ABSTRACT

INTEGRATING TOLERANCES IN G AND M CODES
USING NEURAL NETWORKS

by
Vijay Kumar Sundareshan

Continuous integrated solutions from CAD down to the preparation of NC programs were developed in the recent years. However, if tolerances should be considered, the interaction of human experts is still necessary. A way to fill this gap in the production process is shown in this thesis. The study builds a relationship between the given design tolerances and including these tolerances in machining by generating respective G and M codes. The study focuses on physical phenomena and their inter-relationship while manufacturing. For example how the speed of machining, torque, power, depth of cut, etc. influences machining under specified tolerances. Artificial neural networks (ANN) have been used to generate required outputs because of their capability to learn from a given set of data points. Four different kinds of neural networks, as a module, have been used with different kinds of learning rules (algorithms) depending on the type of inputs and outputs. The whole model incorporates retrieval of tolerances from a CAD software and running the algorithms for (i) Dimensional tolerance analysis, (ii) Control of feed rate, spindle speed, depth of cut and cutting forces, (iii) Propagation of errors in multi-stage machining, and (iv) Vectorization of geometrical tolerances. Machining processes would include (i) Milling, (ii) Turning, and (iii) Drilling. Then the corresponding outputs are interpreted and analyzed to generate G and M codes. This study has shown how ANN can revolutionize NC machine manufacturing. A case study illustrates the effectiveness of the proposed method.

Blank Page

INTEGRATING TOLERANCES IN G AND M CODES
USING NEURAL NETWORKS

by
Vijay Kumar Sundareshan

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Industrial Engineering

Department of Industrial and Manufacturing Systems Engineering

January 2003

APPROVAL PAGE

INTERGRATING TOLERANCES IN G AND M CODES
USING NEURAL NETWORKS

Vijay Kumar Sundareshan

Dr. George H. Abdou, Thesis Advisor   Date
Associate Professor of Industrial and Manufacturing Engineering, NJIT

Dr. Kevin J. McDermott, Committee Member   Date
Associate Professor of Industrial and Manufacturing Engineering, NJIT

Dr. Paul G. Ranky, Committee Member   Date
Professor of Industrial and Manufacturing Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Vijay Kumar Sundareshan

Degree: Master of Science

Date: January 2003

Undergraduate and Graduate Education:

- Master of Science in Industrial Engineering,
  New Jersey Institute of Technology, Newark, New Jersey, 2002

- Bachelor of Science in Mechanical Engineering,
  People's Education Society Institute of Technology, Bangalore, India, 2000

Major: Industrial Engineering

This thesis is dedicated to all my inspirations in life, you know who you are.

ACKOWLEDGEMENT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The automatic generation of NC programs as a part of the production process is nowadays state of the art. A number of different systems are available in the market. They can be distinguished into universal programming systems, integrated CAD systems and workshop-systems depending on the degree of integration [1]. However, they still require more or less user interaction to enter technological data as feed, cutting speed, way of clamping, etc. Determination of the process parameters is up to now largely based on empirical knowledge. Attaining specific shape and position tolerances is mostly not predictable. In most cases the selection of cutting parameters will be based on recommendations of the cutting tool manufacturers [2].

On the other hand, modern production processes require more then ever a complete manufacturing control. A minimization of production costs and a maximum of quality is a duty for every enterprise. One must assign tolerances as small as necessary but as large as possible. This basic tolerance principle is valid today more than ever. A system, that is able to support the designer and the manufacturing planner as early as possible in the product planning process, is required. Also on the economic point, interest is great at gaining precise cost relevant information.

The basic process of planning for NC programming is clarified by the scheme represented in Figure 1.1 (according to Mischo). After reading the work-piece data from a CAD system, the work-piece is interpreted knowledge-based. A blank is selected in the next step, the situation of clamping is examined and the applicable technologies are determined. In the last step the actions must be sequenced and details must be

1

determined. A postprocessor now generates the NC code required for manufacturing. In the point of determination of details, the production parameters influencing the accuracy of the result are determined. Some of the systems used in practice ask in this case the user. The user can now select the production parameters according to his experience and knowledge, hoping they will give him the expected results. However, this procedure has only a small benefit in sense of an automation of the production process. Other systems choose all production parameters stand-alone. An adjustment to tolerances defaulted by the design does not happen in this case. The reason for that is also, that possibilities to define tolerances that meet the requirements of a CAD system do not exist in the most systems up to now. An approach to this is the vectorial tolerance representation [4].
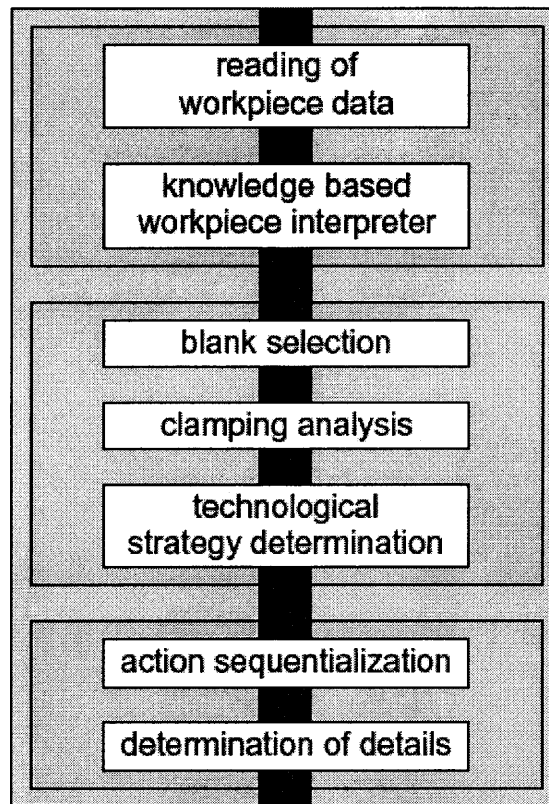
Figure 1.1 A concept for NC planning.
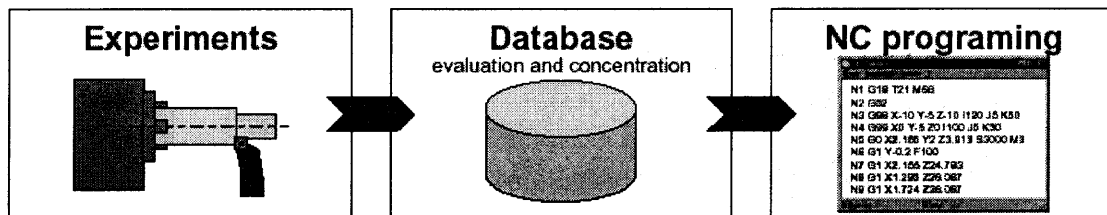
Figure 1.2 Concept for generating NC programs.

In order to gain information about the interdependence between production parameters and attainable accuracy a package of experiments should be carried out. In an experimental circuit from machining - measuring - evaluation (Figure 1.3), technological and economical knowledge can now be collected and prepared. Costs and data structures

of the manufacturing process can correspondingly stored in a database. It is important to separate between random error and systematic error in the step of evaluation. The modification of the NC programs based on the won results closes the experimental circuit. The experimental results in the ANN (Artificial Neural Network) database are now evaluated and summarized.
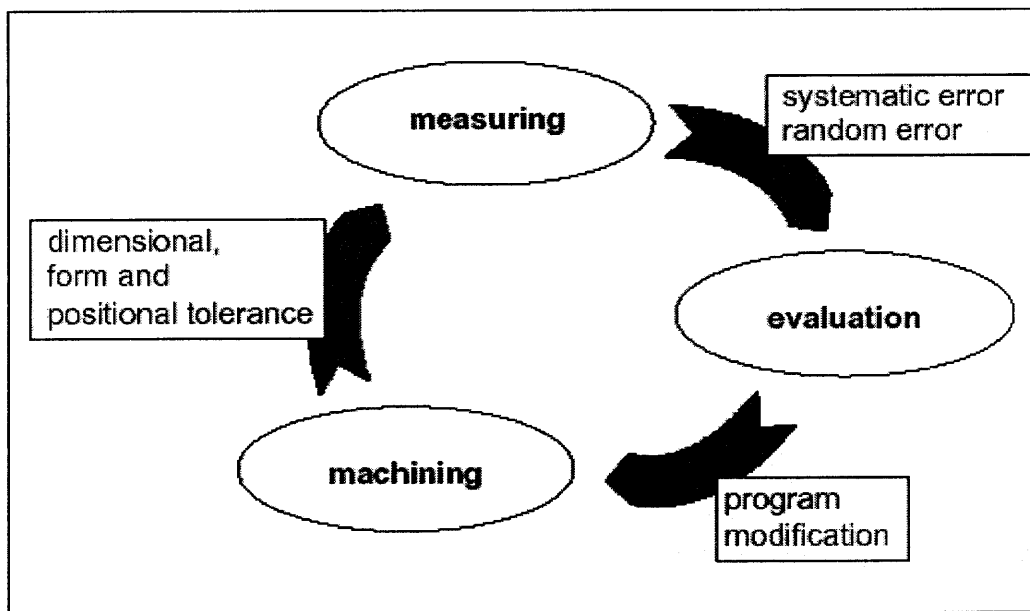
Figure 1.3 Experimental circuit.

In the next step, it is necessary to evaluate and concentrate the collected raw data. Random errors can be extracted in this way. In addition, it is possible to recognize dependencies and regularities (Figure 1.5). In particular the interaction between individual variables must be determined in sense of an effective experimental evaluation.

It is a good idea to structure the data in form of a "feature" and use it to generate

G and M codes for machining. The feature describes the parameters decisive for manufacturing this tolerance.

