	5.986	383.088	24	122.203	391.291
5	11.812	382.584	25	103.319	392.405
6	22.188	383.594	26	122.203	391.289
7	13.333	383.779	27	176.710	393.127
8	22.188	383.612	28	122.246	393.715
9	32.116	384.656	29	176.710	393.127
10	22.942	384.885	30	230.290	395.006
11	32.116	384.665	31	178.377	395.584
12	41.899	385.740	32	230.290	395.006
13	32.667	385.977	33	281.116	397.006
14	41.913	385.732	34	234.696	397.443
15	62.159	387.093	35	281.116	397.006
16	41.928	388.236	36	320.435	399.535
17	62.159	387.091	37	291.420	399.094
18	82.304	388.472	38	320.435	399.535
19	62.652	389.629	39	351.000	399.999
20	82.304	388.472			
Temperature Maximum	399.999	Temperature Minimum	382.438	Temperature Average	389.890
Pressure Maximum	351.000	Pressure Minimum	0	Pressure Average	115.769

Table 7.10 The % difference of the control parameters between *operating condition A* and *operating condition B*.

Parameter Name	Range	% of difference
Maximum Shear Rate (1/s)	Maximum	20.4%
	Minimum	20.1%
	Average	20.5%
Maximum Shear Stress (psi)	Maximum	11.6%
	Minimum	12.5%
	Average	12.1%
Temperature (°F)	Maximum	0.0%
	Minimum	0.5%
	Average	1.0%
Pressure (psi)	Maximum	12.0%.
	Minimum	0.0%
	Average	12.2%
Ended Pressure		12.2%

Table 7.11 Simulation results of the maximum shear rate and the maximum shear stress for each element under *operating condition C*.

Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)	Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)
1	64.046	12.541	25	24.094	6.366
2	62.021	12.321	26	26.540	6.852
3	60.900	12.176	27	37.745	8.692
4	82.096	14.808	28	24.113	6.333
5	56.147	11.637	29	28.889	7.199
6	65.768	12.962	30	28.573	7.112
7	54.558	11.517	31	24.682	6.389
8	65.528	13.060	32	24.156	6.271
9	54.558	11.516	33	19.849	5.444
10	59.075	12.205	34	13.738	4.161
11	86.267	15.417	35	13.739	4.162
12	55.393	11.543	36	20.368	5.563
13	62.494	12.429	37	13.566	4.140
14	48.737	10.482	38	18.485	5.209
15	95.439	16.124	39	13.419	4.122
16	89.026	15.434	40	36.876	8.487
17	36.016	8.347	41	13.418	4.122
18	24.634	6.417	42	19.885	5.505
19	24.423	6.378	43	20.467	5.606
20	37.500	8.617	44	13.567	4.140
21	24.306	6.371	45	15.979	4.664
22	28.939	7.238	46	15.857	4.624
23	24.094	6.366	47	13.767	4.150
24	30.435	7.540	48	13.663	4.127
Shear Rate Maximum	95.439	Shear Rate Minimum	13.418	Shear Rate Average	37.247
Shear Stress Maximum	16.124	Shear Stress Maximum	4.122	Shear Stress Maximum	8.268

Table 7.12 The simulation results of pressure history for *operating condition C*.

Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)	Time (sec)	Pressure (psi)
0	0.00	1.31	244.78	2.57	306.56	4.13	345.26
0.29	65.61	1.70	260.59	3.05	324.49	4.17	346.67
0.63	123.94	2.12	281.71	3.05	326.47	4.71	355.74
0.95	181.75	2.15	285.56	3.57	334.66	4.73	356.98
1.31	239.89	2.56	303.12	3.61	336.38	5.01	366.25

Table 7.13 Simulation result of the temperature and the pressure for each node under *operating conditions C*.

Node No.	Pressure (psi)	Temperature (°F)	Node No.	Pressure (psi)	Temperature (°F)
1	0.000	373.393	21	106.409	380.570
2	0.000	373.415	22	86.620	380.347
3	12.270	373.534	23	106.409	380.646
4	6.215	374.013	24	127.113	381.877
5	12.271	373.570	25	107.462	380.523
6	23.043	374.630	26	127.113	381.754
7	13.850	374.677	27	183.977	384.321
8	23.043	374.592	28	127.172	384.069
9	33.360	375.625	29	183.977	384.321
10	23.826	375.674	30	239.903	386.661
11	33.361	375.669	31	185.720	385.963
12	43.538	376.511	32	239.901	386.658
13	33.929	375.571	33	292.959	388.182
14	43.539	376.448	34	244.494	387.756
15	64.613	377.870	35	292.959	388.182
16	43.557	378.869	36	334.022	389.268
17	64.614	377.892	37	303.717	389.205
18	85.570	379.263	38	334.022	389.268
19	65.124	380.098	39	365.970	390.000
20	85.570	379.247			
Temperature Maximum	390.000	Temperature Minimum	373.393	Temperature Average	380.516
Pressure Maximum	365.970	Pressure Minimum	0	Pressure Average	120.544

Table 7.14 The % difference of the control parameters between *operating condition B* and *operating condition C*.

Parameter Name	Range	% of difference
Maximum Shear Rate (1/s)	Maximum	-0.01%
	Minimum	-0.01%
	Average	-0.01%
Maximum Shear Stress (psi)	Maximum	4.2%
	Minimum	3.8%
	Average	4.2%
Temperature (°F)	Maximum	-2.5%
	Minimum	-2.4%
	Average	-2.4%
Pressure (psi)	Maximum	4.3%.
	Minimum	0%
	Average	4.1%
Ended Pressure		4.3%

Table 7.15 Simulation results of the maximum shear rate and the maximum shear stress for each element under *operating condition D*.

Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)	Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)
1	80.364	14.383	25	30.140	7.324
2	77.787	14.121	26	33.367	7.892
3	76.186	13.936	27	47.515	9.984
4	103.210	16.917	28	30.175	7.297
5	70.271	13.310	29	36.186	8.279
6	82.450	14.795	30	35.801	8.191
7	68.252	13.145	31	30.854	7.367
8	82.368	14.896	32	30.217	7.241
9	68.252	13.145	33	25.141	6.328
10	74.066	13.918	34	17.273	4.836
11	108.569	17.573	35	17.065	4.794
12	69.331	13.204	36	25.825	6.464
13	78.264	14.218	37	17.309	4.859
14	60.988	12.045	38	23.102	5.994
15	119.754	18.404	39	19.224	5.259
16	111.570	17.621	40	46.100	9.672
17	45.293	9.606	41	19.435	5.300
18	30.802	7.392	42	17.296	4.878
19	30.494	7.341	43	25.961	6.510
20	47.188	9.905	44	17.322	4.862
21	30.381	7.335	45	20.121	5.412
22	36.255	8.317	46	19.966	5.370
23	30.141	7.324	47	17.299	4.825
24	38.407	8.690	48	16.946	4.746

Shear Rate Maximum	119.754	Shear Rate Minimum	16.946	Shear Rate Average	46.672
Shear Stress Maximum	18.404	Shear Stress Maximum	4.746	Shear Stress Maximum	9.484

Table 7.16 The simulation results of pressure history for *operating condition D*.

Time	Pressure (psi)	Time	Pressure (psi)	Time	Pressure (psi)	Time	Pressure (psi)
0.00	0.00	1.05	279.75	2.06	350.92	3.30	395.98
0.23	74.99	1.36	298.02	2.44	371.54	3.34	397.19
0.50	141.86	1.69	322.33	2.44	373.86	3.77	408.15
0.76	207.98	1.72	326.75	2.85	383.44	3.78	409.09
1.05	274.26	2.04	346.97	2.88	385.29	4.01	420.31

Table 7.17 Simulation results of the temperature and the pressure for each node under *operating conditions D*.

Node No.	Pressure (psi)	Temperature (°F)	Node No.	Pressure (psi)	Temperature (°F)
1	7.462	377.253	21	122.047	382.874
2	0.000	377.272	22	99.087	382.737
3	14.926	377.357	23	122.044	382.944
4	0.000	377.749	24	146.091	383.913
5	12.863	377.393	25	123.269	382.847
6	25.841	378.234	26	146.099	383.801
7	13.511	378.282	27	211.354	385.750
8	25.223	378.194	28	146.159	385.673
9	37.460	379.004	29	211.351	385.752
10	26.006	379.074	30	275.654	387.518
11	37.276	379.045	31	213.380	387.081
12	49.241	379.715	32	275.654	387.514
13	37.947	378.968	33	336.746	388.659
14	49.174	379.656	34	280.986	388.405
15	73.596	380.772	35	336.745	388.659
16	49.207	381.618	36	384.245	389.505
17	73.576	380.793	37	349.200	389.471
18	97.873	381.864	38	384.248	389.505
19	74.170	382.575	39	420.755	390.000
20	97.866	381.846			
Temperature Maximum	390.000	Temperature Minimum	377.252	Temperature Average	382.801
Pressure Maximum	420.755	Pressure Minimum	0	Pressure Average	138.162

Table 7.18 The % difference of the control parameters between *operating condition C* and *operating condition D*.

Parameter Name	Range	% of difference
Maximum Shear Rate (1/s)	Maximum	25.4%
	Minimum	26.3%
	Average	25.3%
Maximum Shear Stress (psi)	Maximum	14.1%
	Minimum	15.1%
	Average	14.7%
Temperature (°F)	Maximum	0%
	Minimum	1.3%
	Average	0.6%
Pressure (psi)	Maximum	15.0%.
	Minimum	0.0%
	Average	13.4%
Ended Pressure		15.0%

Table 7.19 Simulation results of the maximum shear rate and the maximum shear stress for each element under *operating condition E.*

Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)	Element No.	Maximum Shear Rate (1/s)	Maximum Shear Stress (psi)
1	80.367	14.427	25	30.145	7.355
2	77.787	14.166	26	33.362	7.926
3	76.186	13.980	27	47.511	10.025
4	103.209	16.974	28	30.179	7.327
5	70.273	13.358	29	36.183	8.312
6	82.444	14.850	30	35.797	8.221
7	68.266	13.200	31	30.857	7.394
8	82.341	14.957	32	30.222	7.267
9	68.266	13.200	33	25.139	6.355
10	74.049	13.975	34	17.273	4.856
11	108.564	17.639	35	17.065	4.814
12	69.344	13.253	36	25.823	6.492
13	78.262	14.265	37	17.311	4.881
14	60.988	12.081	38	23.100	6.021
15	119.754	18.458	39	19.225	5.283
16	111.571	17.673	40	46.098	9.717
17	45.288	9.642	41	19.436	5.325
18	30.802	7.420	42	17.295	4.901
19	30.495	7.369	43	25.960	6.540
20	47.185	9.945	44	17.323	4.884
21	30.382	7.364	45	20.119	5.435
22	36.252	8.352	46	19.964	5.392
23	30.146	7.356	47	17.301	4.844
24	38.398	8.727	48	16.949	4.765
Shear Rate Maximum	119.754	Shear Rate Minimum	16.949	Shear Rate Average	46.672
Shear Stress Maximum	18.458	Shear Stress Maximum	4.765	Shear Stress Maximum	9.520

Table 7.20 The simulation results of pressure history for *operating condition E.*

Time	Pressure (psi)	Time	Pressure (psi)	Time	Pressure (psi)	Time	Pressure (psi)
0.00	0.00	1.05	280.74	2.06	352.18	3.30	397.42
0.23	75.21	1.36	299.07	2.44	372.89	3.34	398.64
0.50	142.29	1.69	323.47	2.44	375.22	3.77	409.65
0.76	208.64	1.72	327.91	2.85	384.83	3.78	410.60
1.05	275.20	2.04	348.21	2.88	386.70	4.01	421.87

Table 7.21 Simulation result of the temperature and the pressure for each node under *operating conditions E.*

Node No.	Pressure (psi)	Temperature (°F)	Node No.	Pressure (psi)	Temperature (°F)
1	7.496	376.551	21	122.558	382.465
2	0.000	376.572	22	99.511	382.318
3	14.994	376.660	23	122.555	382.539
4	0.000	377.071	24	146.687	383.558
5	12.922	376.698	25	123.784	382.435
6	25.959	377.582	26	146.694	383.441
7	13.573	377.631	27	212.220	385.504
8	25.338	377.539	28	146.757	385.403
9	37.628	378.390	29	212.217	385.506
10	26.125	378.462	30	276.762	387.372
11	37.443	378.433	31	214.254	386.895
12	49.456	379.137	32	276.762	387.369
13	38.116	378.352	33	338.064	388.578
14	49.389	379.076	34	282.113	388.299
15	73.916	380.250	35	338.062	388.578
16	49.422	381.135	36	385.706	389.467
17	73.896	380.273	37	350.552	389.430
18	98.292	381.400	38	385.709	389.467
19	74.493	382.143	39	422.317	390.000
20	98.284	381.384			

Temperature Maximum	390.000	Temperature Minimum	376.551	Temperature Average	382.394
Pressure Maximum	422.317	Pressure Minimum	0	Pressure Average	138.719

Table 7.22 The % difference of the control parameters between *operating condition D* and *operating condition E.*

Parameter Name	Range	% of difference
Maximum Shear Rate (1/s)	Maximum	0%
	Minimum	0.1%
	Average	0.0%
Maximum Shear Stress (psi)	Maximum	0.2%
	Minimum	0%
	Average	0.4%
Temperature (°F)	Maximum	0%
	Minimum	-0.2%
	Average	-0.1%
Pressure (psi)	Maximum	0.4%.
	Minimum	0.0%
	Average	0.4%
Ended Pressure		0.4%

Table 7.23 File names for the updated parameters.

Parameter Name	File Name
The recommended operating conditions	VARREC.DAT
The user operating conditions	VARUSER.DAT
The correlative weighting factors of the operating variables and the inherent physical properties	VARHPCWF.DAT
The correlative weighting factors of the operating variables and the influencing physical properties	VARFPCWF.DAT
The correlative weighting factors of the operating variables and the flashing deviation	FLVARCWF.DAT
The correlative weighting factors of the inherent physical properties and the flashing deviation	FLFPCWF.DAT

Table 7.24 The updated data when corrective action is "decrease shot to 2.4 in" and the resolution result is "improved".

User operating condition (Shot size)		
Old	New	
2.6 in	2.4 in	
The correlative weighting of the operating variable (cushion) and the flashing deviation		
Old	New	
-0.43	-0.48	
The correlative weighting of the operating variable (cushion) and the inherent physical properties		
	Old	New
Thermal Conductivity - Cushion	0.15	0.20
Thermal Diffusivity - Cushion	0.25	0.30
Specific Volume - Cushion	0.20	0.25
Specific Heat - Cushion	0.30	0.35
The correlative weighting of the operating variable (cushion) and the influencing physical properties		
	Old	New
Shear Rate- Cushion	0.05	0.10
Shear Stress - Cushion	0.10	0.10
Viscosity - Cushion	0.05	0.10

Table 7.25 The updated data when the corrective action is "decrease cushion to 2.4 in" and the resolution result is "unchanged".

User operating condition (Cushion)		
Old	New	
2.6 in	2.4 in	
The correlative weighting of the operating variable (cushion) and the flashing deviation		
Old	New	
-0.43	-0.38	
The correlative weighting of the operating variable (cushion) and the inherent physical properties		
	Old	New
Thermal Conductivity - Cushion	0.15	0.10
Thermal Diffusivity - Cushion	0.25	0.15
Specific Volume - Cushion	0.20	0.15
Specific Heat - Cushion	0.30	0.25
The correlative weighting of the operating variable (cushion) and the influencing physical properties		
	Old	New
Shear Rate- Cushion	0.05	0.05
Shear Stress - Cushion	0.10	0.05
Viscosity - Cushion	0.05	0.05

Table 7.26 The summation of the final updated results of the parameters when the flashing deviation has been corrected by the corrective action "INCREASE REGRIND RATE TO 30%".

Recommended Operating Conditions	NAME	MINIMUM	MAXIMUM
	REGRIND_RATE(%)	30.00	30.00
	SCREW_SPEED(rpm)	60.00	60.00
	NOZZLE_TEMPERATURE(F)	380.00	380.00
	MOLD_TEMPERATURE(F)	160.00	160.00
	BARREL_TEMPERATURE(F)	380.00	380.00
	INJECTION_TIME(sec)	4.00	4.00
	INJECTION_PRESSURE(psi)	5000.00	5000.00
	CUSHION(in)	0.50	0.50
	SHOT_SIZE(in)	2.00	2.00
	MOLD_CLOSED_TIME(sec)	15.00	30.00
	MOLD_OPEN_TIME(sec)	15.00	30.00
	CYCLE_TIME(sec)	30.00	60.00
	DECOMPRESSION(sec)	1.00	4.00
	INJECTION SPEED(1MAX2MED3MIN)	3.00	3.00

Operating Conditions			
	REGRIND_RATE(%)	=	30.00
	SCREW_SPEED(rpm)	=	60.00
	NOZZLE_TEMPERATURE(F)	=	380.00
	MOLD_TEMPERATURE(F)	=	160.00
	BARREL_TEMPERATURE(F)	=	380.00
	INJECTION_TIME(sec)	=	4.00
	INJECTION_PRESSURE(psi)	=	5000.00
	CUSHION(in)	=	0.50
	SHOT_SIZE(in)	=	2.00
	MOLD_CLOSED_TIME(sec)	=	22.00
	MOLD_OPEN_TIME(sec)	=	22.00
	CYCLE_TIME(sec)	=	44.00
	DECOMPRESSION(sec)	=	3.00
	INJECTION SPEED(1MAX2MED3MIN)	=	3.00

Corrective Weighting Factors of the Operating Variables and the Inherent Physical Properties	NAME	THER_COND	THER_DIFF	SPEC_VOL	SPEC_HEAT
	REGRIND_RATE(%)	0.40	0.35	0.15	0.15
	SCREW_SPEED(rpm)	0.25	0.20	0.15	0.45
	NOZZLE_TEMPERATURE(F)	0.60	0.55	0.55	-0.40
	MOLD_TEMPERATURE(F)	0.55	0.55	0.40	-0.50
	BARREL_TEMPERATURE(F)	0.55	0.50	0.55	-0.45
	INJECTION_TIME(sec)	0.55	0.60	-0.55	0.55
	INJECTION_PRESSURE(psi)	-0.60	-0.60	0.55	-0.55
	CUSHION(in)	0.25	0.30	0.35	0.40
	SHOT_SIZE(in)	0.30	0.40	0.35	0.45
	MOLD_CLOSED_TIME(sec)	0.50	0.50	0.50	-0.40
	MOLD_OPEN_TIME(sec)	0.10	0.05	0.05	0.25
	CYCLE_TIME(sec)	0.45	0.45	0.20	-0.45
	DECOMPRESSION(sec)	0.10	0.05	0.05	0.25
	INJECTION SPEED(1MAX2MED3MIN)	-0.25	-0.30	0.35	-0.45

Table 7.26 The summation of the final updated results of the parameters when the flashing deviation has been corrected by the corrective action "INCREASE REGRIND RATE TO 30%" (continued).

Corrective Weighting Factors of the Operating Variables and the Influencing Physical Properties	NAME	SHEAR_RATE	SHEAR_STRESS	VISCOSITY
	REGRIND_RATE(%)	-0.45	0.35	0.55
	SCREW_SPEED(rpm)	0.55	-0.45	-0.55
	NOZZLE_TEMPERATURE(F)	-0.25	-0.20	-0.55
	MOLD_TEMPERATURE(F)	-0.20	-0.15	-0.50
	BARREL_TEMPERATURE(F)	-0.25	-0.20	-0.60
	INJECTION_TIME(sec)	0.55	-0.60	-0.60
	INJECTION_PRESSURE(psi)	-0.55	0.60	0.60
	CUSHION(in)	0.10	0.15	0.10
	SHOT_SIZE(in)	0.20	0.25	0.20
	MOLD_CLOSED_TIME(sec)	0.45	-0.40	-0.45
	MOLD_OPEN_TIME(sec)	0.20	0.20	0.05
	CYCLE_TIME(sec)	0.30	-0.30	0.45
	DECOMPRESSION(sec)	0.20	0.20	0.05
	INJECTION SPEED(1MAX2MED3MIN)	-0.45	0.45	0.40
Corrective Weighting Factors of the Flashing and the Operating Variables	REGRIND_RATE(%)	0.09		
	SCREW_SPEED(rpm)	0.15		
	NOZZLE_TEMPERATURE(F)	-0.21		
	MOLD_TEMPERATURE(F)	-0.32		
	BARREL_TEMPERATURE(F)	-0.32		
	INJECTION_TIME(sec)	-0.37		
	INJECTION_PRESSURE(psi)	-0.60		
	CUSHION(in)	0.45		
	SHOT_SIZE(in)	-0.58		
	MOLD_CLOSED_TIME(sec)	0.00		
	MOLD_OPEN_TIME(sec)	0.00		
	CYCLE_TIME(sec)	0.00		
	DECOMPRESSION(sec)	0.00		
	INJECTION SPEED(1MAX2MED3MIN)	-0.45		
Corrective Weighting Factors of the Flashing and Influencing Physical Properties	SHEAR_RATE	0.45		
	SHEAR_STRESS	0.40		
	VISCOSITY	0.50		

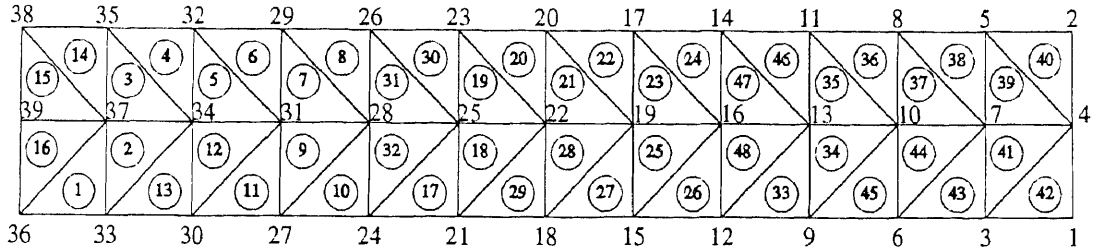


Figure 7.1 A meshing model of molded part uses in MOLDFLOW.

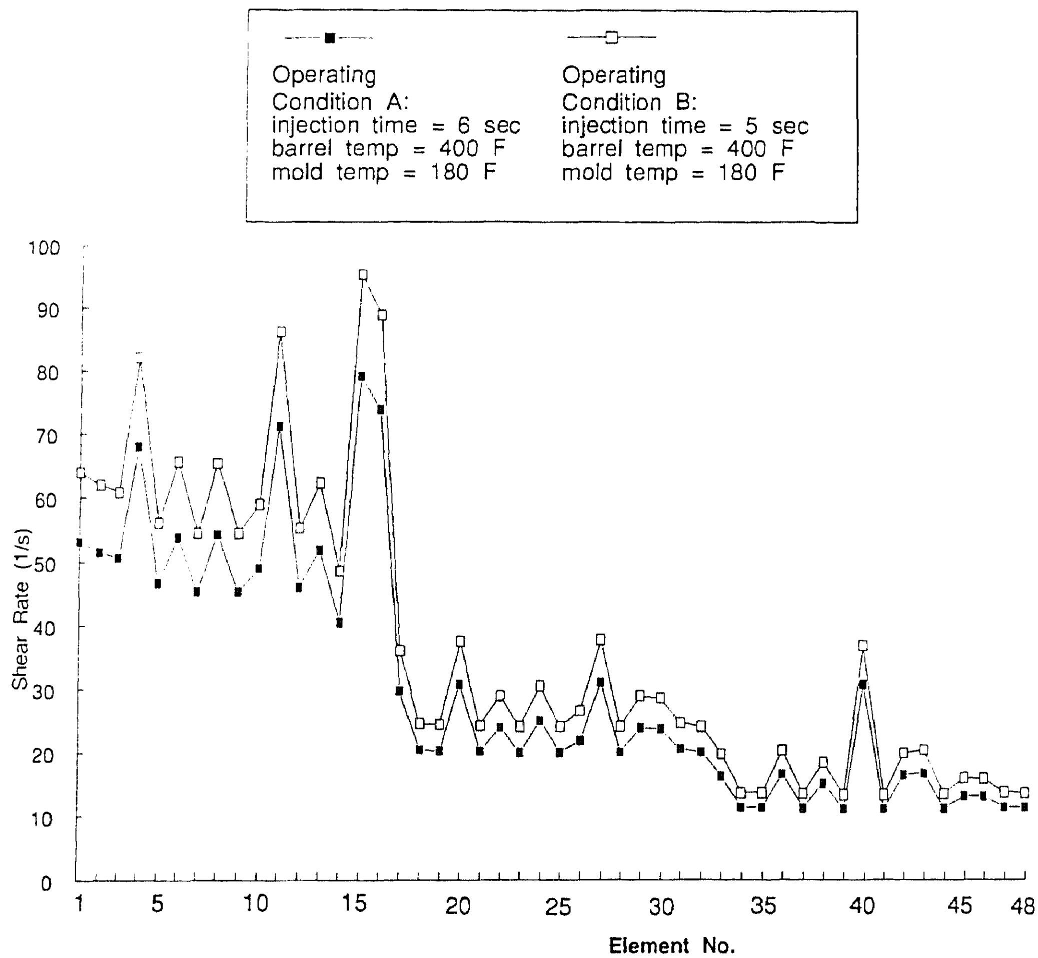


Figure 7.2 The comparison of the maximum shear rate in each element between *operating condition A* and *operating condition B*.

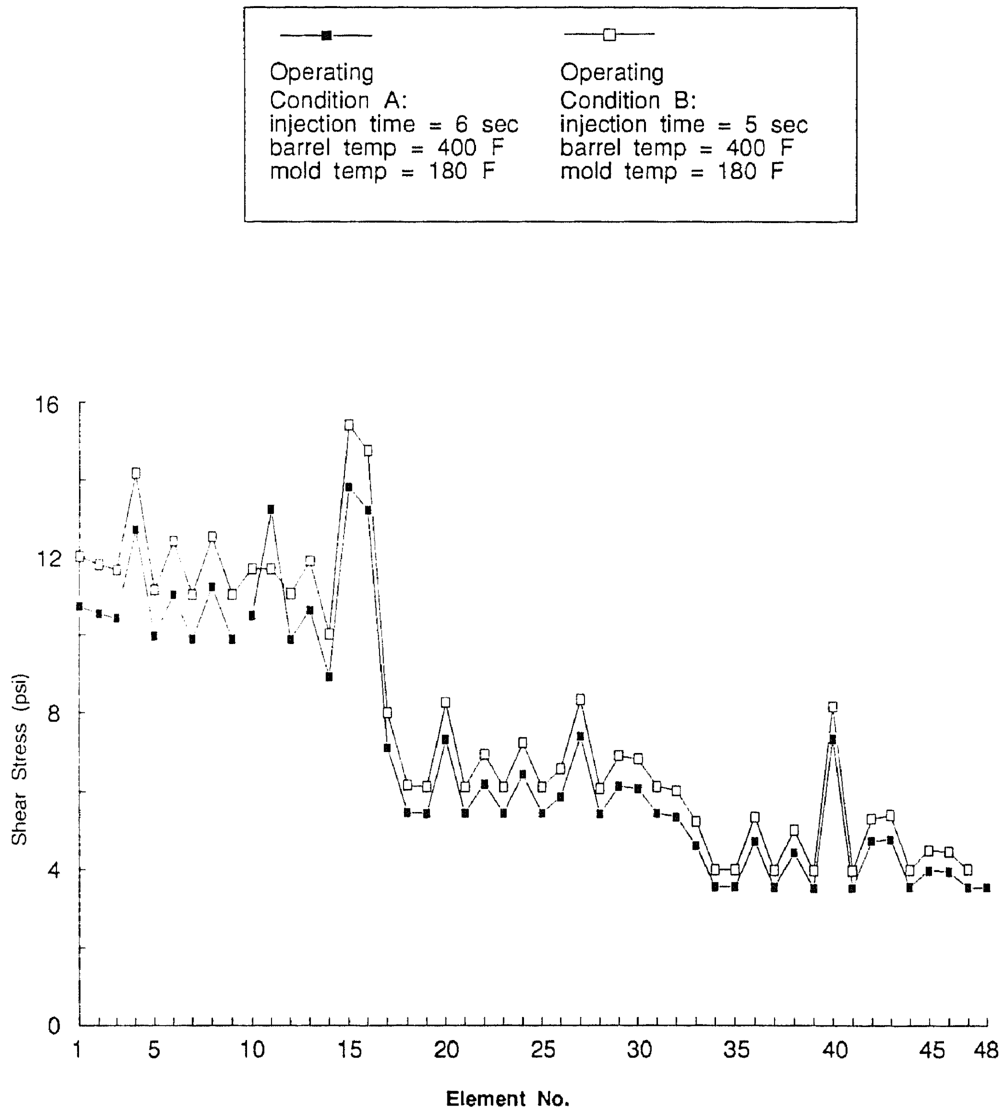


Figure 7.3 The comparison of the maximum shear stress in each element between *operating condition A* and *operating condition B*.

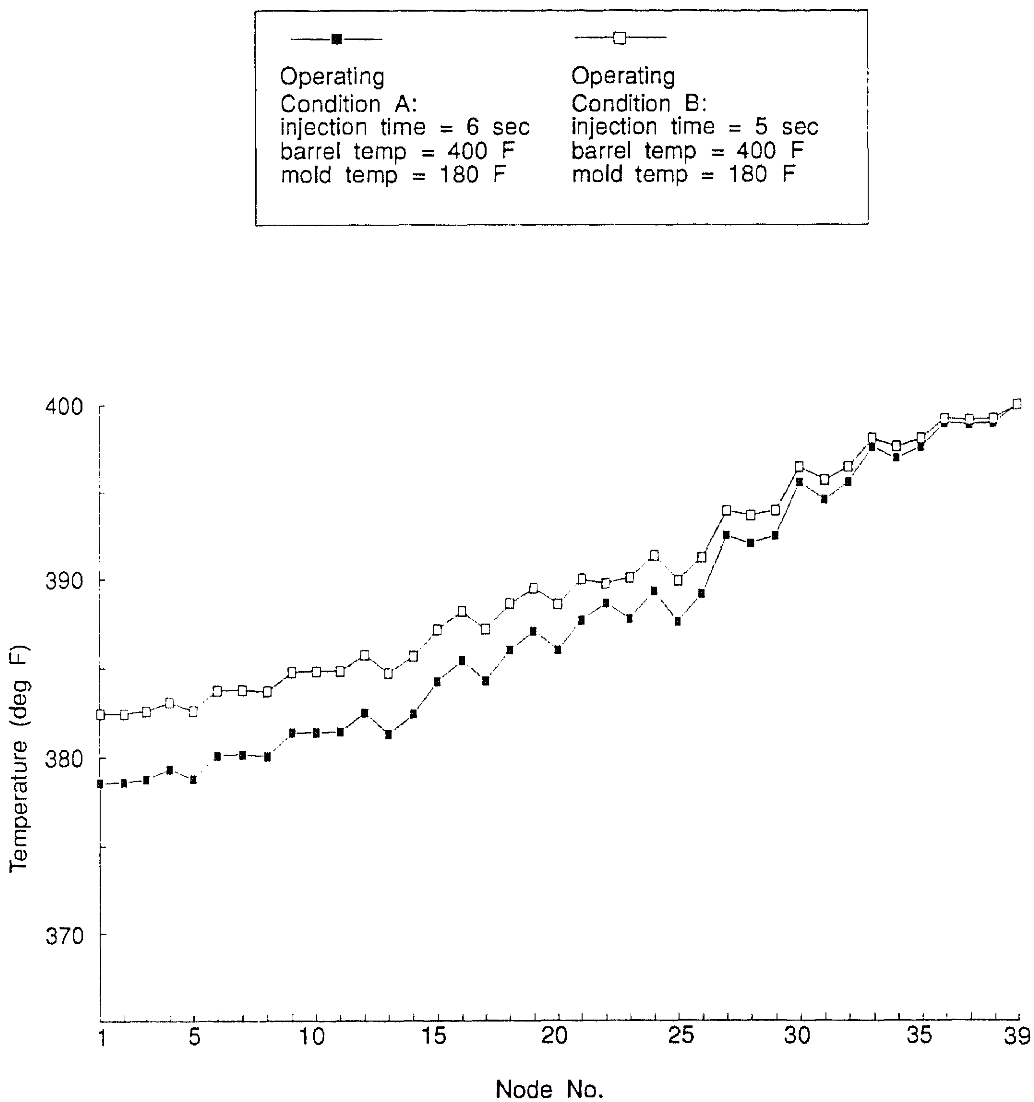


Figure 7.4 The comparison of the temperature in each node between *operating condition A* and the *operating condition B*.

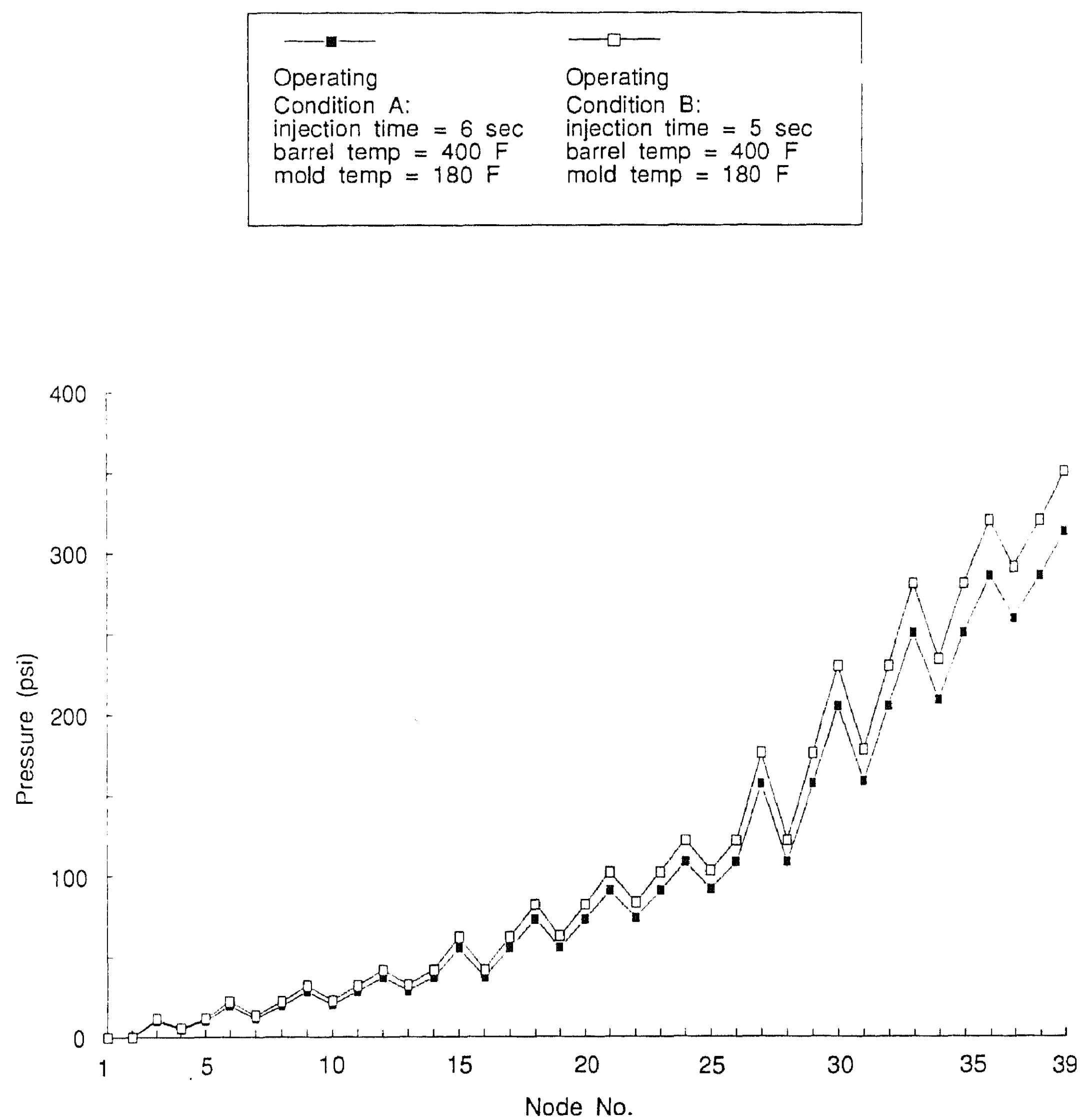


Figure 7.5 The comparison of the pressure in each node between operating *condition A* and *operating condition B*.

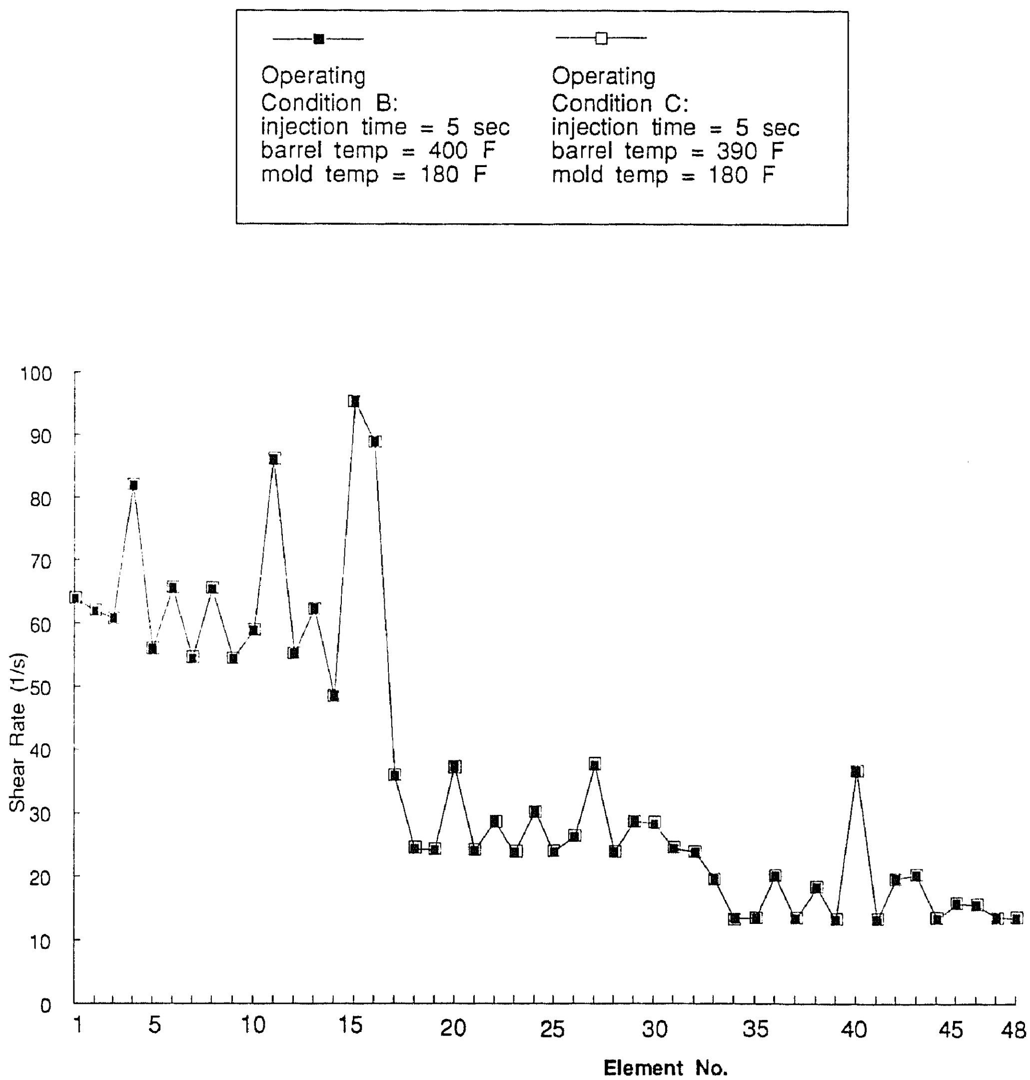


Figure 7.6 The comparison of the maximum shear rate in each element between *operating condition B* and *operating condition C*.

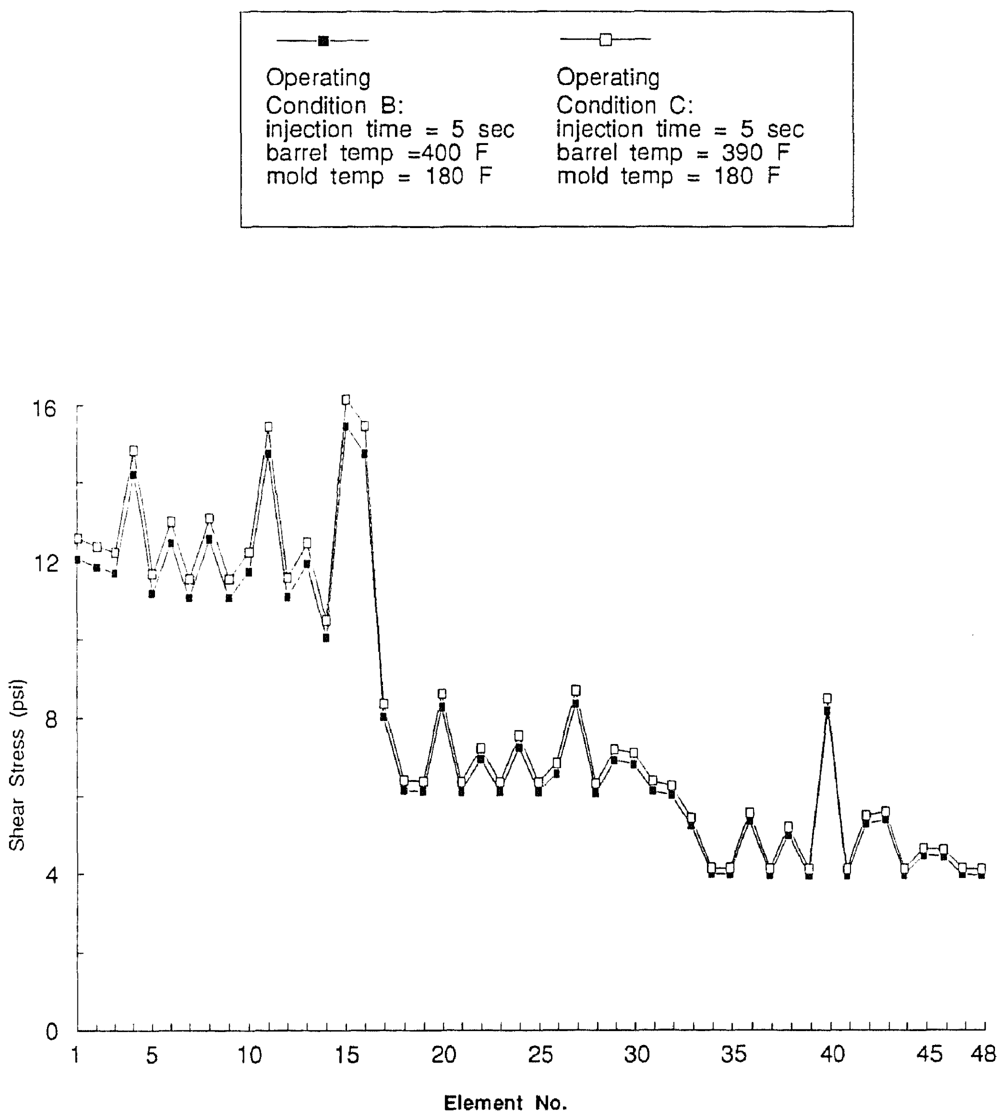


Figure 7.7 The comparison of the maximum shear stress in each element between *operating condition B* and *operating condition C*.

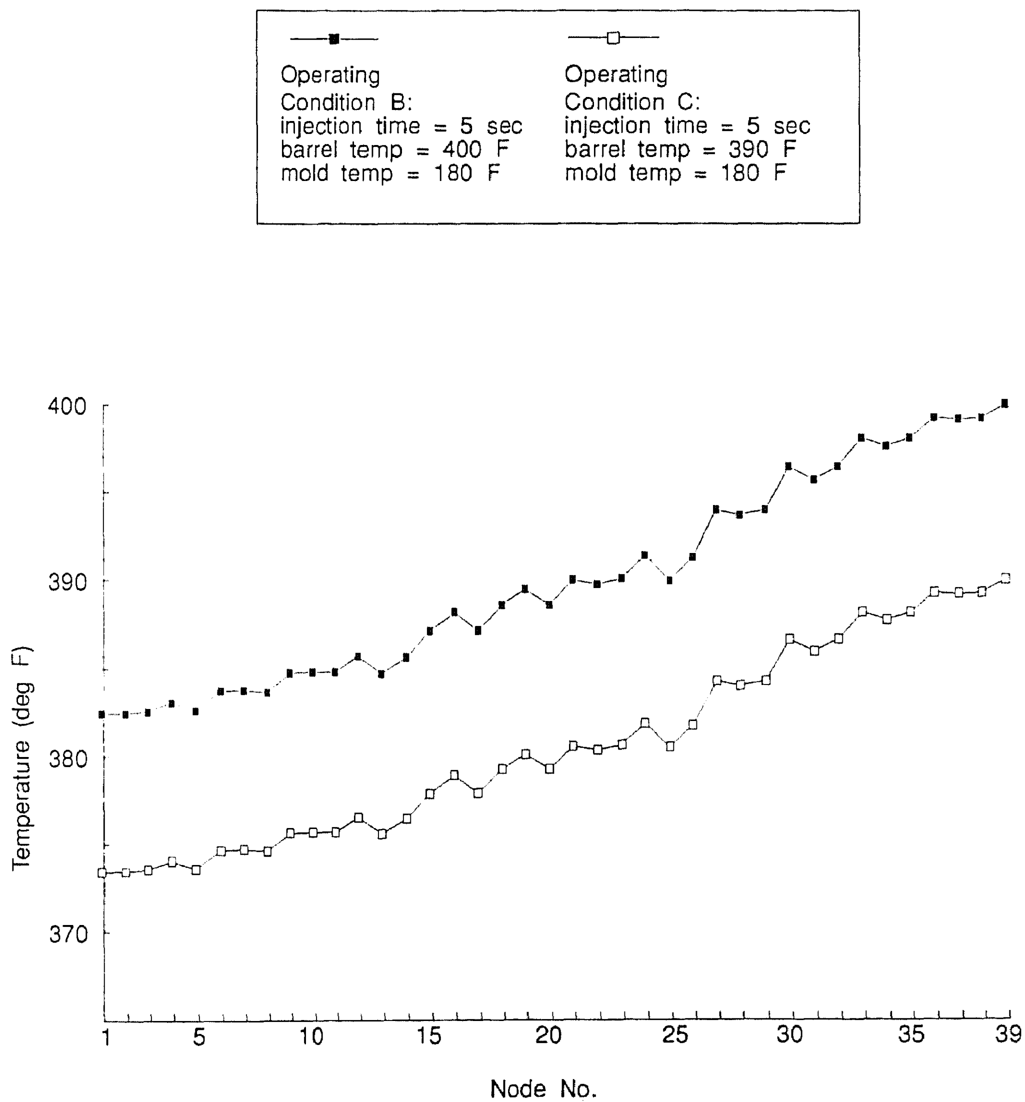


Figure 7.8 The comparison of the temperature in each node between *operating condition B* and *operating condition C*.

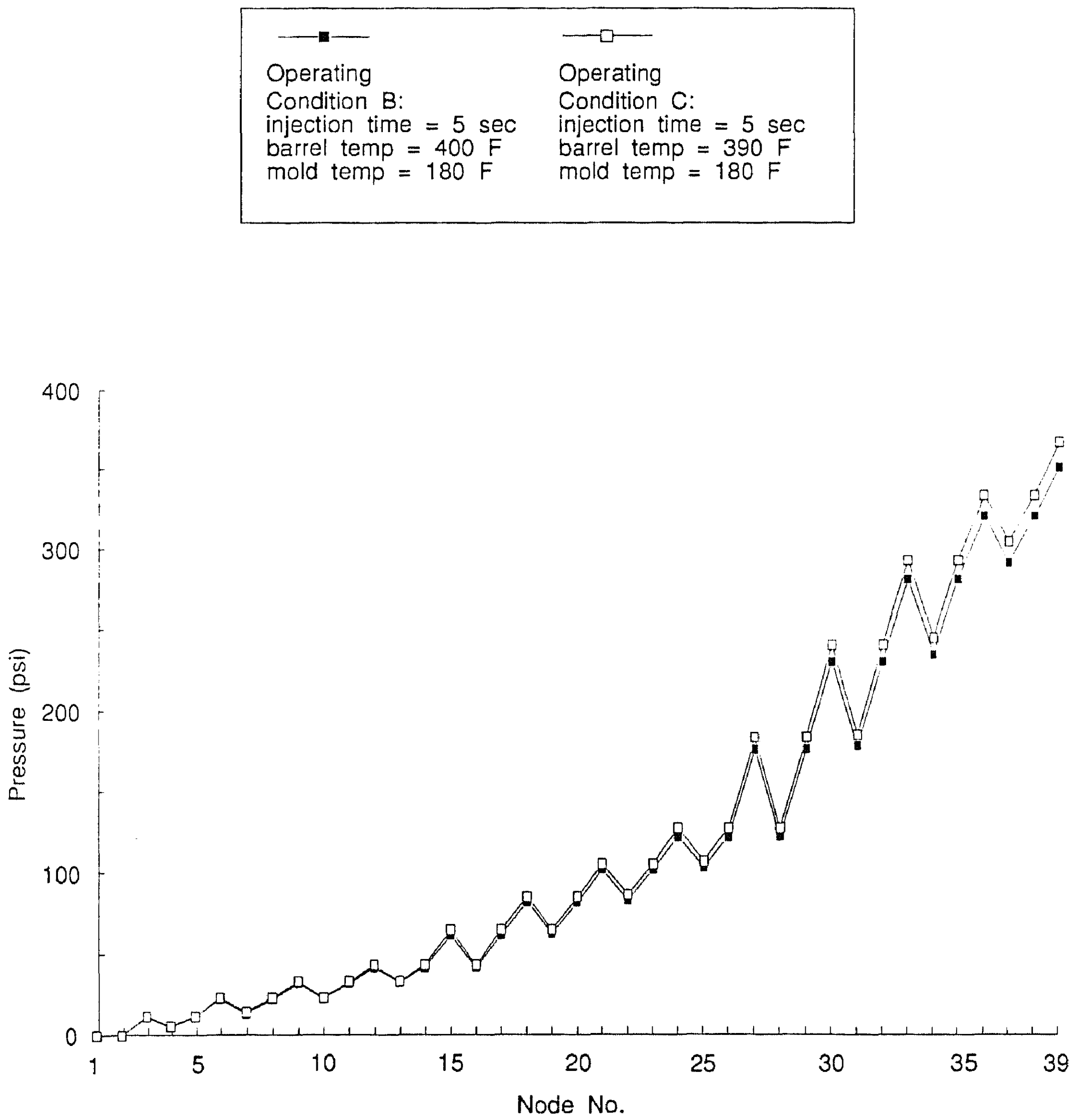


Figure 7.9 The comparison of the pressure in each node between operating *condition B* and *operating condition C*.

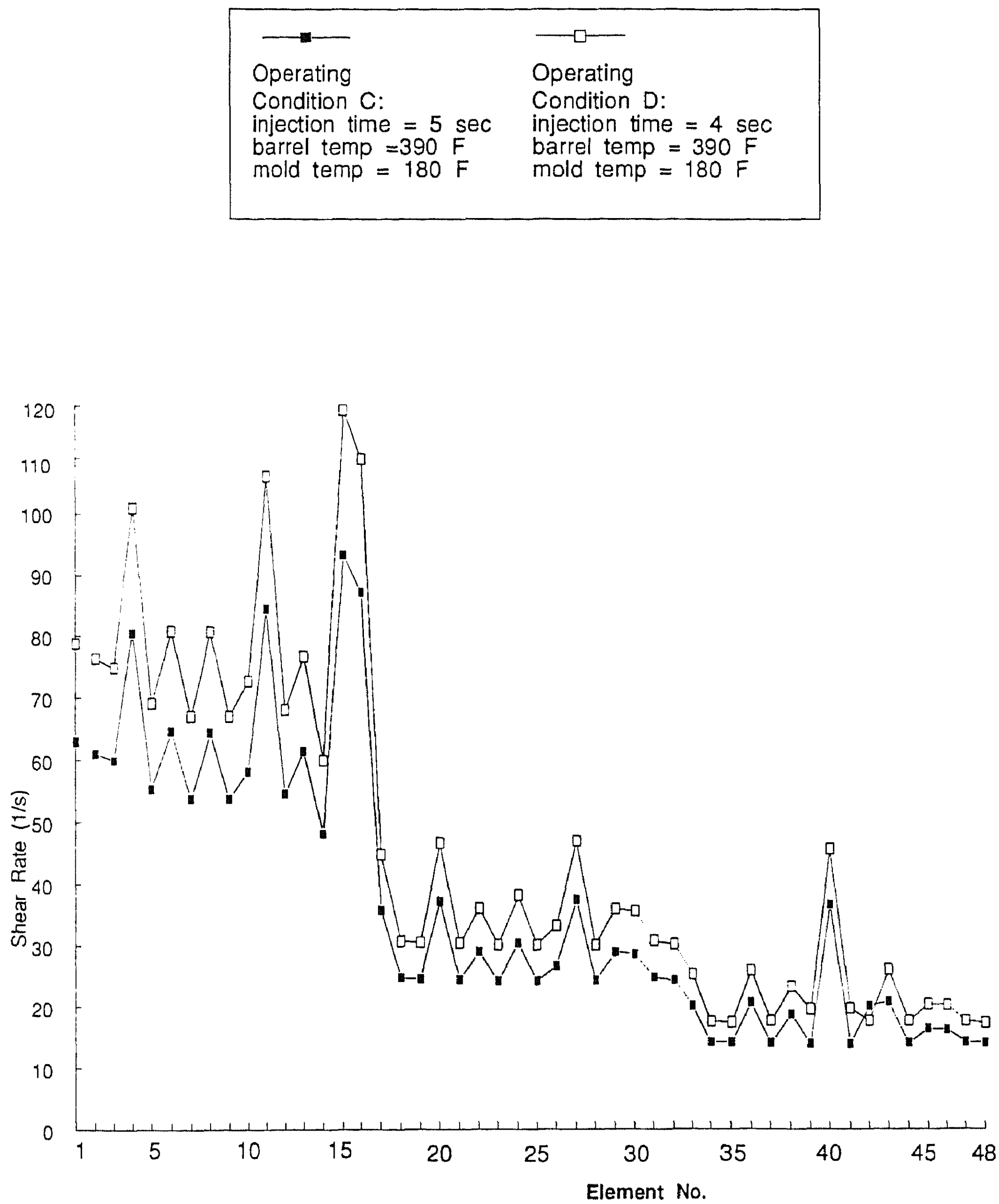


Figure 7.10 The comparison of the maximum shear rate in each element between *operating condition C* and *operating condition D*.

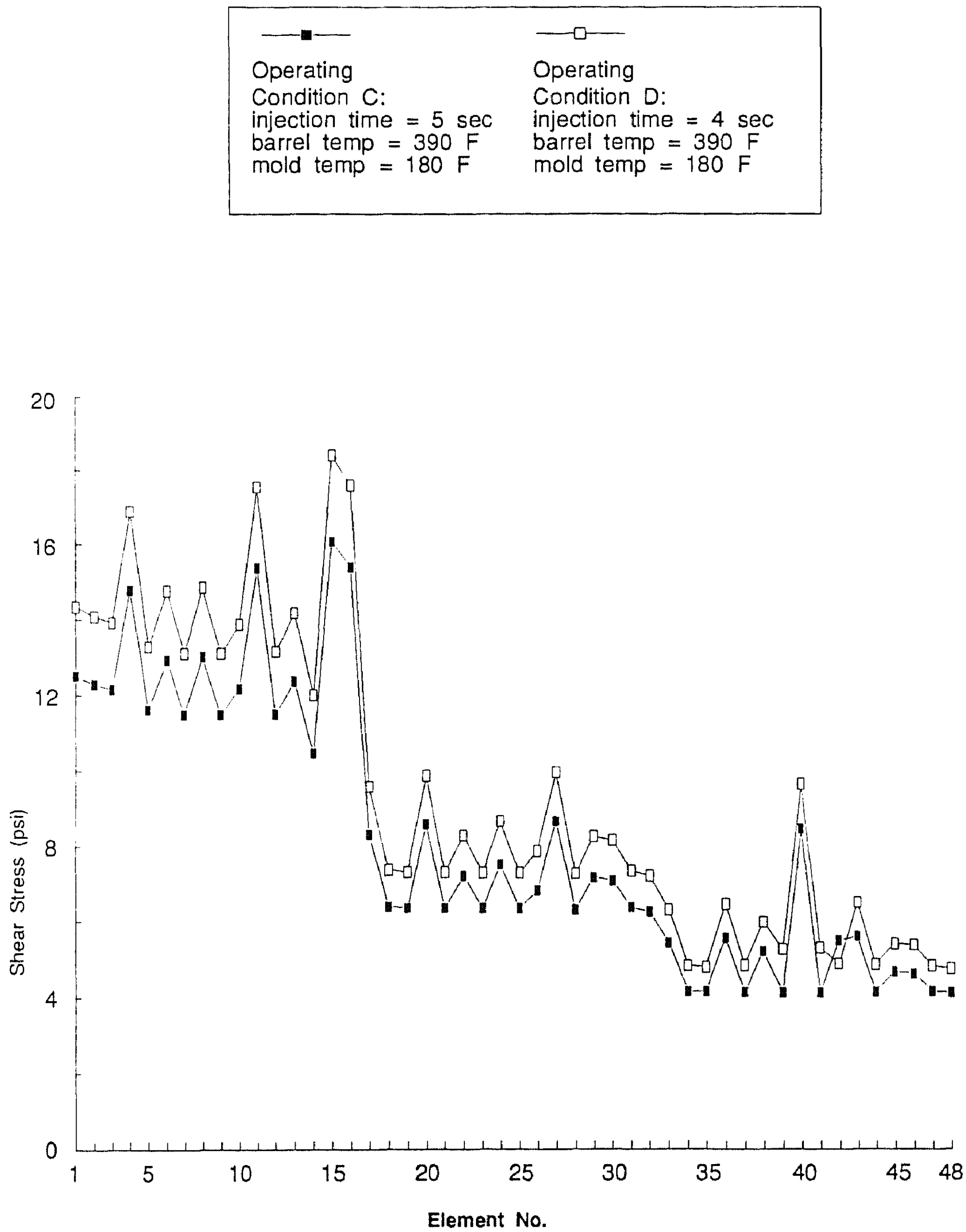


Figure 7.11 The comparison of the maximum shear stress in each element between *operating condition C* and *operating condition D*.

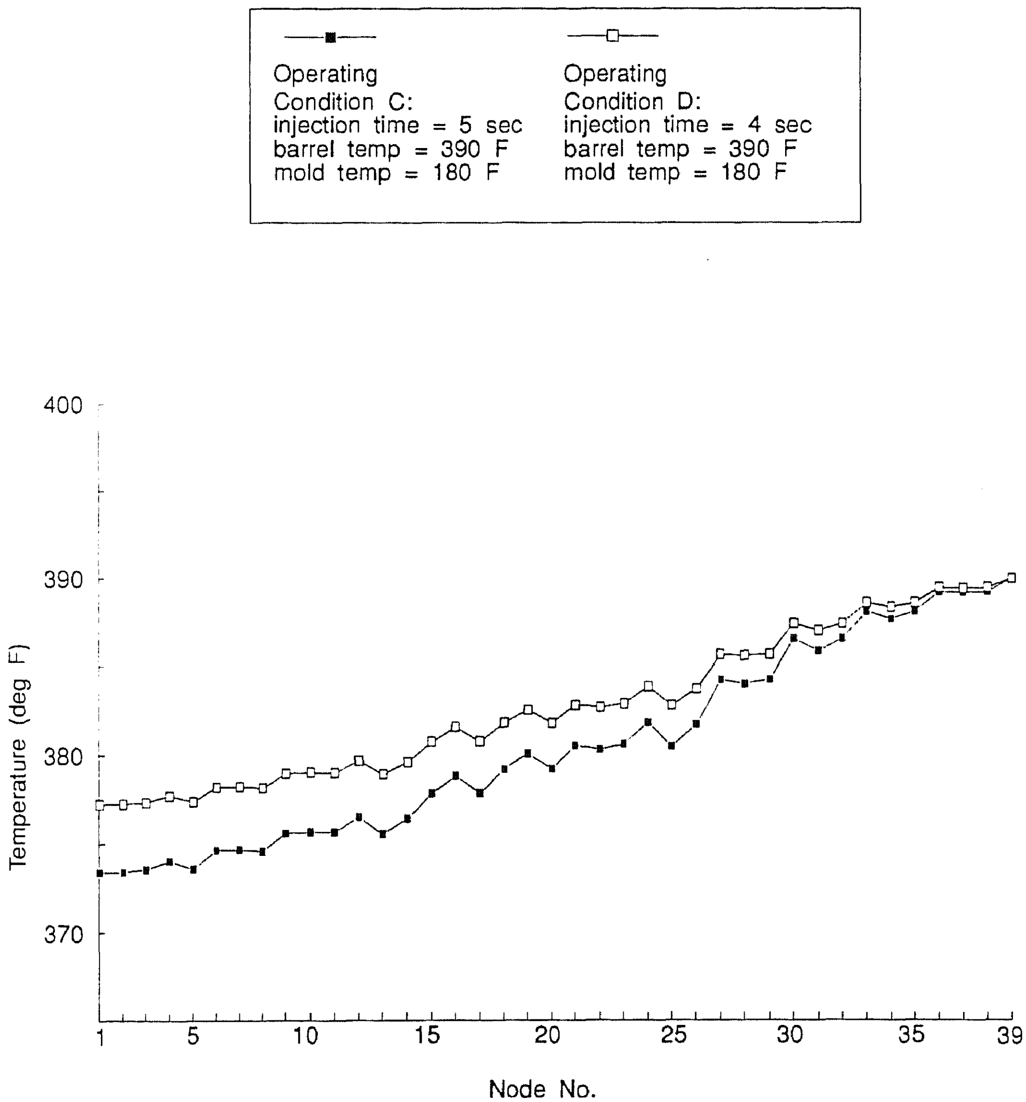


Figure 7.12 The comparison of the temperature in each node between *operating condition C* and *operating condition D*.

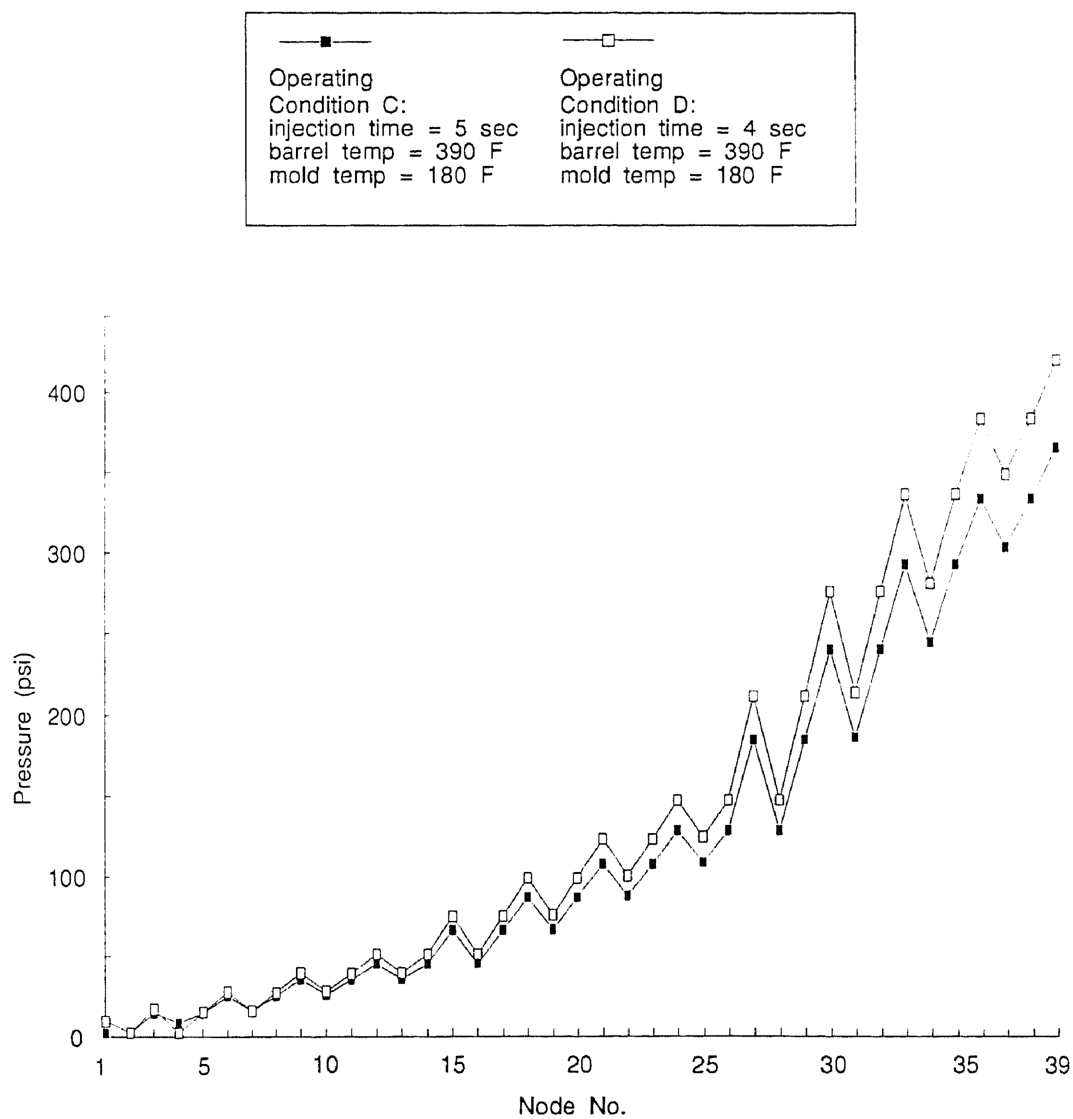


Figure 7.13 The comparison of the pressure in each node between operating *condition C* and *operating condition D*.

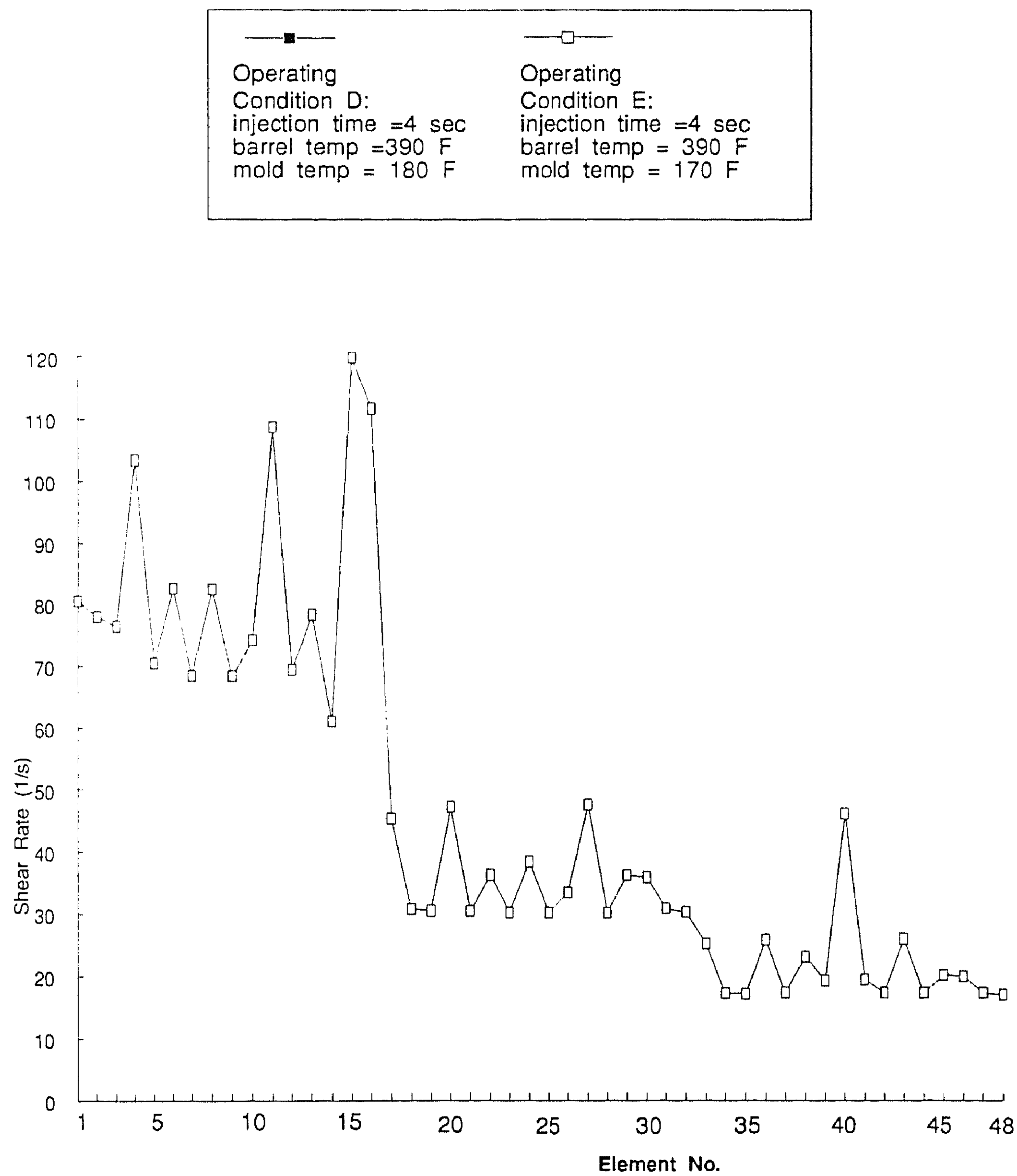


Figure 7.14 The comparison of the maximum shear rate in each element between *operating condition D* and *operating condition E*.

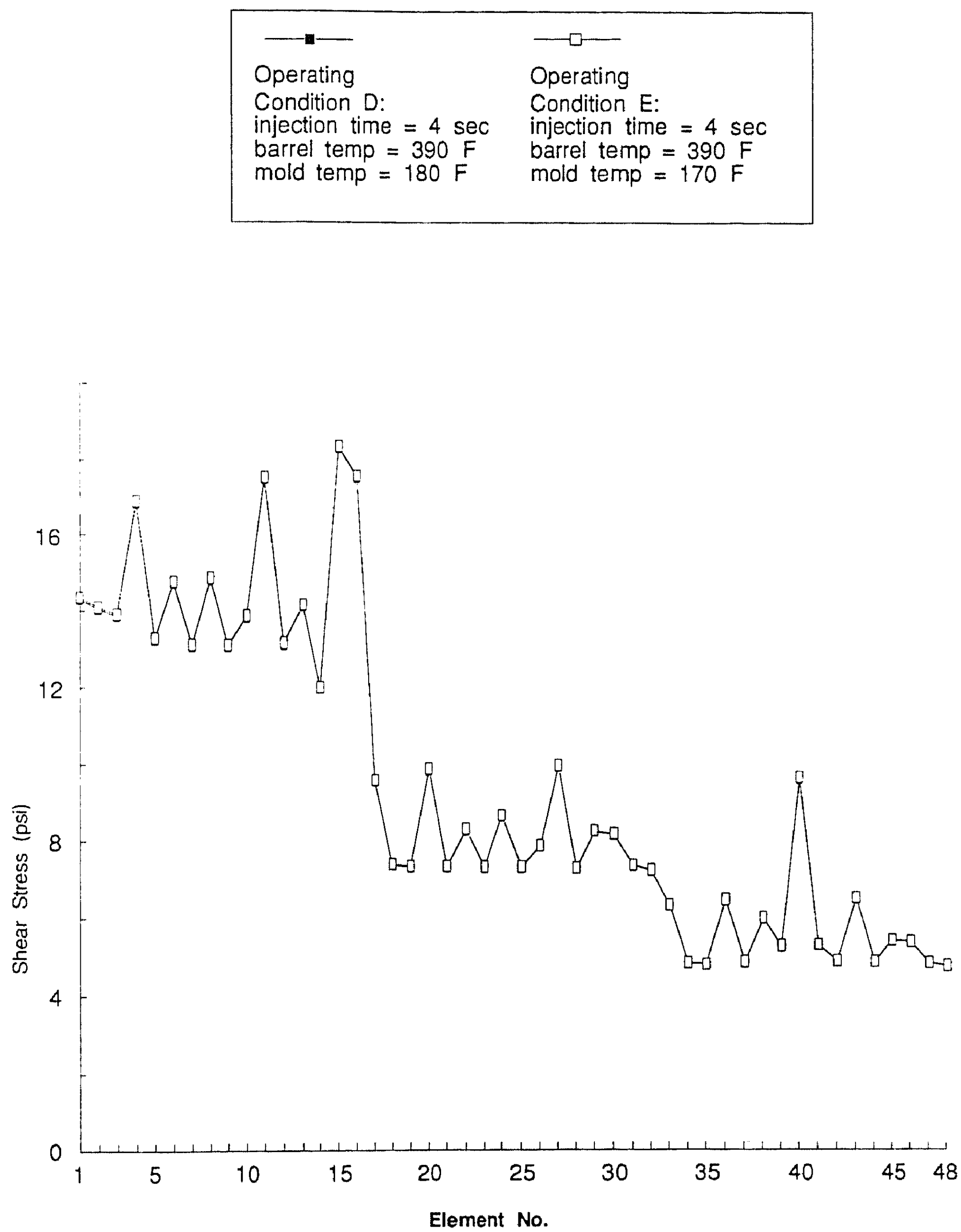


Figure 7.15 The comparison of the maximum shear stress in each element between *operating condition D* and *operating condition E*.

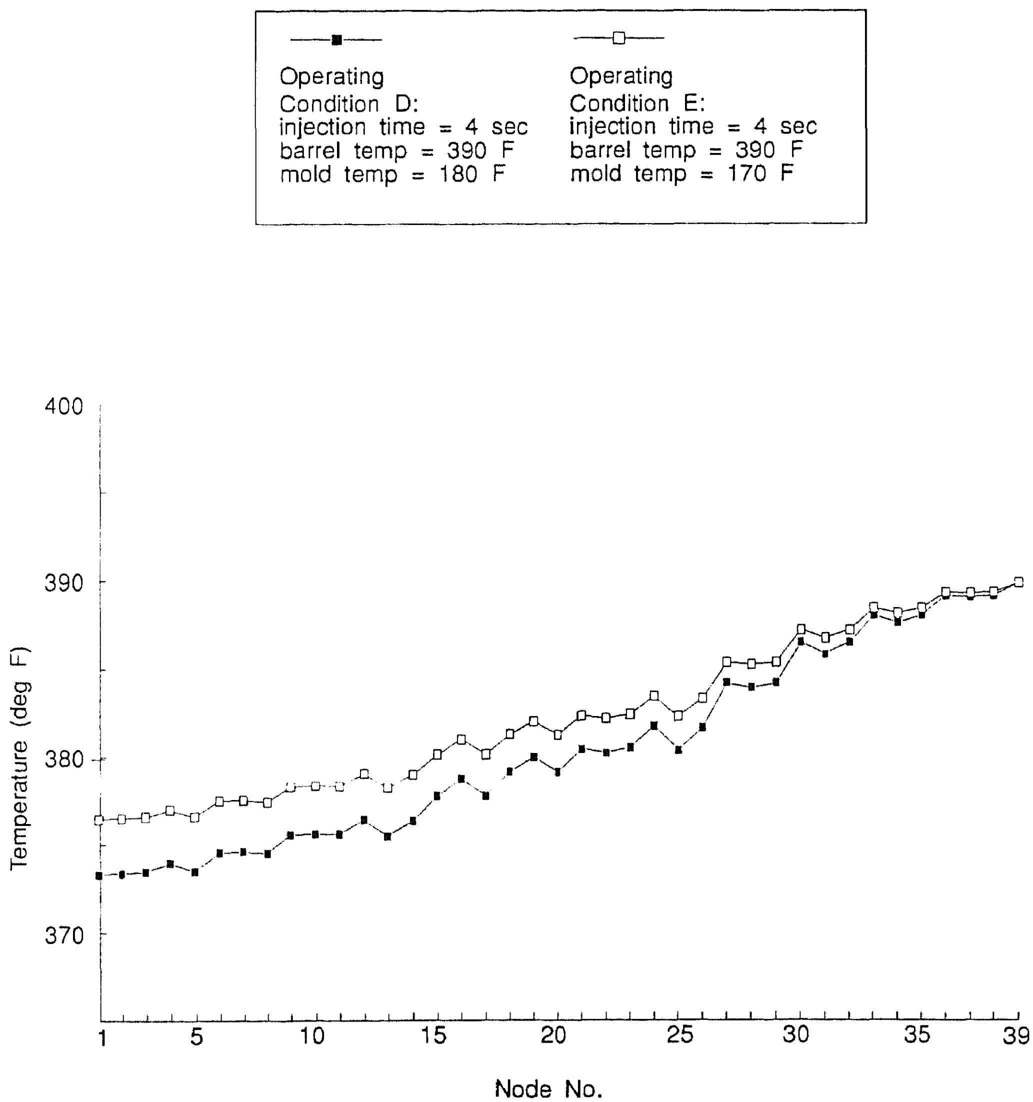


Figure 7.16 The comparison of the temperature in each node between *operating condition D* and *operating condition E*.

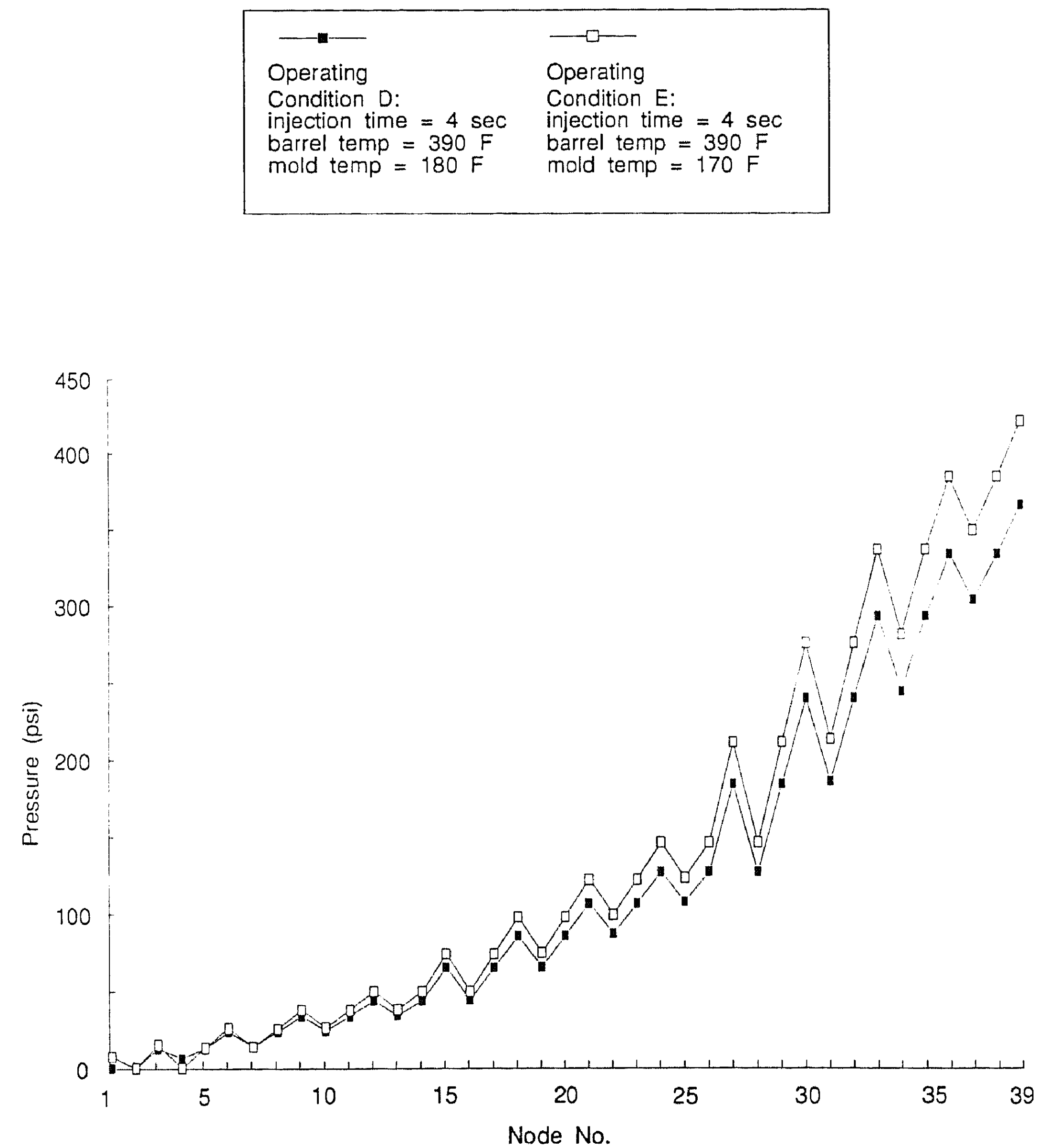


Figure 7.17 The comparison of the pressure in each node between the operating *condition D* and the *operating condition E*.

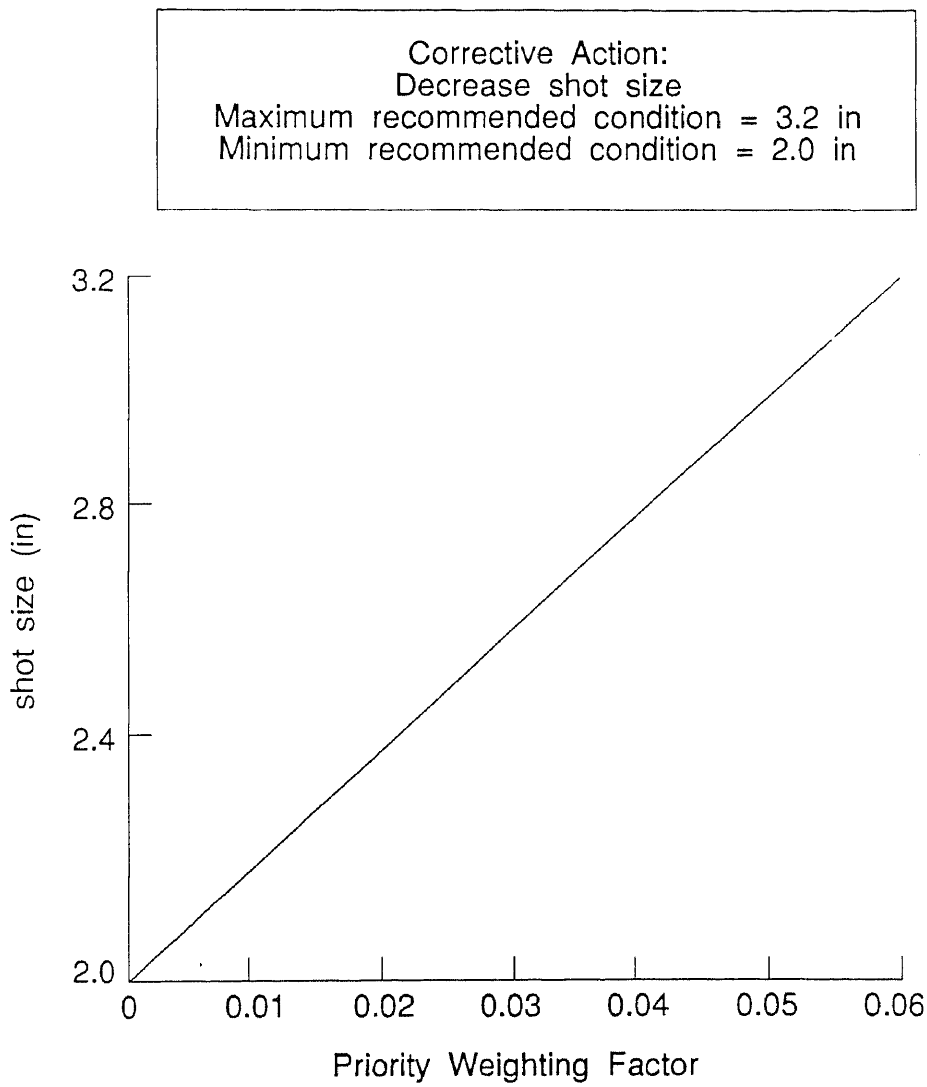


Figure 7.18 The relationship of the operating variable shot size and the priority weighting factor for the flashing deviation.

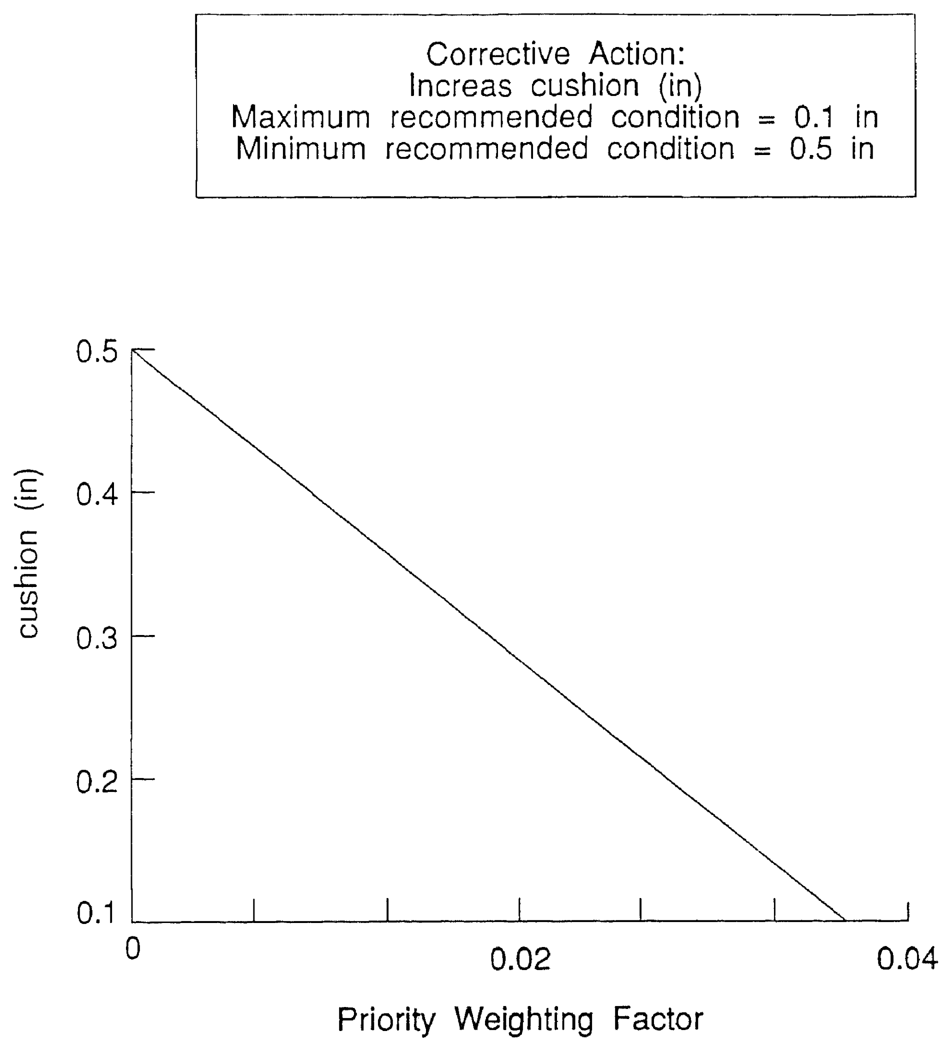


Figure 7.19 The relationship of the operating variable cushion and the priority weighting factor for the flashing deviation.

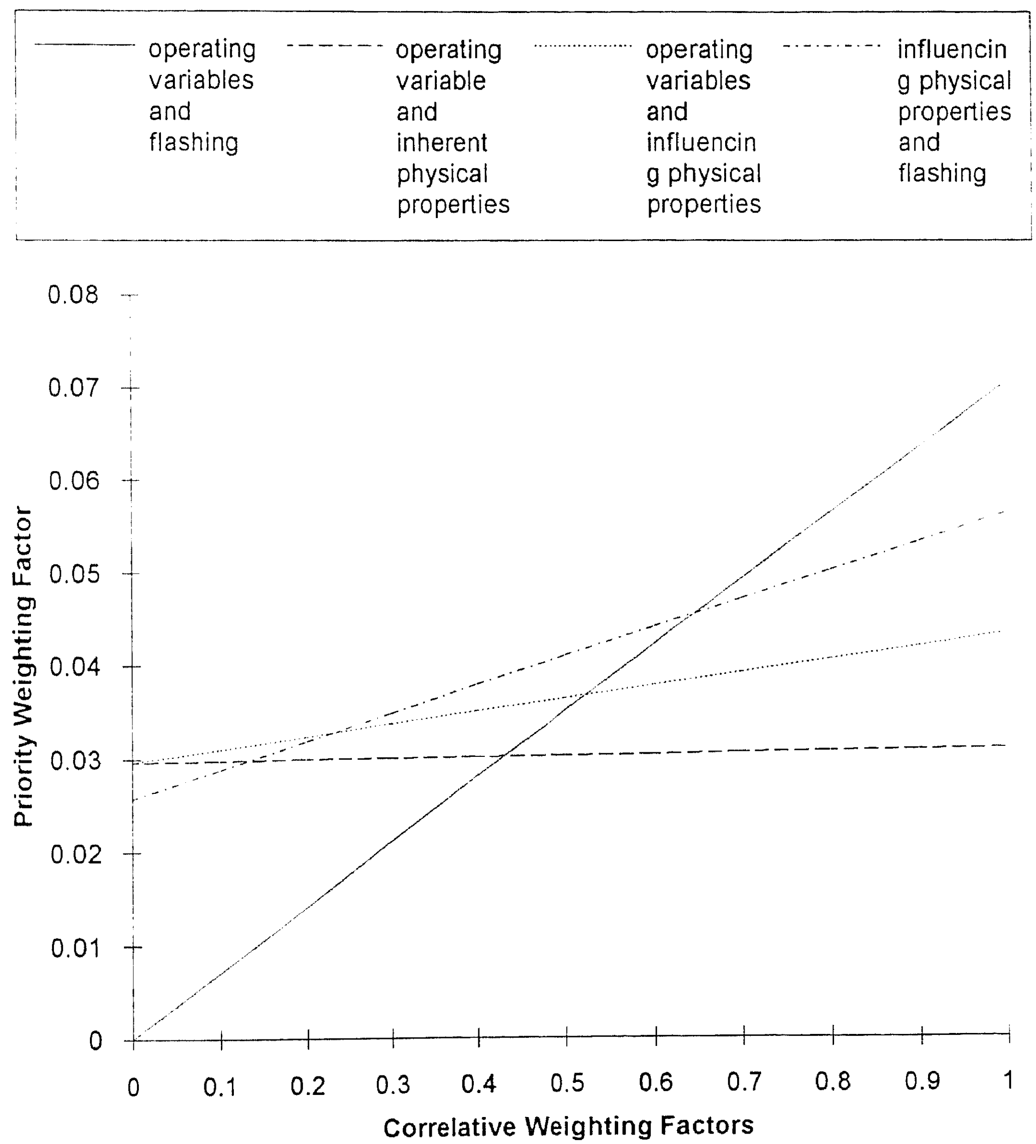


Figure 7.20 The relationship of the correlative weighting factors and the priority weighting factors for the operating variable shot size.

CHAPTER EIGHT
CONCLUSIONS AND FUTURE WORKS

8.1 Conclusions

1. An expert system for the injection molding of engineering thermoplastics has been developed, and the interplay of the controlling variables and the controlled variables has been researched. This expert system can be used in eliminating common deviations such as surface ripples, pit marks, sink marks, voids, flashing, short shots, warpage, distortion, and delamination for engineering resins. This expert system has been examined by the molding experts in the field of injection molding. The results from the system not only simulate the molding experts resolutions skills, but also, can accurately suggest actions for the user to eliminate the deviation.
2. The system has been simulated in the mold filling package, MOLDFLOW. From the simulation results, it can be concluded that the corrective actions provided by the system will directly influence the control parameters for the deviations. Furthermore, in simulations results, the employed sequence of the corrective actions can prioritized by their degree of influence for the controlled parameters.
3. A decision algorithm has been researched in this system. This decision algorithm assists in determining the rank for employing the operating variable corrective actions. From the confirmations of the molding experts and simulation results, this decision algorithm can exactly simulate the molding experts resolution procedures and provide an optimum resolution sequence for the user.
4. A self-learning mechanism has also been also embedded in this system. This self-learning mechanism based on the response of the resolution results modifies the parameters which influence the sequence of the corrective actions. It allows the system to learn from its own resolution procedures, and provides a speedier resolution sequence for the user. From the simulation results, it is confirmed that this self-learning

mechanism can accurately modify the parameters and provide a more significant resolution sequence for the user.

5. A confirmation mechanism has also been embedded in this system. This confirmation mechanism can prevent the incorrect input of parameters which influence resolution procedures. Furthermore, this mechanism allows the user to modify accordingly to their own experience with these parameters. Based on this modification, the system can learn from the user and provide an improved performance.
6. A explanation facility has been also developed in this study. This explanation facility allows the system to interpret the reasons for the corrective actions to the user. These explanations include why the declarative knowledge must be defined and why a particular corrective action is employed for resolving the problem.

8.2 Future Works

Several relative interesting works can be developed in the near future.

1. Expansion of the system to include all grades of all engineering thermoplastics. Currently, the expert system only provides the resolution knowledge of acetal copolymer, Celcon M-90. Allowing the expert system wide application in the field of injection molding, a complete resolution knowledge for all engineering thermoplastics used in injection molding can be developed.
2. Increase the intelligence tool of decision. Currently, the resolution task of the expert system is constrained to a single deviation. However, in the field of the injection molding, multi-deviation often occur. For the expert system technique become a valuable tool in the field of the injection molding, resolution skills for multi-deviation situations must be developed.
3. Input of experiences from actual molding situations. Currently, several correlative weighting factors are based on the consultation of experts. Allowing these correlative

weighting factors to accurately indicate the degree of influence for each correlative weighting factors, a scientific method to determine these correlative weighting factors must be developed.

4. To develop an interface which connects the expert system with the injection molding filling simulation package. Currently, there are several mold filling packages in the field of the injection molding. However, these packages do not have the ability to predict the deviation. Computer aided engineering widely applied in the injection molding process, allows the interface of the expert system and the mold filling simulation package.
5. To develop an interface connecting the expert system with the operating variable controller of the injection molding. Currently, most injection molding machines provide computer control to govern the whole process. Computer aided manufacturing widely applied in the injection molding process, allows the interface of the expert system and the control system to be developed.

APPENDIX A
PROGRAM LISTING

In this study, the system was coded by C programming language and was compiled in Tubro C. This program contains a main function and several functions. The functions include *copo_manuf, copo_celanese_grade, m90, de, di, fl, pm, sm, sp, sr, ss, vi, and wa*. In main function, it contains the declarative procedures of the material type.

In *copo_manuf* function, the declarative procedures of acetal copolymer manufacturer are inclued. Function *copo_celanese_grade* is the set of declarative procedures of the material grades for acetal copolymer, Hoechst Celanese. Function *m90* includes the declarative procedures of the recommended operating conditions, the operating conditions, the correlative weighting factors between the operating variables and the inherent physical properties, and the operating variables and the influencing physical properties, and deviation type for acetal copolymer, Celcon M-90.

The declarative procedures of the correlative weighting factors between the operating variables and the particular deviation, and the influencing physical properties and the particular deviation, and the resolution procedures of each deviation are presented in functions *de, di, fl, pm, sm, sp, sr, ss, vi, and wa* for deviation such as delamination, distortion, flashing, pit marks, sink marks, splay marks, surface ripples, short shots, voids, and warpage respectively.