1. tolerance x
2. average diameter
3. material
4. typ of clamping
5. cutting speed
6. feed
7. classification of the machine
8. ...
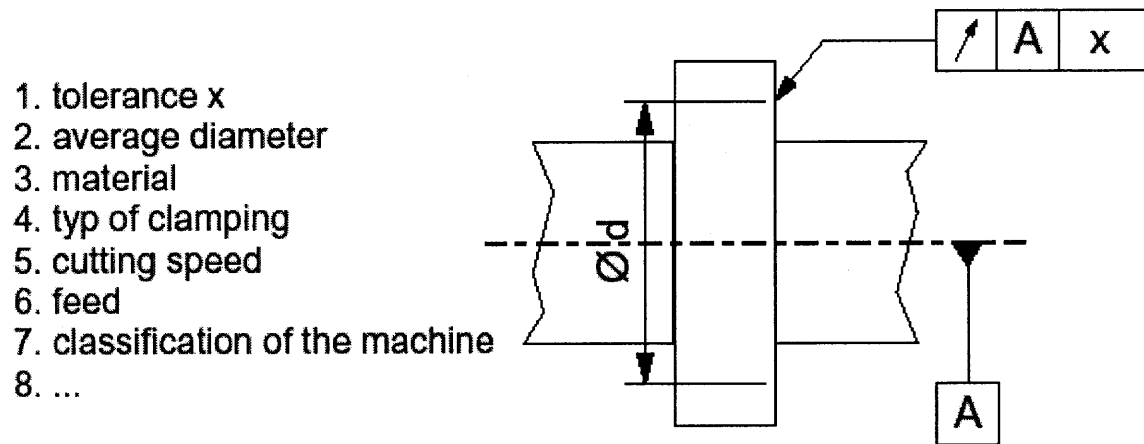
Figure 1.4 Example of Circular Axial Run-out-feature.
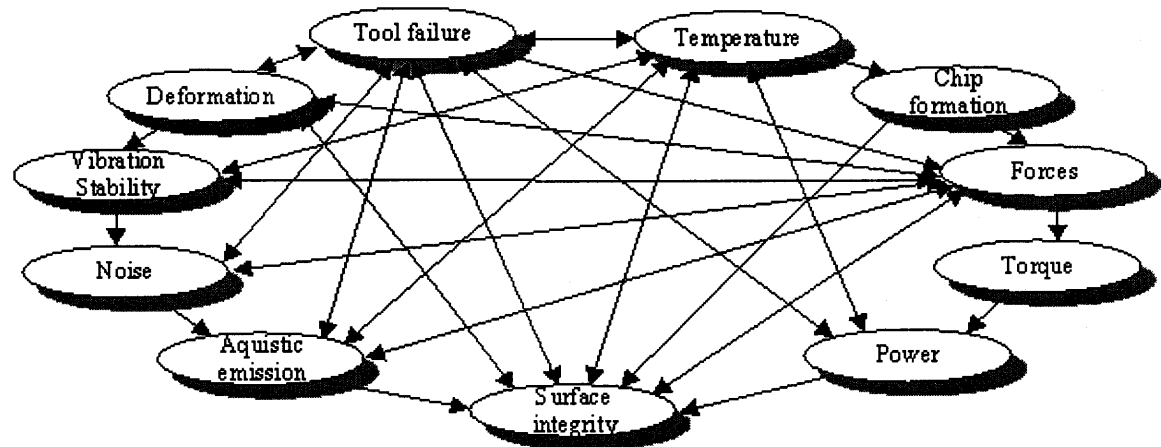
Figure 1.5 Some Physical Phenomena and their Inter-relationship.

CHAPTER 2

LITERATURE REVIEW AND CURRENT PRACTICE

Tolerance is a way to establish within what limits a deviation of measures is allowed to accept a part or a product. There are various methods to calculate the minimal and maximal tolerances with the help of standard tables. So the importance of tolerance combined with tables can be seen. If the designer makes a mistake in the table and the manufacturer makes the part then might get a part, let's say a shaft, which would not fit in the hole resulting in wastage of manpower, material, time and most importantly money.

Typical dimensional tolerances in precision metalworking industries have decreased by three orders of magnitude since World War II, from 0.002" in the 1940s to 20-millionths of an inch in the mid-1990s. Some industries now produce features to tolerances of 2-millionths of an inch. But as dimensional tolerances have decreased, a new quality issue has arisen: Factors of part geometry and surface roughness, once so subtle that they could previously be ignored, are becoming increasingly important. A flat surface that undulates by a mere 10-millionth of an inch may now throw off the accuracy of a part.

Mass production and assembly lines, a vital basis for our modern economy and high standard of living, are founded fundamentally on the subtle concept of producing components within narrow tolerance limits. Tolerances allow for the achievement of acceptable variation in complex assemblies despite variability in the individual components. In fact, Hounshell (1984) shows that it was largely the inability to

manufacture components consistently within narrow limits that held back the widespread use of interchangeable parts and, hence, mass production from its modern beginning in the late eighteenth century to the early twentieth century.

Consider the present day machines used for manufacturing. The latest trend in manufacturing is the NC machines that have reached incredible levels of progress since they first started out in 1951. In recent years the development of NC machines has reached high levels of accuracy and reliability.

But then the available literature concerning CNC programming seems not to comprehend all the problems that can be encountered in such programming. Thus, most studies have focused on the importance of tolerance analysis and have attempted to highlight the significance of tolerance in manufacturing.

Lately CNC machine manufacturers like GE –Fanuc (Chalottesville, Virginia), Makino (Mason, Ohio), Mitsubishi(Vernon Hills, Illinois), etc. are starting to focus on the importance of maintaining the manufactured part within the tolerance limits. These state-of-the-art CNC machine manufacturers have been doing considerable research in high speed machining (HSM), better production, reduction of cycle lead time and not to mention increased accuracy. These machines have the capability to machine directly from the curves, known as the NURBS (Non Uniform Rational B Splines) interpolation. That exchange heretofore has been envisioned as being from one CAD/CAM system to another. But now it also can include an appropriately capable CNC, which brings up the first limitation of linear interpolation—accuracy.

In a conventional NC contour programming process, once the underlying surface geometry is created, the CAM system is going to analyze that geometry, apply a user-specified tolerance and then go about generating line segments joined end-to-end to create the final tool path. The tolerance—sometimes referred to as maximum chordal deviation—means that no point on the line segment will fall further from the reference geometry than the specified value, measured perpendicular to the line segment. So before the part program ever gets out of the CAD/CAM department, the tool path has already deviated from the ideal geometry.

That raises the second, and larger, issue—the tradeoff between accuracy and data volume. If the intent is to machine a smooth contour, a visibly faceted tool path is unacceptable in that it is obviously inaccurate and will require laborious hand finishing that degrades final accuracy even more. So the NC programmer sets the tolerance small to keep the line segments short and make the path as accurate and smooth as possible. But since each line segment corresponds to a block in the part program—expressed as an X-Y-Z coordinate "go-to"—the contouring program swells to enormous size.

Now two more issues arise. First, the gargantuan program probably won't fit in the CNC memory, which necessitates some sort of external memory buffer at the machine tool or DNC link to drip feed the program to the CNC a little at a time. And second, in tight curves large numbers of go-to points will be clustered extremely close together, meaning more blocks for the CNC to process per unit of feed. These two factors form a tag-team that can significantly hamper the speed of the cutting process because the program can't be transferred to the CNC fast enough and/or the blocks can't be executed

fast enough by the CNC to keep up with a desirable programmed feed rate. If the data stream intermittently runs dry for either reason, the feed is going to progress in fits and starts, which degrades surface finish, tool life, and maybe accuracy too if the tool is deflecting in an uncontrollable manner. And so the whole program feed rate is run slower than necessary, or the tolerance is set larger than desired, to avoid these problems.

At least, those are major issues with older controls. Newer high end CNC's have much larger memories, can achieve very high block processing speeds, and apply sophisticated look-ahead capabilities that scan ahead in the program for abrupt changes in cutter path direction. Real-time control algorithms not only see the turns coming, but also lower the feed rate in order to keep the cutter on path and avoid moments of data starvation. These features go a long way to alleviating the accuracy-vs.-data compromise necessitated by linear interpolation. Still, even with these extraordinarily capable CNC's, dense clusters of data points in the part program will significantly reduce the average real feed rate due to block processing limitations and because the control system must execute many abrupt local changes in path direction as it "corners" from each line segment to the next. The accuracies of such capabilities are becoming increasingly well documented in high speed machining applications. Indeed, Siemens claims that "in existing applications, accuracies of 0.5 micron with feeds of 400 ipm were reached" with NURBS interpolation on that company's much-touted 840D control.

But curve interpolation can also be used simply to go flat-out faster than ever before, since the CNC interpolates the original curve at the CNC's interpolation rate. How much faster? It's hard to say at this point, and many machine tool builders are asking

themselves the same question hence getting a tantalizing insight into the issue in the demo area of machine tool builder Makino (Mason, Ohio).

Makino CAD/CAM applications engineer Jeff Wallace believes that NURBS interpolation will result in feed rates being boosted by "at least 30 percent and maybe as much as 50," and has been conducting cutting tests to bear out that hypothesis. To demonstrate, Mr. Wallace showed us the curvy test part being cut on one of Makino's high speed machining centers fit with a GE-Fanuc M C control. To see just how fast the machine could contour the foot-long, P20 piece, the path was programmed at a feed rate of 999 ipm and the feed rate override was set at 200 percent. When planar (X-Z) cuts were first executed in a conventional linear interpolation mode, the actual feed rate topped out at about 650 ipm. Then Mr. Wallace switched over to a NURBS representation of the same path, leaving all other parameters unchanged, and immediately the feed began breaking 1,000 ipm at its fastest moments.

That's not going to happen on just any machine tool, of course. But it does nonetheless illustrate the potential of curve interpolation that has captured the attention of high speed machine builders worldwide.

Some of the company characteristics are highlighted:

1) *GE-Fanuc:* They are one of the leading manufacturers of CNC machines. Whether it's a multi-axis high-speed, high-precision machine, a simple three-axis machine, or a transfer line, GE Fanuc has a control solution to meet the application requirements. From their traditional CNC products to the latest software enhanced OpenFactory systems,

solutions are - faster and easier than ever. Much work has been done especially on NURBS interpolation and in their machines the control reads a very different G-code than that to which machine shops are currently familiar. Rather than the X-Y-Z coordinates of a conventional program block, the NURBS data includes the control points, weights, and knot vectors required to define the curve. The control builder contends that this method of representing curved cutter paths "results in a reduction of program size of 1/10th to 1/100th of a comparable linear interpolation part program and significantly improves the fundamental accuracy issues."

2) *MAKINO*: Makino was the first to bring high-speed spindle technology commercial market. Today, Makino machines and processes are cutting time-to-market for production machinery companies all over the world. With Makino's solutions, not only are the manufacturers cutting metal faster, but you are cutting time to market, too. But cutting time means much more than cutting metal faster. It means having the ability to quickly respond to changing market demands. Makino solutions have that flexibility built in, allowing the company to take on many different parts.