The completed programming listing follows.

```
/*--EXPERT SYSTEM FOR INJECTION MOLDING OF
ENGINEERING THERMOPLASTICS--*/
#include <stdio.h>
#include <float.h>
#include <alloc.h>
#include <string.h>
#include <b:\program\head\choice.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#define material 12
char *MAT_NAME[material];     /*THE MATERIAL
NAME MATRIX*/
char why;                     /*THE INDICATED
ANSWER FOR REQUIRING EXPLAININATION*/
int mc;                       /*THE RESELECT
INDICATED NUMBER OF NEW PROCEDURE*/
main()
{
int mat_choice;               /*THE MATERIAL
INDICATED NUMBER*/
int i,j,k;                    /*THE LOOP
INDICATED NUMBER*/
FILE *mat_name;                /*THE DATA
FILE FOR MATERIAL NAME*/
FILE *fi;
fi = fopen("b:\\program\\output.doc","w");
/*--DECLARE THE MEMORY AREA FOR MATERIAL
MATRIX--*/
for (i=0; i<material; i++)
  MAT_NAME[i] = (char*)malloc(25);
/*--PRINT OUT THE HEAD TITLE AND MATERIAL
CHOICE TITLE--*/
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
space((65 - strlen(INTRO_TITLE1))/2);
printf("%s\n",INTRO_TITLE1);
for (i = 0; i < ((65 -
strlen(INTRO_TITLE1))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",INTRO_TITLE1);
space((65 - strlen(INTRO_TITLE2))/2);
printf("%s\n",INTRO_TITLE2);
for (i = 0; i < ((65 -
strlen(INTRO_TITLE2))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",INTRO_TITLE2);
space((65 - strlen(INTRO_TITLE3))/2);
printf("%s\n",INTRO_TITLE3);
for (i = 0; i < ((65 -
strlen(INTRO_TITLE3))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",INTRO_TITLE3);
space((65 - strlen(INTRO_TITLE4))/2);
printf("%s\n",INTRO_TITLE4);
for (i = 0; i < ((65 -
strlen(INTRO_TITLE4))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",INTRO_TITLE4);
space((65 - strlen(INTRO_TITLE5))/2);
printf("%s\n",INTRO_TITLE5);
for (i = 0; i < ((65 -
strlen(INTRO_TITLE5))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",INTRO_TITLE5);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
do   {
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  space((65 - strlen(MAT_TITLE1))/2);
  printf("%s\n",MAT_TITLE1);
  printf("%s\n",BORDER);
  for (i = 0; i < ((65 -
strlen(MAT_TITLE1))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",MAT_TITLE1);
  fprintf(fi,"%s\n",BORDER);
/*--EXPLANATION FACILITY FOR MATERIAL
CHOICE--*/
  printf("\n%s\n",require);
  fprintf(fi,"\n%s\n",require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why =='?'){
    printf("%s\n",BORDER);
    printf("%s\n",mat_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"%s\n",mat_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
  printf("\n%s\n",MAT_TITLE2);
  fprintf(fi,"\n%s\n",MAT_TITLE2);
/*--READ THE MATERIAL NAME FROM
B:\MATNAME.DAT--*/
  mat_name = fopen
("b:\\program\\matname.dat","r");
  for (i=0; i<material; i++){
    fscanf(mat_name,"%s",MAT_NAME[i]);
    printf("%d%s%s\n",i+1,". ",MAT_NAME[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",MAT_NAME[i]);}
  fclose(mat_name);
  printf("\nMATERIAL CODE NUMBER = ");
```

```
  fprintf(fi,"\nMATERIAL CODE NUMBER = ");
  scanf("%d",&mat_choice);
  fprintf(fi,"%d",mat_choice);
  if (mat_choice == 12)
    break;
  else {
    printf("\n%s%d\n%s%s\n",
         "YOUR MATERIAL INDICATED NUMBER IS :
",mat_choice,
         "YOUR MATERIAL NAME IS :
",MAT_NAME[mat_choice-1]);
    fprintf(fi,"\n%s%d\n%s%s\n",
       "YOUR MATERIAL INDICATED NUMBER IS :
",mat_choice,
       "YOUR MATERIAL NAME IS :
",MAT_NAME[mat_choice-1]);}
  if ( mat_choice !=1 ){
    printf("\n%s",matchoice);
    fprintf(fi,"\n%s",matchoice);
    scanf("%d",&mc);
    fprintf(fi,"%d",mc);
    fprintf(fi,"\n");}
/*--CALL FUNCTION ACETAL COPOLYMER*/
  if (mat_choice ==1 ){
    fclose(fi);
    mc = copo_manuf();}
  }while (mc==1);
printf("\nSTOP THE PROGRAM\n");
fprintf(fi,"\nSTOP THE PROGRAM\n");
return 0;
}

/*--FUNCTION OF COPOLYMER MANUFACTURER--*/
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <b:\program\head\choice.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#define manuf 5
copo_manuf()
{
char *MANUF_NAME[manuf];        /*THE
MANUFACTURER MATRIX*/
char why;                       /*THE
EXPLANATION INDICATED SIGNLE*/
int mc_manuf;
int manuf_choice;               /*THE INDICATED
NUMBER FOR MANUFACTURER*/
int i,j,k;                      /*THE LOOP
INDICATED NUMBER*/
FILE *copolymer_manuf;          /*THE DATA FILE
FOR COPOLYMER MANUFACTURER*/
FILE *fi;
fi =fopen("b:\\program\\output.doc","a");
/*--DECLARE THE MEMORY AREA FOR COPOLYMER
MANUFACTURER MATRIX--*/
for (i=0; i<manuf; i++)
  MANUF_NAME[i] = (char*)malloc(25);
do  {
  mc_manuf = 0;
  copolymer_manuf =
fopen("b:\\program\\ac_manu.dat","r");
  printf("\n%s\n",BORDER);
  space((65 - strlen(MANUF_TITLE1))/2);
  printf("%s\n%s\n",MANUF_TITLE1,BORDER);
  printf("\n%s\n",require);
  fprintf(fi,"\n%s\n",BORDER);
  for (i = 0; i < ((65 -
strlen(MANUF_TITLE1))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n%s\n",MANUF_TITLE1,BORDER)
;
  fprintf(fi,"\n%s\n",require);
/*--THE EXPLANATION STATEMENTS--*/
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("%s\n",BORDER);
    printf("%s\n",manu_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"%s\n",manu_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
  printf("\n%s\n%s\n",MANUF_TITLE2,MANUF_TIT
LE3);
  fprintf(fi,"\n%s\n%s\n",MANUF_TITLE2,MANUF
_TITLE3);
  for (i=0; i<manuf; i++){
    fscanf(copolymer_manuf,"%s"
,MANUF_NAME[i]);
    printf("%d%s%s\n",i+1,".
",MANUF_NAME[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",MANUF_NAME[i]);}
  printf("\nMANUFACTURER CODE NUMBER = ");
  fprintf(fi,"\nMANUFACTURER CODE NUMBER =
");
  scanf("%d",&manuf_choice);
  fprintf(fi,"%d\n",manuf_choice);
  printf("\nYOUR MANUFACTURER CODE NUMBER IS
: %d \nYOUR MANUFACTURER IS
```

```
%s\n",manuf_choice,MANUF_NAME[manuf_choice-
1]);
  fprintf(fi,"\nYOUR MANUFACTURER CODE
NUMBER IS : %d \nYOUR MANUFACTURER IS
%s\n",manuf_choice,MANUF_NAME[manuf_choice-
1]);
  fclose(copolymer_manuf);
  if (manuf_choice != 1){
    printf("\n\n%s",menuchoice);
    fprintf(fi,"\n\n%s",menuchoice);
    scanf("%d",&mc_manuf);
    fprintf(fi,"%d\n");}
  else {
    fclose(fi);
    mc_manuf = copo_celanese_grade();
    printf("%d",mc_manuf);}
  }while (mc_manuf == 2);
return (mc_manuf);
}

/*--FUNCTION OF CELANESE CELCON GRADE--*/
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include <b:\program\head\choice.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#define grade 28
copo_celanese_grade()
{
char *GRADE_NAME[grade];      /*THE GRADE
MATRIX*/
char why;                     /*THE
EXPLANATION INDICATED SIGNLE*/
int mc_grade;
int grade_choice;             /*THE INDICATED
NUMBER FOR GRADE*/
int i,j,k;                    /*THE LOOP
INDICATED NUMBER*/
FILE *celcon_grade;           /*THE DATA FILE
FOR CELCON GRADE*/
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
/*--DECLARE THE MEMORY AREA FOR CELCON GRADE
MATRIX--*/
for (i=0; i<grade; i++)
  GRADE_NAME[i] = (char*)malloc(25);
do  {
  mc_grade = 0;
  celcon_grade =
fopen("b:\\program\\ac_c_gra.dat","r");
  printf("\n%s\n",BORDER);
  space((65 - strlen(GRADE_TITLE1))/2);
  printf("%s\n%s\n",GRADE_TITLE1,BORDER);
  fprintf(fi,"\n%s\n",BORDER);
  for (i = 0; i < ((65 -
strlen(GRADE_TITLE1))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n%s\n",GRADE_TITLE1,BORDER)
;
/*--EXPLANATION STATEMENTS--*/
  printf("\n%s\n",require);
  fprintf(fi,"\n%s\n",require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("%s\n",BORDER);
    printf("%s\n",grade_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"%s\n",grade_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
  printf("\n%s\n%s\n",GRADE_TITLE2,GRADE_TIT
LE3);
  fprintf(fi,"\n%s\n%s\n",GRADE_TITLE2,GRADE
_TITLE3);
  for (i=0; i<grade; i++){
    fscanf(celcon_grade,"%s",
GRADE_NAME[i]);}
  fclose (celcon_grade);
  for (i=0; i<7; i++){
    for (j=0; j<4; j++){
      printf("%2d%s%-12s",i*4+j+1,".
",GRADE_NAME[i*4+j]);
      fprintf(fi,"%2d%s%-12s",i*4+j+1,".
",GRADE_NAME[i*4+j]);}
    printf("\n");
    fprintf(fi,"\n");}
  printf("\nGRADE CODE NUMBER = ");
  fprintf(fi,"\nGRADE CODE NUMBER = ");
  scanf("%d",&grade_choice);
  fprintf(fi,"%d",grade_choice);
  printf("\nYOUR CELCON GRADE CODE IS :
%d\nYOUR CELCON GRADE NAME IS :
%s\n",grade_choice,GRADE_NAME[grade_choice-
1]);
  fprintf(fi,"\nYOUR CELCON GRADE CODE IS :
%d\nYOUR CELCON GRADE NAME IS :
%s\n",grade_choice,GRADE_NAME[grade_choice-
1]);
  if (grade_choice != 14){
    printf("\n\n%s",gradechoice);
    fprintf(fi,"\n\n%s",gradechoice);
```

```
    scanf("%d",&mc_grade);
    fprintf(fi,"%d\n",mc_grade);}
  else {
    fclose(fi);
    mc_grade=m90();}
  }while (mc_grade == 3);
return (mc_grade);
}

/*--CELCON M90 FUNCTION--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#define REC_CHANGE          "DO ANY
RECOMMENDED CONDITIONS NEED TO BE CHANGED"
#define VAR_HP_CWF_CHANGE "DO ANY
CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING\nVARIABLES AND INHERENT PHYSICAL
PROPERTIES NEED TO BE CHANGED"
#define VAR_FP_CWF_CHANGE "DO ANY
CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING\nVARIABLES AND INFLUENCING
PHYSICAL PROPERTIES NEED TO BE CHANGED"
#define VAR_CHANGE          "DO ANY OPERATING
VARIABLES NEED TO BE CHANGED"
#define REC_REQUIRE          "WHICH
RECOMMENDED CONDITIONS NEED TO BE
CHANGED\nPLEASE INDICATE BY ENTERING THE
CODE NUMBER"
#define VAR_HP_CWF_REQUIRE "WHICH
CORRELATIVE WEIGHTING FACTORS NEED TO BE
CHANGED\nPLEASE INDICATE BY ENTERING THE
CODE NUMBER"
#define VAR_FP_CWF_REQUIRE "WHICH
CORRELATIVE WEIGHTING FACTORS NEED TO BE
CHANGED\nPLEASE INDICATE BY ENTERING THE
CODE NUMBER"
#define VAR_REQUIRE          "WHICH OPERATING
VARIABLES NEED TO BE CHANGED\nPLEASE
INDICATE BY ENTERING THE CODE NUMBER"
#define deviation 10    /*THE NUMBER OF
DEVIATION*/
m90()
{
char *VAR_NAME[variable];     /*THE NAME OF
OPERATING VARIABLES*/
char *MOLD_NAME[mold];        /*THE NAME OF
MOLD SIZE*/
char *RANGE_NAME[range];      /*THE NAME OF
OPERATING VARIABLES RANGE*/
char *HP_NAME[hp];            /*THE NAME OF
INHERENT PHYSICAL PROPERTY*/
char *FP_NAME[fp];            /*THE NAME OF
INFLUENCING PHYSICAL PROPERTY*/
char *DEV_NAME[deviation];    /*THE NAME OF
DEVIATION*/
char USER[12];                /*THE USER
OUTPUT DIRECTORY*/
char why;                     /*THE
EXPLANATION INDICATED SINGLE*/
float VAR_REC[variable][range-1]; /*THE
RECOMMENDED CONDITION OF OPERATING
VARIABLE*/
float new_var_rec;            /*THE CHANGED
VALUE OF RECOMMENDED CONDITION*/
float MOLD_REC[mold][range-1];   /*THE
RECOMMENDED CONIDITION OF MOLD SIZE*/
float VAR_HP_CWF[variable][hp]; /*THE CWF
BETWEEN VAR AND HP*/
float new_var_hp_cwf;         /*THE CHANGED
VALUE OF CWF BETWEEN VAR AND HP*/
float VAR_FP_CWF[variable][fp];     /*THE
CWF BETWEEN VAR AND FP*/
float new_var_fp_cwf;          /*THE CHANGED
VALUE OF BETWEEN VAR AND FP*/
float VAR[variable];          /*THE USER
OPERATING VARIABLE*/
float new_var;
float MOLD[mold];
int var_rec_change;           /*THE INDICATED
NUMBER FOR CHANGED VAR_REC*/
int fp_change;            /*THE INDICATED
NUMBER FOR CHANGED VAR-FP*/
int hp_change;            /*THE INDICATED
NUMBER FOR CHANGED VAR_HP*/
int var_change;               /*THE INDICATED
NUMBER FOR CHANGED USER VAR*/
int dev_choice;               /*THE INDICATED
NUMBER FOR DEVIATION*/
int yn;                       /*THE DECISION
INDICATED NUMBER FOR CHANGE OR NOT*/
int i,j,k;                    /*THE ROUTE
INDICATED NUMBER*/
int mc_m90;                   /*THE RESELECT
INDICATED NUMBER*/
int USER_ANSWER;              /*THE USER
CONFIRM KEY FOR MOLDED PART WHETHER BEEN
USED IN THIS SYSTEM OR NOT*/
FILE *var_rec;                /*THE DATA FILE
FOR RECOMMENDED CONDITION*/
FILE *mold_rec;               /*THE DATA FILE
FOR MOLD RECOMMENDED CONDITION*/
```

```
FILE *var_hp_cwf;              /*THE DATA FILE
FOR CWF BETWEEN VAR AND PP*/
FILE *var_fp_cwf;               /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *dev_name;                /*THE DATA FILE
FOR DEVIATION NAME*/
FILE *var;                     /*THE DATA FILE
FOR USER OPERATING VARIABLE*/
FILE *mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
/*--DECLARE MEMORY FOR VAR_NAME AND PP_NAME-
-*/
for (i=0; i<variable; i++){
  VAR_NAME[i] = (char*)malloc(40);
  if ((VAR_NAME[i] = (char*)malloc(40)) ==
0)
    printf("error");}
for (i=0; i<mold; i++){
  MOLD_NAME[i] = (char*)malloc(40);
  if ((MOLD_NAME[i] = (char*)malloc(40)) ==
0)
    printf("error");}
for (i=0; i<hp+1; i++){
  HP_NAME [i] = (char*)malloc(30);
  if ((HP_NAME[i] = (char*)malloc(30)) == 0)
    printf("error");}
for (i=0; i<fp+1; i++){
  FP_NAME[i] = (char*)malloc(20);
  if ((FP_NAME[i] = (char*)malloc(20)) == 0)
    printf("error");}
for (i=0; i<range; i++){
  RANGE_NAME [i] = (char*)malloc(15);
  if ((RANGE_NAME[i] = (char*)malloc(15)) ==
0)
    printf("error");}
for (i=0; i<deviation; i++){
  DEV_NAME[i] = (char*)malloc(30);
  if ((DEV_NAME[i] = (char*)malloc(30)) ==
0)
    printf("error");}
/*RETRIEVE THE OLD DATA INTO THE SYSTEM*/
printf("\n%s\n%s\n%s\n%s\n",
    "DOES THE MOLDED PART HAVE A PRIOR
HISTORY IN THE SYSTEM?",
    "PLEASE INDICATE YOUR ANSWER BY ENTERING
THE CODE NUMBER",
    "1. YES",
    "2. NO");
fprintf(fi,"\n%s\n%s\n%s\n%s\n",
    "DOES THE MOLDED PART HAVE A PRIOR
HISTORY IN THE SYSTEM?",
    "PLEASE INDICATE YOUR ANSWER BY ENTERING
THE CODE NUMBER",
    "1. YES",
    "2. NO");scanf("%d",&USER_ANSWER);
fprintf(fi,"%d\n",USER_ANSWER);
if (USER_ANSWER == 1){
  printf("\n%s\n%s\n%s",
      "YOUR ANSWER IS THAT THE MOLDED PART
HAS A PRIOR HISTORY IN THIS SYSTEM",
      "PLEASE ENTER THE 'OLD' MOLDED PART
NAME",
      "'OLD' MOLDED PART NAME = ");
  fprintf(fi,"\n%s\n%s\n%s",
      "YOUR ANSWER IS THAT THE MOLDED PART
HAS A PRIOR HISTORY IN THIS SYSTEM",
      "PLEASE ENTER THE 'OLD' MOLDED PART
NAME",
      "'OLD' MOLDED PART NAME = ");
  scanf("%s",USER);
  chdir(USER);
  var_rec = fopen("varrec.dat","r");
  mold_rec = fopen ("moldrec.dat","r");
  var_hp_cwf = fopen ("varhpcwf.dat","r");
  var_fp_cwf = fopen ("varfpcwf.dat","r");
  chdir("C:\\program");}
if (USER_ANSWER == 2){
  printf("\n%s\n%s\n%s",
      "YOUR ANSWER IS THAT THE MOLDED PART
HAS NO PRIOR HISTORY IN THIS SYSTEM",
      "PLEASE ENTER THE 'NEW' MOLDED PART
NAME WITH 8 CHARACTERS",
      "'NEW' MOLDED PART NAME = ");
  fprintf(fi,"\n%s\n%s\n%s",
      "YOUR ANSWER IS THAT THE MOLDED PART
HAS NO PRIOR HISTORY IN THIS SYSTEM",
      "PLEASE ENTER THE 'NEW' MOLDED PART
NAME WITH 8 CHARACTERS",
      "'NEW' MOLDED PART NAME = ");
  scanf("%s",USER);
  mkdir(USER);
  var_rec =
fopen("b:\\program\\initial.m90\\varrec.dat"
,"r");
  mold_rec =
fopen("b:\\program\\initial.m90\\moldrec.dat
","r");
  var_hp_cwf = fopen
("b:\\program\\initial.m90\\varhpcwf.dat","r
");
  var_fp_cwf = fopen
("b:\\program\\initial.m90\\varfpcwf.dat","r
");}
/*--FILE OPEN FOR RECOMMENDED CONDITION--*/
for (i=0; i<range; i++)
  fscanf(var_rec,"%s",RANGE_NAME[i]);
for (i=0; i<variable; i++){
```

```
  fscanf(var_rec,"%s",VAR_NAME[i]);
  for (j=0; j<range-1; j++) {
    fscanf(var_rec,"%f",&VAR_REC[i][j]);}}
for (i=0; i<mold; i++){
  fscanf(mold_rec,"%s",MOLD_NAME[i]);
  for (j=0; j<range-1; j++){
    fscanf(mold_rec,"%f",&MOLD_REC[i][j]);}}
fclose(var_rec);
fclose(mold_rec);
/*--TITLE STATEMENTS RECOMMENDED CONDITIONS-
-*/
printf("\n%s\n",BORDER);
space((65 - strlen(REC_TITLE1))/2);
printf("%s\n",REC_TITLE1);
space((65 - strlen(REC_TITLE2))/2);
printf("%s\n",REC_TITLE2);
printf("%s\n",BORDER);
fprintf(fi,"\n%s\n",BORDER);
for (i = 0; i < ((65 -
strlen(REC_TITLE1))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",REC_TITLE1);
for ( i = 0; i < ((65 -
strlen(REC_TITLE2))/2);i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",REC_TITLE2);
fprintf(fi,"%s\n",BORDER);
/*--EXPLANATION STATEMENTS FOR RECOMMENDED
CONDITIONS--*/
printf("\n%s\n",require);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("%s\n",BORDER);
  printf("%s\n",rec_why);
  printf("%s\n",BORDER);
  printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"%s\n",rec_why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
  why = getch();}
/*--PRINTOUT THE RECOMMENDED OPERATING
CONDIITONS--*/
printf("\n%s\n",BORDER);
space((65 - strlen(REC_TITLE3))/2);
printf("%s\n",REC_TITLE3);
printf("%s\n",BORDER);
printf("%32s",RANGE_NAME[0]);
fprintf(fi,"\n%s\n",BORDER);
for (i = 0; i < ((65 -
strlen(REC_TITLE3))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",REC_TITLE3);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"%32s",RANGE_NAME[0]);
for (i=1; i<range; i++){
  printf("%10s",RANGE_NAME[i]);
  fprintf(fi,"%10s",RANGE_NAME[i]);}
printf("\n");
fprintf(fi,"\n");
for (i=0; i<variable; i++){
  printf("%32s",VAR_NAME[i]);
  fprintf(fi,"%32s",VAR_NAME[i]);
  for (j=0; j<range-1; j++) {
    printf("%10.2f",VAR_REC[i][j]);
    fprintf(fi,"%10.2f",VAR_REC[i][j]);}
  printf("\n");
  fprintf(fi,"\n");}
for (i=0; i<mold; i++){
  printf("%32s",MOLD_NAME[i]);
  fprintf(fi,"%32s",MOLD_NAME[i]);
  for (j=0; j<range-1; j++) {
    printf("%10.2f",MOLD_REC[i][j]);
    fprintf(fi,"%10.2f", MOLD_REC[i][j]);}
  printf("\n");
  fprintf(fi,"\n");}
printf("\n%s\n%s\n%s",REC_CHANGE,YN,ANSWER);
fprintf(fi,"\n%s\n%s\n%s",REC_CHANGE,YN,ANSW
ER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fprintf(fi,"\n");
/*--CONFIRM THE RECOMMENDED CONDITION--*/
if (yn == 1){
  do  {
    printf("\n%s\n",REC_REQUIRE);
    fprintf(fi,"\n%s\n",REC_REQUIRE);
    for (i=0; i<variable; i++){
      printf("%d%s%s\n",i+1,"
.",VAR_NAME[i]);
      fprintf(fi,"%d%s%s\n",i+1,"
.",VAR_NAME[i]);}
    for (i=0; i<mold; i++){
      printf("%d%s%s\n",i+variable+1,"
.",MOLD_NAME[i]);
      fprintf(fi,"%d%s%s\n", i+variable+1,"
.",MOLD_NAME[i]);}
    printf("\nOPERATING VARIABLE CODE NUMBER
= ");
    fprintf(fi,"\nOPERATING VARIABLE CODE
NUMBER = ");
    scanf("%d",&var_rec_change);
    fprintf(fi,"%d",var_rec_change);
```

```
    if (var_rec_change <=variable){
      printf("\nOLD RECOMMENDED CONDITIONS
OF %s ARE \n",VAR_NAME[var_rec_change-1]);
      fprintf(fi,"\nOLD RECOMMENDED
CONDITIONS OF %s ARE
\n",VAR_NAME[var_rec_change-1]);
      for (i=1; i<range; i++){
        printf("%10s",RANGE_NAME[i]);
        fprintf(fi,"%10s", RANGE_NAME[i]);}
    printf("\n");
    fprintf(fi,"\n");
    for (i=0; i<range-1; i++){
      printf("%10.2f",
VAR_REC[var_rec_change-1][i]);
      fprintf(fi,"%10.2f",
VAR_REC[var_rec_change-1][i]);}
/*--INPUT THE NEW RECOMMENDED OPERATING
CONDITIONS--*/
    printf("\nPLEASE INPUT YOUR NEW
RECOMMENDED CONDITIONS FOR
%s\n",VAR_NAME[var_rec_change-1]);
    fprintf(fi,"\nPLEASE INPUT YOUR NEW
RECOMMENDED CONDITIONS FOR
%s\n",VAR_NAME[var_rec_change-1]);
    for (i=0; i<range-1; i++){
      printf("%s%s",RANGE_NAME[i+1]," = ");
      fprintf(fi,"%s%s", RANGE_NAME[i+1]," =
");
      scanf("%f",&new_var_rec);
      fprintf(fi,"%f\n",new_var_rec);
      VAR_REC[var_rec_change-1][i] =
new_var_rec;}}
    if (var_rec_change >variable){
      printf("\nOLD RECOMMENDED CONDITIONS
OF %s ARE \n",MOLD_NAME[var_rec_change-
variable-1]);
      fprintf(fi,"\nOLD RECOMMENDED
CONDITIONS OF %s ARE
\n",MOLD_NAME[var_rec_change-variable-1]);
      for (i=1; i<range; i++){
        printf("%10s",RANGE_NAME[i]);
        fprintf(fi,"%10s",RANGE_NAME[i]);}
    printf("\n");
    fprintf(fi,"\n");
    for (i=0; i<range-1; i++){

  printf("%10.2f",MOLD_REC[var_rec_change-
variable-1][i]);
      fprintf(fi,"%10.2f",
MOLD_REC[var_rec_change-variable-1][i]);}
/*--INPUT THE NEW RECOMMENDED OPERATING
CONDITIONS FOR MOLD SIZE--*/
    printf("\nPLEASE INPUT YOUR NEW
RECOMMENDED CONDITIONS FOR
%s\n",MOLD_NAME[var_rec_change-variable-1]);
    fprintf(fi,"\nPLEASE INPUT YOUR NEW
RECOMMENDED CONDITIONS FOR
%s\n",MOLD_NAME[var_rec_change-variable-1]);
    for (i=0; i<range-1; i++){
      printf("%s%s",RANGE_NAME[i+1]," = ");
      fprintf(fi,"%s%s", RANGE_NAME[i+1]," =
");
      scanf("%f",&new_var_rec);
      fprintf(fi,"%f\n",new_var_rec);
      MOLD_REC[var_rec_change-variable-1][i]
= new_var_rec;}}
    printf("%s\n",BORDER);
    space((65 - (strlen(REC_TITLE3)))/2);
    printf("%s\n",REC_TITLE3);
    printf("%s\n",BORDER);
    fprintf(fi,"\n%s\n",BORDER);
    for (i = 0; i < ((65 -
strlen(REC_TITLE3))/2); i++)
      fprintf(fi," ");
    fprintf(fi,"%s\n",REC_TITLE3);
    fprintf(fi,"%s\n",BORDER);
    printf("%32s",RANGE_NAME[0]);
    fprintf(fi,"%32s",RANGE_NAME[0]);
/*--PRINTOUT THE NEW RECOMMENDED OPERATING
CONDITIONS--*/
    for (i=1; i<range; i++){
      printf("%10s",RANGE_NAME[i]);
      fprintf(fi,"%10s", RANGE_NAME[i]);}
    printf("\n");
    fprintf(fi,"\n");
    for (i=0; i<variable; i++){
      printf("%32s",VAR_NAME[i]);
      fprintf(fi,"%32s",VAR_NAME[i]);
      for (j=0; j<range-1; j++) {
        printf("%10.2f",VAR_REC[i][j]);
        fprintf(fi,"%10.2f",
VAR_REC[i][j]);}
      printf("\n");
      fprintf(fi,"\n");}
    for (i=0; i<mold; i++){
      printf("%32s",MOLD_NAME[i]);
      fprintf(fi,"%32s",MOLD_NAME[i]);
      for (j=0; j<range-1; j++) {
        printf("%10.2f", MOLD_REC[i][j]);
        fprintf(fi,"%10.2f",
MOLD_REC[i][j]);}
      printf("\n");
      fprintf(fi,"\n");}
    printf("%s\n%s\n%s",
REC_CHANGE,YN,ANSWER);
```

```
    fprintf(fi,"%s\n%s\n%s",
REC_CHANGE,YN,ANSWER);
    scanf("%d",&yn);
    fprintf(fi,"%d",yn);
    }while (yn==1);
  }
/*--USER INPUT THE OPERATING VARIABLE--*/
/*--EXPLANATION STATEMENTS FOR USER INPUT
OPERATING CONDITIONS--*/
printf("\n%s\n",BORDER);
space((65 - (strlen(VAR_TITLE)))/2);
printf("%s\n%s\n",VAR_TITLE,BORDER);
printf("\n%s\n",require);
fprintf(fi,"\n%s\n",BORDER);
for (i = 0; i < ((65 -
(strlen(VAR_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n%s\n",VAR_TITLE,BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("%s\n",BORDER);
  printf("%s\n",var_why);
  printf("%s\n",BORDER);
  printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"%s\n",var_why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
  why = getch();}
printf("\n%s\n",BORDER);
space((65 - (strlen(VAR_TITLE1)))/2);
printf("%s\n",VAR_TITLE1);
printf("%s\n",BORDER);
fprintf(fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VAR_TITLE1);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  printf("%32s%s",VAR_NAME[i]," = ");
  scanf("%f",&VAR[i]);}
for (i=0; i<mold; i++){
  printf("%32s%s",MOLD_NAME[i]," = ");
  scanf("%f",&MOLD[i]);}
printf("%s\n",BORDER);
space((65 - (strlen(VAR_TITLE2)))/2);
printf("%s\n",VAR_TITLE2);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
```

```
for ( i = 0; i < ((65 -
(strlen(VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  printf("%32s%s%10.2f\n",VAR_NAME[i]," =
",VAR[i]);
  fprintf(fi,"%32s%s%10.2f\n",VAR_NAME[i],"
= ",VAR[i]);}
for (i=0; i<mold; i++){
  printf("%32s%s%10.2f\n",MOLD_NAME[i]," =
",MOLD[i]);
  fprintf(fi,"%32s%s%10.2f\n",MOLD_NAME[i],"
= ",MOLD[i]);}
printf("\n%s\n%s\n%s",VAR_CHANGE,YN,ANSWER);
fprintf(fi,"\n%s\n%s\n%s",VAR_CHANGE,YN,ANSW
ER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fprintf(fi,"\n");
/*--COMFIRMATION FOR THE USER OPERATING
CONDITION--*/
if (yn == 1){
  do{
    printf("\n%s\n",VAR_REQUIRE);
    fprintf(fi,"\n%s\n",VAR_REQUIRE);
    for (i=0; i<variable; i++){
      printf("%d%s%s\n",i+1,"
.",VAR_NAME[i]);
      fprintf(fi,"%d%s%s\n",i+1,"
.",VAR_NAME[i]);}
    for (i=0; i<mold; i++){
      printf("%d%s%s\n",i+variable+1,"
.",MOLD_NAME[i]);
      fprintf(fi,"%d%s%s\n", i+variable+1,"
.",MOLD_NAME[i]);}
    printf("\nOPERATING VARIABLE CODE NUMBER
= ");
    fprintf(fi,"\nOPERATING VARIABLE CODE
NUMBER = ");
    scanf("%d",&var_change);
    fprintf(fi,"%d",var_change);
    if (var_change<=variable){
    printf("\nOLD OPERATING VARIABLE OF %s
IS %8.2f\n",VAR_NAME[var_change-
1],VAR[var_change-1]);
    fprintf(fi,"\nOLD OPERATING VARIABLE OF
%s IS %8.2f\n",VAR_NAME[var_change-
1],VAR[var_change-1]);
    printf("\nPLEASE INPUT YOUR NEW
OPERATING VARIABLE FOR
%s\n",VAR_NAME[var_change-1]);
```

```
    fprintf(fi,"\nPLEASE INPUT YOUR NEW
OPERATING VARIABLE FOR
%s\n",VAR_NAME[var_change-1]);
    scanf("%f",&new_var);
    fprintf(fi,"%f\n",new_var);
    VAR[var_change-1] = new_var;}
    if (var_change>variable){
    printf("\nOLD OPERATING VARIABLE OF %s
IS %8.2f\n",MOLD_NAME[var_change-variable-
1],MOLD[var_change-variable-1]);
    fprintf(fi,"\nOLD OPERATING VARIABLE OF
%s IS %8.2f\n",MOLD_NAME[var_change-
variable-1],MOLD[var_change-variable-1]);
    printf("\nPLEASE INPUT YOUR NEW
OPERATING VARIABLE FOR
%s\n",MOLD_NAME[var_change-variable-1]);
    fprintf(fi,"\nPLEASE INPUT YOUR NEW
OPERATING VARIABLE FOR
%s\n",MOLD_NAME[var_change-variable-1]);
    scanf("%f",&new_var);
    fprintf(fi,"%f\n",new_var);
    MOLD[var_change-variable-1] = new_var;}
    printf("%s\n",BORDER);
    space((65 - (strlen(VAR_TITLE2)))/2);
    printf("%s\n",VAR_TITLE2);
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    for ( i = 0; i < ((65 -
(strlen(VAR_TITLE2)))/2); i++)
      fprintf(fi," ");
    fprintf(fi,"%s\n",VAR_TITLE2);
    fprintf(fi,"%s\n",BORDER);
    for (i=0; i<variable; i++){
      printf("%32s%s%10.2f\n",VAR_NAME[i],"
= ",VAR[i]);
      fprintf(fi,"%32s%s%10.2f",
VAR_NAME[i]," = ",VAR[i]);}
    for (i=0; i<mold; i++){
      printf("%32s%s%10.2f\n",
MOLD_NAME[i]," = ",MOLD[i]);
      fprintf(fi,"%32s%s%10.2f\n",
MOLD_NAME[i]," = ",MOLD[i]);}
    printf("\n%s\n%s\n%s",
VAR_CHANGE,YN,ANSWER);
    fprintf(fi,"\n%s\n%s\n%s",
VAR_CHANGE,YN,ANSWER);
    scanf("%d",&yn);
    fprintf(fi,"%d",yn);
    fprintf(fi,"\n");
    }while (yn == 1);
  }
/*--FILE OPEN FOR CWF BETWEEN VAR AND HP--*/
fscanf(var_hp_cwf,"%s",RANGE_NAME[0]);
for (i=0; i<hp; i++)
  fscanf(var_hp_cwf,"%s",HP_NAME[i]);
for (i=0; i<variable; i++){
  fscanf(var_hp_cwf,"%s",VAR_NAME[i]);
  for (j=0; j<hp; j++) {
    fscanf(var_hp_cwf,"%f",
&VAR_HP_CWF[i][j]);}}
fclose(var_hp_cwf);
/*--PRINTOUT THE CWF BETWEEN VAR AND HP--*/
/*--EXPLANATION STATEMENTS FOR CWF BETWEEN
VAR AND HP--*/
printf("\n%s\n",BORDER);
space((65 - (strlen(CWFVARHP_TITLE1)))/2);
printf("%s\n",CWFVARHP_TITLE1);
space((65 - (strlen(CWFVARHP_TITLE2)))/2);
printf("%s\n%s\n",CWFVARHP_TITLE2,BORDER);
printf("\n%s\n",require);
fprintf(fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(CWFVARHP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",CWFVARHP_TITLE1);
for ( i = 0; i < ((65 -
(strlen(CWFVARHP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n%s\n",CWFVARHP_TITLE2,BORDER
);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"%c",why);
if (why == '?'){
  printf("\n%s\n",BORDER);
  printf("%s\n",var_hp_why);
  printf("%s\n",BORDER);
  printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"%s\n",var_hp_why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
  why = getch();}
printf("\n%s\n",BORDER);
space((65 - (strlen(CWFVARHP_TITLE3)))/2);
printf("%s\n",CWFVARHP_TITLE3);
space((65 - (strlen(CWFVARHP_TITLE4)))/2);
printf("%s\n",CWFVARHP_TITLE4);
printf("%s\n",BORDER);
fprintf(fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(CWFVARHP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",CWFVARHP_TITLE3);
for ( i = 0; i < ((65 -
(strlen(CWFVARHP_TITLE4)))/2); i++)
```

```
  fprintf(fi," ");
fprintf(fi,"%s\n",CWFVARHP_TITLE4);
fprintf(fi,"%s\n",BORDER);
printf("%32s",RANGE_NAME[0]);
fprintf(fi,"%32s",RANGE_NAME[0]);
for (i=0; i<hp; i++){
  printf("%10s",HP_NAME[i]);
  fprintf(fi,"%10s",HP_NAME[i]);}
printf("\n");
fprintf(fi,"\n");
for (i=0; i<variable; i++){
  printf("%32s",VAR_NAME[i]);
  fprintf(fi,"%32s",VAR_NAME[i]);
  for (j=0; j<hp; j++){
    printf("%10.2f",VAR_HP_CWF[i][j]);
    fprintf(fi,"%10.2f", VAR_HP_CWF[i][j]);}
  printf("\n");
  fprintf(fi,"\n");}
printf("\n%s\n%s\n%s",VAR_HP_CWF_CHANGE,YN,A
NSWER);
fprintf(fi,"\n%s\n%s\n%s",VAR_HP_CWF_CHANGE,
YN,ANSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE CORRELATIVE BETWEEN VAR AND
HP--*/
if (yn == 1){
  do  {
    printf("\n%s\n",VAR_HP_CWF_REQUIRE);
    fprintf(fi,"\n%s\n",
VAR_HP_CWF_REQUIRE);
    for (i=0; i<hp; i++){
      printf("%d%s%-15s",i+1,".
",HP_NAME[i]);
      fprintf(fi,"%d%s%-15s",i+1,".
",HP_NAME[i]);}
      printf("\n");
      fprintf(fi,"\n");
    for (i=0; i<4; i++){
      for (j=0; j<3; j++){
        printf("%d%s%-24s",i*3+j+1,"
.",VAR_NAME[i*3+j]);
        fprintf(fi,"%d%s%-24s",i*3+j+1,"
.",VAR_NAME[i*3+j]);}}

    printf("%s%-32s","13. ",VAR_NAME[12]);
    fprintf(fi,"%s%-32s","13.
",VAR_NAME[12]);
    printf("%s%-32s","14. ",VAR_NAME[13]);
    fprintf(fi,"%s%-32s","14.
",VAR_NAME[13]);
    printf("\nINHERENT PHYSICAL PROPERTY
CODE NUMBER = ");
    fprintf(fi,"\nINHERENT PHYSICAL PROPERTY
CODE NUMBER = ");
    scanf("%d",&hp_change);
    fprintf(fi,"%d",hp_change);
    printf("\nOPERATING VARIABLE CODE NUMBER
= ");
    fprintf(fi,"\nOPERATING VARIABLE CODE
NUMBER = ");
    scanf("%d",&var_change);
    fprintf(fi,"%d",var_change);
    printf("\nOLD CORRELATIVE WEIGHTING
FACTORS BETWEEN\n%s AND %s IS:
%4.2f\n",HP_NAME[hp_change-
1],VAR_NAME[var_change-
1],VAR_HP_CWF[var_change-1][hp_change-1]);
    printf("\nPLEASE INPUT YOUR NEW
CORRELATIVE WEIGHTING FACTORS BETWEEN\n%s
AND %s\n",HP_NAME[hp_change-
1],VAR_NAME[var_change-1]);
    fprintf(fi,"\nOLD CORRELATIVE WEIGHTING
FACTORS BETWEEN\n%s AND %s IS:
%4.2f\n",HP_NAME[hp_change-
1],VAR_NAME[var_change-
1],VAR_HP_CWF[var_change-1][hp_change-1]);
    fprintf(fi,"\nPLEASE INPUT YOUR NEW
CORRELATIVE WEIGHTING FACTORS BETWEEN\n%s
AND %s\n",HP_NAME[hp_change-
1],VAR_NAME[var_change-1]);
    scanf("%f",&new_var_hp_cwf);
    fprintf(fi,"%f",new_var_hp_cwf);
    VAR_HP_CWF[var_change-1][hp_change-1] =
new_var_hp_cwf;
    printf("\n");
    fprintf(fi,"\n");
    printf("%s\n",BORDER);
    space((65 -
(strlen(CWFVARHP_TITLE3)))/2);
    printf("%s\n",CWFVARHP_TITLE3);
    space((65 -
(strlen(CWFVARHP_TITLE4)))/2);
    printf("%s\n",CWFVARHP_TITLE4);
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    for ( i = 0; i < ((65 -
(strlen(CWFVARHP_TITLE3)))/2); i++)
      fprintf(fi," ");
    fprintf(fi,"%s\n",CWFVARHP_TITLE3);
    for ( i = 0; i < ((65 -
(strlen(CWFVARHP_TITLE4)))/2); i++)
      fprintf(fi," ");
    fprintf(fi,"%s\n",CWFVARHP_TITLE4);
    fprintf(fi,"%s\n",BORDER);
    printf("%32s",RANGE_NAME[0]);
    fprintf(fi,"%32s",RANGE_NAME[0]);
```

```
    for (i=0; i<hp; i++){
      printf("%10s",HP_NAME[i]);
      fprintf(fi,"%10s",HP_NAME[i]);}
    printf("\n");
    fprintf(fi,"\n");
    for (i=0; i<variable; i++){
      printf("%32s",VAR_NAME[i]);
      fprintf(fi,"%32s",VAR_NAME[i]);
      for (j=0; j<hp; j++){
        printf("%10.2f", VAR_HP_CWF[i][j]);
        fprintf(fi,"%10.2f",
VAR_HP_CWF[i][j]);}
      printf("\n");
      fprintf(fi,"\n");}
    printf("\n%s\n%s\n%s",
VAR_HP_CWF_CHANGE,YN,ANSWER);
    fprintf(fi,"\n%s\n%s\n%s",
VAR_HP_CWF_CHANGE,YN,ANSWER);
    scanf("%d",&yn);
    fprintf(fi,"%d",yn);
  }while (yn==1);
  }
/*--FILE OPEN FOR CWF BETWEEN VAR AND FP--*/
fscanf(var_fp_cwf,"%s",RANGE_NAME[0]);
for (i=0; i<fp; i++)
  fscanf(var_fp_cwf,"%s",FP_NAME[i]);
for (i=0; i<variable; i++){
  fscanf(var_fp_cwf,"%s",VAR_NAME[i]);
  for (j=0; j<fp; j++) {
    fscanf(var_fp_cwf,"%f",
&VAR_FP_CWF[i][j]);}}
fclose(var_fp_cwf);
/*--PRINTOUT THE CWF BETWEEN VAR AND FP--*/
/*--EXPLANATION STATEMENTS FOR CWF BETWEEN
VAR AND FP--*/
printf("\n%s\n",BORDER);
space((65 - (strlen(CWFVARFP_TITLE1)))/2);
printf("%s\n",CWFVARFP_TITLE1);
space((65 - (strlen(CWFVARFP_TITLE2)))/2);
printf("%s\n%s\n",CWFVARFP_TITLE2,BORDER);
printf("\n%s\n",require);
fprintf(fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(CWFVARFP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",CWFVARFP_TITLE1);
for ( i = 0; i < ((65 -
(strlen(CWFVARFP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n%s\n",CWFVARFP_TITLE2,BORDER
);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"%c",why);
if (why == '?'){
  printf("\n%s\n",BORDER);
  printf("%s\n",var_fp_why);
  printf("%s\n",BORDER);
  printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"%s\n",var_fp_why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
  why = getch();}
printf("\n%s\n",BORDER);
space((65 - (strlen(CWFVARFP_TITLE3)))/2);
printf("%s\n",CWFVARFP_TITLE3);
space((65 - (strlen(CWFVARFP_TITLE4)))/2);
printf("%s\n",CWFVARFP_TITLE4);
printf("%s\n",BORDER);
fprintf(fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(CWFVARFP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",CWFVARFP_TITLE3);
for ( i = 0; i < ((65 -
(strlen(CWFVARFP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",CWFVARFP_TITLE4);
fprintf(fi,"%s\n",BORDER);
printf("%32s",RANGE_NAME[0]);
fprintf(fi,"%32s",RANGE_NAME[0]);
for (i=0; i<fp; i++){
  printf("%15s",FP_NAME[i]);
  fprintf(fi,"%15s",FP_NAME[i]);}
printf("\n");
fprintf(fi,"\n");
for (i=0; i<variable; i++){
  printf("%32s",VAR_NAME[i]);
  fprintf(fi,"%32s",VAR_NAME[i]);
  for (j=0; j<fp; j++){
    printf("%15.2f",VAR_FP_CWF[i][j]);
    fprintf(fi,"%15.2f", VAR_FP_CWF[i][j]);}
  printf("\n");
  fprintf(fi,"\n");}
printf("\n%s\n%s\n%s",VAR_FP_CWF_CHANGE,YN,A
NSWER);
fprintf(fi,"\n%s\n%s\n%s",VAR_FP_CWF_CHANGE,
YN,ANSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE CORRELATIVE BETWEEN VAR AND
FP--*/
if (yn == 1){
  do  {
    printf("\n%s\n",VAR_FP_CWF_REQUIRE);
```

```
    fprintf(fi,"\n%s\n",
VAR_FP_CWF_REQUIRE);
    for (i=0; i<fp; i++){
      printf("%d%s%-15s",i+1,".
",FP_NAME[i]);
      fprintf(fi,"%d%s%-15s",i+1,".
",FP_NAME[i]);}
      printf("\n");
      fprintf(fi,"\n");
    for (i=0; i<4; i++){
      for (j=0; j<3; j++){
        printf("%d%s%-24s",i*3+j+1,"
.",VAR_NAME[i*3+j]);
        fprintf(fi,"%d%s%-24s",i*3+j+1,"
.",VAR_NAME[i*3+j]);}}
    printf("%s%-32s","13. ",VAR_NAME[12]);
    fprintf(fi,"%s%-32s","13.
",VAR_NAME[12]);
    printf("%s%-32s","14. ",VAR_NAME[13]);
    fprintf(fi,"%s%-32s","14.
",VAR_NAME[13]);
    printf("\nINFLUENCING PHYSICAL PROPERTY
CODE NUMBER = ");
    fprintf(fi,"\nINFLUENCING PHYSICAL
PROPERTY CODE NUMBER = ");
    scanf("%d",&fp_change);
    fprintf(fi,"%d",fp_change);
    printf("\nOPERATING VARIABLE CODE NUMBER
= ");
    fprintf(fi,"\nOPERATING VARIABLE CODE
NUMBER = ");
    scanf("%d",&var_change);
    fprintf(fi,"%d",var_change);
    printf("\nOLD CORRELATIVE WEIGHTING
FACTORS BETWEEN\n%s AND %s IS:
%4.2f\n",FP_NAME[fp_change-
1],VAR_NAME[var_change-
1],VAR_FP_CWF[var_change-1][fp_change-1]);
    printf("\nPLEASE INPUT YOUR NEW
CORRELATIVE WEIGHTING FACTORS BETWEEN\n%s
AND %s\n",FP_NAME[fp_change-
1],VAR_NAME[var_change-1]);
    fprintf(fi,"\nOLD CORRELATIVE WEIGHTING
FACTORS BETWEEN\n%s AND %s IS:
%4.2f\n",FP_NAME[fp_change-
1],VAR_NAME[var_change-
1],VAR_FP_CWF[var_change-1][fp_change-1]);
    fprintf(fi,"\nPLEASE INPUT YOUR NEW
CORRELATIVE WEIGHTING FACTORS BETWEEN\n%s
AND %s\n",FP_NAME[fp_change-
1],VAR_NAME[var_change-1]);
    scanf("%f",&new_var_fp_cwf);
    fprintf(fi,"%f",new_var_fp_cwf);
    VAR_FP_CWF[var_change-1][fp_change-1] =
new_var_fp_cwf;
    printf("\n");
    fprintf(fi,"\n");
    printf("%s\n",BORDER);
    space((65 -
(strlen(CWFVARFP_TITLE3)))/2);
    printf("%s\n",CWFVARFP_TITLE3);
    space((65 -
(strlen(CWFVARFP_TITLE4)))/2);
    printf("%s\n",CWFVARFP_TITLE4);
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    for ( i = 0; i < ((65 -
(strlen(CWFVARFP_TITLE3)))/2); i++)
      fprintf(fi," ");
    fprintf(fi,"%s\n",CWFVARFP_TITLE3);
    for ( i = 0; i < ((65 -
(strlen(CWFVARFP_TITLE4)))/2); i++)
      fprintf(fi," ");
    fprintf(fi,"%s\n",CWFVARFP_TITLE4);
    fprintf(fi,"%s\n",BORDER);
    printf("%32s",RANGE_NAME[0]);
    fprintf(fi,"%32s",RANGE_NAME[0]);
    for (i=0; i<fp; i++){
      printf("%15s",FP_NAME[i]);
      fprintf(fi,"%15s",FP_NAME[i]);}
    printf("\n");
    fprintf(fi,"\n");
    for (i=0; i<variable; i++){
      printf("%32s",VAR_NAME[i]);
      fprintf(fi,"%32s",VAR_NAME[i]);
      for (j=0; j<fp; j++){
        printf("%15.2f",VAR_FP_CWF[i][j]);

  fprintf(fi,"%15.2f",VAR_FP_CWF[i][j]);}
      printf("\n");
      fprintf(fi,"\n");}
    printf("\n%s\n%s\n%s",
VAR_FP_CWF_CHANGE,YN,ANSWER);
    fprintf(fi,"\n%s\n%s\n%s",
VAR_FP_CWF_CHANGE,YN,ANSWER);
    scanf("%d",&yn);
    fprintf(fi,"%d",yn);
  }while (yn==1);
  }
/*--DEFINE THE DEVIATION--*/
do {
/*--EXPLANATION STATEMENTS FOR DEVIATION
CONFIRMATION--*/

  printf("\n%s\n",BORDER);
  space((65 - (strlen(DEV_TITLE1)))/2);
  printf("%s\n%s\n",DEV_TITLE1,BORDER);
```

```
  printf("\n%s\n",require);
  why = getch();
  fprintf(fi,"\n%s\n",BORDER);
  for ( i = 0; i < ((65 -
(strlen(DEV_TITLE1)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n%s\n",DEV_TITLE1,BORDER);
  fprintf(fi,"\n%s\n",require);
  fprintf(fi,"%c",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
  dev_name = fopen
("b:\\program\\devname.dat","r");
  printf("%s\n",DEV_TITLE2);
  fprintf(fi,"%s\n",DEV_TITLE2);
  for (i=0; i<deviation; i++){
    fscanf(dev_name,"%s",DEV_NAME[i]);
    printf("%d%s%s\n",i+1,". ",DEV_NAME[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",DEV_NAME[i]);}
  printf("DEVIATION CODE NUMBER = ");
  fprintf(fi,"DEVIATION CODE NUMBER = ");
  scanf("%d",&dev_choice);
  fprintf(fi,"%d",dev_choice);
  printf("\nYOUR DEVIATION CODE NUMBER =
%d\n",dev_choice);
  printf("YOUR ARE EXPERIENCING
%s\n",DEV_NAME[dev_choice-1]);
  fprintf(fi,"\nYOUR DEVIATION CODE NUMBER =
%d\n",dev_choice);
  fprintf(fi,"YOUR ARE EXPERIENCING
%s\n",DEV_NAME[dev_choice-1]);
/*--DEVIATION FOR SHORT SHOT--*/
  if (dev_choice == 1){
    fclose(fi);
    mc_m90 =
ss(RANGE_NAME,VAR_NAME,MOLD_NAME,HP_NAME,FP_
NAME,USER,USER_ANSWER,VAR_REC,MOLD_REC,MOLD,
VAR,VAR_HP_CWF,VAR_FP_CWF);
    }
/*--DEVIATION FOR PIT MARKS--*/
  if (dev_choice == 2){
    fclose(fi);
    mc_m90 =
pm(RANGE_NAME,VAR_NAME,MOLD_NAME,HP_NAME,FP_
NAME,USER,USER_ANSWER,VAR_REC,MOLD_REC,MOLD,
VAR,VAR_HP_CWF,VAR_FP_CWF);
    }
/*--DEVIATION FOR SURFACE RIPPLES--*/
  if (dev_choice == 3){
    fclose(fi);
    mc_m90 =
sr(RANGE_NAME,VAR_NAME,MOLD_NAME,HP_NAME,FP_
NAME,USER,USER_ANSWER,VAR_REC,MOLD_REC,MOLD,
VAR,VAR_HP_CWF,VAR_FP_CWF);
    }
/*DEVIATION FOR SPLAY MARKS*/
  if (dev_choice == 4){
    fclose(fi);
    mc_m90 =
sp(RANGE_NAME,VAR_NAME,MOLD_NAME,HP_NAME,FP_
NAME,USER,USER_ANSWER,VAR_REC,MOLD_REC,MOLD,
VAR,VAR_HP_CWF,VAR_FP_CWF);
    }
  if (dev_choice == 5){
    fclose(fi);
    mc_m90 =
wa(RANGE_NAME,VAR_NAME,MOLD_NAME,HP_NAME,FP_
NAME,USER,USER_ANSWER,VAR_REC,MOLD_REC,MOLD,
VAR,VAR_HP_CWF,VAR_FP_CWF);}
  }while (mc_m90 == 4);
  return(mc_m90);
}

/*--REMAND FUNCTION OF SHORT SHOTS FOR
CELCON M90--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\ss_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define SS_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define SS_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define SS_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
```

```
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define SS_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
ss(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float ss_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
SHORT SHOT*/
float ss_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
SHORT SHOT*/
float new_ss_var_cwf;            /*THE
CHANGED CWF OF VAR_SS*/
float new_ss_fp_cwf;             /*THE
CHANGED CWF OF FP_SS*/
float ss_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int ss_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int ss_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_ss;
int result;
FILE *ss_var, *ss_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *ss_var_rec;                /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *ss_mold_rec;               /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *ss_var_hp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *ss_var_fp_cwf;              /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *ss_var_user;                      /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *ss_mold_user;
FILE *fi;
```

```
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  ss_var = fopen
("b:\\program\\initial.m90\\ssvarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  ss_var = fopen ("ssvarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SS_VAR_TITLE1)))/2);
printf("%s\n",SS_VAR_TITLE1);
space((65 - (strlen (SS_VAR_TITLE2)))/2);
printf("%s\n",SS_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SS_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(SS_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_SS--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SS_VAR_TITLE3)))/2);
printf("%s\n",SS_VAR_TITLE3);
space((65 - (strlen (SS_VAR_TITLE4)))/2);
printf("%s\n",SS_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SS_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
```

```
fprintf(fi,"%s\n",SS_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(SS_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(ss_var,"%s%f",var_name[i],&ss_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
ss_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", ss_var_cwf[i]);}
fclose(ss_var);
printf("\n%s\n%s\n%s",SS_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",SS_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE SS_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",SS_VAR_REQUIRE);
  fprintf(fi,"\n%s",SS_VAR_REQUIRE);
  scanf("%d",&ss_var_cwf_change);
  fprintf(fi,"%d",ss_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SHORT SHOT AND %s IS
%5.2f\n",var_name[ss_var_cwf_change-
1],ss_var_cwf[ss_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SHORT SHOT AND %s
IS",var_name[ss_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SHORT SHOT AND %s IS
%5.2f\n",var_name[ss_var_cwf_change-
1],ss_var_cwf[ss_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SHORT SHOT AND %s
IS",var_name[ss_var_cwf_change-1]);
  scanf("%f",&new_ss_var_cwf);
  fprintf(fi,"%8.2f",new_ss_var_cwf);
  ss_var_cwf[ss_var_cwf_change-1] =
new_ss_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (SS_VAR_TITLE3)))/2);
  printf("%s\n",SS_VAR_TITLE3);
  space((65 - (strlen (SS_VAR_TITLE4)))/2);
  printf("%s\n",SS_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(SS_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SS_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(SS_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SS_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",ss_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",ss_var_cwf[i]);}
  printf("\n%s\n%s\n%s",SS_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",SS_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SS_FP_TITLE1)))/2);
printf("%s\n",SS_FP_TITLE1);
space((65 - (strlen (SS_FP_TITLE2)))/2);
printf("%s\n",SS_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SS_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_FP_TITLE1);
for ( i = 0; i < ((65 - (strlen
(SS_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
```

```
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  ss_fp = fopen
("b:\\program\\initial.m90\\ssfpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  ss_fp = fopen ("ssfpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((65 - (strlen (SS_FP_TITLE3)))/2);
printf("%s\n",SS_FP_TITLE3);
space((65 - (strlen (SS_FP_TITLE4)))/2);
printf("%s\n",SS_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SS_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_FP_TITLE3);
for ( i = 0; i < ((65 - (strlen
(SS_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(ss_fp,"%s%f",fp_name[i],&ss_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",ss_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",ss_fp_cwf[i]);}
printf("\n%s\n%s\n%s",SS_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",SS_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(ss_fp);
/*--CONFIRM THE SS_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",SS_FP_REQUIRE);
  fprintf(fi,"\n%s",SS_FP_REQUIRE);
  scanf("%d",&ss_fp_cwf_change);
  fprintf(fi,"%d",ss_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SHORT SHOT AND %s IS
%5.2f\n",fp_name[ss_fp_cwf_change-
1],ss_fp_cwf[ss_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SHORT SHOT AND %s IS
",fp_name[ss_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SHORT SHOT AND %s IS
%5.2f\n",fp_name[ss_fp_cwf_change-
1],ss_fp_cwf[ss_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SHORT SHOT AND %s IS
",fp_name[ss_fp_cwf_change-1]);
  scanf("%f",&new_ss_fp_cwf);
  fprintf(fi,"%8.2f",new_ss_fp_cwf);
  ss_fp_cwf[ss_fp_cwf_change-1] =
new_ss_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (SS_FP_TITLE3)))/2);
  printf("%s\n",SS_FP_TITLE3);
  space((65 - (strlen (SS_FP_TITLE4)))/2);
  printf("%s\n",SS_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((65 - (strlen
(SS_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SS_FP_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(SS_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SS_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",ss_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",ss_fp_cwf[i]);}
  printf("\n%s\n%s\n%s",SS_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",SS_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(SS_TITLE)))/2);
printf("%s\n",SS_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
```

```
for ( i = 0; i < ((65 -
(strlen(SS_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SS_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: CHECK HOPPER FOR
MATERIAL SUPPLY--*/
printf("\n%s\n%s\n%s\n",SUG_ACTION,ss_method
_1,require);
printf("%c\n",why);
fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,ss_me
thod_1,require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("\n%s\n%s\n",REASON,ss_method_why_1
);
  printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  fprintf(fi,"\n%s\n%s\n",REASON,ss_method_w
hy_1);
  fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  why = getch();}
printf("\n%s",AFTER_SUG);
scanf("%d",&answer);
fprintf(fi,"\n%s",AFTER_SUG);
fprintf(fi,"%d",answer);
if (answer == 1){
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",RESPONED1);
  printf("\n%s",re_do);
  scanf("%d",&mc_ss);
  fprintf(fi,"\n%s",re_do);
  fprintf(fi,"%d",mc_ss);
  return(mc_ss);}
/*SUGGSTED ACTION: CLEAN MOLD SURFACE*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,ss_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,ss_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n%s\n",REASON,
ss_method_why_2);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    fprintf(fi,"\n%s\n%s\n",
REASON,ss_method_why_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_ss);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_ss);
    return(mc_ss);}}
/*--SUGGESTED ACTION: USER MAXIMUM INJECTION
SPEED--*/
if (answer == 2){
  if (var[13] !=3){
    answer = 2;
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    printf("\n%s\n",RESPONED3);
    printf("\n%s\n%s\n%s\n",
SUG_ACTION,ss_method_3,require);
    fprintf(fi,"\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n%s\n%s\n",
SUG_ACTION,ss_method_3,require);
    why = getch();
    fprintf(fi,"%c\n",why);
    ss_var_cwf[13] = 0;}
  if (why == '?'){
    printf("\n%s\n%s\n",REASON,
ss_method_why_3);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    fprintf(fi,"\n%s\n%s\n",
REASON,ss_method_why_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  fprintf(fi,"\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_ss);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_ss);
```

```
    return(mc_ss);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,ss_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,ss_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n%s\n",
REASON,ss_method_why_4);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    fprintf(fi,"\n%s\n%s\n",
REASON,ss_method_why_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_ss);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_ss);
    return(mc_ss);}}
/*--SUGGESTED ACTION: CHECK PRESSURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,ss_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,ss_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n%s\n",
REASON,ss_method_why_5);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    fprintf(fi,"\n%s\n%s\n",
REASON,ss_method_why_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_ss);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_ss);
    return(mc_ss);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,ss_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,ss_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n%s\n",REASON,
ss_method_why_6);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    fprintf(fi,"\n%s\n%s\n",REASON,
ss_method_why_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_ss);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_ss);
```

```
    return(mc_ss);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,ss_meth
od_7,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,ss_
method_7,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n%s\n",REASON,
ss_method_why_7);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    fprintf(fi,"\n%s\n%s\n",REASON,
ss_method_why_7);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_ss);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_ss);
    return(mc_ss);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/
  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,ss_var_cwf,ss_fp_cwf,ss_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n", var_name[i],"
= ",ss_var_priority[i]);
  if (ss_var_priority[0] == 0){
    break;}
  if(strcmp(var_name[0],"INJECTION_PRESSURE(
psi)") == 0)
     a = 500;
  if(strcmp(var_name[0],"BARREL_TEMPERATURE(
F)") == 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
     a = 10;
  if(strcmp(var_name[0],"REGRIND_RATE(%)")
== 0||strcmp(var_name[0],"SCREW_SPEED(rpm)")
== 0 )
     a = 5;
  if(strcmp(var_name[0],"CYCLE_TIME(sec)")
== 0)
     a = 2;
  if(strcmp(var_name[0],"INJECTION_TIME(sec)
") == 0)
     a = 1;
  if (strcmp(var_name[0],"SHOT_SIZE(in)") ==
0 || strcmp(var_name[0],"CUSHION(in)") == 0)
     a = 0.2;
  if (ss_var_cwf[0] > 0 &&
ss_var_priority[0] != 0 ){
     if (var[0] < var_rec[0][0])
       var[0] = var_rec[0][0];
     if (var[0] < var_rec[0][1])
       var[0] = var[0] + a;
     if (var[0] >= var_rec[0][1])
         var[0] = var_rec[0][1];
  }
  if (ss_var_cwf[0] < 0 &&
ss_var_priority[0] != 0 ){
     if (var[0] > var_rec[0][1]){
       var[0] = var_rec[0][1];}
     if (var[0] > var_rec[0][0]){
       var[0] = var[0] - a;}
     if (var[0] <= var_rec[0][0]){
       var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if(strcmp(var_name[0]
,"BARREL_TEMPERATURE (F)") == 0){
```

```
        printf("\n%s\n%s%8.2f\n\n%s\n",
SUG_ACTION,ss_inc_bar,var[0],require);
        fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",
SUG_ACTION,ss_inc_bar,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_inc_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_inc_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_inc_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_inc_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_inc_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,ss_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",ss_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_inc_shot,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_inc_shot_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_shot_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_dec_cus,var[0],require);
```

```
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_dec_cus_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_dec_cus_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"REGRIND_RATE(%)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_dec_reg,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_dec_reg,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_dec_reg_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_dec_reg_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_inc_screw,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",ss_inc_screw_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_screw_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",ss_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"CYCLE_TIME(sec)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
ss_inc_cyc_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,ss_inc_cyc_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",ss_inc_cyc_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",ss_inc_cyc_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
```

```
  self_learn(answer, ss_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] < mold_rec[0][1]){
    mold_user[0] = mold_rec[0][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,ss
_inc_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,ss_inc_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,ss_inc_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,ss_inc_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] >
mold_rec[1][0]){
    mold_user[1] = mold_rec[1][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,ss
_dec_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,ss_dec_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,ss_dec_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,ss_dec_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] <
mold_rec[2][1]){
    mold_user[2] = mold_rec[2][1];
    printf("%s\n",BORDER);

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,ss
_inc_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,ss_inc_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,ss_inc_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,ss_inc_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[3] <
mold_rec[3][1]){
    mold_user[3] = mold_rec[3][1];
```

```
  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,ss
_inc_vent,mold_user[3],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,ss_inc_vent,mold_user[3],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,ss_inc_vent_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,ss_inc_vent
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  ss_var_rec = fopen("varrec.dat","w");
  ss_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (ss_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (ss_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(ss_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(ss_var_rec,"%10s",range_name[i]);
  fprintf(ss_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(ss_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(ss_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(ss_var_rec,"\n");}
  fclose(ss_var_rec);
  for (i=0; i<mold; i++){

  fprintf(ss_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(ss_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(ss_mold_rec,"\n");}
  fclose(ss_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
ss_var_user = fopen ("uservar.dat","w");
ss_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(ss_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(ss_var_user);
for (i=0; i<mold; i++){
  fprintf(ss_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(ss_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  ss_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(ss_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(ss_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(ss_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(ss_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(ss_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(ss_var_hp_cwf,"\n");}
  fclose(ss_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  ss_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(ss_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)
```

```
  fprintf(ss_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(ss_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(ss_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(ss_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(ss_var_fp_cwf,"\n");}
  fclose(ss_var_fp_cwf);
/*UPDATE THE SS_VAR_CWF*/
  chdir (user);
  ss_var = fopen ("ssvarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(ss_var,"%32s%6.2f\n",var_name[i],
ss_var_cwf[i]);}
  fclose(ss_var);chdir (user);
/*UPDATE THE SS_FP_CWF*/
  chdir (user);
  ss_fp = fopen ("ssfpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (ss_fp,"%20s%6.2f\n",fp_name[i],
ss_fp_cwf[i]);}
  fclose(ss_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",ss_material,ss_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",ss_material,ss_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_ss);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_ss);
return(mc_ss);
}
/*--REMAND FUNCTION OF PITMARKS FOR CELCON
M90--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\pm_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define PM_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define PM_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define PM_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define PM_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
pm(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
```

```
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float pm_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
PIT MARKS*/
float pm_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
PIT MARKS*/
float new_pm_var_cwf;            /*THE
CHANGED CWF OF VAR_PM*/
float new_pm_fp_cwf;             /*THE
CHANGED CWF OF FP_PM*/
float pm_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int pm_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int pm_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_pm;
int result;
FILE *pm_var, *pm_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *pm_var_rec;                /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *pm_mold_rec;               /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *pm_var_hp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *pm_var_fp_cwf;              /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *pm_var_user;                    /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *pm_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  pm_var = fopen
("b:\\program\\initial.m90\\pmvarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  pm_var = fopen ("pmvarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (PM_VAR_TITLE1)))/2);
printf("%s\n",PM_VAR_TITLE1);
space((65 - (strlen (PM_VAR_TITLE2)))/2);
printf("%s\n",PM_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(PM_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(PM_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_PM--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (PM_VAR_TITLE3)))/2);
printf("%s\n",PM_VAR_TITLE3);
space((65 - (strlen (PM_VAR_TITLE4)))/2);
printf("%s\n",PM_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(PM_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(PM_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(pm_var,"%s%f",var_name[i],&pm_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
pm_var_cwf[i]);
```

```
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", pm_var_cwf[i]);}
fclose(pm_var);
printf("\n%s\n%s\n%s",PM_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",PM_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE PM_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",PM_VAR_REQUIRE);
  fprintf(fi,"\n%s",PM_VAR_REQUIRE);
  scanf("%d",&pm_var_cwf_change);
  fprintf(fi,"%d",pm_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
PIT MARKS AND %s IS
%5.2f\n",var_name[pm_var_cwf_change-
1],pm_var_cwf[pm_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
PIT MARKS AND %s
IS",var_name[pm_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN PIT MARKS AND %s IS
%5.2f\n",var_name[pm_var_cwf_change-
1],pm_var_cwf[pm_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN PIT MARKS AND %s
IS",var_name[pm_var_cwf_change-1]);
  scanf("%f",&new_pm_var_cwf);
  fprintf(fi,"%8.2f",new_pm_var_cwf);
  pm_var_cwf[pm_var_cwf_change-1] =
new_pm_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (PM_VAR_TITLE3)))/2);
  printf("%s\n",PM_VAR_TITLE3);
  space((65 - (strlen (PM_VAR_TITLE4)))/2);
  printf("%s\n",PM_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(PM_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",PM_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(PM_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",PM_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",pm_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",pm_var_cwf[i]);}
  printf("\n%s\n%s\n%s",PM_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",PM_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (PM_FP_TITLE1)))/2);
printf("%s\n",PM_FP_TITLE1);
space((65 - (strlen (PM_FP_TITLE2)))/2);
printf("%s\n",PM_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(PM_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_FP_TITLE1);
for ( i = 0; i < ((65 - (strlen
(PM_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  pm_fp = fopen
("b:\\program\\initial.m90\\pmfpcwf.dat","r"
);
if (user_answer == 1){
```

```
  chdir (user);
  pm_fp = fopen ("pmfpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((65 - (strlen (PM_FP_TITLE3)))/2);
printf("%s\n",PM_FP_TITLE3);
space((65 - (strlen (PM_FP_TITLE4)))/2);
printf("%s\n",PM_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(PM_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_FP_TITLE3);
for ( i = 0; i < ((65 - (strlen
(PM_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(pm_fp,"%s%f",fp_name[i],&pm_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",pm_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",pm_fp_cwf[i]);}
printf("\n%s\n%s\n%s",PM_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",PM_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(pm_fp);
/*--CONFIRM THE PM_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",PM_FP_REQUIRE);
  fprintf(fi,"\n%s",PM_FP_REQUIRE);
  scanf("%d",&pm_fp_cwf_change);
  fprintf(fi,"%d",pm_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
PIT MARKS AND %s IS
%5.2f\n",fp_name[pm_fp_cwf_change-
1],pm_fp_cwf[pm_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
PIT MARKS AND %s IS
",fp_name[pm_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN PIT MARKS AND %s IS
%5.2f\n",fp_name[pm_fp_cwf_change-
1],pm_fp_cwf[pm_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN PIT MARKS AND %s IS
",fp_name[pm_fp_cwf_change-1]);
  scanf("%f",&new_pm_fp_cwf);
  fprintf(fi,"%8.2f",new_pm_fp_cwf);
  pm_fp_cwf[pm_fp_cwf_change-1] =
new_pm_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (PM_FP_TITLE3)))/2);
  printf("%s\n",PM_FP_TITLE3);
  space((65 - (strlen (PM_FP_TITLE4)))/2);
  printf("%s\n",PM_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((65 - (strlen
(PM_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",PM_FP_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(PM_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",PM_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",pm_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",pm_fp_cwf[i]);}
  printf("\n%s\n%s\n%s",PM_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",PM_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(PM_TITLE)))/2);
printf("%s\n",PM_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(PM_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",PM_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: CHECK HOPPER FOR
MATERIAL SUPPLY--*/
printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_method
_1,require);
printf("%c\n",why);
```

```
fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_me
thod_1,require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("\n%s\n%s\n",REASON,pm_method_why_1
);
  printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_1);
  fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  why = getch();}
printf("\n%s",AFTER_SUG);
scanf("%d",&answer);
fprintf(fi,"\n%s",AFTER_SUG);
fprintf(fi,"%d",answer);
if (answer == 1){
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",RESPONED1);
  printf("\n%s",re_do);
  scanf("%d",&mc_pm);
  fprintf(fi,"\n%s",re_do);
  fprintf(fi,"%d",mc_pm);
  return(mc_pm);}
/*SUGGSTED ACTION: CLEAN MOLD SURFACE*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_pm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_pm);
    return(mc_pm);}}
/*--SUGGESTED ACTION: USER MAXIMUM INJECTION
SPEED--*/
if (answer == 2){
  if (var[13] !=3){
    answer = 2;
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    printf("\n%s\n",RESPONED3);

  printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_meth
od_3,require);
    fprintf(fi,"\n%s\n",RESPONED3);

  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_
method_3,require);
    why = getch();
    fprintf(fi,"%c\n",why);
    pm_var_cwf[13] = 0;}
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  fprintf(fi,"\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_pm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_pm);
    return(mc_pm);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
```

```
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_pm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_pm);
    return(mc_pm);}}
/*--SUGGESTED ACTION: CHECK PREPMURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_pm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_pm);
    return(mc_pm);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
```

```
    scanf("%d",&mc_pm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_pm);
    return(mc_pm);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,pm_meth
od_7,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,pm_
method_7,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_method_why_7
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_method_w
hy_7);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_pm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_pm);
    return(mc_pm);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,pm_var_cwf,pm_fp_cwf,pm_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",pm_var_priority[i]);
  if (pm_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if (strcmp(var_name[0],"REGRIND_RATE(%)")
== 0 ||
strcmp(var_name[0],"SCREW_SPEED(rpm)") == 0
)
    a = 5;
  if (strcmp(var_name[0],"CYCLE_TIME(sec)")
== 0)
    a = 2;
  if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0)
    a = 1;
  if (strcmp(var_name[0],"SHOT_SIZE(in)") ==
0 || strcmp(var_name[0],"CUSHION(in)") == 0)
    a = 0.2;
  if (pm_var_cwf[0] > 0 &&
pm_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
  if (pm_var_cwf[0] < 0 &&
pm_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
```

```
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_bar,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_inc_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_inc_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_inc_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_inc_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_inc_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_inc_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,pm_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",pm_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_inc_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_shot,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_inc_shot_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");
```

```
  fprintf(fi,"\n%s\n",pm_inc_shot_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
     }
     if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_dec_cus,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_dec_cus_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_dec_cus_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
     }
     if
(strcmp(var_name[0],"REGRIND_RATE(%)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_dec_reg,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_dec_reg,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_dec_reg_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_dec_reg_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
     }
     if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_screw,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",pm_inc_screw_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_inc_screw_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
     }
     if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",pm_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",pm_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
     }
     if
(strcmp(var_name[0],"CYCLE_TIME(sec)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
pm_inc_cyc_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,pm_inc_cyc_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",pm_inc_cyc_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");
```

```
  fprintf(fi,"\n%s\n",pm_inc_cyc_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
  self_learn(answer, pm_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] < mold_rec[0][1]){
    mold_user[0] = mold_rec[0][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,pm
_inc_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,pm_inc_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_inc_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_inc_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
```

```
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] >
mold_rec[1][0]){
    mold_user[1] = mold_rec[1][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,pm
_dec_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,pm_dec_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_dec_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_dec_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] <
mold_rec[2][1]){
    mold_user[2] = mold_rec[2][1];
    printf("%s\n",BORDER);

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,pm
_inc_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,pm_inc_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_inc_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
```

```
  fprintf(fi,"\n%s\n%s\n",REASON,pm_inc_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[3] <
mold_rec[3][1]){
    mold_user[3] = mold_rec[3][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,pm
_inc_vent,mold_user[3],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,pm_inc_vent,mold_user[3],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,pm_inc_vent_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,pm_inc_vent
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  pm_var_rec = fopen("varrec.dat","w");
  pm_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (pm_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (pm_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(pm_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(pm_var_rec,"%10s",range_name[i]);
  fprintf(pm_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(pm_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(pm_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(pm_var_rec,"\n");}
  fclose(pm_var_rec);
  for (i=0; i<mold; i++){

  fprintf(pm_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(pm_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(pm_mold_rec,"\n");}
  fclose(pm_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
pm_var_user = fopen ("uservar.dat","w");
pm_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(pm_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(pm_var_user);
for (i=0; i<mold; i++){
  fprintf(pm_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(pm_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  pm_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(pm_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(pm_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(pm_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){
```

```
  fprintf(pm_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(pm_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(pm_var_hp_cwf,"\n");}
  fclose(pm_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  pm_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(pm_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(pm_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(pm_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(pm_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(pm_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(pm_var_fp_cwf,"\n");}
  fclose(pm_var_fp_cwf);
/*UPDATE THE PM_VAR_CWF*/
  chdir (user);
  pm_var = fopen ("pmvarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(pm_var,"%32s%6.2f\n",var_name[i],
pm_var_cwf[i]);}
  fclose(pm_var);chdir (user);
/*UPDATE THE PM_FP_CWF*/
  chdir (user);
  pm_fp = fopen ("pmfpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (pm_fp,"%20s%6.2f\n",fp_name[i],
pm_fp_cwf[i]);}
  fclose(pm_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",pm_material,pm_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",pm_material,pm_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_pm);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_pm);
return(mc_pm);
}

/*--REMAND FUNCTION OF SURFACE RIPPLES FOR
CELCON M90--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\sr_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define SR_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
```

```
#define SR_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING
PHYSICAL PROPERTIES AND DEVIATION NEED TO BE
CHANGED"
#define SR_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define SR_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
sr(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float sr_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
SURFACE RIPPLES*/
float sr_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
SURFACE RIPPLES*/
float new_sr_var_cwf;            /*THE
CHANGED CWF OF VAR_SR*/
float new_sr_fp_cwf;             /*THE
CHANGED CWF OF FP_SR*/
float sr_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int sr_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int sr_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_sr;
int result;
FILE *sr_var, *sr_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *sr_var_rec;                /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *sr_mold_rec;               /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *sr_var_hp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *sr_var_fp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *sr_var_user;                   /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *sr_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  sr_var = fopen
("b:\\program\\initial.m90\\srvarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  sr_var = fopen ("srvarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SR_VAR_TITLE1)))/2);
printf("%s\n",SR_VAR_TITLE1);
space((65 - (strlen (SR_VAR_TITLE2)))/2);
printf("%s\n",SR_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SR_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(SR_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_SR--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SR_VAR_TITLE3)))/2);
```

```
printf("%s\n",SR_VAR_TITLE3);
space((65 - (strlen (SR_VAR_TITLE4)))/2);
printf("%s\n",SR_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SR_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(SR_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(sr_var,"%s%f",var_name[i],&sr_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
sr_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", sr_var_cwf[i]);}
fclose(sr_var);
printf("\n%s\n%s\n%s",SR_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",SR_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE SR_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",SR_VAR_REQUIRE);
  fprintf(fi,"\n%s",SR_VAR_REQUIRE);
  scanf("%d",&sr_var_cwf_change);
  fprintf(fi,"%d",sr_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SURFACE RIPPLES AND %s IS
%5.2f\n",var_name[sr_var_cwf_change-
1],sr_var_cwf[sr_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SURFACE RIPPLES AND %s
IS",var_name[sr_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SURFACE RIPPLES AND %s IS
%5.2f\n",var_name[sr_var_cwf_change-
1],sr_var_cwf[sr_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SURFACE RIPPLES AND %s
IS",var_name[sr_var_cwf_change-1]);
  scanf("%f",&new_sr_var_cwf);
  fprintf(fi,"%8.2f",new_sr_var_cwf);
  sr_var_cwf[sr_var_cwf_change-1] =
new_sr_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (SR_VAR_TITLE3)))/2);
  printf("%s\n",SR_VAR_TITLE3);
  space((65 - (strlen (SR_VAR_TITLE4)))/2);
  printf("%s\n",SR_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(SR_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SR_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(SR_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SR_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",sr_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",sr_var_cwf[i]);}
  printf("\n%s\n%s\n%s",SR_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",SR_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((70 - (strlen (SR_FP_TITLE1)))/2);
printf("%s\n",SR_FP_TITLE1);
space((70 - (strlen (SR_FP_TITLE2)))/2);
printf("%s\n",SR_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(SR_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_FP_TITLE1);
for ( i = 0; i < ((70 - (strlen
(SR_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
```

```
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  sr_fp = fopen
("b:\\program\\initial.m90\\srfpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  sr_fp = fopen ("srfpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((70 - (strlen (SR_FP_TITLE3)))/2);
printf("%s\n",SR_FP_TITLE3);
space((70 - (strlen (SR_FP_TITLE4)))/2);
printf("%s\n",SR_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(SR_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_FP_TITLE3);
for ( i = 0; i < ((70 - (strlen
(SR_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(sr_fp,"%s%f",fp_name[i],&sr_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",sr_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",sr_fp_cwf[i]);}
printf("\n%s\n%s\n%s",SR_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",SR_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(sr_fp);
/*--CONFIRM THE SR_FP CWF--*/
if( yn == 1){
```

```
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",SR_FP_REQUIRE);
  fprintf(fi,"\n%s",SR_FP_REQUIRE);
  scanf("%d",&sr_fp_cwf_change);
  fprintf(fi,"%d",sr_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SURFACE RIPPLES AND %s IS
%5.2f\n",fp_name[sr_fp_cwf_change-
1],sr_fp_cwf[sr_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SURFACE RIPPLES AND %s IS
",fp_name[sr_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SURFACE RIPPLES AND %s IS
%5.2f\n",fp_name[sr_fp_cwf_change-
1],sr_fp_cwf[sr_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SURFACE RIPPLES AND %s IS
",fp_name[sr_fp_cwf_change-1]);
  scanf("%f",&new_sr_fp_cwf);
  fprintf(fi,"%8.2f",new_sr_fp_cwf);
  sr_fp_cwf[sr_fp_cwf_change-1] =
new_sr_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (SR_FP_TITLE3)))/2);
  printf("%s\n",SR_FP_TITLE3);
  space((65 - (strlen (SR_FP_TITLE4)))/2);
  printf("%s\n",SR_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((70 - (strlen
(SR_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SR_FP_TITLE3);
  for ( i = 0; i < ((70 - (strlen
(SR_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SR_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",sr_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",sr_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",SR_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",SR_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
```

```
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(SR_TITLE)))/2);
printf("%s\n",SR_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(SR_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SR_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: CHECK HOPPER FOR
MATERIAL SUPPLY--*/
printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_method
_1,require);
printf("%c\n",why);
fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_me
thod_1,require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("\n%s\n%s\n",REASON,sr_method_why_1
);
  printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_1);
  fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  why = getch();}
printf("\n%s",AFTER_SUG);
scanf("%d",&answer);
fprintf(fi,"\n%s",AFTER_SUG);
fprintf(fi,"%d",answer);
if (answer == 1){
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",RESPONED1);
  printf("\n%s",re_do);
  scanf("%d",&mc_sr);
  fprintf(fi,"\n%s",re_do);
  fprintf(fi,"%d",mc_sr);
  return(mc_sr);}
/*SUGGSTED ACTION: CLEAN MOLD SURFACE*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sr);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sr);
    return(mc_sr);}}
/*--SUGGESTED ACTION: USER MAXIMUM INJECTION
SPEED--*/
if (answer == 2){
  if (var[13] !=3){
    answer = 2;
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    printf("\n%s\n",RESPONED3);

  printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_meth
od_3,require);
    fprintf(fi,"\n%s\n",RESPONED3);

  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_
method_3,require);
    why = getch();
    fprintf(fi,"%c\n",why);
    sr_var_cwf[13] = 0;}
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
```

```
  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  fprintf(fi,"\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sr);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sr);
    return(mc_sr);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sr);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sr);
    return(mc_sr);}}
/*--SUGGESTED ACTION: CHECK PRESRURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sr);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sr);
    return(mc_sr);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
```

```
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sr);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sr);
    return(mc_sr);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sr_meth
od_7,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sr_
method_7,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_method_why_7
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_method_w
hy_7);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
```

```
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sr);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sr);
    return(mc_sr);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,sr_var_cwf,sr_fp_cwf,sr_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",sr_var_priority[i]);
  if (sr_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if (strcmp(var_name[0],"REGRIND_RATE(%)")
== 0 ||
strcmp(var_name[0],"SCREW_SPEED(rsr)") == 0
)
    a = 5;
  if (strcmp(var_name[0],"CYCLE_TIME(sec)")
== 0)
    a = 2;
```

```
 if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0)
    a = 1;
  if (strcmp(var_name[0],"SHOT_SIZE(in)") ==
0 || strcmp(var_name[0],"CUSHION(in)") == 0)
    a = 0.2;
  if (sr_var_cwf[0] > 0 &&
sr_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
  if (sr_var_cwf[0] < 0 &&
sr_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_bar,var[0],require);
        why = getch();
                    fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sr_inc_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sr_inc_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sr_inc_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,sr_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sr_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_inj_pre_why);
```

```
         fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
         why = getch();}
     }
     if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_shot,var[0],require);
       why = getch();
                     fprintf(fi,"%c",why);
       if (why == '?'){
         printf("\n%s\n",sr_inc_shot_why);
         printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_shot_why);
         fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
         why = getch();}
     }
     if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_dec_cus,var[0],require);
       why = getch();
       fprintf(fi,"%c",why);
       if (why == '?'){
         printf("\n%s\n",sr_dec_cus_why);
         printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_dec_cus_why);
         fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
         why = getch();}
     }
     if
(strcmp(var_name[0],"REGRIND_RATE(%)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_dec_reg,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_dec_reg,var[0],require);
       why = getch();
       fprintf(fi,"%c",why);
       if (why == '?'){
         printf("\n%s\n",sr_dec_reg_why);
         printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_dec_reg_why);
         fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
         why = getch();}
     }
     if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_screw,var[0],require);
       why = getch();
       fprintf(fi,"%c",why);
       if (why == '?'){
         printf("\n%s\n",sr_inc_screw_why);
         printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_screw_why);
         fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
         why = getch();}
     }
     if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_inj_time,var[0],require);
       why = getch();
       fprintf(fi,"%c",why);
       if (why == '?'){

  printf("\n%s\n",sr_inc_inj_time_why);
         printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_inj_time);
         fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
         why = getch();}
     }
```

```
      if
(strcmp(var_name[0],"CYCLE_TIME(sec)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sr_inc_cyc_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sr_inc_cyc_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sr_inc_cyc_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sr_inc_cyc_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
  self_learn(answer, sr_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] < mold_rec[0][1]){
    mold_user[0] = mold_rec[0][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sr
_inc_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sr_inc_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_inc_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_inc_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] >
mold_rec[1][0]){
    mold_user[1] = mold_rec[1][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sr
_dec_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sr_dec_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_dec_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_dec_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] <
mold_rec[2][1]){
    mold_user[2] = mold_rec[2][1];
    printf("%s\n",BORDER);
```

```
  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sr
_inc_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sr_inc_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_inc_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_inc_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[3] <
mold_rec[3][1]){
    mold_user[3] = mold_rec[3][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sr
_inc_vent,mold_user[3],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sr_inc_vent,mold_user[3],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sr_inc_vent_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sr_inc_vent
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  sr_var_rec = fopen("varrec.dat","w");
  sr_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (sr_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (sr_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(sr_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(sr_var_rec,"%10s",range_name[i]);
  fprintf(sr_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(sr_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(sr_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(sr_var_rec,"\n");}
  fclose(sr_var_rec);
  for (i=0; i<mold; i++){

  fprintf(sr_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(sr_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(sr_mold_rec,"\n");}
  fclose(sr_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
sr_var_user = fopen ("uservar.dat","w");
sr_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(sr_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(sr_var_user);
```

```
for (i=0; i<mold; i++){
  fprintf(sr_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(sr_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  sr_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(sr_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(sr_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(sr_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(sr_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(sr_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(sr_var_hp_cwf,"\n");}
  fclose(sr_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  sr_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(sr_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(sr_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(sr_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(sr_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(sr_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(sr_var_fp_cwf,"\n");}
  fclose(sr_var_fp_cwf);
/*UPDATE THE SR_VAR_CWF*/
  chdir (user);
  sr_var = fopen ("srvarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(sr_var,"%32s%6.2f\n",var_name[i],
sr_var_cwf[i]);}
  fclose(sr_var);chdir (user);
/*UPDATE THE SR_FP_CWF*/
  chdir (user);
  sr_fp = fopen ("srfpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (sr_fp,"%20s%6.2f\n",fp_name[i],
sr_fp_cwf[i]);}
  fclose(sr_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",sr_material,sr_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",sr_material,sr_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_sr);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_sr);
return(mc_sr);
}

/*--REMAND FUNCTION OF SPLAY MARKSFOR CELCON
M90--*/
#include <stdio.h>
```

```
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\sp_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define SP_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define SP_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define SP_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define SP_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
sp(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float sp_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
SPLAY MARKS*/
float sp_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
SPLAY MARKS*/
float new_sp_var_cwf;            /*THE
CHANGED CWF OF VAR_SP*/
float new_sp_fp_cwf;             /*THE
CHANGED CWF OF FP_SP*/
float sp_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int sp_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int sp_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_sp;
int result;
FILE *sp_var, *sp_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *sp_var_rec;                /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *sp_mold_rec;               /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *sp_var_hp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *sp_var_fp_cwf;              /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *sp_var_user;                    /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *sp_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  sp_var = fopen
("b:\\program\\initial.m90\\spvarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  sp_var = fopen ("spvarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SP_VAR_TITLE1)))/2);
printf("%s\n",SP_VAR_TITLE1);
space((65 - (strlen (SP_VAR_TITLE2)))/2);
printf("%s\n",SP_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SP_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(SP_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
```

```
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_SP--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SP_VAR_TITLE3)))/2);
printf("%s\n",SP_VAR_TITLE3);
space((65 - (strlen (SP_VAR_TITLE4)))/2);
printf("%s\n",SP_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SP_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(SP_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(sp_var,"%s%f",var_name[i],&sp_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
sp_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", sp_var_cwf[i]);}
fclose(sp_var);
printf("\n%s\n%s\n%s",SP_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",SP_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE SP_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",SP_VAR_REQUIRE);
  fprintf(fi,"\n%s",SP_VAR_REQUIRE);
  scanf("%d",&sp_var_cwf_change);
  fprintf(fi,"%d",sp_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SPLAY MARKS AND %s IS
%5.2f\n",var_name[sp_var_cwf_change-
1],sp_var_cwf[sp_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SPLAY MARKS AND %s
IS",var_name[sp_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SPLAY MARKS AND %s IS
%5.2f\n",var_name[sp_var_cwf_change-
1],sp_var_cwf[sp_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SPLAY MARKS AND %s
IS",var_name[sp_var_cwf_change-1]);
  scanf("%f",&new_sp_var_cwf);
  fprintf(fi,"%8.2f",new_sp_var_cwf);
  sp_var_cwf[sp_var_cwf_change-1] =
new_sp_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (SP_VAR_TITLE3)))/2);
  printf("%s\n",SP_VAR_TITLE3);
  space((65 - (strlen (SP_VAR_TITLE4)))/2);
  printf("%s\n",SP_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(SP_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SP_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(SP_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SP_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",sp_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",sp_var_cwf[i]);}
  printf("\n%s\n%s\n%s",SP_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",SP_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SP_FP_TITLE1)))/2);
printf("%s\n",SP_FP_TITLE1);
space((65 - (strlen (SP_FP_TITLE2)))/2);
printf("%s\n",SP_FP_TITLE2);
printf("%s\n",BORDER);
```

```
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SP_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_FP_TITLE1);
for ( i = 0; i < ((65 - (strlen
(SP_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  sp_fp = fopen
("b:\\program\\initial.m90\\spfpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  sp_fp = fopen ("spfpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((65 - (strlen (SP_FP_TITLE3)))/2);
printf("%s\n",SP_FP_TITLE3);
space((65 - (strlen (SP_FP_TITLE4)))/2);
printf("%s\n",SP_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SP_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_FP_TITLE3);
for ( i = 0; i < ((65 - (strlen
(SP_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(sp_fp,"%s%f",fp_name[i],&sp_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",sp_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",sp_fp_cwf[i]);}
printf("\n%s\n%s\n%s",SP_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",SP_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(sp_fp);
/*--CONFIRM THE SP_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",SP_FP_REQUIRE);
  fprintf(fi,"\n%s",SP_FP_REQUIRE);
  scanf("%d",&sp_fp_cwf_change);
  fprintf(fi,"%d",sp_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SPLAY MARKS AND %s IS
%5.2f\n",fp_name[sp_fp_cwf_change-
1],sp_fp_cwf[sp_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SPLAY MARKS AND %s IS
",fp_name[sp_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SPLAY MARKS AND %s IS
%5.2f\n",fp_name[sp_fp_cwf_change-
1],sp_fp_cwf[sp_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SPLAY MARKS AND %s IS
",fp_name[sp_fp_cwf_change-1]);
  scanf("%f",&new_sp_fp_cwf);
  fprintf(fi,"%8.2f",new_sp_fp_cwf);
  sp_fp_cwf[sp_fp_cwf_change-1] =
new_sp_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (SP_FP_TITLE3)))/2);
  printf("%s\n",SP_FP_TITLE3);
  space((65 - (strlen (SP_FP_TITLE4)))/2);
  printf("%s\n",SP_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((65 - (strlen
(SP_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SP_FP_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(SP_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
```

```
  fprintf(fi,"%s\n",SP_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",sp_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",sp_fp_cwf[i]);}
  printf("\n%s\n%s\n%s",SP_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",SP_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(SP_TITLE)))/2);
printf("%s\n",SP_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(SP_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SP_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sp_meth
od_1,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sp_
method_1,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_method_why_1
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_method_w
hy_1);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sp);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sp);
    return(mc_sp);}
/*--SUGGESTED ACTION: CHECK PRESSURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sp_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sp_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sp);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sp);
    return(mc_sp);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
```

```
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sp_meth
od_3,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sp_
method_3,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sp);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sp);
    return(mc_sp);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sp_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sp_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sp);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sp);
    return(mc_sp);}}
/*--SUGGESTED ACTION: USE A VENTED BARREL--
*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sp_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sp_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sp);
    fprintf(fi,"\n%s",re_do);
```

```
    fprintf(fi,"%d",mc_sp);
    return(mc_sp);}}
/*--SUGGESTED ACTION: USE A SMALL NOZZLE
ORIFIC--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sp_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sp_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_method_w
hy_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sp);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sp);
    return(mc_sp);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/
  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,sp_var_cwf,sp_fp_cwf,sp_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",sp_var_priority[i]);
  if (sp_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if (strcmp(var_name[0],"SCREW_SPEED(rpm)")
== 0 )
    a = 5;
  if
(strcmp(var_name[0],"DCOMPRESSION(sec)") ==
0)
    a = 1;
  if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0)
    a = 1;
  if (strcmp(var_name[0],"CUSHION(in)") ==
0)
    a = 0.2;
  if (sp_var_cwf[0] > 0 &&
sp_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
  if (sp_var_cwf[0] < 0 &&
sp_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
```

```
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_dec_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_dec_bar,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sp_dec_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_dec_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_dec_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_dec_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sp_dec_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_dec_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_dec_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_dec_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sp_dec_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_dec_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_dec_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,sp_dec_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sp_dec_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_dec_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_dec_cus,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sp_dec_cus_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_dec_cus_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
```

```
      if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_inc_screw,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sp_inc_screw_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_inc_screw_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_dec_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_dec_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sp_dec_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_dec_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"DECOMPRESSION(sec)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sp_inc_dec_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sp_inc_dec_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sp_inc_dec_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sp_inc_dec_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
  self_learn(answer, sp_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
    if (answer != 1 && mold_user[1] <
mold_rec[1][1]){
    mold_user[1] = mold_rec[1][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sp
_inc_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sp_inc_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_inc_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_inc_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
```

```
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[3] <
mold_rec[3][1]){
    mold_user[3] = mold_rec[3][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sp
_inc_vent,mold_user[3],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sp_inc_vent,mold_user[3],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sp_inc_vent_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sp_inc_vent
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  sp_var_rec = fopen("varrec.dat","w");
  sp_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (sp_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (sp_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(sp_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(sp_var_rec,"%10s",range_name[i]);
  fprintf(sp_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(sp_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(sp_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(sp_var_rec,"\n");}
  fclose(sp_var_rec);
  for (i=0; i<mold; i++){

  fprintf(sp_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(sp_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(sp_mold_rec,"\n");}
  fclose(sp_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
sp_var_user = fopen ("uservar.dat","w");
sp_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(sp_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(sp_var_user);
for (i=0; i<mold; i++){
  fprintf(sp_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(sp_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  sp_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(sp_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(sp_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(sp_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(sp_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(sp_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
```

```
  fprintf(sp_var_hp_cwf,"\n");}
  fclose(sp_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  sp_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(sp_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(sp_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(sp_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(sp_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(sp_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(sp_var_fp_cwf,"\n");}
  fclose(sp_var_fp_cwf);
/*UPDATE THE SP_VAR_CWF*/
  chdir (user);
  sp_var = fopen ("spvarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(sp_var,"%32s%6.2f\n",var_name[i],
sp_var_cwf[i]);}
  fclose(sp_var);chdir (user);
/*UPDATE THE SP_FP_CWF*/
  chdir (user);
  sp_fp = fopen ("spfpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (sp_fp,"%20s%6.2f\n",fp_name[i],
sp_fp_cwf[i]);}
  fclose(sp_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",sp_material,sp_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",sp_material,sp_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_sp);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_sp);
return(mc_sp);
}

/*--REMAND FUNCTION OF WARPAGEG FOR CELCON
M90--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\wa_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define WA_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define WA_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define WA_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
```

```
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define WA_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
wa(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float wa_var_cwf[variable];     /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
WARPAGE*/
float wa_fp_cwf[fp];            /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
WARPAGE*/
float new_wa_var_cwf;           /*THE
CHANGED CWF OF VAR_WA*/
float new_wa_fp_cwf;            /*THE
CHANGED CWF OF FP_WA*/
float wa_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int wa_var_cwf_change;          /*THE
CHANGED INDICATED NUMBER*/
int wa_fp_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_wa;
int result;
FILE *wa_var, *wa_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *wa_var_rec;               /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *wa_mold_rec;              /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *wa_var_hp_cwf;            /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *wa_var_fp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *wa_var_user;                   /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *wa_mold_user;
FILE *fi;
```

```
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  wa_var = fopen
("b:\\program\\initial.m90\\wavarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  wa_var = fopen ("wavarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (WA_VAR_TITLE1)))/2);
printf("%s\n",WA_VAR_TITLE1);
space((65 - (strlen (WA_VAR_TITLE2)))/2);
printf("%s\n",WA_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(WA_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(WA_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_WA--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (WA_VAR_TITLE3)))/2);
printf("%s\n",WA_VAR_TITLE3);
space((65 - (strlen (WA_VAR_TITLE4)))/2);
printf("%s\n",WA_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(WA_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
```

```
fprintf(fi,"%s\n",WA_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(WA_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(wa_var,"%s%f",var_name[i],&wa_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
wa_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", wa_var_cwf[i]);}
fclose(wa_var);
printf("\n%s\n%s\n%s",WA_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",WA_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE WA_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",WA_VAR_REQUIRE);
  fprintf(fi,"\n%s",WA_VAR_REQUIRE);
  scanf("%d",&wa_var_cwf_change);
  fprintf(fi,"%d",wa_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
WARPAGE AND %s IS
%5.2f\n",var_name[wa_var_cwf_change-
1],wa_var_cwf[wa_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
WARPAGE AND %s
IS",var_name[wa_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN WARPAGE AND %s IS
%5.2f\n",var_name[wa_var_cwf_change-
1],wa_var_cwf[wa_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN WARPAGE AND %s
IS",var_name[wa_var_cwf_change-1]);
  scanf("%f",&new_wa_var_cwf);
  fprintf(fi,"%8.2f",new_wa_var_cwf);
  wa_var_cwf[wa_var_cwf_change-1] =
new_wa_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (WA_VAR_TITLE3)))/2);
  printf("%s\n",WA_VAR_TITLE3);
  space((65 - (strlen (WA_VAR_TITLE4)))/2);
  printf("%s\n",WA_VAR_TITLE4);
```

```
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(WA_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",WA_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(WA_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",WA_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",wa_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",wa_var_cwf[i]);}
  printf("\n%s\n%s\n%s",WA_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",WA_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((70 - (strlen (WA_FP_TITLE1)))/2);
printf("%s\n",WA_FP_TITLE1);
space((70 - (strlen (WA_FP_TITLE2)))/2);
printf("%s\n",WA_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(WA_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_FP_TITLE1);
for ( i = 0; i < ((70 - (strlen
(WA_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
```

```
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  wa_fp = fopen
("b:\\program\\initial.m90\\wafpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  wa_fp = fopen ("wafpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((70 - (strlen (WA_FP_TITLE3)))/2);
printf("%s\n",WA_FP_TITLE3);
space((70 - (strlen (WA_FP_TITLE4)))/2);
printf("%s\n",WA_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(WA_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_FP_TITLE3);
for ( i = 0; i < ((70 - (strlen
(WA_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(wa_fp,"%s%f",fp_name[i],&wa_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",wa_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",wa_fp_cwf[i]);}
printf("\n%s\n%s\n%s",WA_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",WA_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(wa_fp);
/*--CONFIRM THE WA_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",WA_FP_REQUIRE);
  fprintf(fi,"\n%s",WA_FP_REQUIRE);
  scanf("%d",&wa_fp_cwf_change);
  fprintf(fi,"%d",wa_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
WARPAGE AND %s IS
%5.2f\n",fp_name[wa_fp_cwf_change-
1],wa_fp_cwf[wa_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
WARPAGE AND %s IS
",fp_name[wa_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN WARPAGE AND %s IS
%5.2f\n",fp_name[wa_fp_cwf_change-
1],wa_fp_cwf[wa_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN WARPAGE AND %s IS
",fp_name[wa_fp_cwf_change-1]);
  scanf("%f",&new_wa_fp_cwf);
  fprintf(fi,"%8.2f",new_wa_fp_cwf);
  wa_fp_cwf[wa_fp_cwf_change-1] =
new_wa_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (WA_FP_TITLE3)))/2);
  printf("%s\n",WA_FP_TITLE3);
  space((65 - (strlen (WA_FP_TITLE4)))/2);
  printf("%s\n",WA_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((70 - (strlen
(WA_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",WA_FP_TITLE3);
  for ( i = 0; i < ((70 - (strlen
(WA_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",WA_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",wa_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",wa_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",WA_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",WA_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(WA_TITLE)))/2);
printf("%s\n",WA_TITLE);
printf("%s\n",BORDER);
```

```
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(WA_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",WA_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: JIG THE PART AND COOL
UNIFORMLY--*/
printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_method
_1,require);
printf("%c\n",why);
fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_me
thod_1,require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("\n%s\n%s\n",REASON,wa_method_why_1
);
  printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_1);
  fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  why = getch();}
printf("\n%s",AFTER_SUG);
scanf("%d",&answer);
fprintf(fi,"\n%s",AFTER_SUG);
fprintf(fi,"%d",answer);
if (answer == 1){
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",RESPONED1);
  printf("\n%s",re_do);
  scanf("%d",&mc_wa);
  fprintf(fi,"\n%s",re_do);
  fprintf(fi,"%d",mc_wa);
  return(mc_wa);}
/*SUGGSTED ACTION: CHECK EJECTOR PINS
MARKS*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){
  printf("\n%s\n%s\n",REASON,wa_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_wa);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_wa);
    return(mc_wa);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_meth
od_3,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_
method_3,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
```

```
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_wa);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_wa);
    return(mc_wa);}}
/*--SUGGESTED ACTION: CHECK PRESSURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_wa);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_wa);
    return(mc_wa);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_wa);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_wa);
    return(mc_wa);}}
/*--SUGGESTED ACTION: SET UNIFORM
TEMPERATURE IN BOTH HAVLES OF MOLD--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
```

```
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_wa);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_wa);
    return(mc_wa);}}
/*--SUGGESTED ACTION: RELOCAT GATE NEARER
HEAVY SECTION--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,wa_meth
od_7,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,wa_
method_7,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_method_why_7
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_method_w
hy_7);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_wa);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_wa);
    return(mc_wa);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/
  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,wa_var_cwf,wa_fp_cwf,wa_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",wa_var_priority[i]);
  if (wa_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if (strcmp(var_name[0],"SCREW_SPEED(rpm)")
== 0 )
    a = 5;
  if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0 )
    a = 1;
  if
(strcmp(var_name[0],"MOLD_CLOSED_TIME(sec)")
== 0)
    a = 2;
  if (wa_var_cwf[0] > 0 &&
wa_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
```

```
  if (wa_var_cwf[0] < 0 &&
wa_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_dec_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,wa_dec_bar,var[0],require);
        why = getch();
                        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",wa_dec_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_dec_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_dec_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,wa_dec_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",wa_dec_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_dec_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_dec_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,wa_dec_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",wa_dec_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_dec_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,wa_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",wa_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_inc_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,wa_inc_screw,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
```

```
        if (why == '?'){
          printf("\n%s\n",wa_inc_screw_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_inc_screw_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,wa_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",wa_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_CLOSED_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
wa_inc_mold_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,wa_inc_mold_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",wa_inc_mold_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",wa_inc_mold_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
```

```
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
  self_learn(answer, wa_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] > mold_rec[0][0]){
    mold_user[0] = mold_rec[0][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,wa
_dec_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,wa_dec_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_dec_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_dec_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] <
mold_rec[1][1]){
    mold_user[1] = mold_rec[1][1];
```

```
  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,wa
_inc_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,wa_inc_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_inc_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_inc_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] >
mold_rec[2][0]){
    mold_user[2] = mold_rec[2][0];
    printf("%s\n",BORDER);

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,wa
_dec_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,wa_dec_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,wa_dec_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,wa_dec_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
```

```
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  wa_var_rec = fopen("varrec.dat","w");
  wa_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (wa_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (wa_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(wa_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(wa_var_rec,"%10s",range_name[i]);
  fprintf(wa_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(wa_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(wa_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(wa_var_rec,"\n");}
  fclose(wa_var_rec);
  for (i=0; i<mold; i++){

  fprintf(wa_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(wa_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(wa_mold_rec,"\n");}
  fclose(wa_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
wa_var_user = fopen ("uservar.dat","w");
wa_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(wa_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
```

```
fclose(wa_var_user);
for (i=0; i<mold; i++){
  fprintf(wa_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(wa_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  wa_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(wa_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(wa_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(wa_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(wa_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(wa_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(wa_var_hp_cwf,"\n");}
  fclose(wa_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  wa_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(wa_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(wa_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(wa_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(wa_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(wa_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(wa_var_fp_cwf,"\n");}
  fclose(wa_var_fp_cwf);
/*UPDATE THE WA_VAR_CWF*/
  chdir (user);
  wa_var = fopen ("wavarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(wa_var,"%32s%6.2f\n",var_name[i],
wa_var_cwf[i]);}
  fclose(wa_var);chdir (user);
/*UPDATE THE WA_FP_CWF*/
  chdir (user);
  wa_fp = fopen ("wafpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (wa_fp,"%20s%6.2f\n",fp_name[i],
wa_fp_cwf[i]);}
  fclose(wa_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",wa_material,wa_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",wa_material,wa_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_wa);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_wa);
return(mc_wa);
}

/*--REMAND FUNCTION OF SINK MARKS FOR CELCON
M90--*/
#include <stdio.h>
```

```
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\sm_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define SM_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define SM_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define SM_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define SM_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
sm(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float sm_var_cwf[variable];       /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
SINK MARKS*/
float sm_fp_cwf[fp];              /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
SINK MARKS*/
float new_sm_var_cwf;             /*THE
CHANGED CWF OF VAR_SM*/
float new_sm_fp_cwf;              /*THE
CHANGED CWF OF FP_SM*/
float sm_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int sm_var_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int sm_fp_cwf_change;             /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_sm;
int result;
FILE *sm_var, *sm_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *sm_var_rec;                 /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *sm_mold_rec;                /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *sm_var_hp_cwf;              /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *sm_var_fp_cwf;               /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *sm_var_user;                       /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *sm_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  sm_var = fopen
("b:\\program\\initial.m90\\smvarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  sm_var = fopen ("smvarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SM_VAR_TITLE1)))/2);
printf("%s\n",SM_VAR_TITLE1);
space((65 - (strlen (SM_VAR_TITLE2)))/2);
printf("%s\n",SM_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SM_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(SM_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_SM--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (SM_VAR_TITLE3)))/2);
printf("%s\n",SM_VAR_TITLE3);
space((65 - (strlen (SM_VAR_TITLE4)))/2);
printf("%s\n",SM_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(SM_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(SM_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(sm_var,"%s%f",var_name[i],&sm_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
sm_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", sm_var_cwf[i]);}
fclose(sm_var);
printf("\n%s\n%s\n%s",SM_VAR_CHANGE,YN,ANSWE
R);
```

```
fprintf(fi,"\n%s\n%s\n%s",SM_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE SM_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",SM_VAR_REQUIRE);
  fprintf(fi,"\n%s",SM_VAR_REQUIRE);
  scanf("%d",&sm_var_cwf_change);
  fprintf(fi,"%d",sm_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SINK MARKS AND %s IS
%5.2f\n",var_name[sm_var_cwf_change-
1],sm_var_cwf[sm_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SINK MARKS AND %s
IS",var_name[sm_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SINK MARKS AND %s IS
%5.2f\n",var_name[sm_var_cwf_change-
1],sm_var_cwf[sm_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SINK MARKS AND %s
IS",var_name[sm_var_cwf_change-1]);
  scanf("%f",&new_sm_var_cwf);
  fprintf(fi,"%8.2f",new_sm_var_cwf);
  sm_var_cwf[sm_var_cwf_change-1] =
new_sm_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (SM_VAR_TITLE3)))/2);
  printf("%s\n",SM_VAR_TITLE3);
  space((65 - (strlen (SM_VAR_TITLE4)))/2);
  printf("%s\n",SM_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(SM_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SM_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(SM_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SM_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",sm_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",sm_var_cwf[i]);}
  printf("\n%s\n%s\n%s",SM_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",SM_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((70 - (strlen (SM_FP_TITLE1)))/2);
printf("%s\n",SM_FP_TITLE1);
space((70 - (strlen (SM_FP_TITLE2)))/2);
printf("%s\n",SM_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
```

```
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(SM_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_FP_TITLE1);
for ( i = 0; i < ((70 - (strlen
(SM_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  sm_fp = fopen
("b:\\program\\initial.m90\\smfpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  sm_fp = fopen ("smfpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((70 - (strlen (SM_FP_TITLE3)))/2);
printf("%s\n",SM_FP_TITLE3);
space((70 - (strlen (SM_FP_TITLE4)))/2);
printf("%s\n",SM_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(SM_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_FP_TITLE3);
for ( i = 0; i < ((70 - (strlen
(SM_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(sm_fp,"%s%f",fp_name[i],&sm_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",sm_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",sm_fp_cwf[i]);}
printf("\n%s\n%s\n%s",SM_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",SM_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(sm_fp);
/*--CONFIRM THE SM_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",SM_FP_REQUIRE);
```

```
  fprintf(fi,"\n%s",SM_FP_REQUIRE);
  scanf("%d",&sm_fp_cwf_change);
  fprintf(fi,"%d",sm_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
SINK MARKS AND %s IS
%5.2f\n",fp_name[sm_fp_cwf_change-
1],sm_fp_cwf[sm_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
SINK MARKS AND %s IS
",fp_name[sm_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN SINK MARKS AND %s IS
%5.2f\n",fp_name[sm_fp_cwf_change-
1],sm_fp_cwf[sm_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN SINK MARKS AND %s IS
",fp_name[sm_fp_cwf_change-1]);
  scanf("%f",&new_sm_fp_cwf);
  fprintf(fi,"%8.2f",new_sm_fp_cwf);
  sm_fp_cwf[sm_fp_cwf_change-1] =
new_sm_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (SM_FP_TITLE3)))/2);
  printf("%s\n",SM_FP_TITLE3);
  space((65 - (strlen (SM_FP_TITLE4)))/2);
  printf("%s\n",SM_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((70 - (strlen
(SM_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SM_FP_TITLE3);
  for ( i = 0; i < ((70 - (strlen
(SM_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",SM_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",sm_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",sm_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",SM_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",SM_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
}

/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(SM_TITLE)))/2);
printf("%s\n",SM_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(SM_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sm_meth
od_1,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sm_
method_1,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sm_method_why_1
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sm_method_w
hy_1);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sm);
    return(mc_sm);}
/*--SUGGESTED ACTION: CHECK PRESRURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sm_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sm_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sm_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sm_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sm);
    return(mc_sm);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
```

```
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sm_meth
od_3,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sm_
method_3,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sm_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sm_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sm);
    return(mc_sm);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--SUGGESTED ACTION: RELOCATE GATE NEARER
HEAVY SECTION--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,sm_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,sm_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,sm_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sm_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_sm);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_sm);
    return(mc_sm);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,sm_var_cwf,sm_fp_cwf,sm_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",sm_var_priority[i]);
  if (sm_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0)
    a = 1;
  if (strcmp(var_name[0],"SHOT_SIZE(in)") ==
0 || strcmp(var_name[0],"CUSHION(in)") == 0)
    a = 0.2;
  if (sm_var_cwf[0] > 0 &&
sm_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
       var[0] = var_rec[0][1];
  }
  if (sm_var_cwf[0] < 0 &&
sm_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
     if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_inc_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sm_inc_bar,var[0],require);
       why = getch();
```

```
                    fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sm_inc_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_inc_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_inc_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sm_inc_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sm_inc_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_inc_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_inc_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sm_inc_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sm_inc_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_inc_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,sm_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sm_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_inc_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_inc_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sm_inc_shot,var[0],require);
        why = getch();
                    fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sm_inc_shot_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_inc_shot_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sm_dec_cus,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",sm_dec_cus_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_dec_cus_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
        if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
sm_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,sm_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",sm_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",sm_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 1){
      printf("\n%s\n",RESPONED1);
```

```
        fprintf(fi,"\n%s\n",RESPONED1);}
      if (answer == 2){
        printf("\n%s\n",RESPONED2);
        fprintf(fi,"\n%s\n",RESPONED2);}
      if (answer == 3){
        printf("\n%s\n",RESPONED3);
        fprintf(fi,"\n%s\n",RESPONED3);}

  self_learn(answer, sm_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] < mold_rec[0][1]){
    mold_user[0] = mold_rec[0][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sm
_inc_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sm_inc_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sm_inc_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sm_inc_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] <
mold_rec[1][1]){
    mold_user[1] = mold_rec[1][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,sm
_inc_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,sm_inc_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,sm_inc_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,sm_inc_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);}}

/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  sm_var_rec = fopen("varrec.dat","w");
  sm_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (sm_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (sm_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(sm_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(sm_var_rec,"%10s",range_name[i]);
  fprintf(sm_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(sm_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++)  {

  fprintf(sm_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(sm_var_rec,"\n");}
  fclose(sm_var_rec);
  for (i=0; i<mold; i++){

  fprintf(sm_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++)  {

  fprintf(sm_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(sm_mold_rec,"\n");}
  fclose(sm_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
sm_var_user = fopen ("uservar.dat","w");
sm_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(sm_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(sm_var_user);
for (i=0; i<mold; i++){
  fprintf(sm_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(sm_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  sm_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(sm_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(sm_var_hp_cwf,"%10s",hp_name[i]);
```

```
  fprintf(sm_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(sm_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(sm_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(sm_var_hp_cwf,"\n");}
  fclose(sm_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  sm_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(sm_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(sm_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(sm_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(sm_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(sm_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(sm_var_fp_cwf,"\n");}
  fclose(sm_var_fp_cwf);
/*UPDATE THE SM_VAR_CWF*/
  chdir (user);
  sm_var = fopen ("smvarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(sm_var,"%32s%6.2f\n",var_name[i],
sm_var_cwf[i]);}
  fclose(sm_var);chdir (user);
/*UPDATE THE SM_FP_CWF*/
  chdir (user);
  sm_fp = fopen ("smfpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (sm_fp,"%20s%6.2f\n",fp_name[i],
sm_fp_cwf[i]);}
  fclose(sm_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",sm_material,sm_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",sm_material,sm_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_sm);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_sm);
return(mc_sm);
}

/*--REMAND FUNCTION OF DISTORTION FOR CELCON
M90--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\di_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define DI_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define DI_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define DI_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define DI_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
di(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float di_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
DISTORTION*/
float di_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
DISTORTION*/
float new_di_var_cwf;            /*THE
CHANGED CWF OF VAR_DI*/
float new_di_fp_cwf;             /*THE
CHANGED CWF OF FP_DI*/
```

```
float di_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int di_var_cwf_change;          /*THE
CHANGED INDICATED NUMBER*/
int di_fp_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_di;
int result;
FILE *di_var, *di_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *di_var_rec;               /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *di_mold_rec;              /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *di_var_hp_cwf;            /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *di_var_fp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *di_var_user;                   /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *di_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  di_var = fopen
("b:\\program\\initial.m90\\divarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  di_var = fopen ("divarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (DI_VAR_TITLE1)))/2);
printf("%s\n",DI_VAR_TITLE1);
space((65 - (strlen (DI_VAR_TITLE2)))/2);
printf("%s\n",DI_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(DI_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(DI_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_DI--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (DI_VAR_TITLE3)))/2);
printf("%s\n",DI_VAR_TITLE3);
space((65 - (strlen (DI_VAR_TITLE4)))/2);
printf("%s\n",DI_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(DI_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(DI_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(di_var,"%s%f",var_name[i],&di_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
di_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", di_var_cwf[i]);}
fclose(di_var);
printf("\n%s\n%s\n%s",DI_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",DI_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE DI_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",DI_VAR_REQUIRE);
  fprintf(fi,"\n%s",DI_VAR_REQUIRE);
  scanf("%d",&di_var_cwf_change);
  fprintf(fi,"%d",di_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
DISTORTION AND %s IS
%5.2f\n",var_name[di_var_cwf_change-
1],di_var_cwf[di_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
DISTORTION AND %s
IS",var_name[di_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN DISTORTION AND %s IS
%5.2f\n",var_name[di_var_cwf_change-
1],di_var_cwf[di_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN DISTORTION AND %s
IS",var_name[di_var_cwf_change-1]);
  scanf("%f",&new_di_var_cwf);
  fprintf(fi,"%8.2f",new_di_var_cwf);
  di_var_cwf[di_var_cwf_change-1] =
new_di_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (DI_VAR_TITLE3)))/2);
  printf("%s\n",DI_VAR_TITLE3);
  space((65 - (strlen (DI_VAR_TITLE4)))/2);
  printf("%s\n",DI_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(DI_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DI_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(DI_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DI_VAR_TITLE4);
```

```
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",di_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",di_var_cwf[i]);}
  printf("\n%s\n%s\n%s",DI_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",DI_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((70 - (strlen (DI_FP_TITLE1)))/2);
printf("%s\n",DI_FP_TITLE1);
space((70 - (strlen (DI_FP_TITLE2)))/2);
printf("%s\n",DI_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(DI_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_FP_TITLE1);
for ( i = 0; i < ((70 - (strlen
(DI_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  di_fp = fopen
("b:\\program\\initial.m90\\difpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  di_fp = fopen ("difpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((70 - (strlen (DI_FP_TITLE3)))/2);
printf("%s\n",DI_FP_TITLE3);
space((70 - (strlen (DI_FP_TITLE4)))/2);
printf("%s\n",DI_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(DI_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_FP_TITLE3);
for ( i = 0; i < ((70 - (strlen
(DI_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(di_fp,"%s%f",fp_name[i],&di_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",di_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",di_fp_cwf[i]);}
printf("\n%s\n%s\n%s",DI_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",DI_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(di_fp);
/*--CONFIRM THE DI_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",DI_FP_REQUIRE);
  fprintf(fi,"\n%s",DI_FP_REQUIRE);
  scanf("%d",&di_fp_cwf_change);
  fprintf(fi,"%d",di_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
DISTORTION AND %s IS
%5.2f\n",fp_name[di_fp_cwf_change-
1],di_fp_cwf[di_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
DISTORTION AND %s IS
",fp_name[di_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN DISTORTION AND %s IS
%5.2f\n",fp_name[di_fp_cwf_change-
1],di_fp_cwf[di_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN DISTORTION AND %s IS
",fp_name[di_fp_cwf_change-1]);
  scanf("%f",&new_di_fp_cwf);
  fprintf(fi,"%8.2f",new_di_fp_cwf);
  di_fp_cwf[di_fp_cwf_change-1] =
new_di_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (DI_FP_TITLE3)))/2);
  printf("%s\n",DI_FP_TITLE3);
  space((65 - (strlen (DI_FP_TITLE4)))/2);
  printf("%s\n",DI_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((70 - (strlen
(DI_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DI_FP_TITLE3);
  for ( i = 0; i < ((70 - (strlen
(DI_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DI_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",di_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",di_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",DI_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",DI_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
```

```
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(DI_TITLE)))/2);
printf("%s\n",DI_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(DI_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DI_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: JIG THE PART AND COOL
UNIFORMLY--*/
printf("\n%s\n%s\n%s\n",SUG_ACTION,di_method
_1,require);
printf("%c\n",why);
fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_me
thod_1,require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("\n%s\n%s\n",REASON,di_method_why_1
);
  printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_1);
  fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  why = getch();}
printf("\n%s",AFTER_SUG);
scanf("%d",&answer);
fprintf(fi,"\n%s",AFTER_SUG);
fprintf(fi,"%d",answer);
if (answer == 1){
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",RESPONED1);
  printf("\n%s",re_do);