SOFTWARES:
There are different types of software available in the market presently. Let us have a look at a few of them.
CNCez: It is a world class real-time 3-D simulator for computer numerical controlled utilizing the openGL engine for exceptional real time rendering and and 3-D visualization of tool-cuts. Unlike common CNC simulators, it provides real time control over view-point change and light properties. A user-defined pre-processor for customized G-codes and a macro language for custom cycles.

An interactive tool library editor allows the creation and 3-D preview of the customized tools. A tool turret –editor allows direct manipulation of the current tool

turret. An interactive work-piece property allows the user to specify dimensions, material properties, coloring as well as creating material library.

There are numerous other companies that provide CAD/CAM technology like the famous GIBBS system.

GIBBS CAM: Continuing in the GibbsCAM tradition of delivering power, speed and efficiency without sacrificing ease of use, GibbsCAM 2000 incorporates several new enhancements designed to make NC programming faster and easier through powerful functionality. Most notable are the Wizards technology, Generation II Tool-path Engine, CATIA and VDA-FS CAD interoperability options, and the Reporter.

As a summary, these publications have provided insight into the importance of the tolerances in manufacturing as a quality issue. However, there is a lack in respecting the design tolerances when it comes to manufacturing of parts (even complex surfaces) and further work needs to be done.

This research would focus on expanding the horizons in present CNC machines and to enable manufacturers to machine parts within the desired dimensional and geometric tolerances. After exploring various avenues, this research chose ANN as a strong tool to generate NC codes because of its learning abilities.

CHAPTER 3

RESEARCH OBJECTIVES

The research will focus on building a prototype for the inclusion of tolerance in an NC machine.

The proposal is, let us assume a designer allocates dimensions (with the tolerance) for a component, and say 15 mm with tolerance of 0.005 mm and geometric tolerance for angularity of 45 degrees and 0.002 degrees. Subsequently, while manufacturing the manufacturer would be prompted to enter the tolerances in the NC machine. There would be a program that would automatically generate the G and M codes for the NC machine. By the optimal allotment of tolerances a manufacturing unit can accentuate its time, cost and quality. All the three are vital elements for the successful functioning of the unit.

The G and M codes would be generated for the following processes-

1) Milling

2) Turning

3) Drilling

The whole module would try to incorporate the following manufacturing processes for given design tolerance:

1) Machining processes

2) Machines

3) Tools

4) Machining parameters

5) Process capability

CHAPTER 4
PROPOSED METHODOLOGIES

The research would focus on expanding the horizons in present CNC machines and to enable manufacturers to machine parts within the desired dimensional and geometric tolerances. After exploring various avenues, this research chose ANN as a strong tool to generate NC codes because of its learning abilities.

1) The tolerances are obtained from the CAD system by a command IGES out.

2) Once the tolerances are obtained it is fed into a generic algorithm which would convert these files into two parts;

    a) Geometric tolerances

    b) Dimensional tolerances

3) Then these tolerances are fed into a neural network.

4) The philosophy of the neural network can be summarized as follows-

It would have a set of rules within the network. This is done by assigning specific weights to each neuron and that neuron would be activated as per the information received by it. The network would then work out the "error" by comparing the information received with the set of rules already present within the system. Subsequently the network would do the required compensation to the information received to it.

5) The neural network would be an unsupervised type of network. It would consist of three layers, viz.,

a) Input layer

b) Hidden layer

c) Output layer

The input layer consists of input PE (processing elements). These neurons would just receive information from the environment and transfer it to the intermediate PE's in the intermediate layer. It would be done with the use of a sigmoid function.

Next, the intermediate neurons would transfer the weighted information into the output PE's which in turn would transfer it back to the environment, the weighted difference. The output would again have to be fed into a generic algorithm which would change it into G and M codes for CAM machining purposes, these can also be changed into a text file later on if necessary.

The inputs to the neural network would be in the form of sigmoid function. This paper would deal with understanding of the ANN (artificial neural networks) used for generating G and M codes. First , dealing with the basics of ANN and subsequently the use of numerical analysis in understanding the geometry of surfaces and tool path generation.

This thesis would deal with neural networks for the following:

1) Dimensional tolerance analysis and control

2) Control of feed rate, spindle speed, depth of cut and cutting forces.

3) Propagation of errors in multi step machining

4) Geometrical tolerance control using vectorization

The procedures would be discussed in more details in the following chapters.

4.1 Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. Neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. (Figure 4.1)

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown below. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are used, in this *supervised learning*, to train a network.
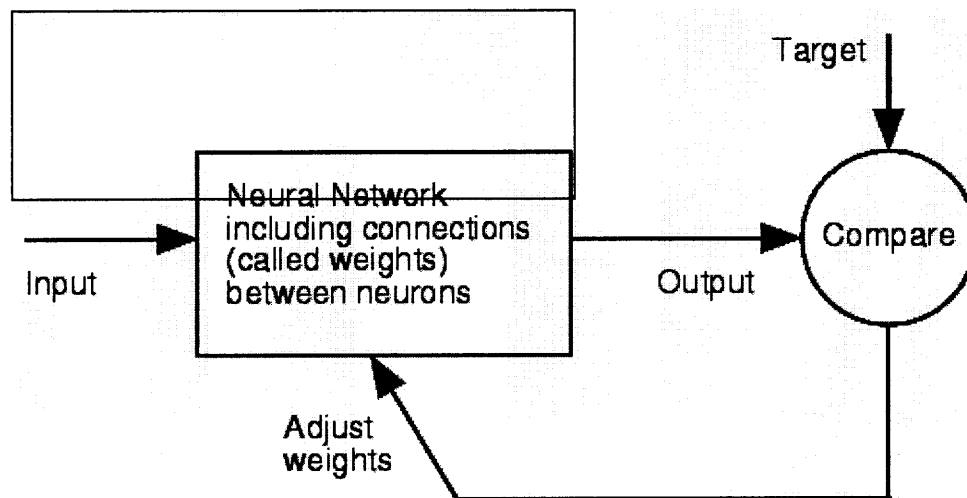
Figure 4.1 Neural Network.

Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as "on line" or "adaptive" training.

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

Today, neural networks can be trained to solve problems that are difficult for conventional computers or human beings. Throughout the toolbox emphasis is placed on neural network paradigms that build up to or are themselves used in engineering, financial and other practical applications.

The supervised training methods are commonly used, but other networks can be obtained from *unsupervised training* techniques or from direct *design* methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make.

The field of neural networks has a history of some five decades but has found solid application only in the past fifteen years, and the field is still developing rapidly.

Thus, it is distinctly different from the fields of control systems or optimization where the terminology, basic mathematics, and design procedures have been firmly established and applied for many years.

4.2 Aspects of Neural Networks in Neural Networks

The first area is the approximation theory. If K is a compact set in R^n, for some n, then it is proved that a semi linear feed-forward any continuous function in C (K) to any required accuracy.

The second area considered is that of learning algorithm. A detailed analysis of an algorithm (the delta rule) will be given. Indeed, computation has inspired considerable advances in this branch of mathematics (Taylor, 1993). The structure of classification space can be analyzed using statistical decision theory (Amari, 1990).

DENSITY AND APPROXIMATION:

It has been proved that a multilayer perceptron can separate any finite sets of points in R^n.

Let us describe this a little more carefully. Let A and B be two finite sets in R^n.
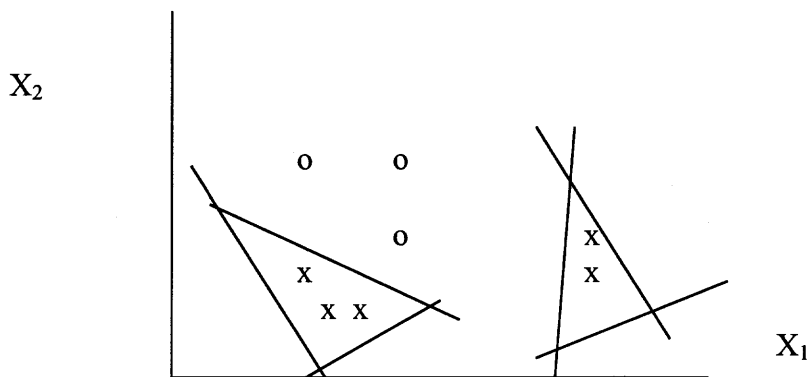
Figure 4.2 Density and Approximation.

'A' might consist of the points labeled "o" and B contains points labeled "x". Assume to wish the network to produce output 1 for the points in A and 0 for points in B. Clearly it is possible to construct a finite set of polygons P1..........Pk such that

$A \subset Q := UP_j$ for j=1,.....,k

Each P_j consists of a finite intersection of half spaces. It can be thus be obtained by a network computing the logical AND function which is linearly separable. The union to include A can then be obtained by a network computing the OR function: OR is also linearly separable. This approach is natural and simple but it is difficult to take it too far. As before, assuming the inputs as vectors in R^n. The output y of the network is a vector in R^m, where m<< n (in many cases m=1). The network then computes a function g: $R^n \rightarrow R^m$ which is regarded as an approximation to some other function f: $R^n \rightarrow R^m$.

"This paper proposes a perceptron with back propagation to calculate any deviations in tool path. It has been proved by Cybenko (1989), Hornik et al (1989) and Funahashi (1989) that one hidden layer is sufficient for approximation. Before explaining the way the neural network functions mathematics of surfaces needs to be understood, since the diagram divides each surface on the part diagram and represents it by an equation."

4.3 Mathematics of surfaces

Looking at the geometry of the surfaces rather closely, as it is the essence of the inputs into the neural networks. This would not only fill in for geometrical tolerances but also dimensional tolerances. But as a foresight there might be a problem for representing intersecting surfaces and complex surfaces, hence this paper stresses on how to correctly represent the surfaces by their equations.

I) <u>INTRODUCTION:</u> Now let us briefly discuss the different forms of representation of surface equations.

The parametric form of a surface equation is

$\underline{P} = \underline{f}(u,v)$

These are vectors and can be represented in multi-dimension. If the value of u is fixed and v is varied, the point P traces a curve in the surface. The partial derivative $\delta p/\delta v$ is a vector tangent to the curve. Similarly, keeping v fixed and u varying then the partial derivative of p w.r.t. u would be a tangent to the above curve. The normal to this plane (also to the surface) can be calculated by the cross product of the partial derivatives.

This defines a curve frame. Similarly, a surface frame with the required tolerance can be defined.

II) SURFACE CURVATURE: The curvature of a surface is rather more complicated than the curvature of the curve, since the curvature of a path across a surface depends on that path as well as on the surface itself. Different paths passing through the same point in different directions will have different normal curvatures, but it is found that the normal curvature takes on the maximum and minimum values for two directions, which are always at right angles to each other. There are two useful scalar measures of curvature. One is mean curvature (J) and gaussian curvature K.

$J = K_1 + K_2 /2 \qquad ; K = K_1.K_2$

Appropriate knowledge of the co-ordinate system and the non-Cartesian co-ordinate system is essential.

III) PARAMETRIC CURVES AND SURFACES: Trying to introduce parametric geometry of free form curves and surfaces, the primary reason to choose parametric representations in this context is that it is possible to express curves and surfaces in terms of linear combinations of scalar functions of the parameters, with vector-valued coefficients. This is appropriate as pointed out by Forrest [16], 'shape is independent of frame of reference'.

Various curves can be defined like,

1) Cubic curves.

2) Cubic B-spline curve.

3) Composite cubic curve.

4) Bezier curve. Etc.

IV)   SURFACE/SURFACE INTERSECTION: As discussed earlier the CAD system (AUTO CAD) has solid modellers in which the object being designed is represented by means of a unified data structure in the computer.(Requicha & Voelcker, Pratt). This structure must contain details of all the face, edges and vertices of the object. This paper proposes the use of different methods for the evaluation of equation of the surfaces and intricacies in the neural network to give the right output for these inputs.

1) Define the surface according to its control limits. Determine whether a given point X,Y and Z lies on the surface. When the surface is implicitly defined as f(X,Y,Z) = 0 it is only necessary to substitute X,Y and Z to determine whether or not they satisfy the equation. In the parametric case, however, points can be computed easily, which do lie on the surface. A search procedure must be used to determine whether X,Y,Z is a point on the surface, but this is clearly much more cumbersome than the implicit function.

2) MARCHING METHOD: This technique generates a sequence of points on or near the required intersection curve by stepping from the current point in a direction controlled by the local differential geometry of the surface or surfaces involved. Jordan, Lennon and Holm [11] developed an algorithm of this kind for directly displaying a curve f(x,y)=0. From a given start point (x_0,y_0) the algorithm steps into one of the neighbouring eight points of the square grid, using the signs of the partial derivatives to select the quadrant moved to, and stepping to the position within that quadrant for which the value of $|f(x + \delta x, y+\delta y)|$ is smallest.

These methods work well for straight lines and conics, but run into troubles with more complex curves. Finding suitable starting points and termination conditions become harder.

A second class of marching methods may be thought of as repeatedly solving a umber of simultaneous equations, one of which is a step constraint. For intersecting surfaces, the surface created by the intersection of these two surfaces can be considered as another surface. A similar concept is used in numerically controlled machining using programs such as APT(Faux and Pratt[6]). Here the cutting tool is driven by in contact

with a part surface and a drive surface which correspond to the two intersecting surfaces, while a sequence of 'pseudo-check surfaces' is used provide step constraints.

Now dealing with how the neural network is going to do such complex mathematical calculations?

The network that this paper is going to study is called 'On interval weighted three-layer neural network'.

In solving application problems, the data sets used to train an ANN may not be hundred percent precise but within certain ranges. Representing data sets with intervals, have interval neural networks. By analyzing the mathematical model, categorize general three-layer ANN training problems into two types. One can be solved by finding numerical solutions of non-linear systems of equations. The other can be transformed into non-linear optimization problems.

Proposed algorithms:

1) For extraction of tolerances from CAD software-

Description of data in tolerance information (Table 4.1).

The below-mentioned variable names would be used in the programs for the tolerance extraction from CAD software.

The processes considered in the research are-

1) Milling

2) Turning

3) Drilling

These processes have a fixed upper and lower dimensional tolerances which are generally used in the industries. [Doyle] (Table 4.2)

Table 4.1 Description of Variables

VARIABLE NAME	DESCRIPTION
N	Total no. surfaces
I	Surface no.
D(I)	Dimension value
T(I)	Dimensional tol. Value
STATE(I)	Type of dimension
X1(I)	X-coordinate of 1st exten. Point
X2(I)	X-coord of 2nd ext. point in X axis
Y1(I)	Y-coord of 1st ext. point I
Y2(I)	Y-coord of 2nd ext. point
SR(I)	Surface roughness
MINCL(I)	Minimum clearance
MAXST(I)	Max straightness
MINST(I)	Min str.
MAXCI(I)	Max circularity
MINCI(I)	Min circ.
MAXCO(I)	Max concentricity
MINCO(I)	Min conc.
MMC(I)	Max mat. Condition
LMC(I)	Least mat. Cond
T_PART	Dim. Tol of total length of part
LENGTH	Total length of part

Table 4.2 List of Tolerances

PROCESS	LTL(in)	UTL (in)	LSR(μ in)	USR(μ in)
Milling	0.00170	0.02000	32	250
Drilling	0.00200	0.10000	63	250
F_turning	0.00170	0.01000	2	16
R_turning	0.01000	0.02000	32	1000

UTL= Upper tolerance limit

LTL= Lower tolerance limit

LSR= Lower surface roughness

USR= Upper surface roughness

ALGORITHM:

1) Input N.

2) Read N.

3) Input I.

4) Read I.

5) Input the dimension and Dim. Tol for I.

6) Read D(I) andT(I).

7) Open "part1DIM.DXF"

8) Read X1,X2,Y1,Y2.

9) Link I and D(I),T(I).

10) Store in DDIM.DXF

11) I=I+1

12) Is I=N

13) If yes goto 15.

14) If no goto 5.

15) End

Now algorithm for geometric tolerances:

1) Input N.

2) Read N.

3) Input I.

4) Read I.

5) Input the dimension and Geometric Tol for I.

6) Read D(I) andT(I).

7) Open "part1DIM.DXF"

8) Read X1,X2,Y1,Y2.

9) Link I and D(I),T(I).

10) Store in GDIM.DXF

11) I=I+1

12) Is I=N

13) If yes goto 15.

14) If no goto 5.

Conditions for geometric tolerances: CSA standards

1) MAXST = 2*T(s) – [MINIMUM HOLE SIZE – MAX SHAFT DIA]

2) MAXCY = 0.5 [MINIMUM HOLE SIZE – MAX SHAFT DIA]

3) MAXCO = [MAX HOLE SIZE – MIN SHAFT DIA]

    MAXCO = [MIN HOLE SIZE- MAX SHAFT SIZE]

4) MMC HOLE = [VIRTUAL CONDITION SHAFT + GEOM. TOL. FOR

    HOLES] …(a)

Equation (a)- Virtual condition shaft implies worst mating boundary.

Conditions for dimensional tolerances : CSA STANDARDS

Table 4.3 CSA standard tolerances

BASIC SIZE(mm)	IT01	IT0	IT1	IT2
0-3	0.0003	0.0005	0.0008	0.0012
3-6	0.0004	0.0006	0.001	0.0015

And so on.

Run CAGT (Computer Assisted Geometric Tolerance). [G.Abdou and R.Cheng]
This module would recreate the part drawing in AUTOCAD with specific reference to
the tolerances.

4.4 Douglas Peucker Method for Line Approximation

This method is suggested for use on complex parts. It is a powerful tool, which can

drastically reduce the generated data curve depending on the error tolerances. The DP

line simplification algorithm was originally proposed in [5]. John Hershberger and Jack

Snoeyink have implemented it in C in [6] as a package named DPSimp:

DPSIMP *is a Douglas-Peucker line simplification algorithm implementation by John*

Hershberger and Jack Snoeyink. They analyze the line simplification algorithm reported

by Douglas and Peucker and show that its worst case is quadratic in n, the number of

input points. The algorithm is based on path hulls, that uses the geometric structure of

the problem to attain a worst-case running time proportional to nlog_2(n), which is the best case of the Douglas algorithm.

STEPS IN DP LINE APPROXIMATION:

1) Generate the data as an array.

2) Approximate the line.

3) Simplify the line.

4) Inject line approximation in original data.

5) Get an array with new points.

6) Store it in DPHULL.h.

ALGORITHM FOR CONVERSION OF DIMENSIONS AND TOLERANCES INTO BINARY FORM :

1) Open DDIM.DXF.

2) Read I,D(I),T(I)

3) Convert D(I),T(I) into binary DIGITS.

4) Store in BIN1.DXF

5) Open GDIM.DXF

6) Read D(I),T(I),I.

7) Convert D(I), T(I) into binary digits.

8) Store in BIN2.DXF.

9) Link BIN1.DXF and BIN.DXF for each I.

10) Store it in BIN3.DXF.

Now the values of BIN3.DXF are fed into the neural network.

4.5 Interpolation

NC machine tools are point driven. Curves (including straight lines) are generated by defining key points on the curve. In the case of lines for example, the end points must be specified. To generate an arc, the end points plus the location of the center point must be specified. Free form curves are generated by specifying control points which may not be on the curve.

The process of generating a curve given specific key points is called interpolation. Most NC control hardware is programmed at the factory to do linear and circular interpolation. Many controllers are able to do helical interpolation (circular motion in X-Y with simultaneous linear in Z). Some controllers are also able to generate tool paths which follow conic sections other than the circle (ellipse, parabola, hyperbola). Special controllers have been designed to generate curves which interpolate cubic and higher order polynomial equations. These latter types of controllers are not common. Higher order curves (all but lines and circles) are generally approximated by straight lines or arc segments.

Standard controllers can be used to generate complex curves using either linear or circular interpolation. In linear interpolation points on the complex curve are first calculated then a program is created which drives the tool from point to point in a straight line. The desired curve can be approximated to any degree of accuracy by using a

sufficient number of points. Circular interpolation is similar except arcs are generated between the calculated points.

LINEAR INTERPOLATION OF CURVES:

Curves may be approximated by a series of straight-line segments. There are two methods for defining the maximum error when using a line to approximate a curve. Both tolerances specify the maximum deviation allowed between the straight line approximating curve and the desired programmed curve.