  scanf("%d",&mc_di);
  fprintf(fi,"\n%s",re_do);
  fprintf(fi,"%d",mc_di);
  return(mc_di);}
/*SUGGSTED ACTION: CHECK EJECTOR PINS
MARKS*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,di_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_di);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_di);
    return(mc_di);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,di_meth
od_3,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_
method_3,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_di);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_di);
    return(mc_di);}}
/*--SUGGESTED ACTION: CHECK PRESSURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,di_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_4);
```

```
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_di);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_di);
    return(mc_di);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,di_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_di);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_di);
    return(mc_di);}}
/*--SUGGESTED ACTION: SET UNIFORM
TEMPERATURE IN BOTH HAVLES OF MOLD--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,di_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_di);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_di);
    return(mc_di);}}
/*--SUGGESTED ACTION: RELOCAT GATE NEARER
HEAVY SECTION--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,di_meth
od_7,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,di_
method_7,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_method_why_7
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_method_w
hy_7);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_di);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_di);
    return(mc_di);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
```

```
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,di_var_cwf,di_fp_cwf,di_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",di_var_priority[i]);
  if (di_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if (strcmp(var_name[0],"SCREW_SPEED(rpm)")
== 0 )
    a = 5;
  if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0 )
    a = 1;
  if
(strcmp(var_name[0],"MOLD_CLOSED_TIME(sec)")
== 0)
    a = 2;
  if (di_var_cwf[0] > 0 &&
di_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
       var[0] = var_rec[0][1];
  }
  if (di_var_cwf[0] < 0 &&
di_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_dec_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,di_dec_bar,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",di_dec_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_dec_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_dec_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,di_dec_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",di_dec_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_dec_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_dec_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,di_dec_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",di_dec_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_dec_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,di_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",di_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_inc_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){
```

```
  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,di_inc_screw,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",di_inc_screw_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_inc_screw_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,di_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",di_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_CLOSED_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
di_inc_mold_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,di_inc_mold_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",di_inc_mold_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",di_inc_mold_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 1){
      printf("\n%s\n",RESPONED1);
      fprintf(fi,"\n%s\n",RESPONED1);}
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
  self_learn(answer, di_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] > mold_rec[0][0]){
    mold_user[0] = mold_rec[0][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,di
_dec_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,di_dec_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_dec_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_dec_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] <
mold_rec[1][1]){
    mold_user[1] = mold_rec[1][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,di
_inc_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,di_inc_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_inc_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_inc_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
```

```
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] >
mold_rec[2][0]){
    mold_user[2] = mold_rec[2][0];
    printf("%s\n",BORDER);

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,di
_dec_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,di_dec_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,di_dec_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,di_dec_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  di_var_rec = fopen("varrec.dat","w");
  di_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (di_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (di_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(di_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(di_var_rec,"%10s",range_name[i]);
  fprintf(di_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(di_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(di_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(di_var_rec,"\n");}
  fclose(di_var_rec);
  for (i=0; i<mold; i++){

  fprintf(di_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(di_mold_rec,"%10.2f",mold_rec[i][j
]);}
```

```
    fprintf(di_mold_rec,"\n");}
  fclose(di_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
di_var_user = fopen ("uservar.dat","w");
di_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(di_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(di_var_user);
for (i=0; i<mold; i++){
  fprintf(di_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(di_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  di_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(di_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(di_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(di_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(di_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(di_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(di_var_hp_cwf,"\n");}
  fclose(di_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  di_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(di_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(di_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(di_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(di_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(di_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(di_var_fp_cwf,"\n");}
  fclose(di_var_fp_cwf);
/*UPDATE THE DI_VAR_CWF*/
  chdir (user);
  di_var = fopen ("divarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(di_var,"%32s%6.2f\n",var_name[i],
di_var_cwf[i]);}
  fclose(di_var);chdir (user);
/*UPDATE THE DI_FP_CWF*/
  chdir (user);
  di_fp = fopen ("difpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (di_fp,"%20s%6.2f\n",fp_name[i],
di_fp_cwf[i]);}
  fclose(di_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
```

```
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",di_material,di_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",di_material,di_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_di);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_di);
return(mc_di);
}

/*--REMAND FUNCTION OF VOIDS FOR CELCON M90-
-*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\vi_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define VI_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define VI_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define VI_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define VI_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
vi(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float vi_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
VOIDS*/
float vi_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
VOIDS*/
float new_vi_var_cwf;            /*THE
CHANGED CWF OF VAR_VI*/
float new_vi_fp_cwf;             /*THE
CHANGED CWF OF FP_VI*/
float vi_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int vi_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int vi_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_vi;
int result;
FILE *vi_var, *vi_fp;       /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *vi_var_rec;              /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *vi_mold_rec;             /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *vi_var_hp_cwf;           /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *vi_var_fp_cwf;            /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *vi_var_user;                  /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *vi_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  vi_var = fopen
("b:\\program\\initial.m90\\vivarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  vi_var = fopen ("vivarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (VI_VAR_TITLE1)))/2);
printf("%s\n",VI_VAR_TITLE1);
space((65 - (strlen (VI_VAR_TITLE2)))/2);
printf("%s\n",VI_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(VI_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
```

```
fprintf(fi,"%s\n",VI_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(VI_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_VI--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (VI_VAR_TITLE3)))/2);
printf("%s\n",VI_VAR_TITLE3);
space((65 - (strlen (VI_VAR_TITLE4)))/2);
printf("%s\n",VI_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(VI_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(VI_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(vi_var,"%s%f",var_name[i],&vi_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
vi_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", vi_var_cwf[i]);}
fclose(vi_var);
printf("\n%s\n%s\n%s",VI_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",VI_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE VI_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",VI_VAR_REQUIRE);
  fprintf(fi,"\n%s",VI_VAR_REQUIRE);
  scanf("%d",&vi_var_cwf_change);
  fprintf(fi,"%d",vi_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
VOIDS AND %s IS
%5.2f\n",var_name[vi_var_cwf_change-
1],vi_var_cwf[vi_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
VOIDS AND %s IS",var_name[vi_var_cwf_change-
1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN VOIDS AND %s IS
%5.2f\n",var_name[vi_var_cwf_change-
1],vi_var_cwf[vi_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN VOIDS AND %s
IS",var_name[vi_var_cwf_change-1]);
  scanf("%f",&new_vi_var_cwf);
  fprintf(fi,"%8.2f",new_vi_var_cwf);
  vi_var_cwf[vi_var_cwf_change-1] =
new_vi_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (VI_VAR_TITLE3)))/2);
  printf("%s\n",VI_VAR_TITLE3);
  space((65 - (strlen (VI_VAR_TITLE4)))/2);
  printf("%s\n",VI_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(VI_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",VI_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(VI_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",VI_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",vi_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",vi_var_cwf[i]);}
  printf("\n%s\n%s\n%s",VI_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",VI_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (VI_FP_TITLE1)))/2);
printf("%s\n",VI_FP_TITLE1);
space((65 - (strlen (VI_FP_TITLE2)))/2);
printf("%s\n",VI_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(VI_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_FP_TITLE1);
for ( i = 0; i < ((65 - (strlen
(VI_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
```

```
   why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  vi_fp = fopen
("b:\\program\\initial.m90\\vifpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  vi_fp = fopen ("vifpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((65 - (strlen (VI_FP_TITLE3)))/2);
printf("%s\n",VI_FP_TITLE3);
space((65 - (strlen (VI_FP_TITLE4)))/2);
printf("%s\n",VI_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(VI_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_FP_TITLE3);
for ( i = 0; i < ((65 - (strlen
(VI_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",VI_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(vi_fp,"%s%f",fp_name[i],&vi_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",vi_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",vi_fp_cwf[i]);}
printf("\n%s\n%s\n%s",VI_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",VI_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(vi_fp);
/*--CONFIRM THE VI_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",VI_FP_REQUIRE);
  fprintf(fi,"\n%s",VI_FP_REQUIRE);
  scanf("%d",&vi_fp_cwf_change);
  fprintf(fi,"%d",vi_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
VOIDS AND %s IS
%5.2f\n",fp_name[vi_fp_cwf_change-
1],vi_fp_cwf[vi_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
VOIDS AND %s IS ",fp_name[vi_fp_cwf_change-
1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN VOIDS AND %s IS
%5.2f\n",fp_name[vi_fp_cwf_change-
1],vi_fp_cwf[vi_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN VOIDS AND %s IS
",fp_name[vi_fp_cwf_change-1]);
  scanf("%f",&new_vi_fp_cwf);
  fprintf(fi,"%8.2f",new_vi_fp_cwf);
  vi_fp_cwf[vi_fp_cwf_change-1] =
new_vi_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (VI_FP_TITLE3)))/2);
  printf("%s\n",VI_FP_TITLE3);
  space((65 - (strlen (VI_FP_TITLE4)))/2);
  printf("%s\n",VI_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((65 - (strlen
(VI_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",VI_FP_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(VI_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",VI_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",vi_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",vi_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",VI_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",VI_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(SM_TITLE)))/2);
printf("%s\n",SM_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(SM_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",SM_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,vi_meth
od_1,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,vi_
method_1,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,vi_method_why_1
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,vi_method_w
hy_1);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_vi);
```

```
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_vi);
    return(mc_vi);}
/*--SUGGESTED ACTION: CHECK PRESRURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,vi_meth
od_2,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,vi_
method_2,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,vi_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,vi_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_vi);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_vi);
    return(mc_vi);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,vi_meth
od_3,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,vi_
method_3,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,vi_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,vi_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_vi);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_vi);
    return(mc_vi);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--SUGGESTED ACTION: RELOCATE GATE NEARER
HEAVY SECTION--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,vi_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,vi_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,vi_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,vi_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_vi);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_vi);
    return(mc_vi);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,vi_var_cwf,vi_fp_cwf,vi_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",vi_var_priority[i]);
  if (vi_var_priority[0] == 0){
    break;}
```

```
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0)
    a = 1;
  if (strcmp(var_name[0],"SHOT_SIZE(in)") ==
0 || strcmp(var_name[0],"CUSHION(in)") == 0)
    a = 0.2;
  if (vi_var_cwf[0] > 0 &&
vi_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
  if (vi_var_cwf[0] < 0 &&
vi_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_inc_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,vi_inc_bar,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",vi_inc_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_inc_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_inc_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,vi_inc_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",vi_inc_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_inc_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_inc_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,vi_inc_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",vi_inc_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_inc_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,vi_inc_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",vi_inc_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_inc_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_inc_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,vi_inc_shot,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",vi_inc_shot_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_inc_shot_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
```

```
      if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,vi_dec_cus,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",vi_dec_cus_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_dec_cus_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
        if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
vi_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,vi_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",vi_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",vi_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 1){
      printf("\n%s\n",RESPONED1);
      fprintf(fi,"\n%s\n",RESPONED1);}
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}

  self_learn(answer, vi_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] < mold_rec[0][1]){
    mold_user[0] = mold_rec[0][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,vi
_inc_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,vi_inc_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,vi_inc_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,vi_inc_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] <
mold_rec[1][1]){
    mold_user[1] = mold_rec[1][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,vi
_inc_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,vi_inc_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,vi_inc_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,vi_inc_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);}}

/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  vi_var_rec = fopen("varrec.dat","w");
  vi_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
```

```
    if (vi_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (vi_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(vi_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(vi_var_rec,"%10s",range_name[i]);
  fprintf(vi_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(vi_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++)  {

  fprintf(vi_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(vi_var_rec,"\n");}
  fclose(vi_var_rec);
  for (i=0; i<mold; i++){

  fprintf(vi_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++)  {

  fprintf(vi_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(vi_mold_rec,"\n");}
  fclose(vi_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
vi_var_user = fopen ("uservar.dat","w");
vi_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(vi_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(vi_var_user);
for (i=0; i<mold; i++){
  fprintf(vi_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(vi_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  vi_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(vi_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(vi_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(vi_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(vi_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++)  {

  fprintf(vi_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(vi_var_hp_cwf,"\n");}
  fclose(vi_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  vi_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(vi_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(vi_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(vi_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(vi_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++)  {
  fprintf(vi_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(vi_var_fp_cwf,"\n");}
  fclose(vi_var_fp_cwf);
/*UPDATE THE VI_VAR_CWF*/
  chdir (user);
  vi_var = fopen ("vivarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(vi_var,"%32s%6.2f\n",var_name[i],
vi_var_cwf[i]);}
  fclose(vi_var);chdir (user);
/*UPDATE THE VI_FP_CWF*/
  chdir (user);
  vi_fp = fopen ("vifpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (vi_fp,"%20s%6.2f\n",fp_name[i],
vi_fp_cwf[i]);}
  fclose(vi_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",vi_material,vi_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",vi_material,vi_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_vi);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_vi);
return(mc_vi);
}

/*--REMAND FUNCTION OF FLASHING FOR CELCON
M90--*/
#include <stdio.h>
#include <string.h>
```

```
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\fl_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define FL_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define FL_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define FL_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define FL_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
fl(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float fl_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
FLASHING*/
float fl_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
FLASHING*/
float new_fl_var_cwf;            /*THE
CHANGED CWF OF VAR_FL*/
float new_fl_fp_cwf;             /*THE
CHANGED CWF OF FP_FL*/
float fl_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int fl_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int fl_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_fl;
int result;
FILE *fl_var, *fl_fp;      /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *fl_var_rec;              /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *fl_mold_rec;             /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *fl_var_hp_cwf;           /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *fl_var_fp_cwf;            /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *fl_var_user;                  /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *fl_mold_user;
```

```
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  fl_var = fopen
("b:\\program\\initial.m90\\flvarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  fl_var = fopen ("flvarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (FL_VAR_TITLE1)))/2);
printf("%s\n",FL_VAR_TITLE1);
space((65 - (strlen (FL_VAR_TITLE2)))/2);
printf("%s\n",FL_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(FL_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(FL_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_FL--*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (FL_VAR_TITLE3)))/2);
printf("%s\n",FL_VAR_TITLE3);
space((65 - (strlen (FL_VAR_TITLE4)))/2);
printf("%s\n",FL_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(FL_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_VAR_TITLE3);
for ( i = 0; i < ((65 - (strlen
(FL_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<variable; i++){
  fscanf
(fl_var,"%s%f",var_name[i],&fl_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
fl_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", fl_var_cwf[i]);}
fclose(fl_var);
printf("\n%s\n%s\n%s",FL_VAR_CHANGE,YN,ANSWE
R);
```

```
fprintf(fi,"\n%s\n%s\n%s",FL_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE FL_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",FL_VAR_REQUIRE);
  fprintf(fi,"\n%s",FL_VAR_REQUIRE);
  scanf("%d",&fl_var_cwf_change);
  fprintf(fi,"%d",fl_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
FLASHING AND %s IS
%5.2f\n",var_name[fl_var_cwf_change-
1],fl_var_cwf[fl_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
FLASHING AND %s
IS",var_name[fl_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN FLASHING AND %s IS
%5.2f\n",var_name[fl_var_cwf_change-
1],fl_var_cwf[fl_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN FLASHING AND %s
IS",var_name[fl_var_cwf_change-1]);
  scanf("%f",&new_fl_var_cwf);
  fprintf(fi,"%8.2f",new_fl_var_cwf);
  fl_var_cwf[fl_var_cwf_change-1] =
new_fl_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (FL_VAR_TITLE3)))/2);
  printf("%s\n",FL_VAR_TITLE3);
  space((65 - (strlen (FL_VAR_TITLE4)))/2);
  printf("%s\n",FL_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((65 - (strlen
(FL_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",FL_VAR_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(FL_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",FL_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",fl_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",fl_var_cwf[i]);}
  printf("\n%s\n%s\n%s",FL_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",FL_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (FL_FP_TITLE1)))/2);
printf("%s\n",FL_FP_TITLE1);
space((65 - (strlen (FL_FP_TITLE2)))/2);
printf("%s\n",FL_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
```

```
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(FL_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_FP_TITLE1);
for ( i = 0; i < ((65 - (strlen
(FL_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  fl_fp = fopen
("b:\\program\\initial.m90\\flfpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  fl_fp = fopen ("flfpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((65 - (strlen (FL_FP_TITLE3)))/2);
printf("%s\n",FL_FP_TITLE3);
space((65 - (strlen (FL_FP_TITLE4)))/2);
printf("%s\n",FL_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(FL_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_FP_TITLE3);
for ( i = 0; i < ((65 - (strlen
(FL_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(fl_fp,"%s%f",fp_name[i],&fl_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",fl_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",fl_fp_cwf[i]);}
printf("\n%s\n%s\n%s",FL_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",FL_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
fprintf(fi,"%d",yn);
fclose(fl_fp);
/*--CONFIRM THE FL_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",FL_FP_REQUIRE);
```

```
  fprintf(fi,"\n%s",FL_FP_REQUIRE);
  scanf("%d",&fl_fp_cwf_change);
  fprintf(fi,"%d",fl_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
FLASHING AND %s IS
%5.2f\n",fp_name[fl_fp_cwf_change-
1],fl_fp_cwf[fl_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
FLASHING AND %s IS
",fp_name[fl_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN FLASHING AND %s IS
%5.2f\n",fp_name[fl_fp_cwf_change-
1],fl_fp_cwf[fl_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN FLASHING AND %s IS
",fp_name[fl_fp_cwf_change-1]);
  scanf("%f",&new_fl_fp_cwf);
  fprintf(fi,"%8.2f",new_fl_fp_cwf);
  fl_fp_cwf[fl_fp_cwf_change-1] =
new_fl_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (FL_FP_TITLE3)))/2);
  printf("%s\n",FL_FP_TITLE3);
  space((65 - (strlen (FL_FP_TITLE4)))/2);
  printf("%s\n",FL_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((65 - (strlen
(FL_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",FL_FP_TITLE3);
  for ( i = 0; i < ((65 - (strlen
(FL_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",FL_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",fl_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",fl_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",FL_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",FL_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(FL_TITLE)))/2);
printf("%s\n",FL_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(FL_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",FL_TITLE);
fprintf(fi,"%s\n",BORDER);
/*--SUGGESTED ACTION: USER CLEAN MOLD
SURFACE--*/
printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_method
_3,require);
printf("%c\n",why);
fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_me
thod_3,require);
why = getch();
fprintf(fi,"%c\n",why);
if (why == '?'){
  printf("\n%s\n%s\n",REASON,fl_method_why_3
);
  printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_3);
  fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
  why = getch();}
printf("\n%s",AFTER_SUG);
scanf("%d",&answer);
fprintf(fi,"\n%s",AFTER_SUG);
fprintf(fi,"%d",answer);
if (answer == 1){
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",RESPONED1);
  printf("\n%s",re_do);
  scanf("%d",&mc_fl);
  fprintf(fi,"\n%s",re_do);
  fprintf(fi,"%d",mc_fl);
  return(mc_fl);}
/*--SUGGESTED ACTION: USING MAXIMUM
CLAMPPING FORCE--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_meth
od_1,require);
  why = getch();
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_
method_1,require);
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_method_why_1
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_1);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_fl);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_fl);
    return(mc_fl);}}
/*--SUGGESTED ACTION: USER MINIMUM INJECTION
SPEED--*/
if (answer == 2){
  if (var[13] !=1){
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    printf("\n%s\n",RESPONED3);

  printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_meth
od_2,require);
    fprintf(fi,"\n%s\n",RESPONED3);

  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_
method_2,require);
```

```
    why = getch();
    fprintf(fi,"%c\n",why);
    fl_var_cwf[13] = 0;}
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  fprintf(fi,"\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_fl);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_fl);
    return(mc_fl);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_fl);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_fl);
    return(mc_fl);}}
/*--SUGGESTED ACTION: CHECK PRESSURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_fl);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_fl);
    return(mc_fl);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_meth
od_7,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_
method_7,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_method_why_7
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_7);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_fl);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_fl);
    return(mc_fl);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
```

```
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,fl_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,fl_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_method_w
hy_6);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_fl);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_fl);
    return(mc_fl);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,fl_var_cwf,fl_fp_cwf,fl_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",fl_var_priority[i]);
  if (fl_var_priority[0] == 0){
    break;}
  if (var[0] >=1000)
    a = 500;
  if (var[0] >=100 && var[0]<1000)
    a = 10;
  if (var[0] >= 10 && var[0] <100)
    a = 5;
  if (var[0] < 10 && var[0]>=4)
    a = 1;
  if (var[0] < 4 && var[0]>=1.5)
    a = 0.2;
  if (var[0]<1.5)
    a = 0.1;
  if (fl_var_cwf[0] > 0 &&
fl_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
  if (fl_var_cwf[0] < 0 &&
fl_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_dec_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_dec_bar,var[0],require);
        why = getch();
                      fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_dec_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_dec_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_dec_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_dec_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_dec_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_dec_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_dec_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_dec_noz,var[0],require);
```

```
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_dec_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_dec_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_dec_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,fl_dec_inj_pre,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",fl_dec_inj_pre_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_dec_inj_pre_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_dec_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_dec_shot,var[0],require);
        why = getch();
                    fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_dec_shot_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_dec_shot_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_inc_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_inc_cus,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_inc_cus_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_inc_cus_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"REGRIND_RATE(%)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_inc_reg,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_inc_reg,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_inc_reg_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_inc_reg_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_inc_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_inc_screw,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",fl_inc_screw_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_inc_screw_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"INJECTION_TIME(sec)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
fl_inc_inj_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,fl_inc_inj_time,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){

  printf("\n%s\n",fl_inc_inj_time_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",fl_inc_inj_time);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
    printf("\n%s",AFTER_SUG_VAR);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG_VAR);
    fprintf(fi,"%d",answer);
    if (answer == 1){
      printf("\n%s\n",RESPONED1);
```

```
      fprintf(fi,"\n%s\n",RESPONED1);}
    if (answer == 2){
      printf("\n%s\n",RESPONED2);
      fprintf(fi,"\n%s\n",RESPONED2);}
    if (answer == 3){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}
  self_learn(answer, fl_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] > mold_rec[0][0]){
    mold_user[0] = mold_rec[0][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,fl
_dec_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,fl_dec_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_dec_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_dec_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] <
mold_rec[1][1]){
    mold_user[1] = mold_rec[1][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,fl
_inc_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,fl_inc_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_inc_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_inc_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] >
mold_rec[2][0]){
    mold_user[2] = mold_rec[2][0];
    printf("%s\n",BORDER);

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,fl
_dec_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,fl_dec_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_dec_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_dec_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[3] <
mold_rec[3][1]){
    mold_user[3] = mold_rec[3][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,fl
_inc_vent,mold_user[3],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,fl_inc_vent,mold_user[3],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,fl_inc_vent_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,fl_inc_vent
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
      printf("\n%s\n",RESPONED3);
      fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
```

```
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  fl_var_rec = fopen("varrec.dat","w");
  fl_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (fl_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (fl_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(fl_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(fl_var_rec,"%10s",range_name[i]);
  fprintf(fl_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(fl_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(fl_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(fl_var_rec,"\n");}
  fclose(fl_var_rec);
  for (i=0; i<mold; i++){

  fprintf(fl_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(fl_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(fl_mold_rec,"\n");}
  fclose(fl_mold_rec);
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
fl_var_user = fopen ("uservar.dat","w");
fl_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(fl_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(fl_var_user);
for (i=0; i<mold; i++){
  fprintf(fl_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(fl_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  fl_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(fl_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(fl_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(fl_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(fl_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(fl_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(fl_var_hp_cwf,"\n");}
  fclose(fl_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  fl_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(fl_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(fl_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(fl_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(fl_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(fl_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(fl_var_fp_cwf,"\n");}
  fclose(fl_var_fp_cwf);
/*UPDATE THE FL_VAR_CWF*/
  chdir (user);
  fl_var = fopen ("flvarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(fl_var,"%32s%6.2f\n",var_name[i],
fl_var_cwf[i]);}
  fclose(fl_var);chdir (user);
/*UPDATE THE FL_FP_CWF*/
  chdir (user);
  fl_fp = fopen ("flfpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (fl_fp,"%20s%6.2f\n",fp_name[i],
fl_fp_cwf[i]);}
  fclose(fl_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",fl_material,fl_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",fl_material,fl_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
```

```
scanf("%d",&mc_fl);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_fl);
return(mc_fl);
}

/*--REMAND FUNCTION OF DELAMINATION FOR
CELCON M90--*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <dir.h>
#include <b:\program\head\de_sug.h>
#include <b:\program\head\respond.h>
#include <b:\program\head\printout.h>
#include <b:\program\head\title.h>
#include <b:\program\head\explan.h>
#include <b:\program\head\choice.h>
#define DE_VAR_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nOPERATING VARIABLE AND DEVIATION
NEED TO BE CHANGED"
#define DE_FP_CHANGE "DOES HERE HAVE ANY
CORRELATIVE WEIGHTING FACTOR
BETWEEN\nINFLUENCING PHYSICAL PROPERTIES AND
DEVIATION NEED TO BE CHANGED"
#define DE_VAR_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE OPERATING VARIABLE\nBY ENTERING
THE CODE NUMBER\nCODE NUMBER = "
#define DE_FP_REQUIRE "WHICH CORRELATIVE
WEIGHTING FACTOR NEED TO BE CHANGED\nPLEASE
INDICATE THE INFLUENCING PHYSICAL
PROPERTIES\nBY ENTERING THE CODE
NUMBER\nCODE NUMBER = "
de(char *range_name[range], char
*var_name[variable], char *mold_name[mold],
char *hp_name[hp], char *fp_name[fp], char
user[12], int user_answer, float
var_rec[variable][range-1], float
mold_rec[mold][range-1], float
mold_user[mold], float var[variable], float
var_hp_cwf[variable][hp], float
var_fp_cwf[variable][fp])
{
char why;
float de_var_cwf[variable];      /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN VAR AND
DELAMINATION*/
float de_fp_cwf[fp];             /*THE
CORRELATIVE WEIGHTING FACTOR BETWEEN FP AND
DELAMINATION*/
float new_de_var_cwf;            /*THE
CHANGED CWF OF VAR_DE*/
float new_de_fp_cwf;             /*THE
CHANGED CWF OF FP_DE*/
float de_var_priority[variable]; /*THE TOTAL
CWF OF OPERATING VARIABLE*/
float a;
int de_var_cwf_change;           /*THE
CHANGED INDICATED NUMBER*/
int de_fp_cwf_change;            /*THE
CHANGED INDICATED NUMBER*/
int i,j,k;
int yn;
int answer;
int mc_de;
int result;
FILE *de_var, *de_fp;      /*DATA FILE FOR
VAR_CWF AND PP_CWF*/
FILE *de_var_rec;                /*THE DATA
FILE FOR RECOMMENDED CONDITION*/
FILE *de_mold_rec;               /*THE DATA
FILE FOR MOLD RECOMMENDED CONDITION*/
FILE *de_var_hp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN VAR AND PP*/
FILE *de_var_fp_cwf;             /*THE DATA
FILE FOR CWF BETWEEN PP AND PP*/
FILE *de_var_user;                   /*THE
DATA FILE FOR USER OPERATING VARIABLE*/
FILE *de_mold_user;
FILE *fi;
fi = fopen("b:\\program\\output.doc","a");
if (user_answer == 2)
  de_var = fopen
("b:\\program\\initial.m90\\devarcwf.dat","r
");
if (user_answer == 1){
  chdir (user);
  de_var = fopen ("devarcwf.dat","r");
  chdir ("b:\\program");}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN VAR AND DEV --*/
printf ("\n%s\n",BORDER);
space((65 - (strlen (DE_VAR_TITLE1)))/2);
printf("%s\n",DE_VAR_TITLE1);
space((65 - (strlen (DE_VAR_TITLE2)))/2);
printf("%s\n",DE_VAR_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
why = getch();
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((65 - (strlen
(DE_VAR_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_VAR_TITLE1);
for ( i = 0; i < ((65 - (strlen
(DE_VAR_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_VAR_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
fprintf(fi,"%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_var_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_var_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE VAR_DE--*/
printf ("\n%s\n",BORDER);
space((70 - (strlen (DE_VAR_TITLE3)))/2);
printf("%s\n",DE_VAR_TITLE3);
space((70 - (strlen (DE_VAR_TITLE4)))/2);
printf("%s\n",DE_VAR_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(DE_VAR_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_VAR_TITLE3);
for ( i = 0; i < ((70 - (strlen
(DE_VAR_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_VAR_TITLE4);
fprintf(fi,"%s\n",BORDER);
```

```
for (i=0; i<variable; i++){
  fscanf
(de_var,"%s%f",var_name[i],&de_var_cwf[i]);
  printf ("%32s%s%5.2f\n",var_name[i]," = ",
de_var_cwf[i]);
  fprintf (fi,"%32s%s%5.2f\n",var_name[i],"
= ", de_var_cwf[i]);}
fclose(de_var);
printf("\n%s\n%s\n%s",DE_VAR_CHANGE,YN,ANSWE
R);
fprintf(fi,"\n%s\n%s\n%s",DE_VAR_CHANGE,YN,A
NSWER);
scanf("%d",&yn);
fprintf(fi,"%d\n",yn);
/*--CONFIRM THE DE_VAR CWF--*/
if (yn ==1 ){
do {
  for (i=0; i<variable; i++){
    printf("%d%s%s\n",i+1,". ",var_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",var_name[i]);}
  printf("\n%s",DE_VAR_REQUIRE);
  fprintf(fi,"\n%s",DE_VAR_REQUIRE);
  scanf("%d",&de_var_cwf_change);
  fprintf(fi,"%d",de_var_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
DELAMINATION AND %s IS
%5.2f\n",var_name[de_var_cwf_change-
1],de_var_cwf[de_var_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
DELAMINATION AND %s
IS",var_name[de_var_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN DELAMINATION AND %s IS
%5.2f\n",var_name[de_var_cwf_change-
1],de_var_cwf[de_var_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN DELAMINATION AND %s
IS",var_name[de_var_cwf_change-1]);
  scanf("%f",&new_de_var_cwf);
  fprintf(fi,"%8.2f",new_de_var_cwf);
  de_var_cwf[de_var_cwf_change-1] =
new_de_var_cwf;
  printf ("%s\n",BORDER);
  space((65 - (strlen (DE_VAR_TITLE3)))/2);
  printf("%s\n",DE_VAR_TITLE3);
  space((65 - (strlen (DE_VAR_TITLE4)))/2);
  printf("%s\n",DE_VAR_TITLE4);
  printf("%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for( i = 0; i < ((70 - (strlen
(DE_VAR_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DE_VAR_TITLE3);
  for ( i = 0; i < ((70 - (strlen
(DE_VAR_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DE_VAR_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<variable; i++){
    printf ("%32s%s%5.2f\n",var_name[i]," =
",de_var_cwf[i]);
    fprintf
(fi,"%32s%s%5.2f\n",var_name[i]," =
",de_var_cwf[i]);}
  printf("\n%s\n%s\n%s",DE_VAR_CHANGE,YN,ANS
WER);
  fprintf(fi,"\n%s\n%s\n%s",DE_VAR_CHANGE,YN
,ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
```

```
}
/*-- THE EXPLANATION STATEMENT FOR CWF
BETWEEN FP AND DEV --*/
printf ("\n%s\n",BORDER);
space((70 - (strlen (DE_FP_TITLE1)))/2);
printf("%s\n",DE_FP_TITLE1);
space((70 - (strlen (DE_FP_TITLE2)))/2);
printf("%s\n",DE_FP_TITLE2);
printf("%s\n",BORDER);
printf("\n%s\n",require);
fprintf (fi,"\n%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(DE_FP_TITLE1)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_FP_TITLE1);
for ( i = 0; i < ((70 - (strlen
(DE_FP_TITLE2)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_FP_TITLE2);
fprintf(fi,"%s\n",BORDER);
fprintf(fi,"\n%s\n",require);
why = getch();
fprintf(fi,"\n%c\n",why);
  if (why == '?'){
    printf("\n%s\n",BORDER);
    printf("%s\n",dev_fp_why);
    printf("%s\n",BORDER);
    printf("PLEASE ENTER ANY KEY TO
CONTINUE\n");
    fprintf(fi,"\n%s\n",BORDER);
    fprintf(fi,"%s\n",dev_fp_why);
    fprintf(fi,"%s\n",BORDER);
    fprintf(fi,"PLEASE ENTER ANY KEY TO
CONTINUE\n");
    why = getch();}
/*--PRINTOUT THE TITLE FOR FP_DEVIATION--*/
if (user_answer == 2)
  de_fp = fopen
("b:\\program\\initial.m90\\defpcwf.dat","r"
);
if (user_answer == 1){
  chdir (user);
  de_fp = fopen ("defpcwf.dat","r");
  chdir ("b:\\program");}
printf ("%s\n",BORDER);
space((70 - (strlen (DE_FP_TITLE3)))/2);
printf("%s\n",DE_FP_TITLE3);
space((70 - (strlen (DE_FP_TITLE4)))/2);
printf("%s\n",DE_FP_TITLE4);
printf("%s\n",BORDER);
fprintf (fi,"%s\n",BORDER);
for ( i = 0; i < ((70 - (strlen
(DE_FP_TITLE3)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_FP_TITLE3);
for ( i = 0; i < ((70 - (strlen
(DE_FP_TITLE4)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_FP_TITLE4);
fprintf(fi,"%s\n",BORDER);
for (i=0; i<fp; i++){
  fscanf(de_fp,"%s%f",fp_name[i],&de_fp_cwf[
i]);
  printf ("%20s%s%5.2f\n",fp_name[i]," =
",de_fp_cwf[i]);
  fprintf (fi,"%20s%s%5.2f\n",fp_name[i]," =
",de_fp_cwf[i]);}
printf("\n%s\n%s\n%s",DE_FP_CHANGE,YN,ANSWER
);
fprintf(fi,"\n%s\n%s\n%s",DE_FP_CHANGE,YN,AN
SWER);
scanf("%d",&yn);
```

```
fprintf(fi,"%d",yn);
fclose(de_fp);
/*--CONFIRM THE DE_FP CWF--*/
if( yn == 1){
do {
  for (i=0; i<fp; i++){
    printf("%d%s%s\n",i+1,". ",fp_name[i]);
    fprintf(fi,"%d%s%s\n",i+1,".
",fp_name[i]);}
  printf("\n%s",DE_FP_REQUIRE);
  fprintf(fi,"\n%s",DE_FP_REQUIRE);
  scanf("%d",&de_fp_cwf_change);
  fprintf(fi,"%d",de_fp_cwf_change);
  printf("\nOLD CORRELATIVE FACTOR BETWEEN
DELAMINATION AND %s IS
%5.2f\n",fp_name[de_fp_cwf_change-
1],de_fp_cwf[de_fp_cwf_change-1]);
  printf("\nNEW CORRELATIVE FACTOR BETWEEN
DELAMINATION AND %s IS
",fp_name[de_fp_cwf_change-1]);
  fprintf(fi,"\nOLD CORRELATIVE FACTOR
BETWEEN DELAMINATION AND %s IS
%5.2f\n",fp_name[de_fp_cwf_change-
1],de_fp_cwf[de_fp_cwf_change-1]);
  fprintf(fi,"\nNEW CORRELATIVE FACTOR
BETWEEN DELAMINATION AND %s IS
",fp_name[de_fp_cwf_change-1]);
  scanf("%f",&new_de_fp_cwf);
  fprintf(fi,"%8.2f",new_de_fp_cwf);
  de_fp_cwf[de_fp_cwf_change-1] =
new_de_fp_cwf;
  printf ("\n%s\n",BORDER);
  space((65 - (strlen (DE_FP_TITLE3)))/2);
  printf("%s\n",DE_FP_TITLE3);
  space((65 - (strlen (DE_FP_TITLE4)))/2);
  printf("%s\n",DE_FP_TITLE4);
  printf("\n%s\n",BORDER);
  fprintf (fi,"%s\n",BORDER);
  for ( i = 0; i < ((70 - (strlen
(DE_FP_TITLE3)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DE_FP_TITLE3);
  for ( i = 0; i < ((70 - (strlen
(DE_FP_TITLE4)))/2); i++)
    fprintf(fi," ");
  fprintf(fi,"%s\n",DE_FP_TITLE4);
  fprintf(fi,"%s\n",BORDER);
  for (i=0; i<fp; i++){
    printf ("%20s%s%5.2f\n",fp_name[i]," =
",de_fp_cwf[i]);
    fprintf (fi,"%20s%s%5.2f\n",fp_name[i],"
= ",de_fp_cwf[i]);}

  printf("\n%s\n%s\n%s",DE_FP_CHANGE,YN,ANSW
ER);
  fprintf(fi,"\n%s\n%s\n%s",DE_FP_CHANGE,YN,
ANSWER);
  scanf("%d",&yn);
  fprintf(fi,"%d",yn);
  }while (yn == 1);
}
/*--SUGGESTED ACTION FROM METHOD ACTION--*/
printf("%s\n",BORDER);
space ((65 - (strlen(DE_TITLE)))/2);
printf("%s\n",DE_TITLE);
printf("%s\n",BORDER);
fprintf(fi,"%s\n",BORDER);
for ( i = 0; i < ((65 -
(strlen(DE_TITLE)))/2); i++)
  fprintf(fi," ");
fprintf(fi,"%s\n",DE_TITLE);
fprintf(fi,"%s\n",BORDER);
/*SUGGSTED ACTION: CLEAN MOLD SURFACE*/
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  fprintf(fi,"%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,de_meth
od_1,require);
  printf("%c\n",why);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,de_
method_1,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_method_why_1
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_method_w
hy_1);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_de);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_de);
    return(mc_de);}
/*--SUGGESTED ACTION: USER MAXIMUM INJECTION
SPEED--*/
if (answer == 2){
  if (var[13] !=3){
    answer = 2;
    printf("%s\n",BORDER);
    fprintf(fi,"%s\n",BORDER);
    printf("\n%s\n",RESPONED3);

  printf("\n%s\n%s\n%s\n",SUG_ACTION,de_meth
od_2,require);
    fprintf(fi,"\n%s\n",RESPONED3);

  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,de_
method_2,require);
    why = getch();
    fprintf(fi,"%c\n",why);
    de_var_cwf[13] = 0;}
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_method_why_2
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_method_w
hy_2);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  fprintf(fi,"\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
```

```
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_de);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_de);
    return(mc_de);}}
/*--SUGGESTED ACTION: CHECK TEMPERATURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,de_meth
od_3,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,de_
method_3,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_method_why_3
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_method_w
hy_3);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_de);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_de);
    return(mc_de);}}
/*--SUGGESTED ACTION: CHECK PRESSURE
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,de_meth
od_4,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,de_
method_4,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_method_why_4
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_method_w
hy_4);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_de);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_de);
    return(mc_de);}}
/*--SUGGESTED ACTION: CHECK SCREW SPEED
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,de_meth
od_5,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,de_
method_5,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_method_why_5
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_method_w
hy_5);
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_de);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_de);
    return(mc_de);}}
/*--SUGGESTED ACTION: CHECK SCREW POSITION
INDICATOR--*/
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);
  printf("%s\n",BORDER);
  printf("\n%s\n%s\n%s\n",SUG_ACTION,de_meth
od_6,require);
  printf("%c\n",why);
  fprintf(fi,"%s\n",BORDER);
  fprintf(fi,"\n%s\n%s\n%s\n",SUG_ACTION,de_
method_6,require);
  why = getch();
  fprintf(fi,"%c\n",why);
  if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_method_why_6
);
    printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_method_w
hy_6);
```

```
    fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
    why = getch();}
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);
  if (answer == 1){
    printf("\n%s\n",RESPONED1);
    fprintf(fi,"\n%s\n",RESPONED1);
    printf("\n%s",re_do);
    scanf("%d",&mc_de);
    fprintf(fi,"\n%s",re_do);
    fprintf(fi,"%d",mc_de);
    return(mc_de);}
  if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
/*--END SUGGESTED ACTION FROM METHOD
CORRECTIVE ACTIONS--*/
/*--BEGIN THE SUGGESTED ACTIONS FROM
OPERATING VARIABLES--*/
if (answer != 1){
  printf("\n%s\nBEGIN THE OPERATING VARIABLE
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE OPERATING
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
do{
/*--CALCULATION OF THE PRIORITY WEIGHTING
FACTOR FOR OPERATING VARIABLE--*/

  decision(var_name,var_rec,var,var_hp_cwf,v
ar_fp_cwf,de_var_cwf,de_fp_cwf,de_var_priori
ty);
  for (i=0; i<variable; i++)
    fprintf(fi,"%s%s%10.4f\n",var_name[i],"
= ",de_var_priority[i]);
  if (de_var_priority[0] == 0){
    break;}
  if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0)
    a = 500;
  if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0 ||
strcmp(var_name[0],"MOLD_TEMPERATURE(F)") ==
0 ||
strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0)
    a = 10;
  if (strcmp(var_name[0],"REGRIND_RATE(%)")
== 0 ||
strcmp(var_name[0],"SCREW_SPEED(rpm)") == 0
)
    a = 5;
  if (strcmp(var_name[0],"CYCLE_TIME(sec)")
== 0)
    a = 2;
  if (strcmp(var_name[0],"SHOT_SIZE(in)") ==
0 || strcmp(var_name[0],"CUSHION(in)") == 0)
    a = 0.2;
  if (de_var_cwf[0] > 0 &&
de_var_priority[0] != 0 ){
    if (var[0] < var_rec[0][0])
      var[0] = var_rec[0][0];
    if (var[0] < var_rec[0][1])
      var[0] = var[0] + a;
    if (var[0] >= var_rec[0][1])
        var[0] = var_rec[0][1];
  }
  if (de_var_cwf[0] < 0 &&
de_var_priority[0] != 0 ){
    if (var[0] > var_rec[0][1]){
      var[0] = var_rec[0][1];}
    if (var[0] > var_rec[0][0]){
      var[0] = var[0] - a;}
    if (var[0] <= var_rec[0][0]){
      var[0] = var_rec[0][0];}
  }
  printf("\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",BORDER);
      if
(strcmp(var_name[0],"BARREL_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_inc_bar,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_inc_bar,var[0],require);
        why = getch();
                    fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",de_inc_bar_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_inc_bar_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"MOLD_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_inc_mold,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_inc_mold,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",de_inc_mold_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_inc_mold_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
          why = getch();}
      }
      if
(strcmp(var_name[0],"NOZZLE_TEMPERATURE(F)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_inc_noz,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_inc_noz,var[0],require);
        why = getch();
        fprintf(fi,"%c",why);
        if (why == '?'){
          printf("\n%s\n",de_inc_noz_why);
          printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_inc_noz_why);
          fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
```

```
            why = getch();}
        }
        if
(strcmp(var_name[0],"INJECTION_PRESSURE(psi)
") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_inc_inj_pre,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n%s\n",SUG_ACTIO
N,de_inc_inj_pre,var[0],require);
          why = getch();
          fprintf(fi,"%c",why);
          if (why == '?'){

  printf("\n%s\n",de_inc_inj_pre_why);
            printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_inc_inj_pre_why);
            fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
            why = getch();}
        }
        if
(strcmp(var_name[0],"SHOT_SIZE(in)") == 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_inc_shot,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_inc_shot,var[0],require);
          why = getch();
                        fprintf(fi,"%c",why);
          if (why == '?'){
            printf("\n%s\n",de_inc_shot_why);
            printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_inc_shot_why);
            fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
            why = getch();}
        }
        if (strcmp(var_name[0],"CUSHION(in)")
== 0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_dec_cus,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_dec_cus,var[0],require);
          why = getch();
          fprintf(fi,"%c",why);
          if (why == '?'){
            printf("\n%s\n",de_dec_cus_why);
            printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_dec_cus_why);
            fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
            why = getch();}
        }
        if
(strcmp(var_name[0],"REGRIND_RATE(%)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_dec_reg,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_dec_reg,var[0],require);
          why = getch();
          fprintf(fi,"%c",why);
          if (why == '?'){
            printf("\n%s\n",de_dec_reg_why);
            printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_dec_reg_why);
            fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
            why = getch();}
        }
        if
(strcmp(var_name[0],"SCREW_SPEED(rpm)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_dec_screw,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_dec_screw,var[0],require);
          why = getch();
          fprintf(fi,"%c",why);
          if (why == '?'){
            printf("\n%s\n",de_dec_screw_why);
            printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_dec_screw_why);
            fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
            why = getch();}
        }
        if
(strcmp(var_name[0],"CYCLE_TIME(sec)") ==
0){

  printf("\n%s\n%s%8.2f\n\n%s\n",SUG_ACTION,
de_inc_cyc_time,var[0],require);

  fprintf(fi,"\n%s\n%s%8.2f\n\n%s\n",SUG_ACT
ION,de_inc_cyc_time,var[0],require);
          why = getch();
          fprintf(fi,"%c",why);
          if (why == '?'){

  printf("\n%s\n",de_inc_cyc_time_why);
            printf("\nPLAEASE ENTER ANY KEY TO
CONTINUE\n");

  fprintf(fi,"\n%s\n",de_inc_cyc_time);
            fprintf(fi,"\nPLAEASE ENTER ANY
KEY TO CONTINUE\n");
            why = getch();}
        }
      printf("\n%s",AFTER_SUG_VAR);
      scanf("%d",&answer);
      fprintf(fi,"\n%s",AFTER_SUG_VAR);
      fprintf(fi,"%d",answer);
      if (answer == 1){
        printf("\n%s\n",RESPONED1);
        fprintf(fi,"\n%s\n",RESPONED1);}
      if (answer == 2){
        printf("\n%s\n",RESPONED2);
        fprintf(fi,"\n%s\n",RESPONED2);}
      if (answer == 3){
        printf("\n%s\n",RESPONED3);
        fprintf(fi,"\n%s\n",RESPONED3);}
```

```
  self_learn(answer, de_var_cwf,
var_hp_cwf,var_fp_cwf);
  }while(answer != 1);
}
/*BEGIN MOLD CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MOLD CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MOLD
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  if (mold_user[0] < mold_rec[0][1]){
    mold_user[0] = mold_rec[0][1];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,de
_inc_gate,mold_user[0],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,de_inc_gate,mold_user[0],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_inc_gate_why
);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_inc_gate
_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[1] >
mold_rec[1][0]){
    mold_user[1] = mold_rec[1][0];

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,de
_dec_cooling,mold_user[1],require);
    printf("%c\n",why);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,de_dec_cooling,mold_user[1],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_dec_cooling_
why);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_dec_cool
ing_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}
  if (answer != 1 && mold_user[2] <
mold_rec[2][1]){
    mold_user[2] = mold_rec[2][1];
    printf("%s\n",BORDER);

  printf("\n%s\n%s%5.2f\n%s\n",SUG_ACTION,de
_inc_runner,mold_user[2],require);
    printf("%c\n",why);
    fprintf(fi,"%s\n",BORDER);

  fprintf(fi,"\n%s\n%s%5.2f\n%s\n",SUG_ACTIO
N,de_inc_runner,mold_user[2],require);
    why = getch();
    fprintf(fi,"%c\n",why);
    if (why == '?'){

  printf("\n%s\n%s\n",REASON,de_inc_runner_w
hy);
      printf("\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");

  fprintf(fi,"\n%s\n%s\n",REASON,de_inc_runn
er_why);
      fprintf(fi,"\nPLAEASE ENTER ANY KEY TO
CONTIUNE\n");
      why = getch();}
    printf("\n%s",AFTER_SUG);
    scanf("%d",&answer);
    fprintf(fi,"\n%s",AFTER_SUG);
    fprintf(fi,"%d",answer);
    if (answer == 2){
    printf("\n%s\n",RESPONED3);
    fprintf(fi,"\n%s\n",RESPONED3);}}}
/*END MOLD CORRECTION ACTIONS*/
/*UPDATE THE CHANGED VARIABLES*/
if (answer == 1 || answer == 2){
  printf("\n%s\nCURRENTLY, THE SYSTEM UPDATE
YOUR DATA. PLEASE
WAIT!\n%s\n",BORDER,BORDER);
/*UPDATE THE RECOMMENDED OPERATING
CONDITIONS*/
  chdir(user);
  de_var_rec = fopen("varrec.dat","w");
  de_mold_rec = fopen("moldrec.dat","w");
  chdir("b:\\program");
  for (i=0; i<variable; i++){
    if (de_var_cwf[i] > 0)
      var_rec[i][0] = var[i];
    if (de_var_cwf[i] < 0)
      var_rec[i][1] = var[i];}
  fprintf(de_var_rec,"%32s",range_name[0]);
  for (i=1; i<range; i++)

  fprintf(de_var_rec,"%10s",range_name[i]);
  fprintf(de_var_rec,"\n");
  for (i=0; i<variable; i++){
    fprintf(de_var_rec,"%32s",var_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(de_var_rec,"%10.2f",var_rec[i][j])
;}
  fprintf(de_var_rec,"\n");}
  fclose(de_var_rec);
  for (i=0; i<mold; i++){

  fprintf(de_mold_rec,"%32s",mold_name[i]);
    for (j=0; j<range-1; j++) {

  fprintf(de_mold_rec,"%10.2f",mold_rec[i][j
]);}
    fprintf(de_mold_rec,"\n");}
  fclose(de_mold_rec);
```

```
/*--UPDATE THE USER'S OPERATING VARIABLE--*/
chdir(user);
de_var_user = fopen ("uservar.dat","w");
de_mold_user = fopen ("usermold.dat","w");
chdir("b:\\program");
for (i=0; i<variable; i++){
  fprintf(de_var_user,"%32s%s%10.2f\n",var_n
ame[i]," = ",var[i]);}
fclose(de_var_user);
for (i=0; i<mold; i++){
  fprintf(de_mold_user,"%32s%s%10.2f\n",mold
_name[i]," = ",mold_user[i]);}
fclose(de_mold_user);
/*UPDATE THE VAR_HP_CWF*/
  chdir (user);
  de_var_hp_cwf = fopen("varhpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(de_var_hp_cwf,"%32s",range_name[0]
);
  for (i=0; i<hp; i++)

  fprintf(de_var_hp_cwf,"%10s",hp_name[i]);
  fprintf(de_var_hp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(de_var_hp_cwf,"%32s",var_name[i]);
    for (j=0; j<hp; j++) {

  fprintf(de_var_hp_cwf,"%10.2f",var_hp_cwf[
i][j]);}
  fprintf(de_var_hp_cwf,"\n");}
  fclose(de_var_hp_cwf);
/*UPDATE THE VAR_FP_CWF*/
  chdir (user);
  de_var_fp_cwf = fopen("varfpcwf.dat","w");
  chdir ("b:\\program");
  fprintf(de_var_fp_cwf,"%32s",range_name[0]
);
  for (i=0; i<fp; i++)

  fprintf(de_var_fp_cwf,"%15s",fp_name[i]);
  fprintf(de_var_fp_cwf,"\n");
  for (i=0; i<variable; i++){

  fprintf(de_var_fp_cwf,"%32s",var_name[i]);
    for (j=0; j<fp; j++) {

  fprintf(de_var_fp_cwf,"%15.2f",var_fp_cwf[
i][j]);}
    fprintf(de_var_fp_cwf,"\n");}
  fclose(de_var_fp_cwf);
/*UPDATE THE DE_VAR_CWF*/
  chdir (user);
  de_var = fopen ("devarcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<variable; i++){
    fprintf
(de_var,"%32s%6.2f\n",var_name[i],
de_var_cwf[i]);}
  fclose(de_var);chdir (user);
/*UPDATE THE DE_FP_CWF*/
  chdir (user);
  de_fp = fopen ("defpcwf.dat","w");
  chdir ("b:\\program");
  for (i=0; i<fp; i++){
    fprintf (de_fp,"%20s%6.2f\n",fp_name[i],
de_fp_cwf[i]);}
  fclose(de_fp);
  }
/*BEGIN MATERIAL CORRECTION ACTIONS*/
if (answer != 1){
  printf("\n%s\nBEGIN THE MATERIAL
CORRECTION ACTIONS\n%s\n",BORDER,BORDER);
  fprintf(fi,"\n%s\nBEGIN THE MATERIAL
VARIABLE CORRECTION
ACTIONS\n%s\n",BORDER,BORDER);
  printf("\n%s\n%s\n",de_material,de_materia
l_caution);
  fprintf(fi,"\n%s\n%s\n",de_material,de_mat
erial_caution);
  printf("\n%s",AFTER_SUG);
  scanf("%d",&answer);
  fprintf(fi,"\n%s",AFTER_SUG);
  fprintf(fi,"%d",answer);}
if (answer == 2){
  printf("\n%s\n",RESPONED3);
  fprintf(fi,"\n%s\n",RESPONED3);}
/*END MATERIAL CORRECTION ACTIONS*/
/*THE FINAL SUGGESTED STATEMENT*/
if (answer !=1){
  printf("\n%s",BORDER);
  printf("\nTHERE IS NO FURTHER CORRECTION
ACTION AVAIABLE.\nPLEASE CONSULT WITH THE
MOLDING EXPERT\nOR THE RAW MATERIAL SUPPLIER
TO RESOLVE THE PROBLEM");
  fprintf(fi,"\n%s",BORDER);
  fprintf(fi,"\nTHERE IS NO FURTHER
CORRECTION ACTION AVAIABLE.\nPLEASE CONSULT
WITH THE MOLDING EXPERT\nOR THE RAW MATERIAL
SUPPLIER TO RESOLVE THE PROBLEM");}
if (answer == 1){
  printf("\n%s\n",BORDER);
  printf("\n%s\n",RESPONED1);
  fprintf(fi,"\n%s\n",BORDER);
  fprintf(fi,"\n%s\n",RESPONED1);}
printf("\n%s\n",BORDER);
printf("\n\n\n%s",re_do);
scanf("%d",&mc_de);
fprintf(fi,"\n%s\n",BORDER);
fprintf(fi,"\n%s",re_do);
fprintf(fi,"%d",mc_de);
return(mc_de);
}

/*FUNCTION OF DECISION ALGORITHM*/
#include <stdio.h>
#include <string.h>
#include <float.h>
#include <alloc.h>
#include <math.h>
#include <b:\PROGRAM\HEAD\suggest.h>
#include <b:\PROGRAM\HEAD\printout.h>
decision (char *dec_var_name[variable],
float dec_var_rec[variable][range-1],float
dec_var[variable],float
dec_var_hp_cwf[variable][hp],float
dec_var_fp_cwf[variable][fp],float
dec_dev_var_cwf[variable],float
dec_dev_fp_cwf[fp],float
var_priority[variable])
{
float total_var_cwf[variable];    /*THE
TOTAL RANKED CWF OF OPERATING VARIABLE*/
float var_fp[variable][fp];       /*THE
MATRIX OF DEV_VAR * VAR_FP_CWF*/
float dev_var_fp[variable][fp];   /*THE
MATRIX OF var_fp*DEV_FP*/
float var_hp[variable][hp];       /*THE
MATRIX OF DEV_VAR_CWF * VAR_HP_CWF*/
float a,b,c,d,e,f,g,h;            /*THE
TRANSFER SPACE FOR ARRANGE THE PRORITY
VALUE*/
```

```
float r;                              /*THE
CALCULATED VALUE OF TOTAL RANKED CWF OF THE
OPERATING VARIABLE*/
char y;
char *x;                               /*THE
TRANSFER SPACE FOR NAME*/
int i,j,k,l;
for (i=0; i<variable; i++){
  for (j=0; j<fp; j++){
    var_fp[i][j] = dec_dev_fp_cwf[j] *
dec_var_fp_cwf[i][j];}}
for (i=0; i<fp; i++){
  for (j=0; j<variable; j++){
    dev_var_fp[j][i] = dec_dev_var_cwf[j] *
var_fp[j][i];}}
for (i=0; i<hp; i++){
  for (j=0; j<variable; j++){
    var_hp[j][i] = dec_dev_var_cwf[j] *
dec_var_hp_cwf[j][i];}}
for (i=0; i<variable; i++){
  a = 0;
  b = 0;
  for (j=0; j<fp; j++){
    a = a + dev_var_fp[i][j];}
  for (k=0; k<hp; k++){
    b = b + var_hp[i][k];}
  total_var_cwf[i] = (a + b)/(hp + fp);}
/*--CALCULATION THE VALUE OF PRIORITY
WEIGHTING FACTOR--*/
for (i=0; i<variable; i++){
  a = (dec_var_rec[i][0] +
dec_var_rec[i][1]) / 2;
  b = (dec_var_rec[i][1] -
dec_var_rec[i][0]);
  if (dec_dev_var_cwf [i] > 0 && b != 0){
    var_priority[i] = fabs(total_var_cwf[i])
* (dec_var_rec[i][1] - dec_var[i] )/b;}
  if (dec_dev_var_cwf [i] <= 0 && b != 0){
    var_priority[i] = fabs(total_var_cwf[i])
* (dec_var[i] - dec_var_rec[i][0])/b;}
  if (b==0){
    var_priority[i] = 0;}}
/*--ARRANGE THE PRORITY WEIGHTING FACTOR--*/
for (i=0; i<variable-2; i++){
  for (j =0; j<variable-1; j++){
    if (var_priority[j] <
var_priority[j+1]){
      c = var_priority[j];
      var_priority[j] = var_priority[j+1];
      var_priority[j+1] = c;
      d = dec_dev_var_cwf[j];
      dec_dev_var_cwf[j] =
dec_dev_var_cwf[j+1];
      dec_dev_var_cwf[j+1] = d;
      e = dec_var[j];
      dec_var[j] = dec_var[j+1];
      dec_var[j+1] = e;
      for (k=0; k<range-1; k++){
        f = dec_var_rec[j][k];
        dec_var_rec[j][k] =
dec_var_rec[j+1][k];
        dec_var_rec[j+1][k] = f;}
      for (l=0; l<fp; l++){
        g = dec_var_fp_cwf[j][l];
        dec_var_fp_cwf[j][l] =
dec_var_fp_cwf[j+1][l];
        dec_var_fp_cwf[j+1][l] = g;}
      for (l=0; l<hp; l++){
        h = dec_var_hp_cwf[j][l];
        dec_var_hp_cwf[j][l] =
dec_var_hp_cwf[j+1][l];
        dec_var_hp_cwf[j+1][l] = h;}
      x = dec_var_name[j];
      dec_var_name[j] = dec_var_name[j+1];
      dec_var_name[j+1] = x;}}}
return;
}

/*FUNCTION OF SELF-LEARNING MECHANISM*/
#include <stdio.h>
#include <math.h>
#include <dir.h>
#include <b:\program\head\printout.h>
self_learn(int self_answer, float
self_dev_var_cwf[variable], float
self_var_hp_cwf[variable][hp], float
self_var_fp_cwf[variable][fp])
{
int i,j,k,l;
if (self_answer == 3){
  if (self_dev_var_cwf[0] >= 0.1){
    self_dev_var_cwf[0] =
self_dev_var_cwf[0] - 0.05;}
  if (self_dev_var_cwf[0] <= -0.1){
    self_dev_var_cwf[0] =
self_dev_var_cwf[0] + 0.05;}
  for (i=0; i<hp; i++){
    if (self_var_hp_cwf[0][i]>= 0.1 ){
      self_var_hp_cwf[0][i] =
self_var_hp_cwf[0][i] - 0.05;}
    if (self_var_hp_cwf[0][i] <= -0.1){
      self_var_hp_cwf[0][i] =
self_var_hp_cwf[0][i] + 0.05;}}
  for (i=0; i<fp; i++){
    if (self_var_fp_cwf[0][i]>= 0.1 ){
      self_var_fp_cwf[0][i] =
self_var_fp_cwf[0][i] - 0.05;}
    if (self_var_fp_cwf[0][i] <=- 0.1 ){
      self_var_fp_cwf[0][i] =
self_var_fp_cwf[0][i] + 0.05;}}
  return;}
if (self_answer == 2){
  if (self_dev_var_cwf[0] > 0.0){
    self_dev_var_cwf[0] =
self_dev_var_cwf[0] + 0.05;}
  if (self_dev_var_cwf[0] < 0.0){
    self_dev_var_cwf[0] =
self_dev_var_cwf[0] - 0.05;}
  for (i=0; i<hp; i++){
    if (self_var_hp_cwf[0][i] > 0){
      self_var_hp_cwf[0][i] =
self_var_hp_cwf[0][i] + 0.05;}
    if (self_var_hp_cwf[0][i] < 0){
      self_var_hp_cwf[0][i] =
self_var_hp_cwf[0][i] - 0.05;}}
  for (j=0; j<fp; j++){
    if (self_var_fp_cwf[0][j] > 0){
      self_var_fp_cwf[0][j] =
self_var_fp_cwf[0][j] + 0.05;}
    if (self_var_fp_cwf[0][j] < 0){
      self_var_fp_cwf[0][j] =
self_var_fp_cwf[0][j] - 0.05;}}
  return;}
return;
}

/*FUNCTION OF HEADING PRINTOUT*/
#include <stdio.h>
space(int number)
{
int i;
for (i=0; i<number; i++){
  putch(' ');}
return;}
```