1. Inner Tolerance: The error is on the part side. Undercutting results from this type of tolerance.

2. Outer Tolerance: The error is on the tool side. Excess stock is left with this type of tolerance.

Both tolerances may be specified and have different values. Either however, not both tolerances may be zero.

If the curve to be approximated is an arc, the following equation can be derived to determine the number of approximating lines needed for a specified inner and/or outer tolerance.

From trigonometry (see figure 2):

$$\cos(\theta / 2) = \frac{(R - t_i)}{(R + t_o)}$$

which yields:

$$\theta = 2\cos^{-1}\frac{(R - t_i)}{(R + t_o)}$$

Example:

How many lines does it require to approximate a 5 in diameter circle to an inner tolerance of .05 inches and an outer tolerance of 0.0 inches?

From the above equation then:

$$\theta = 2\cos^{-1}\frac{(2.5 - .05)}{(2.5 + 0.0)} = 22.96$$

For this circle then the minimum number of chords needed is: 360/22.96 = 15.68 . Since the number of chords must be an integer, the calculated value must be rounded up. The correct answer is 16 chords are needed to approximate this circle to the specified tolerances.

Moving Circle Interpolation: A free form curve may be approximated by creating arcs between given points on the curve. This technique is known as the moving circle interpolation method. Free form curves created by this technique are smooth. They have both zero order (end points touch) and first order (arcs are tangent) continuity.

Given a set of points (p_1 to p_n) to be connected by a smooth curve, moving circle interpolation may be accomplished as follows.

1. Select the first three points (p_1, p_2, p_3) and determine the center of the arc that passes through these points.

2. Construct (machine an arc using G2 or G3) an arc through the first two points (p_1, p_2). Use the center points coordinates for your I and J values.

3. Drop the first point and select the next point (p_4). Determine the center of a second arc passing through these three points (p_2, p_3, p_4).

4. Construct an arc through the first two (p_2, p_3) of the three points. Using the calculated center of the arc as your I and J values again.

5. Repeat this process with the succeeding points.

6. Construct the arc through the last three points.

An Algorithm for Finding the Center of an Arc :

Given: Three points that lie on an arc P_1, P_2, P_3

Find: The center of the specified arc

1. Connect points P_1 and P_2 with a line L_a

2. Connect points P_2 and P_3 with a line L_b

3. Determine the coordinates (P_a, P_b) of the center of each line.

4. Determine the slopes (m_a, m_b) of lines L_a and L_b

5. Determine the slopes of line L_1 and L_2 perpendicular to L_a and L_b passing through points P_a and P_b.

6. Determine the equations for lines L_1 and L_2

LINEAR INTERPOLATION ALGORITHM

To approximate a zero of a continuous function f(x) on an interval [a,b] where f(a) and f(b) have opposite signs.

INPUT:

    a   - left endpoint of interval

    b   - right endpoint of interval

    MAXITS - maximum number of iterations to allow

    TOLER - tolerance to stop iterating

1. input a, b, TOLER, MAXITS

2. set xL = a

   set xR = b

   set yL = f(xL)

   set yR = f(xR)

3. Repeat for N from 1 to MAXITS

    a. set xm = xR - yR * (xR-xL)/(yR-yL)

      set ym = f(xm)

    b. IF ym and yR have opposite signs, then

        set xL = xm

        set yL = ym

      otherwise

        set xR = xm

        set yR = ym

    c. print N, xm, ym, |xR - xL|

    d. IF |xR - xL| < TOLER or |ym| < TOLER, then

        stop (method converged to within tolerance)

      otherwise

      repeat loop 3 until N = MAXITS

4. Stop (maximum number of iterations exceeded).

Two quantities u and v have opposite signs if their product is negative.

CHAPTER 5

FRAMEWORK OF NEURAL NETWORKS

The ANN module used represents the following and gives the desired output which would be useful to generate G and M codes.

- Dimensional tolerance analysis and control.

- Control of feed rate, spindle speed, depth of cut and cutting forces.

- Propagation of errors in multi step machining.

- Geometrical tolerance control using vectorization.

5.1 Optimization of Process Chains by Neural Networks

In this system customer's requirements are handled by quality control charts. These charts may be used not only for the required (e.g. dimensional tolerances, surface finish of the end product) parameters but also for the "internal" ones (e.g. quality of rough cutting). In the paper, process optimization based on adaptive process models trained by reinforcement learning technique and quality control charts is proposed.

Introduction

In the continuously growing international competition quality and cost issues are of fundamental importance. In manufacturing, quality features of products are usually measurable physical parameters. Customer's requirements are specified by quality control charts using tolerances.

36

"INTERNAL" PARAMETERS

Products are manufactured in successive operation steps, e.g. on a production line. Each operation has some input and some output parameters. During an operation the set of the physical parameters of a work-piece are changed. The modification of these parameters can be considered as a parameter flow in the production line. (Figure 5.1.)
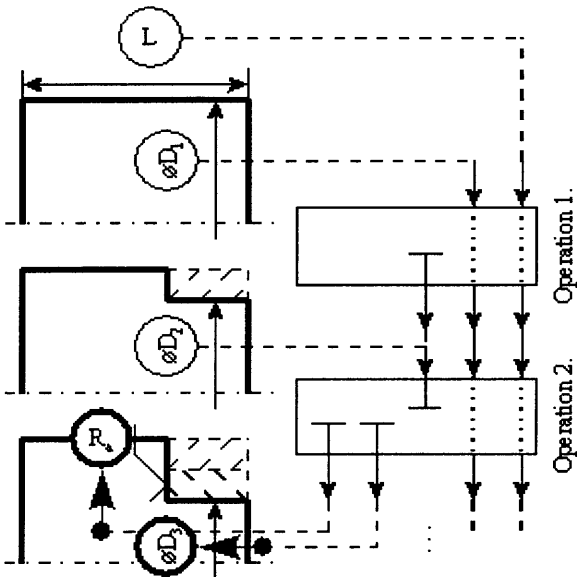

Figure 5.1 "Internal" Parameters (D_2) and End-product Parameters (R_a, D_3).

Output parameters of an operation can be

- Input parameters of the next operation or

- Parameters of the end-product.

Concerning input and output parameters of an operation, the following cases can be distinguished:

- some input parameters are terminated,

- some input parameters become without any changes output parameters,

- new output parameters are generated which were not present at the previous steps of the production.

The customer sets requirements usually on a part of the end-product parameters only. This fact does not follow that only these parameters are to be controlled during production. Other, not directly specified parameters are called as "internal" parameters. To minimize the production cost, these parameters are also subjects of process control.

CONTROL CHART PARAMETERS

This ISO 9000 standard applies quality control charts for describing quality parameters (Pham & Oztemel, 1996). Five of the most important parameters are :

- n - the number of the patterns from which the data of the control charts are calculated

- LTL - lower tolerance of the data

- UTL - upper tolerance of the data

- $C_P = \dfrac{UTL - LTL}{6 \cdot \delta}$

- $C_{PK} = \min\left(\dfrac{UTL - \mu}{3 \cdot \delta}, \dfrac{\mu - LTL}{3 \cdot \delta} \right)$

where μ is the mean value and δ is the deviation of the measured data. This model can be used if the physical parameters follow the normal distribution and if "n" is large

enough and in the case of characteristic data. The above five parameters are subject of control during the production.

OPERATION MODELS - REINFORCEMENT LEARNING

Artificial neuronal networks (ANNs) proved to be applicable for process modeling (Monostori et al., 1996). In batch learning, a set of training data is necessary and used during the learning phase. To get this data set, extensive and costly experiments are to be performed. For continuous learning, the reinforcement learning approach can be advantageously used, which moreover does not need the concrete target values of the network, but only some indication of its functioning (Nuttin et al., 1995).

PROCESS MODELS BASED ON STATISTICAL PARAMETERS OF PHYSICAL QUANTITIES: Usual process models employ values of physical parameters on their inputs and outputs (Westkamper, 1995). Here, the use of their mean values and standard deviations is proposed. By the way, the new model can be created also on the base of the ordinary, value based model, namely in two steps. (Figure 5.2)
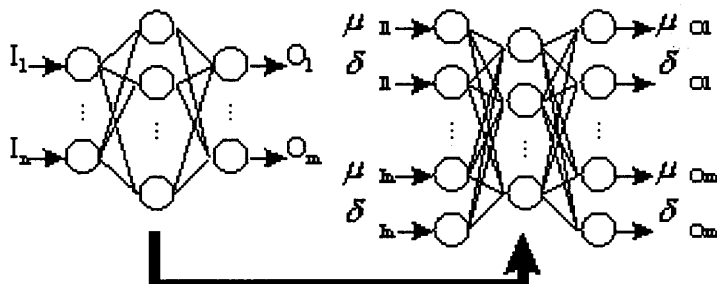

Figure 5.2 Creation of the New Model on the Base of the Ordinary, Value Based Model.

Firstly, the training (input and output) data sets including the above statistical measures are created. Secondly, the new model learns the mapping between these data sets. In the first step the required mean values and deviations are determined. According to these values, input data for the ordinary model are generated. On the base of the outputs of the ordinary model mean values and deviations of the output parameters can be calculated. With this method the demanded training data set of the new model can be built. By learning of this new data set the ANN can be initialized. To increase its precision the reinforcement learning is responsible.

SETTING OF OPERATION AND CHART PARAMETERS:

In order to reach a global optimum,

1. Relations among input, output and operation parameters (e.g. tool type, feed rate) have to be determined for every operation by using artificial neural networks.

2. Right control charts and process parameters have to be selected along the process chain (Osanna et al., 1996).

5.2 Control of Feed Rate, Spindle Speed and Depth of Cut ~~(considering various parameters)~~.

The model introduces a concept for quality-oriented, comprehensive modeling of machining processes. It incorporates a large number of variables grouped into input, output and in-process categories. Fundamental features of the concept are the ability to learn from experience, and the flexibility in realizing various, task dependent mappings with their inherent model building capability.

INTRODUCTION

Very intensive research activities are conducted all over the world for the modeling of machining processes. Process models are considered as abstract representations of processes linking causes and effects or transforming process inputs into outputs. They can be classified in two groups: fundamental or micro models and applied or macro models. Our goal is to develop a framework for applied modeling, which is able to manage the cutting processes in their whole complexity.