APPENDIX B
KNOWLEDGE BASE

In this program, the #define command in the C programming language is used to store the knowledge. The knowledge includes the corrective actions, and the explanation statements of the declarative procedures and of the corrective actions. In the following, the completed listing of the knowledge is presented.

LISTING OF KNOWLEDGE BASE

```
/*--THE EXPLANATION STATEMENTS FOR MATERIAL CHOICE--*/
#define mat_why "ACTION: MATERIAL IDENTIFICATION\nREASON: SINCE DIFFERENT MATERIALS HAVE
DIFFERENT PHYSICAL PROPERTIES\nWHICH CAUSE DIFFERENT DEGREES OF INFLUENCE FOR THE SPECIFIC
DEVIATION\nTHE SYSTEM REQUIRES THE MATERIAL TO BE IDENTIFIED"

/*--THE EXPLANATION STATEMENTS FOR MANUFACTURER CHOICE--*/
#define manu_why "ACTION: MATERIAL MANUFACTURER IDENTIFICATION\nREASON: SINCE DIFFERENT
MATERIAL MANUFACTURERS PRODUCTS HAVE\nDIFFERENT PHYSICAL PROPERTIES WHICH CAUSE DIFFERENT
DEGREES OF INFLUENCE\nFOR THE SPECIFIC DEVIATION\nTHE SYSTEM REQUIRES THE MANUFACTURER TO
BE IDENTIFIED"

/*--THE EXPLANATION STATEMENTS FOR GRADE CHOICE--*/
#define grade_why "ACTION: MATERIAL GRADE IDENTIFICATION\nREASON: SINCE DIFFERENT MATERIAL
GRADES HAVE DIFFERENT PHYSICAL PROPERTIES\nWHICH CAUSE DIFFERENT DEGREES OF INFLUENCE FOR
THE SPECIFIC DEVIATION\nDIFFERENT MATERIAL GRADES HAVE DIFFERENT RECOMMENDED
OPERATING\nCONDITIONS WHICH WILL BE EMPLOYED INTO DECISION ALGORITHM TO\nCALCULATE THE
OPERATING VARIABLE PRIORITY WEIGHTING FACTORS WHICH\nINFLUENCE THE DEVIATION RESOLUTION
PROCEDURES\nTHE SYSTEM REQUIRES THE GRADE TO BE IDENTIFIED"

/*THE EXPLANATION STATEMENTS FOR CONFIRMATION OF RECOMMENDED OPERATING CONDITIONS--*/
#define rec_why "ACTION: CONFIRMATION OF THE RECOMMENDED OPERATING CONDITIONS\nREASON:
SINCE THE VALUES OF THE RECOMMENDED OPERATING CONDITIONS\nWILL BE EMPLOYED IN THE DECISION
ALGORITHM TO CALCULATE\nTHE OPERATING VARIABLE PRIORITY WEIGHTING FACTORS WHICH\nINFLUENCE
THE DEVIATION RESOLUTION PROCEDURES\nTHE SYSTEM REQUIRES THE ACCURACY OF RECOMMENDED
OPERATING CONDITIONS"

/*THE EXPLANATION STATEMENT FOR THE USER INPUT DATA--*/
#define var_why "ACTION: INPUT THE OPERATING CONDITIONS\nREASON:SINCE THE OPERATING
CONDITIONS WILL BE EMPLOYED IN THE\nDECISION ALGORITHM TO CALCULATE THE OPERATING
VARIABLES PRIORITY\nWEIGHTING FACTOR WHICH INFLUENCES THE DEVIATION RESOLUTION
PROCEDURES\nTHE SYSTEM REQUIRES THE OPERATING CONDITIONS TO BE IDENTIFIED"

/*THE EXPLANATION STATEMENTS FOR THE CONFIRMATION OF CORRELATIVE WEIGHTING FACTORS BETWEEN
OPERATING VARIABLE AND PHYSICAL PROPERTIES--*/
#define var_hp_why "ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS
BETWEEN\nOPERATING VARIABLES AND INHERENT PHYSICAL PROPERTIES\nREASON:SINCE THE
CORRELATIVE WEIGHTING FACTORS BETWEEN OPERATING VARIABLES\nAND INHERENT PHYSICAL
PROPERTIES WILL BE EMPLOYED IN THE DECISION\nALGORITHM TO CALCULATE THE OPERATING VARIABLE
PRIORITY WEIGHTING\nFACTORS WHICH INFLUENCES THE DEVIATION RESOLUTION PROCEDURES\nTHE
SYSTEM REQUIRES THE CORRELATIVE WEIGHTING FACTORS BETWEEN\nOPERATING VARIABLES AND
INHERENT PHYSICAL PROPERTIES\nTO BE IDENTIFIED"

/*--THE EXPLANATION STATEMENTS FOR THE CONFIRMATION OF CWF BETWEEN PP AND PP--*/
#define var_fp_why "ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS
BETWEEN\nOPERATING VARIABLES AND INFLUENCING PHYSICAL PROPERTIES\nREASON: SINCE THE
CORRELATIVE WEIGHTING FACTORS BETWEEN THE OPERATING\nVARIABLES AND INFLUENCING PHYSICAL
PROPERTIES WILL BE EMPLOYED IN\nTHE DECISION ALGORITHM TO CALCULATE THE OPERATING VARIABLE
PRIORITY\nWEIGHTING FACTORS WHICH INFLUENCE THE DEVIATION RESOLUTION PROCEDURES\nTHE
SYSTEM REQUIRES THE CORRELATIVE WEIGHTING FACTORS BETWEEN\nOPERATING VARIABLES AND
INFLUENCING PHYSICAL PROPERTIES\nTO BE IDENTIFIED"
```

```
/*--THE EXPLANATION STATEMENTS FOR IDENTIFICATION OF DEVIATION--*/
#define dev_why "ACTION: DEVIATION IDENTIFICATION\nREASON: SINCE DIFFERENT DEVIATIONS HAVE
DIFFERENT PROCEDURES\nFOR RESOLUTION\nTHE SYSTEM REQURIES THE DEVIATION TO BE IDENTIFIED"

/*--THE EXPLANATION STATEMENTS FOR THE CONFIRMATION OF CWF BETWEEN DEV AND VAR--*/
#define dev_var_why "ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS
BETWEEN\nTHE DEVIATION AND THE OPERATING VARIABLES\nREASON: SINCE THE CORRELATIVE
WEIGHTING FACTORS BETWEEN THE DEVIATION\nAND THE OPERATING VARIABLES ARE EMPLOYED IN THE
DECISION ALGORITHM\nTO CALCULATE THE OPERATING VARIABLE PRIORITY WEIGHTING\nFACTORS WHICH
INFLUENCES THE DEVIATION RESOLTION PROCEDURE\nTHE SYSTEM REQUIRES THE CORRELATIVE
WEIGHTING FACTORS BETWEEN\nDEVIATION AND OPERATING VARIABLES TO BE IDENTIFIED"

/*--THE EXPLANATION STATEMENTS FOR THE CONFIRMATION OF CWF BETWEEN DEV AND PP--*/
#define dev_fp_why "ACTION: CONFIRMATION OF THE CORRELATIVE WEIGHTING FACTORS BETWEEN\nTHE
DEVIATION AND THE INFLUENCING PHYSICAL PROPERTIES\nREASON: SINCE THE CORRELATIVE WEIGHTING
FACTORS BETWEEN THE DEVIATION\nAND THE INFLUENCING PHYSICAL PROPERTIES ARE EMPLOYED IN
THE\nDECISION ALGORITHM TO CALCULATE THE OPERATING VARIABLE PRIORITY\nWEIGHTING FACTORS
WHICH INFLUENCES THE DEVIATION RESOLUTION PROCEDURE\nTHE SYSTEM RERUIRES THE CORRELATIVE
WEIGHTING FACTORS BETWEEN\nTHE DEVIATION AND THE INFLUENCING PHYSICAL PROPERTIES TO BE
IDENTIFIED"

/KNOWLEDGE BASE FOR SHORT SHOT DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*CHECK HOPPER FOR MATERIAL SUPLLY*/
#define ss_method_1 "IS YOUR HOPPER EMPTY OR NOT\nIF IT IS EMPTY, PLAESE ADD THE MATERIAL
INTO THE HOPPER\n"
#define ss_method_why_1 "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSE FOR\nSHORT SHOT.
\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*CLEAN THE MOLD SURFACE*/
#define ss_method_2 "DOES THE MOLD SURFACE STICK WITH MATERIAL\nOR/AND FOREIGN
CONTAMINATION\nIF IT DOES, PLEASE CLEAN THE MOLD SURFACE"
#define ss_method_why_2 "WHEN MOLD SURFACE STICK WITH MATERIAL OR/AND FOREIGN
CONTAMINATION,\nIT RESULTS IN THE MATERIAL CAN FLOW THROUGH IN THE MOLD CAVITY.\nTHE STICK
MATERIAL REQUIRES TO BE REMOVED.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*USE THE MAXIMUM INJECTION SPEED*/
#define ss_method_3 "PLEASE USE MAXIMUM BOSTER PRESSURE OR INJECTION SPEED"
#define ss_method_why_3 "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSE FOR
SHORT SHOT\nUSE MAXIMUM BOOSTER PRESSURE OR INJECTION PRESSURE CAN\nINCREASE INJECTION
PRESSURE.\nTHE SYSTEM PROVIDE \nTHIS ACTION AS SEQUENCE"
/*CHECK TEMPERATUER INDICATORS--*/
#define ss_method_4 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define ss_method_why_4 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL
TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES SHORT
SHOT.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL
TEMPERATURE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*CHECK PRESSURE INDICATORS--*/
#define ss_method_5 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING
CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define ss_method_why_5 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n.
IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES SHORT SHOT\nCHECKING THE
PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THE SYSTEM PROVIDES THIS
ACTION AS SEQEUNCE"
/*CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define ss_method_6 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW
OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define ss_method_why_6 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\nOF THE
MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF THE INJECTION\nMATERIAL AND CAUSES SHORT
SHOT\nCHECKING THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION
MATERIAL\nTHE SYSTEM PROVIDES THIS ACTION AS SEQEUNCE"
/*CHECK SCREW POITION INDICATORS--*/
#define ss_method_7 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW
POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define ss_method_why_7 "THE INCORRECT SCREW POSITION READING MISLEADS THE QUANTITY\nOF
THE INJECTION MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF\nTHE INJECTION MATERIAL AND
CAUSES SHORT SHOT.\nCHECKING THE SCREW POSITION READING CAN ENSURE AN ACCURANCY
QUANTITY\nOF THE INJECTION MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE"
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*INCREASE SHOT SIZE--*/
#define ss_inc_shot "INCREASE SHOT SIZE (in) TO "
```

```
#define ss_inc_shot_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSHORT
SHOT.\nTHE ACTION, INCREASING SHOT SIZE, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY.
\nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define ss_dec_cus "DECREASE CUSHION (in) TO "
#define ss_dec_cus_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSHORT
SHOT.\nTHE ACTION, DECREASING CUSHION, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY.
\nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define ss_inc_bar "INCREASE BARREL TEMPERATURE (deg F) TO "
#define ss_inc_bar_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SHORT SHOT.\nTHE ACTION, INCREASING THE BARREL TEMPERATURE, IT WILL
DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define ss_inc_mold "INCREASE MOLD TEMPERATURE (deg F) TO "
#define ss_inc_mold_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SHORT SHOT.\nTHE ACTION, INCREASING THE MOLD TEMPERATURE, IT WILL
DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define ss_inc_noz "INCREASE NOZZLE TEMPERATURE (deg F) TO "
#define ss_inc_noz_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SHORT SHOT.\nTHE ACTION, INCREASING THE NOZZLE TEMPERATURE, IT WILL
DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define ss_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define ss_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSES\nFOR
SHORT SHOT.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE REGRIND RATE--*/
#define ss_dec_reg "DECREASE REGRIND RATE (%) TO "
#define ss_dec_reg_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SHORT SHOT.\nTHE ACTION, DECREASE THE REGRIND RATE, IT WILL DECREASE
THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE THE \nMATERIAL FILLING
SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS
SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE SCREW SPEED--*/
#define ss_inc_screw "DECREASE SCREW SPEED (rpm) TO "
#define ss_inc_screw_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SHORT SHOT.\nTHE ACTION, INCREASE THE SCREW SPEED, IT WILL DECREASE
THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL FILLING
SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS
SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define ss_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define ss_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE SHORT SHOT.\nTHE ACTION, INCREASE INJECTION
TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define ss_inc_cyc_time "INCREASE CYCLE TIME (sec) TO "
#define ss_inc_cyc_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE SHORT SHOT.\nTHE ACTION, INCREASE CYCLE
TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE GATE SIZE*/
#define ss_inc_gate "INCCREASE GATE SIZE (in) TO "
#define ss_inc_gate_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSHORT
SHOT.\nTHE ACTION, INCREASING GATE SIZE, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY.
\nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR DECREASE COOLING CHANNEL SIZE*/
#define ss_dec_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define ss_dec_cooling_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE
OF THE MAJOR CAUSES FOR SHORT SHOT.\nTHE ACTION, DECREASE THE COOLING CHANNEL SIZE, IT
WILL DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE
\nMATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES
THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE RUNNER SIZE*/
#define ss_inc_runner "DECREAS RUNNER SIZE (in) TO "
```

```
#define ss_inc_runner_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSHORT
SHOT.\nTHE ACTION, INCREASING RUNNER SIZE, IT WILL INCREASE THE FILLING \nMATERIAL
QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE VENTING CHANNEL SIZE*/
#define ss_inc_vent "INCREASE VENTING CHANNEL SIZE (in) TO "
#define ss_inc_vent_why "SINCE THE EXCEEDED CAVITY PRESSURE IS THE ONE OF MAJOR CAUSES
FOR\nSHORT SHOT.\nINCREASING VENTING CHANNEL WILL DECREASE THE CAVITY PRESSURE.\nTHE
SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define ss_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE
ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define ss_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/KNOWLEDGE BASE FOR PIT MRAKS DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*CHECK HOPPER FOR MATERIAL SUPLLY*/
#define pm_method_1 "IS YOUR HOPPER EMPTY OR NOT\nIF IT IS EMPTY, PLAESE ADD THE MATERIAL
INTO THE HOPPER\n"
#define pm_method_why_1 "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSE FOR\nPIT MARKS.
\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*CLEAN THE MOLD SURFACE*/
#define pm_method_2 "DOES THE MOLD SURFACE STICK WITH MATERIAL\nOR/AND FOREIGN
CONTAMINATION\nIF IT DOES, PLEASE CLEAN THE MOLD SURFACE"
#define pm_method_why_2 "WHEN MOLD SURFACE STICK WITH MATERIAL OR/AND FOREIGN
CONTAMINATION,\nIT RESULTS IN THE MATERIAL CAN FLOW THROUGH IN THE MOLD CAVITY.\nTHE STICK
MATERIAL REQUIRES TO BE REMOVED.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*USE THE MAXIMUM INJECTION SPEED*/
#define pm_method_3 "PLEASE USE MAXIMUM BOSTER PRESSURE OR INJECTION SPEED"
#define pm_method_why_3 "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSE FOR PIT
MARKS\nUSE MAXIMUM BOOSTER PRESSURE OR INJECTION PRESSURE CAN\nINCREASE INJECTION
PRESSURE.\nTHE SYSTEM PROVIDE \nTHIS ACTION AS SEQUENCE"
/*CHECK TEMPERATUER INDICATORS--*/
#define pm_method_4 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define pm_method_why_4 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL
TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES PIT
MARKS.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL
TEMPERATURE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*CHECK PRESSURE INDICATORS--*/
#define pm_method_5 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING
CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define pm_method_why_5 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n.
IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES PIT MARKS\nCHECKING THE
PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THE SYSTEM PROVIDES THIS
ACTION AS SEQEUNCE"
/*CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define pm_method_6 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW
OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define pm_method_why_6 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\nOF THE
MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF THE INJECTION\nMATERIAL AND CAUSES PIT
MARKS\nCHECKING THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION
MATERIAL\nTHE SYSTEM PROVIDES THIS ACTION AS SEQEUNCE"
/*CHECK SCREW POITION INDICATORS--*/
#define pm_method_7 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW
POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define pm_method_why_7 "THE INCORRECT SCREW POSITION READING MISLEADS THE QUANTITY\nOF
THE INJECTION MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF\nTHE INJECTION MATERIAL AND
CAUSES PIT MARKS.\nCHECKING THE SCREW POSITION READING CAN ENSURE AN ACCURANCY
QUANTITY\nOF THE INJECTION MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE"
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*INCREASE SHOT SIZE--*/
#define pm_inc_shot "INCREASE SHOT SIZE (in) TO "
#define pm_inc_shot_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nPIT
MARKS.\nTHE ACTION, INCREASING SHOT SIZE, IT WILL INCREASE THE FILLING \nMATERIAL
QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define pm_dec_cus "DECREASE CUSHION (in) TO "
```

```
#define pm_dec_cus_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nPIT
MARKS.\nTHE ACTION, DECREASING CUSHION, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY.
\nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define pm_inc_bar "INCREASE BARREL TEMPERATURE (deg F) TO "
#define pm_inc_bar_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR PIT MARKS.\nTHE ACTION, INCREASING THE BARREL TEMPERATURE, IT WILL
DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define pm_inc_mold "INCREASE MOLD TEMPERATURE (deg F) TO "
#define pm_inc_mold_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR PIT MARKS.\nTHE ACTION, INCREASING THE MOLD TEMPERATURE, IT WILL
DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define pm_inc_noz "INCREASE NOZZLE TEMPERATURE (deg F) TO "
#define pm_inc_noz_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR PIT MARKS.\nTHE ACTION, INCREASING THE NOZZLE TEMPERATURE, IT WILL
DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE."
#define pm_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define pm_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSES\nFOR
PIT MARKS.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE REGRIND RATE--*/
#define pm_dec_reg "DECREASE REGRIND RATE (%) TO "
#define pm_dec_reg_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR PIT MARKS.\nTHE ACTION, DECREASE THE REGRIND RATE, IT WILL DECREASE
THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE THE \nMATERIAL FILLING
SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS
SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE SCREW SPEED--*/
#define pm_inc_screw "DECREASE SCREW SPEED (rpm) TO "
#define pm_inc_screw_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR PIT MARKS.\nTHE ACTION, INCREASE THE SCREW SPEED, IT WILL DECREASE
THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL FILLING
SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS
SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define pm_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define pm_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE PIT MARKS.\nTHE ACTION, INCREASE INJECTION
TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define pm_inc_cyc_time "INCREASE CYCLE TIME (sec) TO "
#define pm_inc_cyc_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE PIT MARKS.\nTHE ACTION, INCREASE CYCLE TIME
WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE GATE SIZE*/
#define pm_inc_gate "INCCREASE GATE SIZE (in) TO "
#define pm_inc_gate_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nPIT
MARKS.\nTHE ACTION, INCREASING GATE SIZE, IT WILL INCREASE THE FILLING \nMATERIAL
QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR DECREASE COOLING CHANNEL SIZE*/
#define pm_dec_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define pm_dec_cooling_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE
OF THE MAJOR CAUSES FOR PIT MARKS.\nTHE ACTION, DECREASE THE COOLING CHANNEL SIZE, IT WILL
DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE RUNNER SIZE*/
#define pm_inc_runner "DECREAS RUNNER SIZE (in) TO "
#define pm_inc_runner_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nPIT
MARKS.\nTHE ACTION, INCREASING RUNNER SIZE, IT WILL INCREASE THE FILLING \nMATERIAL
QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE VENTING CHANNEL SIZE*/
#define pm_inc_vent "INCREASE VENTING CHANNEL SIZE (in) TO "
```

```
#define pm_inc_vent_why "SINCE THE EXCEEDED CAVITY PRESSURE IS THE ONE OF MAJOR CAUSES FOR\nPIT MARKS.\nINCREASING VENTING CHANNEL WILL DECREASE THE CAVITY PRESSURE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define pm_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define pm_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE \nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR SURFACE RIPPLES DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*CHECK HOPPER FOR MATERIAL SUPLLY*/
#define sr_method_1 "IS YOUR HOPPER EMPTY OR NOT\nIF IT IS EMPTY, PLAESE ADD THE MATERIAL INTO THE HOPPER\n"
#define sr_method_why_1 "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSE FOR\nSURFACE RIPPLES. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*CLEAN THE MOLD SURFACE*/
#define sr_method_2 "DOES THE MOLD SURFACE STICK WITH MATERIAL\nOR/AND FOREIGN CONTAMINATION\nIF IT DOES, PLEASE CLEAN THE MOLD SURFACE"
#define sr_method_why_2 "WHEN MOLD SURFACE STICK WITH MATERIAL OR/AND FOREIGN CONTAMINATION,\nIT RESULTS IN THE MATERIAL CAN FLOW THROUGH IN THE MOLD CAVITY.\nTHE STICK MATERIAL REQUIRES TO BE REMOVED.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*USE THE MAXIMUM INJECTION SPEED*/
#define sr_method_3 "PLEASE USE MAXIMUM BOSTER PRESSURE OR INJECTION SPEED"
#define sr_method_why_3 "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSE FOR SURFACE RIPPLES\nUSE MAXIMUM BOOSTER PRESSURE OR INJECTION PRESSURE CAN\nINCREASE INJECTION PRESSURE.\nTHE SYSTEM PROVIDE \nTHIS ACTION AS SEQUENCE"
/*CHECK TEMPERATUER INDICATORS--*/
#define sr_method_4 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sr_method_why_4 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES SURFACE RIPPLES.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL TEMPERATURE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*CHECK PRESSURE INDICATORS--*/
#define sr_method_5 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sr_method_why_5 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n. IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES SURFACE RIPPLES\nCHECKING THE PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THE SYSTEM PROVIDES THIS ACTION AS SEQEUNCE"
/*CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define sr_method_6 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sr_method_why_6 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\nOF THE MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF THE INJECTION\nMATERIAL AND CAUSES SURFACE RIPPLES\nCHECKING THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION MATERIAL\nTHE SYSTEM PROVIDES THIS ACTION AS SEQEUNCE"
/*CHECK SCREW POITION INDICATORS--*/
#define sr_method_7 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sr_method_why_7 "THE INCORRECT SCREW POSITION READING MISLEADS THE QUANTITY\nOF THE INJECTION MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF\nTHE INJECTION MATERIAL AND CAUSES SURFACE RIPPLES.\nCHECKING THE SCREW POSITION READING CAN ENSURE AN ACCURANCY QUANTITY\nOF THE INJECTION MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE"
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*INCREASE SHOT SIZE--*/
#define sr_inc_shot "INCREASE SHOT SIZE (in) TO "
#define sr_inc_shot_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSURFACE RIPPLES.\nTHE ACTION, INCREASING SHOT SIZE, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define sr_dec_cus "DECREASE CUSHION (in) TO "
#define sr_dec_cus_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSURFACE RIPPLES.\nTHE ACTION, DECREASING CUSHION, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define sr_inc_bar "INCREASE BARREL TEMPERATURE (deg F) TO "
```

```
#define sr_inc_bar_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SURFACE RIPPLES.\nTHE ACTION, INCREASING THE BARREL TEMPERATURE, IT
WILL DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE
MATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES
THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define sr_inc_mold "INCREASE MOLD TEMPERATURE (deg F) TO "
#define sr_inc_mold_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SURFACE RIPPLES.\nTHE ACTION, INCREASING THE MOLD TEMPERATURE, IT
WILL DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE
MATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES
THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define sr_inc_noz "INCREASE NOZZLE TEMPERATURE (deg F) TO "
#define sr_inc_noz_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SURFACE RIPPLES.\nTHE ACTION, INCREASING THE NOZZLE TEMPERATURE, IT
WILL DECREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE
MATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES
THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define sr_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define sr_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSES\nFOR
SURFACE RIPPLES.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE REGRIND RATE--*/
#define sr_dec_reg "DECREASE REGRIND RATE (%) TO "
#define sr_dec_reg_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SURFACE RIPPLES.\nTHE ACTION, DECREASE THE REGRIND RATE, IT WILL
DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN INCREASE THE \nMATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE SCREW SPEED--*/
#define sr_inc_screw "DECREASE SCREW SPEED (rpm) TO "
#define sr_inc_screw_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF
THE MAJOR CAUSES FOR SURFACE RIPPLES.\nTHE ACTION, INCREASE THE SCREW SPEED, IT WILL
DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define sr_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define sr_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE SURFACE RIPPLES.\nTHE ACTION, INCREASE
INJECTION TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS
SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define sr_inc_cyc_time "INCREASE CYCLE TIME (sec) TO "
#define sr_inc_cyc_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE SURFACE RIPPLES.\nTHE ACTION, INCREASE
CYCLE TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS
SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE GATE SIZE*/
#define sr_inc_gate "INCCREASE GATE SIZE (in) TO "
#define sr_inc_gate_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSURFACE
RIPPLES.\nTHE ACTION, INCREASING GATE SIZE, IT WILL INCREASE THE FILLING \nMATERIAL
QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR DECREASE COOLING CHANNEL SIZE*/
#define sr_dec_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define sr_dec_cooling_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE
OF THE MAJOR CAUSES FOR SURFACE RIPPLES.\nTHE ACTION, DECREASE THE COOLING CHANNEL SIZE,
IT WILL DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE
\nMATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES
THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE RUNNER SIZE*/
#define sr_inc_runner "DECREAS RUNNER SIZE (in) TO "
#define sr_inc_runner_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSURFACE
RIPPLES.\nTHE ACTION, INCREASING RUNNER SIZE, IT WILL INCREASE THE FILLING \nMATERIAL
QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE VENTING CHANNEL SIZE*/
#define sr_inc_vent "INCREASE VENTING CHANNEL SIZE (in) TO "
#define sr_inc_vent_why "SINCE THE EXCEEDED CAVITY PRESSURE IS THE ONE OF MAJOR CAUSES
FOR\nSURFACE RIPPLES.\nINCREASING VENTING CHANNEL WILL DECREASE THE CAVITY PRESSURE.\nTHE
SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
```

```
#define sr_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define sr_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE \nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR SPLAY MARKS DEVIATION*/
/*--THE SUGGESTED ACTION FOR CHECK TEMPERATUER INDICATORS--*/
#define sp_method_1 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sp_method_why_1 "THE INCORRECT TEMPERATURE READING MISLEADS\nTHE HEATING AMOUNT OF THE MATERIAL, IT MAYBE RESULTS IN\nSPLAY MARKS DEVIATION.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE SUGGESTED ACTION FOR CHECK PRESSURE INDICATORS--*/
#define sp_method_2 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sp_method_why_2 "THE INCORRECT PRESSURE READING MISLEADS\nTHE INJECTION PRESSURE READING. IT MAYBE RESULTS IN\nSPLAY MARKS DEVIATION.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE SUGGESTED ACTION FOR CHECK SCREW POITION INDICATORS--*/
#define sp_method_3 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sp_method_why_3 "THE INCORRECT SCREW POSITION READING MISLEADS\nTHE QUANTITY OF THE INJECTION MATERIAL. IT MAYBE RESULTS\nIN SPLAY MARKS DEVIATION\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE SUGGESTED ACTION FOR CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define sp_method_4 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sp_method_why_4 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\n OF THE MATERIAL. IT MAYBE RESULTS IN SPLAY MARKS DEVIATIN.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE SUGGESTED ACTION FOR CHECK PRESSURE INDICATORS--*/
#define sp_method_5 "PLEASE USE A VENTED BARREL"
#define sp_method_why_5 "THE EXCEEDED MATERIAL MOISTURE MAYBE RESULTS IN\nSPLAY MARKS DEVIATION. USING A VENTED BARREL ALLOWS\nMATERIAL MOISTURE ESCAPE FROM THE BARREL DURING\nPLASTICATION STAGE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE SUGGESTED ACTION FOR CHECK SCREW POITION INDICATORS--*/
#define sp_method_6 "PLEASE CHANGE A SMALL NOZZLE ORIFICE"
#define sp_method_why_6 "TOO A LARGE NOZZLE ORIFICE RESULTS IN THE INJECTION\nSPEED TOO SLOW AND MAYBE CAUSES SPLAY MARKS DEVIAITON\nUSING A SMALL NOZZLE ORIFICE, IT CAN INCREASE INJECTION SPEED.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE CUSHION--*/
#define sp_dec_cus "DECREASE CUSHION (in) TO "
#define sp_dec_cus_why "THE MATERIAL SHORTAGE IS A CAUSE FOR THE SPLAY MARKS.\nTHE ACTION, DECREASING CUSHION WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE BARREL TEMPERATURE--*/
#define sp_dec_bar "DECREASE BARREL TEMPERATURE (deg F) TO "
#define sp_dec_bar_why "THE MATERIAL FILLING SPEED TOO HIGH IN THE MOLD CAVITY IS A CAUSES\nFOR SPLAY MARKS. THE ACTION, DECREASING THE BARREL TEMPERATURE,\nIT WILL DECREASE THE MATERIAL FILLING SPEED IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE MOLD TEMPERATURE--*/
#define sp_dec_mold "DECREASE MOLD TEMPERATURE (deg F) TO "
#define sp_dec_mold_why "THE MATERIAL FILLING SPEED TOO HIGH IN THE MOLD CAVITY IS A CAUSES\nFOR SPLAY MARKS. THE ACTION, DECREASING THE MOLD TEMPERATURE,\nIT WILL DECREASE THE MATERIAL FILLING SPEED IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE NOZZLE TEMPERATURE --*/
#define sp_dec_noz "DECREASE NOZZLE TEMPERATURE (deg F) TO "
#define sp_dec_noz_why "THE MATERIAL FILLING SPEED TOO HIGH IN THE MOLD CAVITY IS A CAUSES\nFOR SPLAY MARKS. THE ACTION, DECREASING THE NOZZLE TEMPERATURE,\nIT WILL DECREASE THE MATERIAL FILLING SPEED IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR DECREASE INJECTION PRESSURE --*/
#define sp_dec_inj_pre "DECREASE INJECTION PRESSURE (psi) TO "
#define sp_dec_inj_pre_why "THE MATERIAL FILLING SPEED TOO HIGH IN THE MOLD CAVITY IS A CAUSES\nFOR SPLAY MARKS. THE ACTION, DECREASING THE INJECTION PRESSURE,\nIT WILL DECREASE THE MATERIAL FILLING SPEED IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
```

```
/*--THE RESOLVED STATEMENTS FOR INCREASE SCREW SPEED--*/
#define sp_inc_screw "DECREASE SCREW SPEED (rpm) TO "
#define sp_inc_screw_why "THE EXCEEDED MATERIAL MOISTURE IS A CAUSE FOR THE SPLAY
MARKS.\nTHE ACTION, DECREASING THE SCREW SPEED,\nIT ALLOWS THE MATERIAL MOISTURE ESCAPE
DURING PLASTICATION STAGE\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR DECREASE INJECTION TIME--*/
#define sp_dec_inj_time "DECREASE INJECTION TIME (sec) TO "
#define sp_dec_inj_time_why "THE HIGHER PRESSURE GRADIENT IS A CAUSE FOR SPLAY MARKS.
\nTHE ACTION, DECREASE INJECTION TIME WILL DECREASE \nTHE PRESSURE GRADIENT. \nTHE SYSTEM
PROVIDES THIS ACTION AS SEQUENCE"
/*INCREASE DECOMPRESSION TIME*/
#define sp_inc_dec_time "INCREASE DECOMPRESSION TIME(sec) TO "
#define sp_inc_dec_time_why "THE EXCEEDED MATERIAL MOISTURE IS A CAUSE FOR THE SPLAY
MARKS.\nTHE ACTION, INCREASING DECOMPRESSION TIME,\nIT ALLOWS THE MATERIAL MOISTURE ESCAPE
DURING PLASTICATION STAGE\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE COOLING CHANNEL SIZE*/
#define sp_inc_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define sp_inc_cooling_why "THE MATERIAL FILLING SPEED TOO HIGH IN THE MOLD CAVITY IS A
CAUSES\nFOR SPLAY MARKS. THE ACTION, INCREASING THE COOLING CHANNEL,\nIT WILL DECREASE THE
MATERIAL FILLING SPEED IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE VENTING CHANNEL SIZE*/
#define sp_inc_vent "INCREASE VENTING CHANNEL SIZE (in) TO "
#define sp_inc_vent_why "THE EXCEEDED MATERIAL MOISTURE IS A CAUSE FOR THE SPLAY
MARKS.\nTHE ACTION, INCREASING VENTING CHANNEL SIZE,\nIT ALLOWS THE MATERIAL MOISTURE
ESCAPE DURING PACKING AND COOLING STAGE\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define sp_material "PLEASE CHANGE MATERIAL TO ONE WITH A LOWER THERMAL CONDUCTIVITY\nTHE
ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define sp_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR WARPAGE DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*JIG THE PART AND COOL UNIFORMLY*/
#define wa_method_1 "PLAESE JIG THE PART AND COOL UNIFORMLY\n"
#define wa_method_why_1 "THE UNEVEN COOLING IS ONE OF THE MAJOR CAUSE FOR\n WARPAGE. \nTHE
SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*CHECK EJECTOR PINS MARKS*/
#define wa_method_2 "OBSEVER THE EJECTOR PINS MARKS, IF EJECTOR PINS MARKS IS\nUNEVEN
PLEASE ADJUST THE EJECTOR PINS LOCATION"
#define wa_method_why_2 "WHEN EJECTOR PINS LOCATION IS UNEVEN, IT RESULTS IN\nTHE EJECTOR
FORCE UNEVEN APPLY IN THE MOLD SURFACE\nAND CAUSES THE WARPAGE.\nTHE SYSTEM PROVIDES THIS
ACTION AS SEQENCE."
/*CHECK TEMPERATUER INDICATORS--*/
#define wa_method_3 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define wa_method_why_3 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL
TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES PIT
MARKS.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL
TEMPERATURE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQENCE."
/*CHECK PRESSURE INDICATORS--*/
#define wa_method_4 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING
CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define wa_method_why_4 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n.
IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES PIT MARKS\nCHECKING THE
PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THE SYSTEM PROVIDES THIS
ACTION AS SEQEUNCE"
/*CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define wa_method_5 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW
OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define wa_method_why_5 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\nOF THE
MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF THE INJECTION\nMATERIAL AND CAUSES PIT
MARKS\nCHECKING THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION
MATERIAL\nTHE SYSTEM PROVIDES THIS ACTION AS SEQEUNCE"
/*SET UNIFORM TEMPERATURE IN BOTH HAVLES OF MOLD*/
#define wa_method_6 "PLEASE SET UNIFORM TEMPERATURE IN BOTH HAVLES OF MOLD"
#define wa_method_why_6 "THE UNEVEN COOLING RESULT IS ONE OF THE MAJOR CAUSE FOR
WARPAGE.\nTHE SYSTEM PROVIDE \nTHIS ACTION AS SEQUENCE"
/*RELOCATE GATE NEARER HEAVY SECTION*/
```

```
#define wa_method_7 "PLEASE RELOCATE GATE NEARER HEAVY SECTION"
#define wa_method_why_7 "THE UNEVEN COOLING RESULT IS ONE OF THE MAJOR CAUSE FOR
WARPAGE.\nTHE SYSTEM PROVIDE \nTHIS ACTION AS SEQUENCE"
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define wa_dec_bar "DECREASE BARREL TEMPERATURE (deg F) TO "
#define wa_dec_bar_why "THE HIGH THERMAL SHEAR STRESS IS ONE OF THE MAJOR\nCAUSES OF
WARPAGE.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL DECREASE \nTHE DECREASE
THERMAL SHEAR STRESS.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define wa_dec_mold "DECREASE MOLD TEMPERATURE (deg F) TO "
#define wa_dec_mold_why "THE HIGH THERMAL SHEAR STRESS IS ONE OF THE MAJOR\nCAUSES OF
WARPAGE.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL DECREASE \nTHE DECREASE
THERMAL SHEAR STRESS.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define wa_dec_noz "DECREASE NOZZLE TEMPERATURE (deg F) TO "
#define wa_dec_noz_why "THE HIGH THERMAL SHEAR STRESS IS ONE OF THE MAJOR\nCAUSES OF
WARPAGE.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL DECREASE \nTHE DECREASE
THERMAL SHEAR STRESS.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define wa_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define wa_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSES\nFOR
WARPAGE.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR DECREASE SCREW SPEED--*/
#define wa_inc_screw "DECREASE SCREW SPEED (rpm) TO "
#define wa_inc_screw_why "THE HIGH MATERIAL TEMPERATURE IS ONE OF THE MAJOR\nCAUSES FOR
WARPAGE.\nTHE ACTION, INCREASE THE SCREW SPEED, IT WILL DECREASE THE\nMATERIAL TEMPERATURE
IN THE MOLD CAVITY.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define wa_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define wa_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE WILL CAUSE WARPAGE.\nTHE ACTION, INCREASE INJECTION
TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE MOLD CLOSED TIME--*/
#define wa_inc_mold_time "INCREASE MOLD CLOSED TIME (sec) TO "
#define wa_inc_mold_time_why "THE MATERIAL IS NOT SOLIDIFIED COMPLETELY IS ONE OF
THE\nMAJOR CAUSE FOR WARPAGE.\nTHE ACTION, INCREASE MOLD CLOSED TIME PROVIDES MORE
TIME\nFOR COOLING MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR DECREASE GATE SIZE*/
#define wa_dec_gate "DECCREASE GATE SIZE (in) TO "
#define wa_dec_gate_why "THE HIGHER MATERIAL SPEED IN THE MOLD CAVITY IS ONE OF\nMAJOR
CAUSE FOR WARPAGE.\nTHE ACTION, DECREASING GATE SIZE, IT WILL DECREASE THE FILLING \nSPEED
OF THE MATERIAL. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE COOLING CHANNEL SIZE*/
#define wa_inc_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define wa_inc_cooling_why "THE MATERIAL IS NOT SOLIDIFIED COMPLETELY IS ONE OF THE\nMAJOR
CAUSE FOR WARPAGE.\nTHE ACTION, INCREASE COOLING CHANNEL SIZE PROVIDES MORE\nCOOLANT IN
COOLING THE MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR DECREASE RUNNER SIZE*/
#define wa_dec_runner "DECREASE RUNNER SIZE (in) TO "
#define wa_dec_runner_why "THE HIGHER MATERIAL SPEED IN THE MOLD CAVITY IS ONE OF\nMAJOR
CAUSE FOR WARPAGE.\nTHE ACTION, DECREASING RUNNER SIZE, IT WILL DECREASE THE FILLING
\nSPEED OF THE MATERIAL. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define wa_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE
ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define wa_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR SINK MARKS DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*CHECK TEMPERATUER INDICATORS--*/
#define sm_method_1 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sm_method_why_1 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL
TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES SINK
MARKS.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL
TEMPERATURE.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*CHECK PRESSURE INDICATORS--*/
```

```
#define sm_method_2 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sm_method_why_2 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n. IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES SINK MARKS\nCHECKING THE PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*CHECK SCREW POITION INDICATORS--*/
#define sm_method_3 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define sm_method_why_3 "THE INCORRECT SCREW POSITION READING MISLEADS THE QUANTITY\nOF THE INJECTION MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF\nTHE INJECTION MATERIAL AND CAUSES SINK MARKS.\nCHECKING THE SCREW POSITION READING CAN ENSURE AN ACCURANCY QUANTITY\nOF THE INJECTION MATERIAL.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION"
/*RELOCATE GATE NEARER HEAVY SECTION*/
#define sm_method_4 "PLEASE RELOCATE GATE NEARER HEAVY SECTION"
#define sm_method_why_4      "WHEN MATERIAL FLOW CHANGES FROM THIN SECTION TO THICK SECTION\nCAUSE INTERNAL STRESS CONCENTRATE IN THIS AREA.\nCHANGING GATE LOCATION NEARER THE HEAVY SECTION CAN AVOID THIS SITUATION.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define sm_inc_shot "INCREASE SHOT SIZE (in) TO "
#define sm_inc_shot_why "THE INSUFFICIENT MATERIAL SUPPLY DURING THE PACKING STAGE CAUSES \nSINK MARKS.\nTO INCREASE CUSHION INCREASES THE MATERIAL SUPLLY DURING\nTHE PACKING STAGE.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define sm_dec_cus "DECREASE CUSHION (in) TO "
#define sm_dec_cus_why "THE INSUFFICIENT MATERIAL SUPPLY DURING THE PACKING STAGE CAUSES \nSINK MARKS.\nTO DECREASE CUSHION INCREASES THE MATERIAL SUPLLY DURING\nTHE PACKING STAGE.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define sm_inc_bar "INCREASE BARREL TEMPERATURE (deg F) TO "
#define sm_inc_bar_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES SINK MARKS.\nTO INCREASE BARREL TEMPERATURE DECREASES THE VISCOSITY OF\nMATERIAL IN THE MOLD CAVITY WHICH INCREASES THE MATERIAL\nFILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define sm_inc_mold "INCREASE MOLD TEMPERATURE (deg F) TO "
#define sm_inc_mold_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES SINK MARKS.\nTO INCREASE MOLD TEMPERATURE DECREASES THE VISCOSITY OF\nMATERIAL IN THE MOLD CAVITY WHICH INCREASES THE MATERIAL\nFILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define sm_inc_noz "INCREASE NOZZLE TEMPERATURE (deg F) TO "
#define sm_inc_noz_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES SINK MARKS.\nTO INCREASE NOZZLE TEMPERATURE DECREASES THE VISCOSITY OF\nMATERIAL IN THE MOLD CAVITY WHICH INCREASES THE MATERIAL\nFILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define sm_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define sm_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW CAUSES SINK MARKS.\nTHE SYSTEM PROVIDES THIS ACTION"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define sm_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define sm_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND THE COMRESSION STAGEECTION PRESSURE CAUSES SINK MARKS.\nTO INCREASE INJECTION TIME INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE RESOLVED STATEMENTS FOR INCREASE GATE SIZE*/
#define sm_inc_gate "INCCREASE GATE SIZE (in) TO "
#define sm_inc_gate_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nSINK MARKS.\nTHE ACTION, INCREASING GATE SIZE, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE COOLING CHANNEL SIZE*/
#define sm_inc_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define sm_inc_cooling_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE OF THE MAJOR CAUSES FOR SINK MARKS.\nTHE ACTION, DECREASE THE COOLING CHANNEL SIZE, IT WILL DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define sm_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
```

```
#define sm_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR DISTORTION DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*JIG THE PART AND COOL UNIFORMLY*/
#define di_method_1 "PLAESE JIG THE PART AND COOL UNIFORMLY\n"
#define di_method_why_1 "THE UNEVEN COOLING IS ONE OF THE MAJOR CAUSE FOR\n DISTORTION.
\nTHE SYSTEM PROVIDESS THIS ACTION."
/*CHECK EJECTOR PINS MARKS*/
#define di_method_2 "OBSEVER THE EJECTOR PINS MARKS, IF EJECTOR PINS MARKS IS\nUNEVEN
PLEASE ADJUST THE EJECTOR PINS LOCATION"
#define di_method_why_2 "WHEN EJECTOR PINS LOCATION IS UNEVEN, IT RESULTS IN\nTHE EJECTOR
FORCE UNEVEN APPLY IN THE MOLD SURFACE\nAND CAUSES THE DISTORTION.\nTHE SYSTEM PROVIDES
THIS ACTION."
/*CHECK TEMPERATUER INDICATORS--*/
#define di_method_3 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define di_method_why_3 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL
TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES PIT
MARKS.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL
TEMPERATURE.\nTHE SYSTEM PROVIDES THIS ACTION."
/*CHECK PRESSURE INDICATORS--*/
#define di_method_4 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING
CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define di_method_why_4 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n.
IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES PIT MARKS\nCHECKING THE
PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THE SYSTEM PROVIDES THIS
ACTION."
/*CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define di_method_5 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW
OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define di_method_why_5 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\nOF THE
MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF THE INJECTION\nMATERIAL AND CAUSES PIT
MARKS\nCHECKING THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION
MATERIAL\nTHE SYSTEM PROVIDES THIS ACTION."
/*SET UNIFORM TEMPERATURE IN BOTH HAVLES OF MOLD*/
#define di_method_6 "PLEASE SET UNIFORM TEMPERATURE IN BOTH HAVLES OF MOLD"
#define di_method_why_6 "THE UNEVEN COOLING RESULT IS ONE OF THE MAJOR CAUSE FOR
DISTORTION.\nTHE SYSTEM PROVIDES THIS ACTION."
/*RELOCATE GATE NEARER HEAVY SECTION*/
#define di_method_7 "PLEASE RELOCATE GATE NEARER HEAVY SECTION"
#define di_method_why_7 "THE UNEVEN COOLING RESULT IS ONE OF THE MAJOR CAUSE FOR
DISTORTION.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define di_dec_bar "DECREASE BARREL TEMPERATURE (deg F) TO "
#define di_dec_bar_why "THE HIGH THERMAL SHEAR STRESS IS ONE OF THE MAJOR\nCAUSES OF
DISTORTION.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL DECREASE \nTHE
DECREASE THERMAL SHEAR STRESS.\nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define di_dec_mold "DECREASE MOLD TEMPERATURE (deg F) TO "
#define di_dec_mold_why "THE HIGH THERMAL SHEAR STRESS IS ONE OF THE MAJOR\nCAUSES OF
DISTORTION.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL DECREASE \nTHE
DECREASE THERMAL SHEAR STRESS.\nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define di_dec_noz "DECREASE NOZZLE TEMPERATURE (deg F) TO "
#define di_dec_noz_why "THE HIGH THERMAL SHEAR STRESS IS ONE OF THE MAJOR\nCAUSES OF
DISTORTION.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL DECREASE \nTHE
DECREASE THERMAL SHEAR STRESS.\nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define di_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define di_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW IS ONE OF THE MAJOR CAUSES\nFOR
DISTORTION.\nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE SCREW SPEED--*/
#define di_inc_screw "DECREASE SCREW SPEED (rpm) TO "
#define di_inc_screw_why "THE HIGH MATERIAL TEMPERATURE IS ONE OF THE MAJOR\nCAUSES FOR
DISTORTION.\nTHE ACTION, INCREASE THE SCREW SPEED, IT WILL DECREASE THE\nMATERIAL
TEMPERATURE IN THE MOLD CAVITY.\nTHE SYSTEM PROVIDES THIS ACTION."
```

```
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define di_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define di_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND THE COMRESSION STAGEECTION PRESSURE WILL CAUSE DISTORTION.\nTHE ACTION, INCREASE INJECTION TIME WILL INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMETNS FOR INCREASE MOLD CLOSED TIME--*/
#define di_inc_mold_time "INCREASE MOLD CLOSED TIME (sec) TO "
#define di_inc_mold_time_why "THE MATERIAL IS NOT SOLIDIFIED COMPLETELY IS ONE OF THE\nMAJOR CAUSE FOR DISTORTION.\nTHE ACTION, INCREASE MOLD CLOSED TIME PROVIDES MORE TIME\nFOR COOLING MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE RESOLVED STATEMENTS FOR DECREASE GATE SIZE*/
#define di_dec_gate "DECCREASE GATE SIZE (in) TO "
#define di_dec_gate_why "THE HIGHER MATERIAL SPEED IN THE MOLD CAVITY IS ONE OF\nMAJOR CAUSE FOR DISTORTION.\nTHE ACTION, DECREASING GATE SIZE, IT WILL DECREASE THE FILLING \nSPEED OF THE MATERIAL. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION."
/*THE RESOLVED STATEMENTS FOR INCREASE COOLING CHANNEL SIZE*/
#define di_inc_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define di_inc_cooling_why "THE MATERIAL IS NOT SOLIDIFIED COMPLETELY IS ONE OF THE\nMAJOR CAUSE FOR DISTORTION.\nTHE ACTION, INCREASE COOLING CHANNEL SIZE PROVIDES MORE\nCOOLANT IN COOLING THE MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE RESOLVED STATEMENTS FOR DECREASE RUNNER SIZE*/