Paragraph 2 outlines the complicated relations between some physical phenomena of the cutting process. In paragraph 3 classical models are reviewed. A large number of input and output parameters are listed in paragraph 4, which are needed to handle the multivariable character of the cutting process. ANN model is proposed as the basic element of the cutting model framework described.

PHYSICAL PHENOMENA AND THEIR INTERRELATIONSHIPS

Because of the complicated relationships between the phenomena incorporated into the cutting model, the machining process is hard to be decomposed [11] (Figure 5.3).

have been inaccurate and of limited validity, due to the complexity of the object (for instance: tool wear depends on the independent input parameters and some output parameters such as cutting force variation, process stability etc.) and the limitation of the applied approximation. It must be emphasized, that although the deterministic approach helps to answer and understand the basic principles of metal cutting processes, it is important to develop other methods which is able to handle the complexity of cutting, process uncertainty and are able to transform the information into knowledge [13]. To describe the complete machining system [11], one of the most important questions is to determine the input-output features.

To determine all the important input and output parameters, first, the main groups (Figure 5.4), the relevant parameters and their notations and units were determined. Among the parameters are continuous variables and logical "OR" decisions. The proposed model refers to the tool path length where the cutting parameters are not changed. If some parameters change the model is used appropriately.
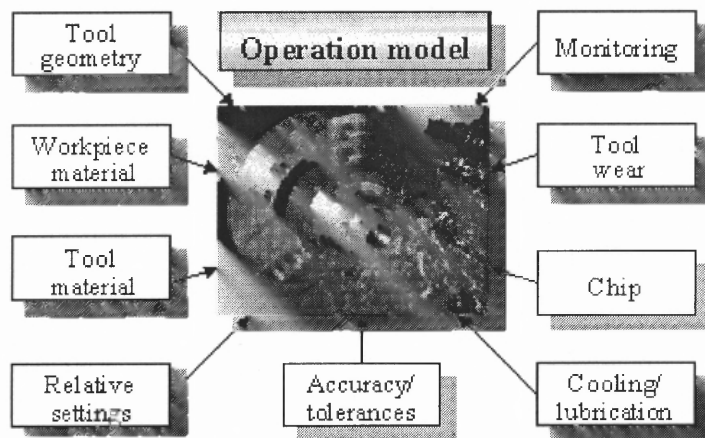

Figure 5.4 Operation Model.

establishing an empirical relationship between the tool life "T" and cutting parameters: cutting speed "v_c", feed "f" and "a" depth of cut (turning operation). For the deterministic tool life model T= f(v_c, f, a) empirically values of the exponents are necessary.

The first shear plane model of the cutting process, (F_c=A*τ_c*[ctgϕ +tg(ϕ +ω)]), which is based on pure theoretical aspects, was developed by Merchant. The generally used cutting force model (F_c=k_1*h^{1-m}*b) developed by Kinzle is based on stress theory and empirical work too. ("k_1", "h", "b" denote the cutting force constant, the chip thickness and the width of chip, respectively).

The models in the applied group may be structured as an exponential empirical formula:

Table 5.1 Linearization of Process Engagement Conditions

The formula:	where the process engagement conditions are:
$I = C \cdot v_c^z \cdot f^x \cdot a_p^y \cdot a_e^m$, or after linearization with logarithm: $I^* = c^* + z \cdot v_c^* + x \cdot f^* + y \cdot a_e^* + m \cdot a_p^*$	$v_{c\,min} < v_c < v_{c\,max}$, $f_{min} < f < f_{max}$, $a_{e\,min} < a_e < a_{e\,max}$, $a_{p\,min} < a_p < a_{p\,max}$ $I_{min} < I < I_{max}$

(Cutting feature "I", depth of cut "a_p", width of cut "a_e") The complex transformation matrix of the linearized model is demonstrated in table 5.1. It must be emphasized that some cutting features (chip breaking, process stability, tool breaking) could not be characterized by this deterministic empirical model. In general, these types of models

have been inaccurate and of limited validity, due to the complexity of the object (for instance: tool wear depends on the independent input parameters and some output parameters such as cutting force variation, process stability etc.) and the limitation of the applied approximation. It must be emphasized, that although the deterministic approach helps to answer and understand the basic principles of metal cutting processes, it is important to develop other methods which is able to handle the complexity of cutting, process uncertainty and are able to transform the information into knowledge [13]. To describe the complete machining system [11], one of the most important questions is to determine the input-output features.

To determine all the important input and output parameters, first, the main groups (Figure 5.4), the relevant parameters and their notations and units were determined. Among the parameters are continuous variables and logical "OR" decisions. The proposed model refers to the tool path length where the cutting parameters are not changed. If some parameters change the model is used appropriately.

Figure 5.4 Operation Model.

Some parameters can be used as input and as output variables, as well. This is the way to follow the changes of these variables along the process. If, e.g., I is one of these variables: I_{input} means the state of the variable before and I_{output} after the cutting process.

The next list shows the parameters incorporated in the investigations.

1. The **tool geometry** group consists of:

   o Micro parameters:

   - Previous machining:

     - Ground: Fine machined "OR" Not fine machined

     - "OR" Not ground.

   - Edge radius ($r\beta$)[μ m]

   o Macro parameters:

   - Monolith:

     - Tool length to be used (l_f)[mm]

     - Group: N "OR" H "OR" W

   - Throw away insert:

     - Positive "OR" Negative

     - Type of chip breaker: None "OR" "OR" PM "OR" PF "OR" PR "OR" MF "OR" MR "OR" QM "OR" QF "OR" QR

     - Inscribe circle diameter (d)[mm]

     - Edge length (l_f)[mm]

     - Insert thickness (S)[mm]

     - Orthogonal rake angle (γ_o)[°]

- Orthogonal clearance angle (α)[°]

- Inclination angle (λ)[°]

- Cutting edge angle (κ)[°]

- Include angle (ε)[°]

- Edge number:

  - Single edge:

    - Corner radius ($r\varepsilon$)[mm]

  - "OR" Multiple edge:

    - Width of fuzette ($r\varepsilon$)[mm]

    - Tool diameter (d_s)[mm]

    - Cutter half cone angle (φ_s)[°]

    - Distance of the corner radius center from the rotation axis (C_r)[mm]

    - Distance of the corner radius center from the tool tip (C_a)[mm]

    - Number of cutting edges (Z_s)[.]

    - Run out - radial - average (μ_r)[mm]

    - Run out - radial - deviation (σ_r)[mm]

    - Run out - axial - average (μ_a)[mm]

    - Run out - axial – deviation (σ_a)[mm]

2. The **work piece material** group consists of:

   o Surface layer:

- Pre-produced: "OR" Casted "OR" Drawn "OR" Rolled "OR" Forged

  - "OR" Machined: Rough "OR" Fined "OR" Finished

  o Heat treatment: Normalized "OR" Tempered "OR" Quenched

  o Ingredients:

    - Impurities (S%)[%]

    - Carbonising:

      - Normal hardening (CN%)[%]

      - Precipitation hardening (CK%)[%]

  o Material parameters:

    - Maximum tensile strength (R_M)[Pas]

    - 0.2 tensile strength $(R_{M\,0.2})$[Pas]

    - Modulus of elasticity (E)[Pas]

    - μ $(\mu$)[Pas]

    - Vickers hardness (HV_{100N})[HV]

    - Impact energy (KC)[KC]

  o Cutting speed constant (C_v)[]

  o Main cutting constant (k_1)[]

  o Main cutting force exponents:

    - (X_F)[]

    - (Y_F)[]

    - (Z_F)[]

3. The **tool material** group consist of:

- o Coating:
  - Not coated
  - "OR" Coated
    - Temperature of the coating: Very law "OR" Low "OR" High
    - Structure of crystallographic:
      - Monocrystal
      - "OR" Policrystal
        - Porosity (VP)[%]
- o Cutting ability:
  - Tool life constant (C_T)[]
  - Tool life exponent (Z_T)[]
- o Ingredients:
  - Impurities (S%)[%]
  - Carbonizing:
    - Normal hardening (CN%)[%]
    - Precipitation hardening (CK%)[%]
- o Material parameters:
  - Maximum tensile strength (R_M)[Pas]
  - 0.2 tensile strength ($R_{M\,0.2}$)[Pas]
  - Modulus of elasticity (E)[Pas]
  - μ (μ)[Pas]
  - Vickers hardness (HV_{100N})[HV]

- Impact energy (KC)[KC]

4. The **relative setting** group consists of:

   o tool path length (L)[mm]

   o Surface first curvature of the work piece (ρ_1)[1/mm]

   o Surface second curvature of the work piece (ρ_2)[1/mm]

   o Immersion (contact) angle (φ)[°]

   o Depth of cut (tool axis direction) (a_p)[mm]

   o Depth of cut (perpendicular to the tool axis) (a_e)[mm]

   o Velocity (cutting speed along the ρ_1) (v_c)[m/sec]

   o Velocity (cutting speed along the ρ_2) (v_f)[m/sec]

   o Velocity (cutting speed along d_s) (v_s)[m/sec]

   o Single edge:

      - Feed per work piece revolution (f)[mm]

   o Multiple edge:

      - Feed per tool revolution (f)[mm]

5. The **accuracy/tolerances** group consists of:

   o positioning accuracy projected to the first surface curvature (V_{P1})[mm]

   o positioning accuracy projected to the second surface curvature (V_{P2})[mm]

   o Main spindle run-out (radial) (e_{rad})[μ m]

   o Main spindle run-out (axial) (e_{ax})[μ m]

   o Average of the surface curvature ρ_1 along machining length (μ_{GM1})[1/mm]

   o Deviation of the surface curvature ρ_1 along machining length (σ_{GM1})[1/mm]

- o Average of the surface curvature ρ_2 along machining length (μ_{GM2})[1/mm]

- o Deviation of the surface curvature ρ_2 along machining length (σ_{GM2})[1/mm]

- o Surface roughness along ρ_1 (R_{a1})[μ m]

- o Surface roughness along ρ_2 (R_{a2})[μ m]

6. The **cooling/lubrication** group consist of:

- o No cooling

- o "OR" Cooling

  - Solid

    - Graphite: There is "OR" There is no graphite

    - "OR" Sulphides: There is "OR" There is no sulphide

    - "OR" Plastic material: There is "OR" There is no plastic material

  - "OR" Fluid

    - Media – coolant: Water "OR" Oil "OR" Spirit "OR" Others

    - Ingredients - lubrication: "OR" Oils "OR" Petroleum "OR" Graphite "OR" Sulphite