#define di_dec_runner "DECREASE RUNNER SIZE (in) TO "
#define di_dec_runner_why "THE HIGHER MATERIAL SPEED IN THE MOLD CAVITY IS ONE OF\nMAJOR CAUSE FOR DISTORTION.\nTHE ACTION, DECREASING RUNNER SIZE, IT WILL DECREASE THE FILLING \nSPEED OF THE MATERIAL. \nTHE SYSTEM PROVIDES THIS SUGGESTED ACTION."
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define di_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define di_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,\nTHE ELECTRICAL PROPERTIES, THE OPTICAL PROPERTIES, AND THE \nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURES THAT THESE\nPROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR VOIDS DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*CHECK TEMPERATUER INDICATORS--*/
#define vi_method_1 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define vi_method_why_1 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES VOIDS.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL TEMPERATURE.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*CHECK PRESSURE INDICATORS--*/
#define vi_method_2 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define vi_method_why_2 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n. IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES VOIDS\nCHECKING THE PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*CHECK SCREW POITION INDICATORS--*/
#define vi_method_3 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define vi_method_why_3 "THE INCORRECT SCREW POSITION READING MISLEADS THE QUANTITY\nOF THE INJECTION MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF\nTHE INJECTION MATERIAL AND CAUSES VOIDS.\nCHECKING THE SCREW POSITION READING CAN ENSURE AN ACCURANCY QUANTITY\nOF THE INJECTION MATERIAL.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION"
/*RELOCATE GATE NEARER HEAVY SECTION*/
#define vi_method_4 "PLEASE RELOCATE GATE NEARER HEAVY SECTION"
#define vi_method_why_4        "WHEN MATERIAL FLOW CHANGES FROM THIN SECTION TO THICK SECTION\nCAUSE INTERNAL STRESS CONCENTRATE IN THIS AREA.\nCHANGING GATE LOCATION NEARER THE HEAVY SECTION CAN AVOID THIS SITUATION.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define vi_inc_shot "INCREASE SHOT SIZE (in) TO "
#define vi_inc_shot_why "THE INSUFFICIENT MATERIAL SUPPLY DURING THE PACKING STAGE CAUSES \nVOIDS.\nTO INCREASE CUSHION INCREASES THE MATERIAL SUPLLY DURING\nTHE PACKING STAGE.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define vi_dec_cus "DECREASE CUSHION (in) TO "
```

```
#define vi_dec_cus_why "THE INSUFFICIENT MATERIAL SUPPLY DURING THE PACKING STAGE CAUSES
\nVOIDS.\nTO DECREASE CUSHION INCREASES THE MATERIAL SUPLLY DURING\nTHE PACKING
STAGE.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define vi_inc_bar "INCREASE BARREL TEMPERATURE (deg F) TO "
#define vi_inc_bar_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
VOIDS.\nTO INCREASE BARREL TEMPERATURE DECREASES THE VISCOSITY OF\nMATERIAL IN THE MOLD
CAVITY WHICH INCREASES THE MATERIAL\nFILLING SPEED AND INJECTION PRESSURE IN THE MOLD
CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define vi_inc_mold "INCREASE MOLD TEMPERATURE (deg F) TO "
#define vi_inc_mold_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
VOIDS.\nTO INCREASE MOLD TEMPERATURE DECREASES THE VISCOSITY OF\nMATERIAL IN THE MOLD
CAVITY WHICH INCREASES THE MATERIAL\nFILLING SPEED AND INJECTION PRESSURE IN THE MOLD
CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define vi_inc_noz "INCREASE NOZZLE TEMPERATURE (deg F) TO "
#define vi_inc_noz_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
VOIDS.\nTO INCREASE NOZZLE TEMPERATURE DECREASES THE VISCOSITY OF\nMATERIAL IN THE MOLD
CAVITY WHICH INCREASES THE MATERIAL\nFILLING SPEED AND INJECTION PRESSURE IN THE MOLD
CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define vi_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define vi_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW CAUSES VOIDS.\nTHE SYSTEM
PROVIDES THIS ACTION"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define vi_inc_inj_time "INCREASE INJECTION TIME (sec) TO "
#define vi_inc_inj_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE CAUSES VOIDS.\nTO INCREASE INJECTION TIME INCREASE THE
PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE RESOLVED STATEMENTS FOR INCREASE GATE SIZE*/
#define vi_inc_gate "INCCREASE GATE SIZE (in) TO "
#define vi_inc_gate_why "THE MATERIAL SHORTAGE IS ONE OF THE MAJOR CAUSES FOR\nVOIDS.\nTHE
ACTION, INCREASING GATE SIZE, IT WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE
SYSTEM PROVIDES THIS SUGGESTED ACTION AS SEQUENCE."
/*THE RESOLVED STATEMENTS FOR INCREASE COOLING CHANNEL SIZE*/
#define vi_inc_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define vi_inc_cooling_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nIS ONE
OF THE MAJOR CAUSES FOR VOIDS.\nTHE ACTION, DECREASE THE COOLING CHANNEL SIZE, IT WILL
DECREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION
AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define vi_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE
ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define vi_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR FLASHING DEVIATION*/
/*--THE SUGGESTED ACTION FOR USING MAXIMUM CLMAPPING FORCE--*/
#define fl_method_1 "IS THE CLAMPPING FORCE MAXIMUM\nIF NOT, PLAESE USE THE MAXIMUM
CLAMPPING FORCE"
#define fl_method_why_1 "THE INJECTION PRESSURE IS GREATER THAN THE CLAMPPING FORCE\nAND
CAUSES FLASHING DEVIATION\nTHEREFORE, USING THE MAXIMUM CLAMPPING FORCE CAN ENSURE\nTHE
INJECTION PRESSURE DOES NOT EXCEED THE CLAMPPING FORCE"
/*--THE SUGGESTED ACTION FOR USING MINIMUM CLMAPPING FORCE--*/
#define fl_method_2 "PLEASE USE MINIMUM INJECTION SPEED"
#define fl_method_why_2 "THE INJECTION PRESSURE IS PORTATIONALLY WITH THE\nINJECTION SPEED
OF THE MATERIAL\nTHERERFORE, USING THE MINIMUM INJECTION SPEED CAN ENSURE THE\nINJECTION
PRESSURE DOES NOT EXCEED THE CLAMPING FORCE"
/*--THE SUGGESTED ACTION FOR CLEANING MOLD SURFACE--*/
#define fl_method_3 "DOES THE MOLD STICKING OCCUR\nOR DOES FOREIGN CONTAMINATION EXIST\nIF
IT DOES, PLEASE CLEAN THE MOLD SURFACE"
#define fl_method_why_3 "WHEN MOLD STICKING MATERIAL OR FOREIGN CONTAMINATION EXIST,\nTHE
MOLD SURFACES CANNOT CLOSE TIGHTLY CAUSES FLASHING OCCURS.\nTHEREFORE, THE MATERIAL MUST
BE REMOVED"
/*--THE SUGGESTED ACTION FOR CHECK TEMPERATUER INDICATORS--*/
#define fl_method_4 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
```

```
#define fl_method_why_4 "SINCE THE INCORRECT TEMPERATURE READING MISLEADS\nTHE HEATING AMOUNT OF THE MATERIAL, IT MAYBE RESULTS IN\nTHE MATERIAL OVERHEAT AND CAUSES FLASHING DEVIATION\nTHEREFORE, TO CHECK THE TEMPERATURE READING CAN ENSURE\nTHE MATERIAL IS NOT OVERHEATED"
/*--THE SUGGESTED ACTION FOR CHECK PRESSURE INDICATORS--*/
#define fl_method_5 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define fl_method_why_5 "SINCE THE INCORRECT PRESSURE READING MISLEADS THE\nINJECTION PRESSURE. IT MAYBE RESULTS IN THE INJECTION PRESSURE\nEXCEED THE CLAMPPING FORCE AND CAUSES FLASHING DEVIATION\nTHEREFORE, TO CHECK THE PRESSURE READING CAN ENSURE\nTHE INJECTION PRESSURE IS NOT EXCEED THE CLAMPPING FORCE"
/*--THE SUGGESTED ACTION FOR CHECK SCREW POITION INDICATORS--*/
#define fl_method_6 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define fl_method_why_6 "SINCE THE INCORRECT SCREW POSITION READING MISLEADS\nTHE QUANTITY OF THE INJECTION MATERIAL. IT MAYBE RESULTS\nIN THE OVER-CHARGE OF THE INJECTION MATERIAL AND \nCAUSES FLASHING DEVIATION\nTHEREFORE, TO CHECK THE SCREW POSITION READING CAN \nENSURE AN ACCURANCY QUANTITY OF THE INJECTION MATERIAL"
/*--THE SUGGESTED ACTION FOR CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define fl_method_7 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define fl_method_why_7 "SINCE THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\n OF THE MATERIAL. IT MAYBE RESULTS IN THE\nOVER-CHARGE OF THE INJECTION MATERIAL AND CAUSE FLASHING DEVIATIN \nTHEREFORE, TO CHECK THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION MATERIAL"
/*--THE RESOLVED STATEMENTS FOR DECREASE SHOT SIZE--*/
#define fl_dec_shot "DECREASE SHOT SIZE (in) TO "
#define fl_dec_shot_why "SINCE THE MATERIAL OVER-CHARGE IS ONE OF THE MAJOR CAUSES FOR\nFLASHING. DECREASING SHOT SIZE WILL DECREASE THE MATERIAL QUANTITY\nOF THE MATERIAL FILLING THE MOLD. \nTHEREFORE, THE SYSTEM PROVIDES THIS SUGGESTED ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE CUSHION--*/
#define fl_inc_cus "INCREASE CUSHION (in) TO "
#define fl_inc_cus_why "SINCE THE MATERIAL OVER-CHARGE IS ONE OF THE MAJOR CAUSES FOR FLASHING.\nINCREASING CUSHION WILL DECREASE THE MATERIAL QUANTITY\nOF THE MATERIAL FILLING THE MOLD.\nTHEREFORE, THE SYSTEM PROVIDES THIS SUGGESTED ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE BARREL TEMPERATURE--*/
#define fl_dec_bar "DECREASE BARREL TEMPERATURE (deg F) TO "
#define fl_dec_bar_why "SINCE THE MATERIAL TOO FREE TO FLOW IN THE MOLD\nCAVITY IS ONE OF THE MAJOR CAUSES FOR FLASHING.\nTHE ACTION, DECREASING THE BARREL TEMPERATURE, IT WILL INCREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE \nTHE MATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE MOLD TEMPERATURE--*/
#define fl_dec_mold "DECREASE MOLD TEMPERATURE (deg F) TO "
#define fl_dec_mold_why "SINCE THE MATERIAL TOO FREE TO FLOW IN THE MOLD\nCAVITY IS ONE OF THE MAJOR CAUSES FOR FLASHING.\nTHE ACTION, DECREASING THE MOLD TEMPERATURE, IT WILL INCREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE \nTHE MATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE NOZZLE TEMPERATURE --*/
#define fl_dec_noz "DECREASE NOZZLE TEMPERATURE (deg F) TO "
#define fl_dec_noz_why "SINCE THE MATERIAL TOO FREE TO FLOW IN THE MOLD\nCAVITY IS ONE OF THE MAJOR CAUSES FOR FLASHING.\nTHE ACTION, DECREASING THE NOZZLE TEMPERATURE, IT WILL INCREASE \nTHE VISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE \nTHE MATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE INJECTION PRESSURE --*/
#define fl_dec_inj_pre "DECREASE INJECTION PRESSURE (psi) TO "
#define fl_dec_inj_pre_why "SINCE THE EXCEEDED INJECTION PRESSURE IS ONE OF THE \nMAJOR CAUSES FOR FLASHING. \nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE REGRIND RATE--*/
#define fl_inc_reg "INCREASE REGRIND RATE (%) TO "
#define fl_inc_reg_why "SINCE THE MATERIAL TOO FREE TO FLOW IN THE MOLD \nCAVITY IS ONE OF THE MAJOR CAUSES FOR THE FLASHING.\nTHE ACTION, INCREASING THE REGRIND RATE, IT WILL INCREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMENTS FOR INCREASE SCREW SPEED--*/
#define fl_inc_screw "INCREASE SCREW SPEED (rpm) TO "
#define fl_inc_screw_why "SINCE THE MATERIAL TOO FREE TO FLOW IN THE MOLD \nCAVITY IS ONE OF THE MAJOR CAUSES FOR THE FLASHING.\nTHE ACTION, DECREASING THE SCREW SPEED, IT WILL INCREASE THE \nVISCOSITY OF MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL
```

```
FILLING SPEED AND INJECTION PRESSURE IN THE MOLD CAVITY. \nTHEREFORE, THE SYSTEM PROVIDE
THIS ACTION AS SEQUENCE"
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define fl_inc_inj_time "DECREASE INJECTION TIME (sec) TO "
#define fl_inc_inj_time_why "SINCE THE HIGHER PRESSURE GRADIENT IS ONE OF THE MAJOR
CAUSES\nFOR FLASHING. \nTHE ACTION, INCREASE INJECTION TIME WILL DECREASE \nTHE PRESSURE
GRADIENT. \nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR DECREASE GATE SIZE*/
#define fl_dec_gate "DECREASE GATE SIZE (in) TO "
#define fl_dec_gate_why "SINCE THE OVERFLOW IN THE MOLD CAVITY IS ONE OF THE\nMAJOR CAUSES
FOR FLASHING.\nDECREASING GATE SIZE WILL RESOLVE THIS CAUSE\nTHEREFORE, THE SYSTEM PROVIDE
THIS ACTION."
/*THE RESOLVED STATEMENTS FOR INCREASE COOLING CHANNEL SIZE*/
#define fl_inc_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define fl_inc_cooling_why "SINCE THE MATERIAL TEMPERATURE TOO HIGH IN THE MOLD\nCAVITY
WILL DECREASE MATERIAL VISOCITY\nINCREASING COOLING CHANNEL SIZE WILL DECREASE THE
MATERIAL/nTEMPERATURE AND DECREASE THE MATERIAL VISOCITY IN/nTHE MOLD CAVITY.\nTHEREFORE,
THE SYSTEM PROVIDE THIS ACTION."
/*THE RESOLVED STATEMENTS FOR DECREASE RUNNER SIZE*/
#define fl_dec_runner "DECREAS RUNNER SIZE (in) TO "
#define fl_dec_runner_why "SINCE THE OVERFLOW IN THE MOLD CAVITY IS ONE OF THE\nMAJOR
CAUSES FOR FLASHING.\nINCREASE RUNNER SIZE WILL RESOLVE THIS CAUSE\nTHEREFORE, THE SYSTEM
PROVIDE THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR INCREASE VENTING CHANNEL SIZE*/
#define fl_inc_vent "INCREASE VENTING CHANNEL SIZE (in) TO "
#define fl_inc_vent_why "SINCE THE EXCEEDED INJECTION PRESSURE IS THE/nONE OF MAJOR CAUSES
FOR FLASHING.\nINCREASING VENTING CHANNEL WILL DECREASE THE INJECTION
PRESSURE.\nTHEREFORE, THE SYSTEM PROVIDE THIS ACTION AS SEQUENCE"
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define fl_material "PLEASE CHANGE MATERIAL TO ONE WITH A LOWER THERMAL CONDUCTIVITY\nTHE
ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define fl_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"

/*KNOWLEDGE BASE FOR DELAMINATION DEVIATION*/
/*METHOD CORRECTIVE ACTIONS*/
/*CLEAN THE MOLD SURFACE*/
#define de_method_1 "DOES THE MOLD STICKING OCCUR\nOR FOREIGN CONTAMINATION EXIST. IF IT
DOES, PLEASE CLEAN THE MOLD SURFACE"
#define de_method_why_1 "WHEN MOLD SURFACE STICKING OCCUR OR FOREIGN CONTAMINATION
EXIST,\nIT RESULTS IN THE MATERIAL CAN FLOW THROUGH IN THE MOLD CAVITY.\nTHE MATERIAL MUST
BE REMOVED.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*USE THE MAXIMUM INJECTION SPEED*/
#define de_method_2 "PLEASE USE MAXIMUM BOSTER PRESSURE OR INJECTION SPEED"
#define de_method_why_2 "THE INJECTION PRESSURE TOO LOW CAUSES DELAMINATION\nUSING MAXIMUM
BOOSTER PRESSURE OR INJECTION PRESSURE CAN\nINCREASE INJECTION PRESSURE.\nTHEREFORE, THE
SYSTEM PROVIDES THIS ACTION."
/*CHECK TEMPERATUER INDICATORS--*/
#define de_method_3 "PLEASE CHECK THE TEMPERATURE INDICATORS READING\nIS THE TEMPERATURE
READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define de_method_why_3 "THE INCORRECT TEMPERATURE READING MISLEADS THE MATERIAL
TEMPERATURE\n. IT MAYBE RESULTS IN THE MATERIAL TEMPERATURE TOO LOW\nAND CAUSES
DELAMINATION.\nCHECKING THE TEMPERATURE READING CAN ENSURE THE CORRECT \nMATERIAL
TEMPERATURE.\nTHE SYSTEM PROVIDES THIS ACTION."
/*CHECK PRESSURE INDICATORS--*/
#define de_method_4 "PLEASE CHECK THE PRESSURE INDICATORS READING\nIS THE PRESSURE READING
CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define de_method_why_4 "THE INCORRECT PRESSURE READING MISLEADS THE INJECTION PRESSURE\n.
IT MAYBE RESULTS IN THE INJECTION PRESSURE TOO LOW\nAND CAUSES DELAMINATION\nCHECKING THE
PRESSURE READING CAN ENSURE THE CORRECT\nINJECTION PRESSURE.\n THE SYSTEM PROVIDES THIS
ACTION."
/*CHECK THE SCREW OR RAM SPEED INDICATORS--*/
#define de_method_5 "PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING\nIS THE SCREW
OR RAM SPEED READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define de_method_why_5 "THE INCORRECT SCREW SPEED MISLEADS THE MELTEN AMOUNT\nOF THE
MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF THE INJECTION\nMATERIAL AND CAUSES
DELAMINATION\nCHECKING THE SCREW SPEED CAN ENSURE AN ACCURANCY \nAMOUNT OF THE INJECTION
MATERIAL\nTHE SYSTEM PROVIDES THIS ACTION."
/*CHECK SCREW POITION INDICATORS--*/
```

```
#define de_method_6 "PLEASE CHECK THE SCREW POSITION INDICATORS READING\nIS THE SCREW
POSITION READING CORRECTLY\nIF IT IS NOT, PLEASE ADJUST ITS ACCURANCY"
#define de_method_why_6 "THE INCORRECT SCREW POSITION READING MISLEADS THE QUANTITY\nOF
THE INJECTION MATERIAL. IT MAYBE RESULTS IN THE SHORTAGE OF\nTHE INJECTION MATERIAL AND
CAUSES DELAMINATION.\nCHECKING THE SCREW POSITION READING CAN ENSURE AN ACCURANCY
QUANTITY\nOF THE INJECTION MATERIAL.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE OPERATING VARIABLE CORRECTIVE ACTIONS*/
/*INCREASE SHOT SIZE--*/
#define de_inc_shot "INCREASE SHOT SIZE (in) TO "
#define de_inc_shot_why "THE MATERIAL SHORTAGE CAUSES DELAMINATION.\nINCREASING SHOT SIZE
WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED
ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE CUSHION--*/
#define de_dec_cus "DECREASE CUSHION (in) TO "
#define de_dec_cus_why "THE MATERIAL SHORTAGE CAUSES DELAMINATION.\nDECREASING CUSHION
WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED
ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE BARREL TEMPERATURE--*/
#define de_inc_bar "INCREASE BARREL TEMPERATURE (deg F) TO "
#define de_inc_bar_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
DELAMINATION. INCREASING THE BARREL TEMPERATURE WILL DECREASE \nTHE VISCOSITY OF MATERIAL
IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL FILLING SPEED AND INJECTION PRESSURE
IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE MOLD TEMPERATURE--*/
#define de_inc_mold "INCREASE MOLD TEMPERATURE (deg F) TO "
#define de_inc_mold_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
DELAMINATION. INCREASING THE MOLD TEMPERATURE WILL DECREASE \nTHE VISCOSITY OF MATERIAL IN
THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL FILLING SPEED AND INJECTION PRESSURE IN
THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR INCREASE NOZZLE TEMPERATURE --*/
#define de_inc_noz "INCREASE NOZZLE TEMPERATURE (deg F) TO "
#define de_inc_noz_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
DELAMINATION. INCREASING THE NOZZLE TEMPERATURE WILL DECREASE \nTHE VISCOSITY OF MATERIAL
IN THE MOLD CAVITY WHICH CAN INCREASE \nTHE MATERIAL FILLING SPEED AND INJECTION PRESSURE
IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION ."
/*--THE RESOLVED STATEMENTS FOR INCREASE INJECTION PRESSURE --*/
#define de_inc_inj_pre "INCREASE INJECTION PRESSURE (psi) TO "
#define de_inc_inj_pre_why "THE INJECTION PRESSURE TOO LOW CAUSES
DELAMINATION.\nTHEREFORE, THE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE REGRIND RATE--*/
#define de_dec_reg "DECREASE REGRIND RATE (%) TO "
#define de_dec_reg_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
DELAMINATION.\nDECREASING THE REGRIND RATE WILL DECREASE THE \nVISCOSITY OF MATERIAL IN
THE MOLD CAVITY WHICH CAN INCREASE THE \nMATERIAL FILLING SPEED AND INJECTION PRESSURE IN
THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMENTS FOR DECREASE SCREW SPEED--*/
#define de_dec_screw "DECREASE SCREW SPEED (rpm) TO "
#define de_dec_screw_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
DELAMINATION.\nINCREASING THE SCREW SPEED WILL DECREASE THE \nVISCOSITY OF MATERIAL IN THE
MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL FILLING SPEED AND INJECTION PRESSURE IN THE
MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*--THE RESOLVED STATEMETNS FOR INCREASE INJECTION TIME--*/
#define de_inc_cyc_time "INCREASE CYCLE TIME (sec) TO "
#define de_inc_cyc_time_why "THE LOWER PRESSURE GRADIENT BETWEEN THE INJECTION STAGE\nAND
THE COMRESSION STAGEECTION PRESSURE CAUSES DELAMINATION.\nINCREASING CYCLE TIME WILL
INCREASE THE PRESSURE GRADIENT.\nTHE SYSTEM PROVIDES THIS ACTION."
/*THE RESOLVED STATEMENTS FOR INCREASE GATE SIZE*/
#define de_inc_gate "INCCREASE GATE SIZE (in) TO "
#define de_inc_gate_why "THE MATERIAL SHORTAGE IS CAUSES DELAMINATION.\nINCREASING GATE
SIZE WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED
ACTION."
/*THE RESOLVED STATEMENTS FOR DECREASE COOLING CHANNEL SIZE*/
#define de_dec_cooling "INCREASE COOLING CHANNEL SIZE (in) TO "
#define de_dec_cooling_why "THE MATERIAL SOLIDIFICATION PRIOR THE CAVITY FULFILL\nCAUSES
DELAMINATION.\nDECREASING THE COOLING CHANNEL SIZE WILL DECREASE THE \nVISCOSITY OF
MATERIAL IN THE MOLD CAVITY WHICH CAN DECREASE THE \nMATERIAL FILLING SPEED AND INJECTION
PRESSURE IN THE MOLD CAVITY. \nTHE SYSTEM PROVIDES THIS ACTION."
/*THE RESOLVED STATEMENTS FOR INCREASE RUNNER SIZE*/
#define de_inc_runner "DECREAS RUNNER SIZE (in) TO "
#define de_inc_runner_why "THE MATERIAL SHORTAGE CAUSES DELAMINATION.\nINCREASING RUNNER
SIZE WILL INCREASE THE FILLING \nMATERIAL QUANTITY. \nTHE SYSTEM PROVIDES THIS SUGGESTED
ACTION."
```

```
/*THE RESOLVED STATEMENTS FOR CHANGE MATERIAL*/
#define de_material "PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY\nTHE
ORIGINAL CONDUCTIVITY OF THE MATERIAL IS\n5.5*10**-4 cal/sec cm deg C"
#define de_material_caution "CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE
CAREFULLY CONSIDER\nTHE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL
PROPERTIES,\nTHE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
\nCHEMICAL\nPROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE\nPROPERTIES ARE
SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT"
```

APPENDIX C
RESOLUTION PROCEDURES

In this appendix, the completed listing of resolution procedures for the deviation such as flashing, surface ripples, pit marks, splay marks, sink marks, voids, short shots, warpage, distortion, and delamination is presented in C.1 to C.10 respectively. In these resolution procedures, the definition of the declarative knowledge is followed the discussion in section 6-2 to section 6-3. Therefore, these definitions of the declarative knowledge will not be listed in this appendix.

Furthermore, in each corrective action level the deviation is assumed the deviation is not corrected. Also, the resolution results in the level of the operating variable corrective action is assumed "IMPROVED".

C.1 RESOLUTION PROCEDURES OF FLASHING DEVIATION

```
******************************************************************
             BEGIN TO RESOLVE THE DEVIATION FLASHING
******************************************************************

SUGGESTED ACTION:
DOES THE MOLD SURFACE STICK WITH MATERIAL
OR/AND FOREIGN CONTAMINATION
IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
IS THE CLAMPPING FORCE IN MAXIMUM
IF IT IS NOT, PLAESE USE THE MAXIMUM CLAMPPING FORCE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
```

```
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
```

```
*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************
*****************************************************************

SUGGESTED ACTION:
DECREASE SHOT SIZE (in) TO     2.40

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SHOT SIZE (in) TO     2.20

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION PRESSURE (psi) TO  5500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SHOT SIZE (in) TO     2.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE CUSHION (in) TO     0.50

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
```

```
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION PRESSURE (psi) TO  5000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION TIME (sec) TO     5.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION TIME (sec) TO     4.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
```

```
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   170.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED

*****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   160.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
```

```
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SCREW SPEED (rpm) TO    55.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SCREW SPEED (rpm) TO    60.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE REGRIND RATE (%) TO    30.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*****************************************************************

SUGGESTED ACTION:
DECREASE GATE SIZE (in) TO  0.10
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
```

```
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DECREAS RUNNER SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ON WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.2 RESOLUTION PROCEDURES OF SURFACE RIPPLES DEVIATION

```
*****************************************************************
```

```
          BEGIN TO RESOLVE THE DEVIATION SURFACE RIPPLES
*****************************************************************

SUGGESTED ACTION:
IS YOUR HOPPER EMPTY OR NOT
IF IT IS EMPTY, PLAESE ADD THE MATERIAL INTO THE HOPPER

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DOES THE MOLD SURFACE STICK WITH MATERIAL
OR/AND FOREIGN CONTAMINATION
IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
```

```
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
```

```
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SHOT SIZE (in) TO     2.80

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
```

```
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SHOT SIZE (in) TO     3.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
```

```
SUGGESTED ACTION:
INCREASE SHOT SIZE (in) TO     3.20

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   190.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   200.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    27.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    29.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
```

```
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE REGRIND RATE (%) TO     5.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE REGRIND RATE (%) TO     0.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    45.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    40.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
```

```
SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    30.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*****************************************************************

SUGGESTED ACTION:
INCCREASE GATE SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DECREAS RUNNER SIZE (in) TO  0.40
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ON WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
```

```
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

******************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
******************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.3 RESOLUTION PROCEDURES OF PIT MARKS DEVIATION

```
******************************************************************
           BEGIN TO RESOLVE THE DEVIATION SHORT SHOTS
******************************************************************

SUGGESTED ACTION:
IS YOUR HOPPER EMPTY OR NOT
IF IT IS EMPTY, PLAESE ADD THE MATERIAL INTO THE HOPPER

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
DOES THE MOLD SURFACE STICK WITH MATERIAL
OR/AND FOREIGN CONTAMINATION
IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
```

```
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

******************************************************************
```

```
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
**************************************************************
**************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
**************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
**************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED

**************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
**************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO      0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO      8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
```

```
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   190.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   200.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    27.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    29.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE REGRIND RATE (%) TO     5.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
DECREASE REGRIND RATE (%) TO     0.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    45.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    40.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    30.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
```

```
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
****************************************************************

SUGGESTED ACTION:
INCCREASE GATE SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
****************************************************************

SUGGESTED ACTION:
DECREAS RUNNER SIZE (in) TO  0.40
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
****************************************************************

PLEASE CHANGE MATERIAL TO ON WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
```

```
*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.4 RESOLUTION PROCEDURES OF SPLAY MARKS DEVIATION

```
*****************************************************************
           BEGIN TO RESOLVE THE DEVIATION SPLAY MARKS
*****************************************************************

YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
```

```
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE USE A VENTED BARREL
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHANGE A SMALL NOZZLE ORIFICE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************
*****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
```

```
SUGGESTED ACTION:
DECREASE INJECTION PRESSURE (psi) TO  5500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.30

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION PRESSURE (psi) TO  5000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
```

```
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION TIME (sec) TO     5.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE INJECTION TIME (sec) TO     4.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   170.00
```

```
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   160.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    45.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    40.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE DECOMPRESSION TIME(sec) TO     4.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
```

```
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
CUSHION(in) =      0.0000
DECOMPRESSION(sec) =      0.0000
SCREW_SPEED(rpm) =      0.0000
MOLD_TEMPERATURE(F) =      0.0000
INJECTION_TIME(sec) =      0.0000
BARREL_TEMPERATURE(F) =      0.0000
INJECTION_PRESSURE(psi) =      0.0000
NOZZLE_TEMPERATURE(F) =      0.0000
MOLD_CLOSE_TIME(sec) =      0.0000
MOLD_OPEN_TIME(sec) =      0.0000
CYCLE_TIME(sec) =      0.0000
SHOT_SIZE(in) =      0.0000
REGRIND_RATE(%) =      0.0000
INJECTION_SPEED(1MAX2MED3MIN) =      0.0000

*******************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*******************************************************************

SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*******************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*******************************************************************

PLEASE CHANGE MATERIAL TO ON WITH A LOWER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
```

```
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.5 RESOLUTION PROCEDURES OF SINK MARKS DEVIATION

```
*****************************************************************
            BEGIN TO RESOLVE THE DEVIATION SINK MARKS
*****************************************************************

YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
```

```
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE RELOCATE GATE NEARER HEAVY SECTION
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
```

```
SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   170.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   160.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
```

```
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE CUSHION (in) TO     2.80

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
```

```
SUGGESTED ACTION:
INCREASE CUSHION (in) TO     3.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************

SUGGESTED ACTION:
INCREASE CUSHION (in) TO     3.20

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
CUSHION(in) =      0.0000
SHOT_SIZE(in) =      0.0000
NOZZLE_TEMPERATURE(F) =      0.0000
BARREL_TEMPERATURE(F) =      0.0000
MOLD_TEMPERATURE(F) =      0.0000
INJECTION_TIME(sec) =      0.0000
INJECTION_PRESSURE(psi) =      0.0000
MOLD_CLOSED_TIME(sec) =      0.0000
MOLD_OPEN_TIME(sec) =      0.0000
CYCLE_TIME(sec) =      0.0000
DECOMPRESSION(sec) =      0.0000
SCREW_SPEED(rpm) =      0.0000
REGRIND_RATE(%) =      0.0000
INJECTION_SPEED(1MAX2MED3MIN) =      0.0000

*******************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*******************************************************************

SUGGESTED ACTION:
INCCREASE GATE SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
```

```
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.6 RESOLUTION PROCEDURES OF VOIDS DEVIATION

```
*****************************************************************
            BEGIN TO RESOLVE THE DEVIATION SINK MARKS
*****************************************************************

YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
```

```
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE RELOCATE GATE NEARER HEAVY SECTION
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************

*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO      7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   170.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO      8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
******************************************************************
```

```
SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   160.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
NOZZLE_TEMPERATURE(F) =      0.0125
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SHOT SIZE (in) TO     2.80

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SHOT SIZE (in) TO     3.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE SHOT SIZE (in) TO     3.20

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
```

```
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*****************************************************************

SUGGESTED ACTION:
INCCREASE GATE SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
```

```
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.7 RESOLUTION PROCEDURES OF SHORT SHOTS DEVIATION

```
*****************************************************************
           BEGIN TO RESOLVE THE DEVIATION SHORT SHOTS
*****************************************************************

SUGGESTED ACTION:
IS YOUR HOPPER EMPTY OR NOT
IF IT IS EMPTY, PLAESE ADD THE MATERIAL INTO THE HOPPER

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DOES THE MOLD SURFACE STICK WITH MATERIAL
OR/AND FOREIGN CONTAMINATION
IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
```

```
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
******************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

******************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
******************************************************************
******************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
```

```
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   190.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
```

```
****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   200.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    27.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    29.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
DECREASE REGRIND RATE (%) TO     5.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
DECREASE REGRIND RATE (%) TO     0.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    45.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    40.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE CYCLE TIME (sec) TO    30.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
```

```
*****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*****************************************************************

SUGGESTED ACTION:
INCCREASE GATE SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DECREAS RUNNER SIZE (in) TO  0.40
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ON WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.8 RESOLUTION PROCEDURES OF WARPAGE DEVIATION

```
*****************************************************************
             BEGIN TO RESOLVE THE DEVIATION WARPAGE
*****************************************************************

SUGGESTED ACTION:
PLAESE JIG THE PART AND COOL UNIFORMLY

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
OBSEVER THE EJECTOR PINS MARKS, IF EJECTOR PINS MARKS IS
UNEVEN PLEASE ADJUST THE EJECTOR PINS LOCATION
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
```

```
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE SET UNIFORM TEMPERATURE IN BOTH HAVLES OF MOLD
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE RELOCATE GATE NEARER HEAVY SECTION
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD CLOSED TIME (sec) TO    20.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD CLOSED TIME (sec) TO    22.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   170.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD CLOSED TIME (sec) TO    24.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   160.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
```

```
SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
```

```
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    55.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    60.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
```

```
*****************************************************************

SUGGESTED ACTION:
DECCREASE GATE SIZE (in) TO  0.10
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DECREASE RUNNER SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ON WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************
```

```
IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

C.9 RESOLUTION PROCEDURES OF DISTORTION DEVIATION

```
*****************************************************************
            BEGIN TO RESOLVE THE DEVIATION DISTORTION
*****************************************************************

SUGGESTED ACTION:
PLAESE JIG THE PART AND COOL UNIFORMLY

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
OBSEVER THE EJECTOR PINS MARKS, IF EJECTOR PINS MARKS IS
UNEVEN PLEASE ADJUST THE EJECTOR PINS LOCATION
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE SET UNIFORM TEMPERATURE IN BOTH HAVLES OF MOLD
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE RELOCATE GATE NEARER HEAVY SECTION
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
*****************************************************************
*****************************************************************

SUGGESTED ACTION:
INCREASE MOLD CLOSED TIME (sec) TO    20.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
```

```
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   170.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  6500.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE MOLD TEMPERATURE (deg F) TO   160.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     7.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
```

```
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION TIME (sec) TO     8.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   390.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE BARREL TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
```

```
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************

SUGGESTED ACTION:
DECREASE NOZZLE TEMPERATURE (deg F) TO   380.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    55.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    60.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*******************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*******************************************************************

SUGGESTED ACTION:
DECCREASE GATE SIZE (in) TO  0.10
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

SUGGESTED ACTION:
INCREASE COOLING CHANNEL SIZE (in) TO  0.50
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE
```

```
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DECREASE RUNNER SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*****************************************************************

PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRICAL PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURES THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*****************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*****************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
```

C.10 RESOLUTION PROCEDURES OF DELAMINATION DEVIATION

```
*****************************************************************
           BEGIN TO RESOLVE THE DEVIATION DELAMINATION
*****************************************************************

YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
```

```
DOES THE MOLD STICKING OCCUR
OR FOREIGN CONTAMINATION EXIST. IF IT DOES, PLEASE CLEAN THE MOLD SURFACE
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE TEMPERATURE INDICATORS READING
IS THE TEMPERATURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE PRESSURE INDICATORS READING
IS THE PRESSURE READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
PLEASE CHECK THE SCREW OR RAM SPEED INDICATORS READING
IS THE SCREW OR RAM SPEED READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************
```

SUGGESTED ACTION:
PLEASE CHECK THE SCREW POSITION INDICATORS READING
IS THE SCREW POSITION READING CORRECTLY
IF IT IS NOT, PLEASE ADJUST ITS ACCURANCY
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

****
BEGIN THE OPERATING VARIABLE CORRECTION ACTIONS
****
****

SUGGESTED ACTION:
INCREASE [illegible]ECTION PRESSURE (psi) TO 6500.00
IF YOU [illegible] TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWI[illegible]ESS ANY KEY TO CONTINUE

AFTER TH[illegible]RRECTIVE ACTION
IS YOUR [illegible]TION NOW
1. CORRE[illegible]
2. IMPROV[illegible]
3. UNCHANG[illegible]
[illegible]LEASE IND[illegible] YOUR ANSWER BY CODE NUMBER
[illegible]DE NUMBER[illegible]
[illegible]R ANSWER [illegible]HAT THE DEVIATION HAS BEEN IMPROVED
**********[illegible]***

SUGGESTED ACT[illegible]
INCREASE NOZZ[illegible]PERATURE (deg F) TO 410.00

IF YOU WANT TO [illegible] WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS [illegible]EY TO CONTINUE

AFTER THIS CORRECT[illegible] ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO 190.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO 410.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'

```
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE INJECTION PRESSURE (psi) TO  7000.00
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE NOZZLE TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
****************************************************************

SUGGESTED ACTION:
INCREASE MOLD TEMPERATURE (deg F) TO   200.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
BARREL_TEMPERATURE(F) =      0.0180
****************************************************************

SUGGESTED ACTION:
INCREASE BARREL TEMPERATURE (deg F) TO   420.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
```

```
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    45.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
SCREW_SPEED(rpm) =      0.0032
CUSHION(in) =      0.0000
BARREL_TEMPERATURE(F) =      0.0000
MOLD_TEMPERATURE(F) =      0.0000
NOZZLE_TEMPERATURE(F) =      0.0000
INJECTION_PRESSURE(psi) =      0.0000
INJECTION_TIME(sec) =      0.0000
MOLD_CLOSED_TIME(sec) =     -0.0000
MOLD_OPEN_TIME(sec) =      0.0000
CYCLE_TIME(sec) =      0.0000
DECOMPRESSION(sec) =      0.0000
SHOT_SIZE(in) =      0.0000
REGRIND_RATE(%) =      0.0000
INJECTION_SPEED(1MAX2MED3MIN) =      0.0000

*****************************************************************

SUGGESTED ACTION:
DECREASE SCREW SPEED (rpm) TO    40.00

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
CUSHION(in) =      0.0000
SCREW_SPEED(rpm) =      0.0000
BARREL_TEMPERATURE(F) =      0.0000
MOLD_TEMPERATURE(F) =      0.0000
NOZZLE_TEMPERATURE(F) =      0.0000
INJECTION_PRESSURE(psi) =      0.0000
INJECTION_TIME(sec) =      0.0000
```

```
MOLD_CLOSED_TIME(sec) =    -0.0000
MOLD_OPEN_TIME(sec) =     0.0000
CYCLE_TIME(sec) =     0.0000
DECOMPRESSION(sec) =     0.0000
SHOT_SIZE(in) =     0.0000
REGRIND_RATE(%) =     0.0000
INJECTION_SPEED(1MAX2MED3MIN) =     0.0000

*****************************************************************

SUGGESTED ACTION:
DECREASE CUSHION (in) TO     0.10

IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. IMPROVED
3. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION HAS BEEN IMPROVED
CUSHION(in) =     0.0000
SCREW_SPEED(rpm) =     0.0000
BARREL_TEMPERATURE(F) =     0.0000
MOLD_TEMPERATURE(F) =     0.0000
NOZZLE_TEMPERATURE(F) =     0.0000
INJECTION_PRESSURE(psi) =     0.0000
INJECTION_TIME(sec) =     0.0000
MOLD_CLOSED_TIME(sec) =    -0.0000
MOLD_OPEN_TIME(sec) =     0.0000
CYCLE_TIME(sec) =     0.0000
DECOMPRESSION(sec) =     0.0000
SHOT_SIZE(in) =     0.0000
REGRIND_RATE(%) =     0.0000
INJECTION_SPEED(1MAX2MED3MIN) =     0.0000

*****************************************************************
BEGIN THE MOLD CORRECTION ACTIONS
*****************************************************************

SUGGESTED ACTION:
INCCREASE GATE SIZE (in) TO  0.20
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
*****************************************************************

SUGGESTED ACTION:
DECREAS RUNNER SIZE (in) TO  0.40
IF YOU WANT TO KNOW WHY TO TAKE THIS ACTION PLEASE ENTER '?'
OTHERWISE PRESS ANY KEY TO CONTINUE


AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED
```

```
*******************************************************************
BEGIN THE MATERIAL VARIABLE CORRECTION ACTIONS
*******************************************************************

PLEASE CHANGE MATERIAL TO ONE WITH A HIGHER THERMAL CONDUCTIVITY
THE ORIGINAL CONDUCTIVITY OF THE MATERIAL IS
5.5*10**-4 cal/sec cm deg C
CAUTION: WHEN YOU CHANGE THE MOLDED MATERIAL, PLEASE CAREFULLY CONSIDER
THE ALTERED MATERIAL PROPERTIES SUCH AS THE MECHANICAL PROPERTIES,
THE ELECTRIC PROPERTIES, THE OPTICAL PROPERTIES, AND THE
CHEMICAL
PROPERTIES. CHECKING THESE PROPERTIES ENSURE THAT THESE
PROPERTIES ARE SUITABLE FOR THE FUNCTIONAL PERFORMANCE OF THE PRODUCT

AFTER THIS CORRECTIVE ACTION
IS YOUR DEVIATION NOW
1. CORRECTED
2. UNCHANGED
PLEASE INDICATED YOUR ANSWER BY CODE NUMBER
CODE NUMBER = 2
YOUR ANSWER IS THAT THE DEVIATION WAS UNCHANGED

*******************************************************************
THERE IS NO FURTHER CORRECTION ACTION AVAIABLE.
PLEASE CONSULT WITH THE MOLDING EXPERT
OR THE RAW MATERIAL SUPPLIER TO RESOLVE THE PROBLEM
*******************************************************************

IF YOU WANT TO RESELECT THE MATERIAL, PLEASE ENTER '1'
IF YOU WANT TO RESELECT THE MANUFACTURER, PLEASE ENTER '2'
IF YOU WANT TO RESELECT THE GRADE, PLEASE ENTER '3'
IF YOU WANT TO REDEFINE THE DEVIATION, PLEASE ENTER '4'
OTHERWISE, TO STOP THE PROGRAM PLEASE ENTER '0'
YOUR CHOICE IS = 0
STOP THE PROGRAM
```

APPENDIX D
SIMULATION RESULTS

In this study, the operating variable corrective actions has been simulated in mold filling package, MOLDFLOW. The summary of simulation results for *operating conditions A*, *operating condition B*, *operating conditions C*, *operating condition D*, and *operating condition E*, which have been defined in chpater severn, are listed in D.1, D.2, D.3, and D.4 respectively.