    - Cooling method:

      - Mist

        - Pressure (Pl)[Pas]

        - Volume (V_e)[m^3]

        - Volume rate (Ql)[m^3/sec]

      - "OR" Flooding

- Pressure (Pl)[Pas]

- Volume rate (Ql)[m^3/sec]

- "OR" Inside

- Pressure (Pl)[Pas]

- Volume rate (Ql)[m^3/sec]

- Media volume divided by ingredient – ratio (V%)[%]

- Gas

  - Media – coolant: Air "OR" Nitrogen

  - Ingredients - lubrication: "OR" Oils "OR" Petroleum

  - Media volume divided by ingredient – ratio (V%)[%]

7. The **chip group** consists of:

   o Chip thickness:

     - Theoretical chip thickness (h)[mm]

     - Theoretical maximum of the chip thickness ($h_{c\ max}$)[mm]

     - Measured chip thickness (h)[mm]

     - Measured maximum of the chip thickness ($h_{c\ max}$)[mm]

   o Chip form:

     - chip ratio (space for chip/theoretical volume of the chip) (K)[]

8. The **tool-wearing** group consists of:

   o Wearing:

     - Average flank wear (VB)[mm]

     - Maximum flank wear (VB_{max})[mm]

     - Total removed volume by this tool (V_c)[mm^3]

- o Tool breakage: Broken "OR" Not broken

9. The **monitoring group** consist of:

- o Force:

  - Along ρ_1:

    - Alteration (max-min) (ΔF_c)[N]

    - Trend (inclination of the line) (m F_c)[]

    - Average (μF_c)[N]

  - Along ρ_2:

    - Alteration (max-min) (ΔF_f)[N]

    - Trend (inclination of the line) (m F_f)[]

    - Average (μF_f)[N]

  - Normal force:

    - Alteration (max-min) (ΔF_p)[N]

    - Trend (inclination of the line) (m F_p)[]

    - Average (μF_p)[N]

- o Cutting power:

  - Cutting power on the main spindle:

    - Alteration (max-min) (ΔP_c)[W]

    - Trend (inclination of the line) (m P_c)[]

    - Average (μP_c)[W]

  - Cutting power on the feed engine:

    - Alteration (max-min) (ΔP_f)[W]

    - Trend (inclination of the line) (m P_f)[]

- Average (μ P_f)[W]

- o Temperature:

  - Alteration (max-min) (Δ T)[C°]

  - Trend (inclination of the line) (mT)[]

  - Average (μ T)[C°]

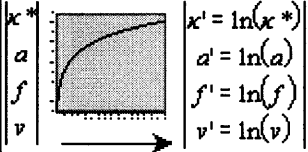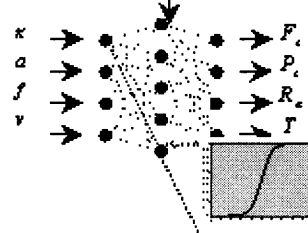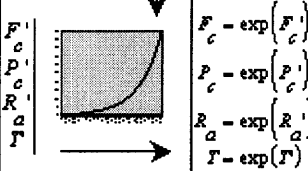## COMPARISON AMONG CUTTING PROCESS MODELLING METHODS

Physical/empirical approach: Theoretical recognition and empirical experience determine this type of basic models. Their coefficients are defined with the help of multiple regression calculations. The model structure used can be regarded as input for the regression calculation as well as the basic experimental data.

Neural network approach: In the field of neural networks various net structures and training methods are used. Neural networks possess most of the following characteristics [14]:

- o powerful parallel computing and mapping structure,

- o strong abilities of learning and self-organization,

- o strong abilities to store and retrieve knowledge by content rather than by address,

- o feasibility for hardware implementation and real-time control,

- o few prior assumptions or specific requirements for modeling.

Table 5.2 Machining Process Modeling

TRANSFORMATIONS OF THE CUTTING MODELS

Learning capability is the reason why a neural network based cutting model is proposed

in the paper. There are techniques as well to transform the knowledge of one of these

models to the other model and vice-versa. One useful knowledge transformation method

can be done with the help of input, output data pairs. If one of these models and the

boundaries (min and max bounds of the parameters) of its use are given, a set of input-

output data pairs can be calculated. Based on these data pairs:

o   Structure and weights of the ANNs can be learnt,

o   empirical function fitting can be calculated by minimal squares method,

5.3 Framework of an ANN based cutting model

The proposed neural network based cutting model has input and output parameters from the data set presented above. It is to be seen that there are two types of parameters: decision variables (e.g. whether there is or there is no cooling) and continuous variables (e.g. Young modulus).

ANNs can successfully handle continuous variables. To handle the validity of an ANN model the possible intervals for each parameter have to be given. A set of minimum and maximum vector pairs can be used to determine the validity of an ANN.

In the case of a single vector pair: one of the vectors consists of minimum and the other of maximum values of parameters. The ANN is useful when each variable of the input vector - given by the user - is above the related minimum and below the related maximum parameters of the given data pair.

But one vector pair determines only one field of validity that's why the storing of a set of min. and max. data pairs is needed to determine several fields of validity.

The use of the ANN based cutting model.
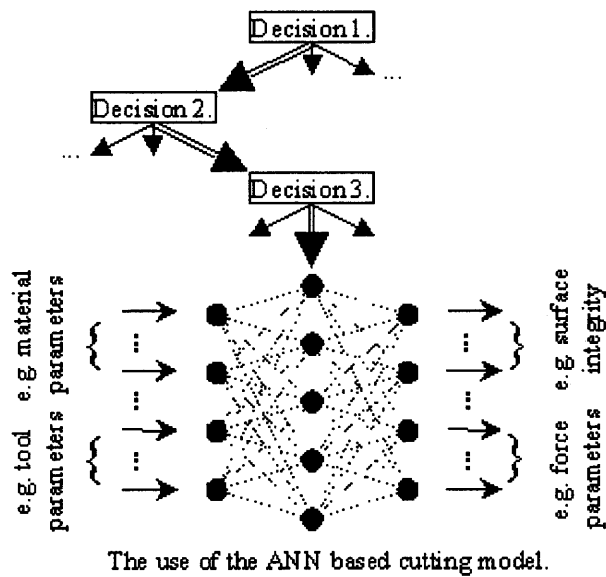
Figure 5.5 The use of Neural Network based cutting model.

To build up this model new data sets have to be given to be learnt by the ANN. The building up process consists of three steps:

1. Determination of the related ANN, based on the decision variables of the new data set.

2. ANN learning, based on the new data set, which consists of data from previous learning and the new data set given by the user.

3. Storing of:

    1. the enlarged min. and max. limits of the cutting model validity

    2. the data pairs used for learning.

The use of the proposed cutting model involves three steps (Figure 5.):

4. Determination of the relevant input, output variables, the related ANN and the limit of the use of this ANN, based on the decision variables. This step is a selection of a leaf on a tree built by the decision variables.

5. Information of the user if the model could be used on the parameter field requested by her/him. The model is valid if there is a single vector pair among the set of minimum and maximum vectors where the ANN is valid.

6. The ANN estimation of the related output variables based on the given input variables.

Figure 5.6 Function Approximation Using Neural Networks.

This model is large to manage the whole cutting process by a large number of decisions and continuous input and output variables, but at one factory, usually, only a part of this model is needed.
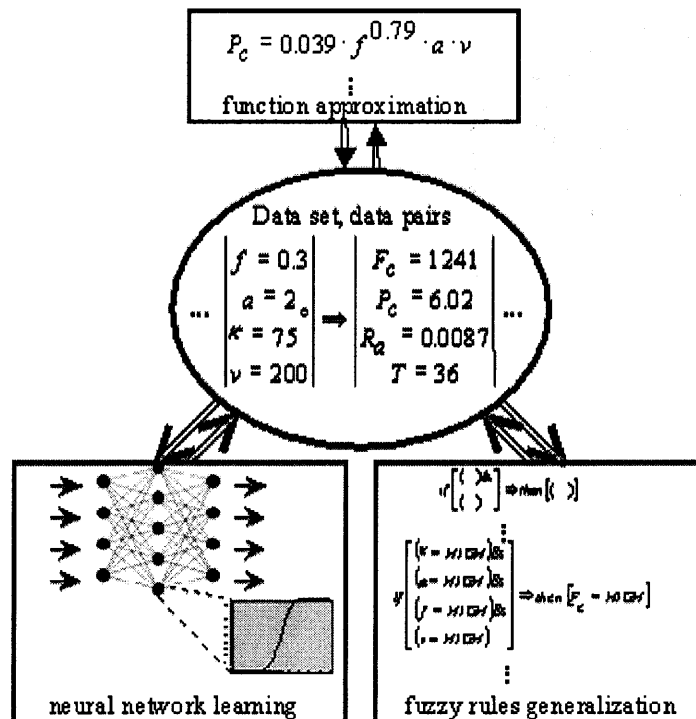
5.4 Propagation of errors in multi step machining

A state space model is developed to describe the dimensional variation propagation of multistage machining processes. A complicated machining system usually contains multiple stages. When the work piece passes through multiple stages, machining errors at each stage will be accumulated and transformed onto the work piece. Differential motion vector, a concept from the robotics field, is used in this model as the state vector to represent the geometric deviation of the work piece. The deviation accumulation and transformation are quantitatively described by the state transition in the state space model. A systematic procedure that builds the model is presented and an experimental validation is also conducted. The validation result is satisfactory.

This model has great potential to be applied to fault diagnosis and process design evaluation for complicated machining processes. This state space model is then represented as input to the neural network that would give an estimate of the error propagating through the work piece. This estimation would be useful for the generation of G and M codes. Refer to the appendix for a detailed explanation of the ANN output. Four inputs and four outputs are considered with one hidden layer.

The outputs are then interpreted using a post processor desigened in VisualBasic.