D.1 Simulation Results of Operating Condition A

```
MFL4 4.0.4    SUMMARY

*************************************************
*                                               *
*  USER SPECIFIED PARAMETERS  -  FULL ANALYSIS  *
*                                               *
*************************************************

Cooling Results
===============
Cooling Result File(s)          : NONE

Model Information
=================
MODEL (master) file name : jason
     39 nodes,     highest no.=    39
     48 ".TRI"el. highest no.=    48
     48 elements, highest no.=    48
Maximum aspect ratio of    2.003 at element :    47
Minimum aspect ratio of    1.902 at element :    33
Average aspect ratio of triangular elements :    1.940

Material Information
====================

STANDARD database accessed is version $V14_0_0
PERSONAL database NOT available


Material DATABASE type  : MATDB <STANDARD>
SUPPLIER/file name      : HOECEL
GRADE code              : HC240
Material MODEL order    : 2
Material description    :
HC240    POM     CELCON M90 GP MI=9        HOECHST CEL VI(200)280 PPI       MAR90
Conductivity          BTU/ft/hr/degF        0.213
Specific Heat         BTU/lb/degF           0.448
Density               lb/cu.ft             76.724
Freeze Temperature    deg.F               258.800
No Flow Temperature   deg.F               291.200

Viscosity
=================
```

Temperature deg.F	Shear Rate 1/s	Viscosity poise
356.000	1000.000	3263.800
392.000	100.000	7854.200
392.000	1000.000	2802.800
392.000	10000.000	727.800

```
       428.000            100.000             6760.500
       428.000           1000.000             2400.000
```

Injection Node(s)

Number of injection nodes : 1
Injection node numbers :
39

Molding Conditions

Mold temperature	:	180.000 deg.F
Melt Temperature	:	400.000 deg.F
Injection time	:	6.000 sec
Total volume	:	1.800 cu.inch
Flow rate	:	0.300 cu.in/s

Analysis Options

Material Compressibility	:	10.0 % (at 14500.0 psi)
Maximum injection pressure	:	72518.8 psi
Maximum clamp force	:	2204.6 sh.ton
Nominal injection profile	:	

% Shot vol.	% Nom. flow rate
100	100

```
   100-##################################################
       |#                                               #
 %     |#                                               #
    80 -#                                               #
 F     |#                                               #
 L     |#                                               #
 O  60 -#                                               #
 W     |#                                               #
       |#                                               #
 R  40 -#                                               #
 A     |#                                               #
 T     |#                                               #
 E  20 -#                                               #
       |#                                               #
       |#                                               #
     0  ---------|---------|---------|---------|---------|
                20        40        60        80        100

                             % SHOT VOLUME
```

Constant Pressure at Shot Volume : 98.00 %

Output Options

No. intermediate results files	:	0
Number of time series nodes	:	0

Numerical Parameters

Pressure Iteration Limit	:	100
Iteration Tolerance	:	0.005000
Viscosity Iteration Limit	:	10
Iteration Tolerance	:	0.005000

Result file format Option : PATRAN

Result File Name : joanne1

```
*************************************************
*                                               *
*         FILLING PHASE RESULTS SUMMARY         *
*                                               *
*************************************************
```

Maximum - Minimum Values

```
========================
Max Pressure      ( at Fill )        :    313.5946 psi
Max Pressure      ( during cycle ) :    313.5946 psi

Max Clamp Force   ( during cycle ) :      0.3026 sh.ton
Total projected area                 :      6.0000 sq.inch

Actual injection time                :      6.0098 sec

Min Temperature   ( at Fill )        :    378.5591 deg.F
Max Temperature   ( at Fill )        :    400.0000 deg.F

Min Temperature   ( flow front )     :    378.5591 deg.F
Max Temperature   ( flow front )     :    400.0000 deg.F

Max Shear Rate    ( at Fill )        :     65.6810 1/s
Max Shear Rate    ( during cycle ) :     79.3045 1/s

Max Shear Stress ( at Fill )         :     12.2596 psi
Max Shear Stress ( during cycle ) :     13.8381 psi

Max Solidification Time (Tri. Elements):     68.19 sec ( Element   48 )
Min Solidification Time (Tri. Elements):     19.38 sec ( Element    8 )

Throughputs
===========

   Node          Throughput
                  [cu.inch]
   39              1.803

Output files produced
=====================

No. Intermediate Results File(s)   :     0
No. of Time Series Result File(s)  :     0
Result file(s) produced            : joannel.mnr
                                   : joannel.mer
Summary File                       : joannel.msu

Actual Injection Profile
========================

  % Shot vol.    % Nom. flow rate
      100            100
      100             99
      100             96

   100-##################################################
       |#                                                #
%      |#                                                #
   80 -#                                                 #
F      |#                                                #
L      |#                                                #
O  60 -#                                                 #
W      |#                                                #
       |#                                                #
R  40 -#                                                 #
A      |#                                                #
T      |#                                                #
E  20 -#                                                 #
       |#                                                #
       |#                                                #
    0  ----------|---------|---------|---------|---------|
                20        40        60        80        100

                          % SHOT VOLUME

Analysis Display Summary
========================

 :----------------------------------------------------------------------:
 :           :         :      Flow rate        :            :            :
 :           :         :-----------------------:            :            :
 :    Time   : Volume  :   Actual   : Nominal  :  Pressure  : Clamp force:
```

sec	%	cu.in/s	%	psi	sh.ton
0.35268	5.88	0.300	100.00	56.20	0.00
0.75647	12.60	0.300	100.00	106.10	0.02
1.14009	18.99	0.300	100.00	155.68	0.05
1.56833	26.12	0.300	100.00	205.73	0.10
1.57323	26.20	0.300	100.00	210.03	0.10
2.03945	33.96	0.300	100.00	223.51	0.12
2.54083	42.30	0.300	100.00	241.61	0.14
2.58516	43.04	0.300	100.00	244.79	0.15
3.06490	51.02	0.300	100.00	259.83	0.17
3.08828	51.41	0.300	100.00	262.86	0.18
3.65289	60.80	0.300	100.00	278.26	0.21
3.65678	60.86	0.300	100.00	279.95	0.21
4.27746	71.19	0.300	100.00	286.81	0.22
4.32775	72.03	0.300	100.00	288.44	0.23
4.94846	82.35	0.300	100.00	295.67	0.24
5.00485	83.29	0.300	100.00	297.25	0.25
5.64793	93.98	0.300	100.00	304.60	0.27
5.67614	94.45	0.300	100.00	306.08	0.27
6.00767	99.97	0.300	100.00	313.59	0.29
6.00980	100.00	0.288	96.09	313.59	0.30

* Change over to Pressure Control at 100.0 % Shot Volume
Pressure = 313.595 psi

Execution Times

EXECUTED 15-JUL-92 12:13
COMPLETED 15-JUL-92 12:18

D.1 Simulation Results of Operating Condition B

MFL4 4.0.4 SUMMARY

```
*************************************************
*                                               *
*  USER SPECIFIED PARAMETERS  -  FULL ANALYSIS  *
*                                               *
*************************************************
```

Cooling Results

Cooling Result File(s) : NONE

Model Information

MODEL (master) file name : jason
39 nodes, highest no.= 39
48 ".TRI"el. highest no.= 48
48 elements, highest no.= 48
Maximum aspect ratio of 2.003 at element : 47
Minimum aspect ratio of 1.902 at element : 33
Average aspect ratio of triangular elements : 1.940

Material Information

STANDARD database accessed is version $V14_0_0
PERSONAL database NOT available

Material DATABASE type : MATDB <STANDARD>
SUPPLIER/file name : HOECEL
GRADE code : HC240
Material MODEL order : 2
Material description :
HC240 POM CELCON M90 GP MI=9 HOECHST CEL VI(200)280 PPI MAR90
Conductivity BTU/ft/hr/degF 0.213

Specific Heat	BTU/lb/degF	0.448
Density	lb/cu.ft	76.724
Freeze Temperature	deg.F	258.800
No Flow Temperature	deg.F	291.200

Viscosity

Temperature deg.F	Shear Rate 1/s	Viscosity poise
356.000	1000.000	3263.800
392.000	100.000	7854.200
392.000	1000.000	2802.800
392.000	10000.000	727.800
428.000	100.000	6760.500
428.000	1000.000	2400.000

Injection Node(s)

Number of injection nodes : 1
Injection node numbers :
39

Molding Conditions

Mold temperature	:	180.000 deg.F
Melt Temperature	:	400.000 deg.F
Injection time	:	5.000 sec
Total volume	:	1.800 cu.inch
Flow rate	:	0.360 cu.in/s

Analysis Options

Material Compressibility : 10.0 % (at 14500.0 psi)
Maximum injection pressure : 72518.8 psi
Maximum clamp force : 2204.6 sh.ton
Nominal injection profile :

% Shot vol.	% Nom. flow rate
100	100

```
   100-###################################################
       |#                                                #
  %    |#                                                #
    80 -#                                                #
  F    |#                                                #
  L    |#                                                #
  O 60 -#                                                #
  W    |#                                                #
       |#                                                #
  R 40 -#                                                #
  A    |#                                                #
  T    |#                                                #
  E 20 -#                                                #
       |#                                                #
       |#                                                #
     0  ----------|---------|---------|---------|---------|
                 20        40        60        80       100

                          % SHOT VOLUME
```

Constant Pressure at Shot Volume : 98.00 %

Output Options

No. intermediate results files : 0
Number of time series nodes : 0

Numerical Parameters

Pressure Iteration Limit	:	100
Iteration Tolerance	:	0.005000
Viscosity Iteration Limit	:	10

```
Iteration Tolerance                  : 0.005000

Result file format Option            : PATRAN
=========================

Result File Name                     : joanne2
=========================

************************************************
*                                              *
*         FILLING PHASE RESULTS SUMMARY        *
*                                              *
************************************************

Maximum - Minimum Values
========================

Max Pressure     ( at Fill )       :    351.2690 psi
Max Pressure     ( during cycle )  :    351.2690 psi

Max Clamp Force  ( during cycle )  :      0.3390 sh.ton
Total projected area               :      6.0000 sq.inch

Actual injection time              :      5.0091 sec

Min Temperature  ( at Fill )       :    382.4384 deg.F
Max Temperature  ( at Fill )       :    400.0000 deg.F

Min Temperature  ( flow front )    :    382.4384 deg.F
Max Temperature  ( flow front )    :    400.0000 deg.F

Max Shear Rate   ( at Fill )       :     78.9486 1/s
Max Shear Rate   ( during cycle )  :     95.4572 1/s

Max Shear Stress ( at Fill )       :     13.6996 psi
Max Shear Stress ( during cycle )  :     15.4439 psi

Max Solidification Time (Tri. Elements):     69.27 sec ( Element   48 )
Min Solidification Time (Tri. Elements):     19.55 sec ( Element    8 )

Throughputs
===========

   Node          Throughput
                  [cu.inch]
   39              1.804

Output files produced
=====================

No. Intermediate Results File(s)  :     0
No. of Time Series Result File(s) :     0
Result file(s) produced           : joanne2.mnr
                                  : joanne2.mer
Summary File                      : joanne2.msu

Actual Injection Profile
========================

  % Shot vol.    % Nom. flow rate
      100           100
      100            99
      100            96

   100-##################################################
       |#                                               #
 %     |#                                               #
   80 -#                                                #
 F     |#                                               #
 L     |#                                               #
 O 60 -#                                                #
 W     |#                                               #
       |#                                               #
 R 40 -#                                                #
 A     |#                                               #
 T     |#                                               #
```

```
E 20 -#                                              #
      |#                                             #
      |#                                             #
   0  ----------|---------|---------|---------|---------|
               20        40        60        80       100

                         % SHOT VOLUME
```

Analysis Display Summary

Time sec	Volume %	Flow rate Actual cu.in/s	Flow rate Nominal %	Pressure psi	Clamp force sh.ton
0.29374	5.87	0.360	100.00	62.80	0.01
0.63014	12.60	0.360	100.00	118.66	0.03
0.95012	18.99	0.360	100.00	174.05	0.06
1.30745	26.13	0.360	100.00	229.78	0.11
1.31085	26.19	0.360	100.00	234.49	0.11
1.70019	33.97	0.360	100.00	249.68	0.13
2.11780	42.30	0.360	100.00	269.97	0.16
2.15431	43.03	0.360	100.00	273.67	0.16
2.55510	51.03	0.360	100.00	290.54	0.19
2.57359	51.40	0.360	100.00	293.85	0.20
3.04518	60.81	0.360	100.00	311.08	0.23
3.04739	60.86	0.360	100.00	312.98	0.23
3.56576	71.20	0.360	100.00	320.86	0.25
3.60594	72.00	0.360	100.00	322.52	0.25
4.12529	82.37	0.360	100.00	331.07	0.27
4.17052	83.27	0.360	100.00	332.43	0.28
4.70851	94.00	0.360	100.00	341.15	0.30
4.72988	94.43	0.360	100.00	342.35	0.30
5.00803	99.98	0.360	100.00	351.27	0.33
5.00913	100.00	0.346	96.19	351.27	0.33

* Change over to Pressure Control at 100.0 % Shot Volume
Pressure = 351.269 psi

Execution Times

EXECUTED 15-JUL-92 12:33
COMPLETED 15-JUL-92 12:38

D.1 Simulation Results of Operating Condition C

MFL4 4.0.4 SUMMARY

```
**************************************************
*                                                *
*  USER SPECIFIED PARAMETERS  -  FULL ANALYSIS   *
*                                                *
**************************************************
```

Cooling Results

Cooling Result File(s) : NONE

Model Information

```
MODEL (master) file name : jason
     39 nodes,     highest no.=    39
     48 ".TRI"el. highest no.=    48
     48 elements, highest no.=    48
Maximum aspect ratio of    2.003 at element :    47
Minimum aspect ratio of    1.902 at element :    33
Average aspect ratio of triangular elements :     1.940
```

Material Information

STANDARD database accessed is version $V14_0_0
PERSONAL database NOT available

Material DATABASE type : MATDB <STANDARD>
SUPPLIER/file name : HOECEL
GRADE code : HC240
Material MODEL order : 2
Material description :
HC240 POM CELCON M90 GP MI=9 HOECHST CEL VI(200)280 PPI MAR90

Conductivity	BTU/ft/hr/degF	0.213
Specific Heat	BTU/lb/degF	0.448
Density	lb/cu.ft	76.724
Freeze Temperature	deg.F	258.800
No Flow Temperature	deg.F	291.200

Viscosity

Temperature deg.F	Shear Rate 1/s	Viscosity poise
356.000	1000.000	3263.800
392.000	100.000	7854.200
392.000	1000.000	2802.800
392.000	10000.000	727.800
428.000	100.000	6760.500
428.000	1000.000	2400.000

Injection Node(s)

Number of injection nodes : 1
Injection node numbers :
39

Molding Conditions

Mold temperature : 180.000 deg.F
Melt Temperature : 390.000 deg.F
Injection time : 5.000 sec
Total volume : 1.800 cu.inch
Flow rate : 0.360 cu.in/s

Analysis Options

Material Compressibility : 10.0 % (at 14500.0 psi)
Maximum injection pressure : 72518.8 psi
Maximum clamp force : 2204.6 sh.ton
Nominal injection profile :

% Shot vol.	% Nom. flow rate
100	100

```
   100-#####################################################
       |#                                                  #
 %     |#                                                  #
    80 -#                                                  #
 F     |#                                                  #
 L     |#                                                  #
 O  60 -#                                                  #
 W     |#                                                  #
       |#                                                  #
 R  40 -#                                                  #
 A     |#                                                  #
 T     |#                                                  #
 E  20 -#                                                  #
       |#                                                  #
       |#                                                  #
     0 ----------|---------|---------|---------|---------|
                20        40        60        80        100

                          % SHOT VOLUME
```

```
Constant Pressure at Shot Volume  :  98.00 %

Output Options
==============

No. intermediate results files     :    0
Number of time series nodes        :    0

Numerical Parameters
====================

Pressure Iteration Limit           :      100
Iteration Tolerance                : 0.005000
Viscosity Iteration Limit          :       10
Iteration Tolerance                : 0.005000

Result file format Option          : PATRAN
=========================

Result File Name                   : joannec
=========================

*************************************************
*                                               *
*          FILLING PHASE RESULTS SUMMARY        *
*                                               *
*************************************************

Maximum - Minimum Values
========================

Max Pressure     ( at Fill )        :    366.2473 psi
Max Pressure     ( during cycle ) :      366.2473 psi

Max Clamp Force  ( during cycle ) :        0.3529 sh.ton
Total projected area                :      6.0000 sq.inch

Actual injection time               :      5.0095 sec

Min Temperature  ( at Fill )        :    373.3936 deg.F
Max Temperature  ( at Fill )        :    390.0000 deg.F

Min Temperature  ( flow front )     :    373.3936 deg.F
Max Temperature  ( flow front )     :    390.0000 deg.F

Max Shear Rate   ( at Fill )        :     78.9557 1/s
Max Shear Rate   ( during cycle ) :       95.4387 1/s

Max Shear Stress ( at Fill )        :     14.3146 psi
Max Shear Stress ( during cycle ) :       16.1360 psi

Max Solidification Time (Tri. Elements):     65.98 sec ( Element   48 )
Min Solidification Time (Tri. Elements):     18.64 sec ( Element    8 )

Throughputs
===========

   Node          Throughput
                  [cu.inch]
   39               1.804

Output files produced
=====================

No. Intermediate Results File(s)  :      0
No. of Time Series Result File(s) :      0
Result file(s) produced            : joannec.mnr
                                   : joannec.mer
Summary File                       : joannec.msu

Actual Injection Profile
========================

  % Shot vol.     % Nom. flow rate
     100              100
     100               99
     100               96
```

```
 100-###################################################
     |#                                                #
 %   |#                                                #
  80 -#                                                #
 F   |#                                                #
 L   |#                                                #
 O 60 -#                                               #
 W   |#                                                #
     |#                                                #
 R 40 -#                                               #
 A   |#                                                #
 T   |#                                                #
 E 20 -#                                               #
     |#                                                #
     |#                                                #
   0  ----------|---------|---------|---------|---------|
               20        40        60        80       100

                         % SHOT VOLUME
```

Analysis Display Summary

Time sec	Volume %	Flow rate Actual cu.in/s	Flow rate Nominal %	Pressure psi	Clamp force sh.ton
0.29375	5.87	0.360	100.00	65.61	0.01
0.63017	12.60	0.360	100.00	123.94	0.03
0.95015	18.99	0.360	100.00	181.75	0.06
1.30751	26.13	0.360	100.00	239.89	0.11
1.31091	26.19	0.360	100.00	244.78	0.12
1.70028	33.97	0.360	100.00	260.59	0.14
2.11792	42.30	0.360	100.00	281.71	0.16
2.15443	43.03	0.360	100.00	285.56	0.17
2.55526	51.03	0.360	100.00	303.12	0.20
2.57376	51.40	0.360	100.00	306.56	0.21
3.04540	60.81	0.360	100.00	324.49	0.24
3.04759	60.86	0.360	100.00	326.47	0.24
3.56604	71.20	0.360	100.00	334.66	0.26
3.60621	72.00	0.360	100.00	336.38	0.26
4.12560	82.37	0.360	100.00	345.26	0.29
4.17084	83.27	0.360	100.00	346.67	0.29
4.70888	94.00	0.360	100.00	355.74	0.31
4.73026	94.43	0.360	100.00	356.98	0.32
5.00842	99.98	0.360	100.00	366.25	0.34
5.00952	100.00	0.346	96.20	366.25	0.35

* Change over to Pressure Control at 100.0 % Shot Volume
Pressure = 366.247 psi

Execution Times

EXECUTED 17-JUL-92 14:35
COMPLETED 17-JUL-92 14:38

D.1 Simulation Results of Operating Condition D

MFL4 4.0.4 SUMMARY

```
**************************************************
*                                                *
*  USER SPECIFIED PARAMETERS  -  FULL ANALYSIS   *
*                                                *
**************************************************
```

Cooling Results

Cooling Result File(s) : NONE

Model Information

```
==================
MODEL (master) file name : jason
     39 nodes,     highest no.=    39
     48 ".TRI"el. highest no.=    48
     48 elements, highest no.=    48
Maximum aspect ratio of    2.003 at element :    47
Minimum aspect ratio of    1.902 at element :    33
Average aspect ratio of triangular elements :     1.940

Material Information
======================

STANDARD database accessed is version $V14_0_0
PERSONAL database NOT available

Material DATABASE type : MATDB <STANDARD>
SUPPLIER/file name     : HOECEL
GRADE code             : HC240
Material MODEL order   : 2
Material description   :
HC240   POM     CELCON M90 GP MI=9       HOECHST CEL VI(200)280 PPI       MAR90
Conductivity         BTU/ft/hr/degF        0.213
Specific Heat        BTU/lb/degF           0.448
Density              lb/cu.ft             76.724
Freeze Temperature   deg.F               258.800
No Flow Temperature  deg.F               291.200

Viscosity
===================

    Temperature           Shear Rate           Viscosity
       deg.F                 1/s                 poise
       356.000            1000.000             3263.800
       392.000             100.000             7854.200
       392.000            1000.000             2802.800
       392.000           10000.000              727.800
       428.000             100.000             6760.500
       428.000            1000.000             2400.000


Injection Node(s)
==================
Number of injection nodes   :   1
Injection node numbers      :
    39

Molding Conditions
===================
Mold temperature           :    180.000 deg.F
Melt Temperature           :    390.000 deg.F
Injection time             :      4.000 sec
Total volume               :      1.800 cu.inch
Flow rate                  :      0.450 cu.in/s

Analysis Options
=================
Material Compressibility          :       10.0 %   (at 14500.0 psi)
Maximum injection pressure        :    72518.8 psi
Maximum clamp force               :     2204.6 sh.ton
Nominal injection profile         :
  % Shot vol.    % Nom. flow rate
      100             100

   100-##################################################
      |#                                                #
 %    |#                                                #
   80 -#                                                #
 F    |#                                                #
 L    |#                                                #
 O 60 -#                                                #
 W    |#                                                #
      |#                                                #
 R 40 -#                                                #
```

```
A    |#                                                        #
T    |#                                                        #
E 20 -#                                                        #
     |#                                                        #
     |#                                                        #
   0  ----------|---------|---------|---------|---------|
               20        40        60        80       100

                        % SHOT VOLUME

Constant Pressure at Shot Volume  :  98.00 %

Output Options
==============

No. intermediate results files    :    0
Number of time series nodes       :    0

Numerical Parameters
====================

Pressure Iteration Limit          :      100
Iteration Tolerance               : 0.005000
Viscosity Iteration Limit         :       10
Iteration Tolerance               : 0.005000

Result file format Option         : PATRAN
=========================

Result File Name                  : joanne6
=========================

*************************************************
*                                               *
*         FILLING PHASE RESULTS SUMMARY         *
*                                               *
*************************************************

Maximum - Minimum Values
========================

Max Pressure     ( at Fill )        :    421.0750 psi
Max Pressure     ( during cycle )   :    421.0750 psi

Max Clamp Force  ( during cycle )   :      0.3987 sh.ton
Total projected area                :      6.0000 sq.inch

Actual injection time               :      4.0087 sec

Min Temperature  ( at Fill )        :    377.2516 deg.F
Max Temperature  ( at Fill )        :    390.0000 deg.F

Min Temperature  ( flow front )     :    377.2534 deg.F
Max Temperature  ( flow front )     :    390.0000 deg.F

Max Shear Rate   ( at Fill )        :     98.9157 1/s
Max Shear Rate   ( during cycle )   :    119.7541 1/s

Max Shear Stress ( at Fill )        :     16.3668 psi
Max Shear Stress ( during cycle )   :     18.4185 psi

Max Solidification Time (Tri. Elements):     67.08 sec ( Element   48 )
Min Solidification Time (Tri. Elements):     18.81 sec ( Element    8 )

Throughputs
===========

   Node          Throughput
                  [cu.inch]
   39               1.805

Output files produced
=====================

No. Intermediate Results File(s)  :     0
No. of Time Series Result File(s) :     0
Result file(s) produced           : joanne6.mnr
```

```
                                          : joanne6.mer
Summary File                              : joanne6.msu
```

Actual Injection Profile

% Shot vol.	% Nom. flow rate
100	100
100	96

```
     100-##################################################
        |#                                                #
 %      |#                                                #
     80 -#                                                #
 F      |#                                                #
 L      |#                                                #
 O   60 -#                                                #
 W      |#                                                #
        |#                                                #
 R   40 -#                                                #
 A      |#                                                #
 T      |#                                                #
 E   20 -#                                                #
        |#                                                #
        |#                                                #
      0  ---------|---------|---------|---------|---------|
                 20        40        60        80       100

                          % SHOT VOLUME
```

Analysis Display Summary

Time sec	Volume %	Flow rate Actual cu.in/s	Flow rate Nominal %	Pressure psi	Clamp force sh.ton
0.23484	5.87	0.450	100.00	74.99	0.01
0.50389	12.59	0.450	100.00	141.86	0.03
0.76017	18.99	0.450	100.00	207.98	0.07
1.04638	26.13	0.450	100.00	274.26	0.13
1.04861	26.19	0.450	100.00	279.75	0.13
1.36086	33.98	0.450	100.00	298.02	0.16
1.69478	42.31	0.450	100.00	322.33	0.19
1.72373	43.03	0.450	100.00	326.75	0.20
2.04482	51.04	0.450	100.00	346.97	0.23
2.05923	51.40	0.450	100.00	350.92	0.24
2.43723	60.82	0.450	100.00	371.54	0.27
2.43834	60.85	0.450	100.00	373.86	0.28
2.85459	71.23	0.450	100.00	383.44	0.30
2.88482	71.98	0.450	100.00	385.29	0.30
3.30196	82.38	0.450	100.00	395.98	0.33
3.33617	83.23	0.450	100.00	397.19	0.33
3.76891	94.02	0.450	100.00	408.15	0.36
3.78395	94.40	0.450	100.00	409.09	0.36
4.00668	99.95	0.450	100.00	420.31	0.40
4.00872	100.00	0.433	96.32	421.08	0.40

* Change over to Pressure Control at 100.0 % Shot Volume
Pressure = 420.310 psi

Execution Times

EXECUTED 15-JUL-92 12:27
COMPLETED 15-JUL-92 12:31

D.1 Simulation Results of Operating Condition E

```
MFL4 4.0.4    SUMMARY

**************************************************
*                                                *
*  USER SPECIFIED PARAMETERS  -  FULL ANALYSIS  *
*                                                *
**************************************************

Cooling Results
===============
Cooling Result File(s)           : NONE

Model Information
=================
MODEL (master) file name : jason
     39 nodes,    highest no.=    39
     48 ".TRI"el. highest no.=    48
     48 elements, highest no.=    48
Maximum aspect ratio of    2.003 at element :    47
Minimum aspect ratio of    1.902 at element :    33
Average aspect ratio of triangular elements :     1.940

Material Information
====================

STANDARD database accessed is version $V14_0_0
PERSONAL database NOT available



Material DATABASE type : MATDB <STANDARD>
SUPPLIER/file name     : HOECEL
GRADE code             : HC240
Material MODEL order   : 2
Material description   :
HC240    POM      CELCON M90 GP MI=9       HOECHST CEL VI(200)280 PPI       MAR90
Conductivity          BTU/ft/hr/degF        0.213
Specific Heat         BTU/lb/degF           0.448
Density               lb/cu.ft             76.724
Freeze Temperature    deg.F               258.800
No Flow Temperature   deg.F               291.200

Viscosity
=========
    Temperature          Shear Rate          Viscosity
       deg.F                1/s                poise
       356.000           1000.000           3263.800
       392.000            100.000           7854.200
       392.000           1000.000           2802.800
       392.000          10000.000            727.800
       428.000            100.000           6760.500
       428.000           1000.000           2400.000


Injection Node(s)
=================
Number of injection nodes  :    1
Injection node numbers      :
    39

Molding Conditions
==================
Mold temperature          :    170.000 deg.F
Melt Temperature          :    390.000 deg.F
Injection time            :      4.000 sec
Total volume              :      1.800 cu.inch
Flow rate                 :      0.450 cu.in/s

Analysis Options
================
Material Compressibility          :        10.0 %   (at 14500.0 psi)
Maximum injection pressure        :     72518.8 psi
```

```
Maximum clamp force               :      2204.6 sh.ton
Nominal injection profile         :
  % Shot vol.     % Nom. flow rate
      100           100

   100-##################################################
      |#                                                #
 %    |#                                                #
   80 -#                                                #
 F    |#                                                #
 L    |#                                                #
 O 60 -#                                                #
 W    |#                                                #
      |#                                                #
 R 40 -#                                                #
 A    |#                                                #
 T    |#                                                #
 E 20 -#                                                #
      |#                                                #
      |#                                                #
    0  ----------|---------|---------|---------|---------|
                20        40        60        80       100

                        % SHOT VOLUME

Constant Pressure at Shot Volume  :  98.00 %

Output Options
==============
No. intermediate results files    :    0
Number of time series nodes       :    0

Numerical Parameters
====================
Pressure Iteration Limit          :      100
Iteration Tolerance               : 0.005000
Viscosity Iteration Limit         :       10
Iteration Tolerance               : 0.005000

Result file format Option         : PATRAN
=========================

Result File Name                  : joannee
=========================
*********************************************
*                                           *
*         FILLING PHASE RESULTS SUMMARY     *
*                                           *
*********************************************
Maximum - Minimum Values
========================
Max Pressure     ( at Fill )      :    422.6382 psi
Max Pressure     ( during cycle ) :    422.6382 psi

Max Clamp Force  ( during cycle ) :      0.4004 sh.ton
Total projected area              :      6.0000 sq.inch

Actual injection time             :      4.0087 sec

Min Temperature  ( at Fill )      :    376.5493 deg.F
Max Temperature  ( at Fill )      :    390.0000 deg.F

Min Temperature  ( flow front )   :    376.5512 deg.F
Max Temperature  ( flow front )   :    390.0000 deg.F

Max Shear Rate   ( at Fill )      :     98.9136 1/s
Max Shear Rate   ( during cycle ) :    119.7543 1/s

Max Shear Stress ( at Fill )      :     16.4138 psi
Max Shear Stress ( during cycle ) :     18.4719 psi

Max Solidification Time (Tri. Elements):      61.81 sec ( Element    48 )
```

Min Solidification Time (Tri. Elements): 17.37 sec (Element 8)

Throughputs

Node	Throughput [cu.inch]
39	1.805

Output files produced

No. Intermediate Results File(s) : 0
No. of Time Series Result File(s) : 0
Result file(s) produced : joannee.mnr
: joannee.mer
Summary File : joannee.msu

Actual Injection Profile

% Shot vol.	% Nom. flow rate
100	100
100	96

```
   100-##################################################
        |#                                                #
%       |#                                                #
    80 -#                                                 #
F       |#                                                #
L       |#                                                #
O   60 -#                                                 #
W       |#                                                #
        |#                                                #
R   40 -#                                                 #
A       |#                                                #
T       |#                                                #
E   20 -#                                                 #
        |#                                                #
        |#                                                #
     0   ----------|---------|---------|---------|---------|
                  20        40        60        80       100

                          % SHOT VOLUME
```

Analysis Display Summary

Time sec	Volume %	Flow rate Actual cu.in/s	Flow rate Nominal %	Pressure psi	Clamp force sh.ton
0.23484	5.87	0.450	100.00	75.21	0.01
0.50389	12.59	0.450	100.00	142.29	0.03
0.76017	18.99	0.450	100.00	208.64	0.07
1.04637	26.13	0.450	100.00	275.20	0.13
1.04862	26.19	0.450	100.00	280.74	0.14
1.36084	33.98	0.450	100.00	299.07	0.16
1.69478	42.31	0.450	100.00	323.47	0.19
1.72375	43.03	0.450	100.00	327.91	0.20
2.04482	51.04	0.450	100.00	348.21	0.23
2.05925	51.40	0.450	100.00	352.18	0.24
2.43721	60.82	0.450	100.00	372.89	0.28
2.43836	60.85	0.450	100.00	375.22	0.28
2.85457	71.23	0.450	100.00	384.83	0.30
2.88485	71.98	0.450	100.00	386.70	0.31
3.30197	82.38	0.450	100.00	397.42	0.33
3.33620	83.23	0.450	100.00	398.64	0.33
3.76892	94.02	0.450	100.00	409.65	0.36
3.78399	94.40	0.450	100.00	410.60	0.37
4.00673	99.95	0.450	100.00	421.87	0.40
4.00875	100.00	0.433	96.32	422.64	0.40

```
 :----------------------------------------------------------------------:
* Change over to Pressure Control at  100.0 % Shot Volume
  Pressure =       421.870 psi

Execution Times
===============
EXECUTED  17-JUL-92  14:36

COMPLETED 17-JUL-92  14:38
```

BIBLIOGRAPHY

Agassant, J.F., H. Alles, S. Philipon, and M. Vincent. 1988. "Experimental and Theoretical Study of the Injection Molding of Thermoplastics Materials," *Polymer Engineering and Science* **28:** 460-468.

Agrawal, A.R., I.O. Pandelidis, and M. Pecht. 1987. "Injection Molding Process Control-A Review," *Polymer Engineering and Science* **27:** 1345-1356.

Agrawal, A.R., and I.O. Pandelidis. 1988. "Observers for Optimal Anticipatory Control of RAM Velocity in Injection Molding," *Polymer Engineering and Science* **28:** 157-164.

Atkinson L., and M. Atkinson. 1990. *Using C, QUE Press*, Carmel, IN.

Austin C. 1990. "Verification of Flow Analysis Software: Theory and Practice," *Proceedings of the 48th Annual Technical Conference of the Society of Plastics Engineers* Dallas **36:**. 209-211.

Bakerdjian, Z., and M.R. Kamal. 1977. "Distribution of Some Physical Properties in Injection Molded Thermoplastics Parts," *Polymer Engineering and Science* **17:** 96-100.

Boldizar, A., and J. Kubat. 1986. "Measurement of Cycle Time in Injection Molding of Filling Thermoplastics," *Polymer Engineering and Science* **26:** 877-885.

Buchanan, B.D., and E.H. Shortliffe. eds. 1984. *Rule-Based Expert System: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA.

Catic I.J. 1979. "Cavity Temperature an Important Parameter in the Injection Molding Process," *Polymer Engineering and Science* **19:** 893-899.

Charniak, E., and D. McDermott. 1987. *Introduction to Artificial Intelligence*, Addison Wesley, New York.

Chen, B.S., and W.H. Liu. 1989. "Numerical Simulation And Experimental Investigation of Injection Molding Filling with Melt Solidification," *Polymer Engineering and Science* **29:** 1039-1050.

Chung, T.S., and M.E. Ryan. 1981. "Analysis of the Packing Stage in Injection Molding," *Polymer Engineering and Science* **21:** 271-275.

Cox, H.W.,and C.C. Mentzer. 1986. "Injection Molding the Effect of Fill Time on Properties," *Polymer Engineering and Science* **26:** 488-498.

Darlington, M.V., A.J. Scott, and A.C. Smith. 1986. "Pressure Losses in the Packing Stage of Injection Molding," *Polymer Engineering and Science* **26:** 1282-1289.

Donovan, R.C. 1971. "A Theoretical Melting Model for a Reciprocating-Screw Injection Molding Machine," *Polymer Engineering and Science* **11:** 361-368.

Donovan, R.C., D.E. Thomas, and L.D. Leversen. 1971. "An Experimental Study of Plasticating in a Reciprocating-Screw Injection Molding Machine," *Polymer Engineering and Science* **11:** 353-360.

DuBois, J.H. 1970. *Plastics History U.S.A.*, Cahners Books, Boston.

DuBois, J.H., and F.W. John. 1974. *Plastics*, 5th edition, Van Nostrand Reinhold, New York.

DuBois, J.H., and W.I. Pribble. 1987. *Plastics Mold Engineering Handbook*, Van Nostrand Reinhold, New York.

Duda, R.O., J. Gaschnig, and P.E. Hart. 1979. *Model Design in the PROSPECTOR Consultant System for Mineral Exploration.* In Michie.

Dym, J.B. 1987. *Injection Molds and Molding*, Van Nostrand Reinhold, New York.

Frados, J. 1976. *Plastics Engineering Handbook*, Van Nostrand Reinhold.

Gogos, C.G., C.-F. Huang, and L.R. Schmidt. 1986. "The Process of Cavity Filling Including the Fountain Flow in Injection Molding," *Polymer Engineering and Science* **26:** 1457-1466.

Greener, J. 1986. "General Consequences of the Packing Phase in Injection Molding," *Polymer Engineering and Science* **26:** 886-892.

Groleau, R.J. 1990 "Testing Flow Analysis Programs Using Empirical Means," *Proceedings of the 48th Annual Technical Conference of the Society of Plastics Engineers* Dallas **36:** 212-217.

Gupta, M., and T.H Kwon. 1990. "3-d Flow Analysis of Non-Newtonian viscous Fluids Using "Enriched" Finite Elements," *Polymer Engineering and Science* **30:** 1420-1430.

Hall, C. 1981. *Polymer Materials an Introduction for Technologists and Scientists*, John Wiley & Sons, New York.

Harry, D.H., and R.G. Parrott. 1970. "Numerical Simulation of Injection Molding Filling," *Polymer Engineering and Science* **10:** 209-214.

Hicks, R., and R. Lee. 1989. *VP-Expert For Business Applications*, Holden-Day Inc., Oakland, CA.

Hu. D. 1988. *C/C++ for Expert System*, MIS Press, Portland, Oregon.

Huilier, D., C. Lenfant, and J. Terrisse. 1988. "Modeling the Packing Stage in Injection Molding of Thermoplastic," *Polymer Engineering and Science* **28:** 1637-1643.

Hyatt, J.W. 1911. "___" *Journal of Industrial and Engineering Chemistry* **6:** 158-159.

Isayev, A.I. et al. 1987. *Injection and Compression Molding Fundaments*, Marcel Dekker, Inc., New York.

Jackson, P. 1990. *Introduction to Expert System*, Addison Wesley, New York.

Jacobi, H.R. 1955. "The Historical Development of Plastics Processing Techniques," *Kunststoffe* **55**: __.

Jacobi, H.R. 1963. *Screw Extrusion of Plastics: Fundamentals and Theory*, Gordon and Breach, New York.

Jan, T.C., and K.T. O'Brien. 1991. "Architecture of an Expert System for Injection Molding Problems," *Proceedings of the 49th Annual Technical Conference of the Society of Plastics Engineers* Montrel **37:** 439-443.

Jan, T.C., and K.T. O'Brien. 1991. "Reducing Surface Defects in the Injection Molding of Thermoplastics Using a Hybrid Expert System," *Proceedings of the 50th Annual Technical Conference of the Society of Plastics Engineers* Detroit **38:** 439-443.

Jan, T.C., and K.T. O'Brien. "A User Friendly Interactive Expert System for the Injection Molding of Engineering Thermoplastics," Accept for Publication, *Advanced Manufacturing Technology.*

Jong, W.R., and K.K. Wang. 1990. "Automatic Interpretation of Weld Lines and Air Traps in Injection Molding by Synthesizing Deterministic and Heuristic Knowledge," *Proceedings of the 48th Annual Technical Conference of the Society of Plastics Engineers* Dallas **36:** 197-202.

Juran, R. 1987. *Modern Plastics Encyclopedia*, McGraw Hill, New York.

Kamal, M.R., and P.C. Lafleur. 1982. "Computer Simulation of Injection Molding," *Polymer Engineering and Science* **22:** 1066-1074.

Kamal, M.R. , and S. Kenig. 1972. "The Injection Molding of Thermoplastics Part 2: Experimental Test of the Model," *Polymer Engineering and Science* **12:** 302-308.

Kamal, M.R., and S. Kenig. 1972. "The Injection Molding of Thermoplastics Part 1: Theoretical Model," *Polymer Engineering and Science* **12:**. 294-301.

Kamal, M.R., Y. Kuo, and P.H. Doan. 1975. "The Injection Molding Behavior of Thermoplastics in The Rectangular Cavities," *Polymer Engineering and Science* **15:** 863-868.

Kroschwitz, J.I. 1987. *Encyclopedia of Polymer Science and Engineering*, John Wiley & Sons, New York.

Lindsay, R.K., B.G. Buchanan, B. G. Feigenbaum, and J. Lederberg. 1980. *Application of Artificial Intelligence for Organic Chemistry: The DENDRAL Project*, McGraw Hill, New York.

Liou, M.J., and N.P Suh. 1989. "Reducing Residual Stresses in Molded Parts," *Polymer Engineering and Science* **29:** 441-449.

Load, H.A., and G. Williams. 1975. "Circular Channels, Part 2, The Transient Flow of Plastics Materials in The Cavities of Injection Molding Dies," *Polymer Engineering and Science* **15:** 569-582.

Luger, G.F., and W.A. Stubblefield. 1989. *Artificial Intelligence and the Design of Expert System*, Benjamin/Cummings, New York.

Manas-Zloczower, I., J.W. Blake, and C.W. Macosko. 1987. "Space-Time Distribution in Filling a Mold," *Polymer Engineering and Science* **27:** 1229-1235.

Manzione, L.T. 1987. *Applications of Computer Aided Engineering in Injection Molding*, Van Nostrand Reinhold, New York.

Martin, J., and S. Oxman. 1988. *Building Expert System: A Tutorial*, Prentice Hall, Englewood Cliffs, NJ.

Mumford, J.K. 1924. *The Story of Bakelite*, Robt L. Stillson Co., New York.

Mylopoulos, J., and H.J. Levesque. 1984. "An Overview of Knowledge Respresentation". In Brodie et al. *On Conceptual Modelling. Spinger-Verlag*, New York.

Oda, K., J.L. White, and E.S. Clark. 1976. "Jetting Phenomena in Injection Mold Filling," *Polymer Engineering and Science* **16:** 585-592.

Ogata, K. 1970. Modern Control Engineering, Prentice-Hall, Inc., Englewood Cliffs, NJ.

Palit, K., and I.-J. Chen. 1990. "The Significance of Rheology in CAE Injection Molding Filling Software," *Proceedings of the 48th Annual Technical Conference of the Society of Plastics Engineers* Dallas **36:** 192-196.

Pandelidis, I.O., and A.R. Argawal. 1988. "Optimal Anticipatory Control of RAM Velocity in Injection Molding," *Polymer Engineering and Science* **28:** 147-156.

Pandelidis, I., and Q. Zou. 1990. "Optimization of Injection Molding Design Part 1: Gate Location Optimization," *Polymer Engineering and Science*, **30:** 873-882.

Pandelidis, I., and Q. Zou. 1990. "Optimization of Injection Molding Design Part 2: Molding Condition Optimization," *Polymer Engineering and Science*, **30:** 883-892.

Parsave, K., and M. Chignell. 1987. *Expert System for Experts*, John Wiley & Sons, New York.

Pople, H.E., J.D. Myers, and R. Miller., 1975. "A DIALOG INTERNIST: A Model of Diagnostic Logic for Internal Medicine", *Proceedings of IJCAI-75*, 849-855.

Rosato, D.V., and D.V. Rosato. 1986. *Injection Molding Handbook,* Van Nostrand Reinhold, New York.

Rubin, I.I. 1972. *Injection Molding Theory and Practice*, John Wiley & Sons, New York.

Sanschagrin, B. 1983. "Process Control of Injection Molding," *Polymer Engineering and Science* **23:** 431-438.

Shafer, D. 1989. *Designing Intelligence Front Ends for Business Software*, John Wiley & Sons, New York.

Siegmann, A., A. Buchman, and S. Kenig. 1982. "Residual Stresses in Polymer 3, The Influence of Injection Molding Process Condition," *Polymer Engineering and Science* **22:** 560-568.

Siegmann, A., S. Kenig, and A Buchman. 1987. "Residual Stresses in Injection Molded Amorphous Polymers," *Polymer Engineering and Science* **27:** 1069-1078.

St. Jacques, M. 1982. "An Analysis of Thermal Warpage in Injection Molding Flat Parts Due to Unbalanced Cooling," *Polymer Engineering and Science* **22:** 241-247.

Tadmor, Z., and C.G. Gogos. 1979. *Principles of Polymer Processing*, John Wiley & Sons, New York.

Tamma, K.K., B.L. Dowler, and S.B. Ralkar. 1988. "Computer Aided Application to Injection Molding Transfinite/Finite Element Thermal/Stress Response Formulations," *Polymer Engineering and Science* **28:** 421-428.

Taylor, W.A. 1987. *What Every Engineer Should Know About Artificial Intelligence*, MIT Press, Cambridge, MA.

Thompson, M., and J.R. White. 1984. "The Effect of a Temperature Gradient on Residual Stresses an Distortion in Injection Molding," *Polymer Engineering and Science* **24:** 227-241.

Thornton, J.S., R.E. Montgomery, C.M. Thompson, and D.A. Dillard. 1988. "Analysis of Interfacial Stresses for Elastomeric Disks in Compression," *Polymer Engineering and Science* **28:** 655-659.

Titomanlio G. 1982. "Estimation of Injection Molding Pressure During Mold Filling," *Polymer Engineering and Science* **22:** 589-593.

Titomanlio, G., S. Piccarolo, and G. Marrucci. 1985. "Analysis of Void Formation in Extruded Bars," *Polymer Engineering and Science* **25:** 91-97.

van Krevelen, D.W. 1972. *Properties of Polymers Correlations with Chemical Structure*, Elsevier Publishing Company, New York.

Waterman, D., 1986. *A Guide to Expert System*. Addison-Wesley, Reading, MA.

Weizenbaum, J. 1976. *Computer Power and Human Reason*. Freeman, San Francisco.

White, J.L. 1975. "Fluid Mechanical Analysis of Injection Mold Filling," *Polymer Engineering and Science* **15:** 44-50.

Williams, G., and H.A. Load. 1975. "Mold Filling Studies for the Injection Molding of Thermoplastics Material Part I, the Flow of Plastics Materials in the Hot/Cold Walled," *Polymer Engineering and Science* **15:** 553-561.

Winston, P.H. 1986 *Artificial Intelligence*, Addison Wesley, New York.

Winston, P.H., and B.K.P. Horn. 1989. *LISP*, Addison Wesley, New York.

Yu, C.J., and J.E. Sunderland. 1992. "Determination of Ejection Temperature and Cooling Time in Injection Molding," *Polymer Engineering and Science* **32:** 191-197.