Given below is the Nomenclature of terms that will be used frequently in this related discusson:

These would represent the fixture co-ordinate system. The fixture coordinate system is divided into two parts

1)   Nominal fixture coordinate system

2)   Actual fixture coordinate system

3)   Homogenuous transformation matrix

4)   Identity matrices

NOMENCLATURE

^{0}FCS, FCS	nominal and actual fixture coordinate system
HTM	homogeneous transformation matrix
H_{n}^{R}, ${}^{0}H_{n}^{R}$	actual and nominal HTM between RCS and LCS$_{n}$
δH_{n}^{R}	deviation HTM defined by $\mathbf{d}_{n}^{R}$ and $\mathbf{\theta}_{n}^{R}$
$I_{n \times n}$, $O_{m \times n}$	n by n identity matrix and m by n zero matrix, respectively
LCS$_{i}$	ith local coordinate system
M, N	total number of key features and machining stages, respectively
RCS	reference coordinate system
R_{n}^{R}, ${}^{0}R_{n}^{R}$	actual and nominal rotational matrix between RCS and LCS$_{n}$

T_1, T_2 transformation matrices for datum-induced error

T_3 transformation matrix for fixture error

VD&T   vectorial dimensioning and tolerancing

$\mathbf{d}_n^R$, $\mathbf{\theta}_n^R$ position and orientation deviation of LCS$_n$ from nominal positions w.r.t. RCS

d_{nx}^R, d_{ny}^R, d_{nz}^R three elements of $\mathbf{d}_n^R$

k stage index. $k = 1 \dots N$

$l(k)$ total number of newly generated feature at stage k

$\mathbf{n}$, $\mathbf{o}$, $\mathbf{a}$ column vectors of a rotational matrix

$\hat{\mathbf{p}}$ a skew symmetric matrix obtained from vector $\mathbf{p}$.

$\tilde{\mathbf{p}}$ an extension of a vector $\mathbf{p}$. $\tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{p}^T & 1 \end{bmatrix}^T$.

$q_i(k)$ feature index for newly generated feature at stage k. $i = 1 \dots l(k)$.

$r_i(k)$ feature index for the primary datum ($i = 1$), secondary datum ($i = 2$), and tertiary datum ($i = 3$) of the kth stage.

$s_i(k)$ feature index for the generated feature up to stage k (not include stage k). i is an integer from 1 to total number of generated features up to stage k

$\mathbf{t}_n^R$, $^0\mathbf{t}_n^R$ actual and nominal vector of the origin of LCS$_n$ expressed in RCS

t_{nx}^R, t_{ny}^R, t_{nz}^R three elements of $\mathbf{t}_n^R$

w.r.t.   with respect to

$\mathbf{x}_n^R$ a differential motion vector representing the deviation of LCS$_n$ in RCS. It is a stack of $\mathbf{d}_n^R$ and $\mathbf{\theta}_n^R$ and is expressed in LCS$_n$.

$\mathbf{x}(k)$ state vector

$\mathbf{x}^n(k)$ a stack of differential motion vectors of newly generated features at stage k.

Δ_n^R, $\overline{\Delta}_n^R$ differential transformation matrix corresponding to δH_n^R and $(\delta H_n^R)^{-1}$

ω_n^R, $^0\omega_n^R$ actual and nominal Euler rotational angles between RCS and LCS$_n$

θ_{nx}^R, θ_{ny}^R, θ_{nz}^R three elements of $\mathbf{\theta}_n^R$

$[\cdot]_{(n)}$ the nth element of a vector in the bracket

$\times$ cross product of two vectors

5.5 Work piece geometric deviation:

Machining process is used to remove materials from the work piece to obtain higher dimensional accuracy, better surface finishing, or more complicated surface form which cannot be obtained by other processes. A complicated machining process is usually a "multistage" machining process, which refers to a machining process where a part will be machined through different setups when it passes through this process. It is not necessary that a multistage machining process contains multiple machining stations. If there are different setups on only one machining station, this machining process is still considered as a multistage machining process. When a work piece passes through certain stage of a multistage machining process, the machining error and fixturing error of this stage will be accumulated on the work piece. These errors will again affect the machining accuracy of the following stages if the datum used by following stage is produced at current stage. Since the work piece carries all the machining error information, a representation of accuracy of a work piece is required to study the complicated interaction of errors among different stages.

Work piece Geometric Deviation Representation

To regulate the deviations of part features, people developed standards for geometric dimensioning and tolerancing (ISO 1101 (1983) or ANSI Y14.5(1982)). However, these conventional geometric tolerances are originated from the hard gauging practice. They are not suitable for the working principle of Coordinate Measurement Machine (CMM) that is now a standard measurement equipment for machining process. In addition, the representation of part feature in the conventional geometric tolerances does not conform to the part representations used in CAD/CAM systems. Recently, some researchers [10]

proposed a vectorial dimensioning tolerancing (VD&T) strategy. The principle of VD&T is based on the concept of substitute elements or substitute. A substitute feature is an imaginary geometrical ideal feature (e.g., plane, circle, line) whose location, orientation, and size (if applicable) are calculated from the measurement data points of the work piece surface. Substitute features are represented by the location vector, orientation vector, and size(s). The location vector indicates the location of a specified point of the substitute feature. The substitute orientation vector is a unit vector that is normal to the substitute plane or parallel to the substitute axis (cylinder, cone, etc). The size is available for some features. For example, the diameter is the size of a circular hole. The VD&T work piece feature representation follows the working principle of CMM and CAD/CAM systems. The measurement data from CMM can be analyzed and compared with the design model directly. The difference between the true feature and the design requirement can be feedback to the manufacturing process directly. It is a better tolerancing method for manufacturing process control [11].

This paper adopts a vectorial feature representation proposed by Yau [12, 13]. The difference between his representation and the ordinary vectorial representation is in the orientation representation. Instead of using a unit direction vector, he used a vector that consists of three Euler rotating angles to represent the orientation of the substitute feature. The representation of using unit direction vector makes it difficult to designate tolerance on the orientation for a general 3D geometric element. Moreover, the direction vector representation violates certain functional requirements that VD&T intends to capture. Another advantage of angular representation of orientation is that there are many mathematical tools available in the fields of robotics and kinematics for

this representation. Therefore, in this paper, a location vector and a vector that consists of three rotating Euler angles are used to represent a work piece feature. Since the size of a feature is usually formed at one machining stage, it is not considered in the following derivation.

Let us look at a few corollaries which are used to develop the model. Refer to the appendix for the proofs of these corollaries.

$$\mathbf{x}(k+1) = A(k)\mathbf{x}(k) + B(k)\mathbf{u}(k) + \mathbf{w}(k)$$
$$\mathbf{y}(k) = C(k)\mathbf{x}(k) + \mathbf{v}(k)$$
$$\tag{1}$$

Corollary 1

Consider a RCS and two features 1 and 2. Given ${}^{0}H_{1}^{R}$, ${}^{0}H_{2}^{1}$ and the deviation of feature 1 w.r.t. RCS, $\mathbf{x}_{1}^{R}$ and the deviation of feature 2 w.r.t. feature 1, $\mathbf{x}_{2}^{1}$, then

$$\mathbf{x}_{2}^{R} = \begin{bmatrix} ({}^{0}R_{2}^{1})^{T} & -({}^{0}R_{2}^{1})^{T} \cdot ({}^{0}\hat{t}_{2}^{1}) & I_{3\times3} & 0 \\ 0 & ({}^{0}R_{2}^{1})^{T} & 0 & I_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1}^{R} \\ \mathbf{x}_{2}^{1} \end{bmatrix}. \tag{2}$$

Corollary 2

Consider a RCS and two features 1 and 2. Given ${}^{0}H_{1}^{R}$ and ${}^{0}H_{2}^{R}$ and the deviation of feature 1 w.r.t. RCS, $\mathbf{x}_{1}^{R}$ and the deviation of feature 2 w.r.t. RCS, $\mathbf{x}_{2}^{R}$, then

$$\mathbf{x}_{2}^{1} = \begin{bmatrix} -({}^{0}R_{2}^{1})^{T} & ({}^{0}R_{2}^{1})^{T} \cdot ({}^{0}\hat{t}_{2}^{1}) & I_{3\times3} & 0 \\ 0 & -({}^{0}R_{2}^{1})^{T} & 0 & I_{3\times3} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{1}^{R} \\ \mathbf{x}_{2}^{R} \end{bmatrix}. \tag{3}$$

These two corollaries are very useful when the RCS switches to another coordinate system.
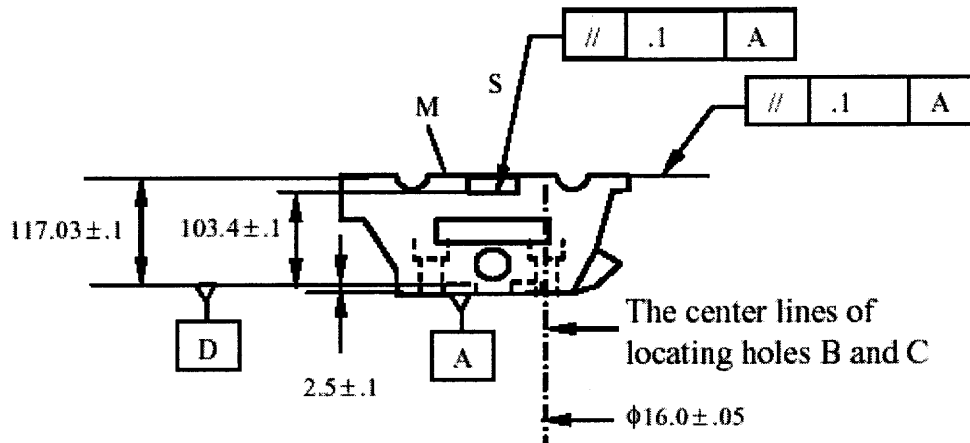
Fig 5.6 Major Design specification.

Table 5.3 Specifications and Sequence of Operation

Characteristics	Specifications(mm)	Operation
1. Distance between M and rough datum surface D	117.03 ± 0.1	1st
2. Distance between A and rough datum surface D	2.5 ± 0.1	2nd
3. Parallelism between M and A	0.1	2nd
4. Distance between slot S and A	103.4 ± 0.1	3rd
5. Parallelism between A and S	0.1	3rd

5.6 Geometrical tolerance control using vectorization:

Geometrical dimensioning and tolerancing currently depends on ISO standards (ISO 286, ISO 1101, ISO 8015, ISO 5459,etc.).However, ambiguity and absence of a sound mathematical representation of the conventional tolerance scheme have fostered the use of vectorial tolerance description methods for tolerance zones.
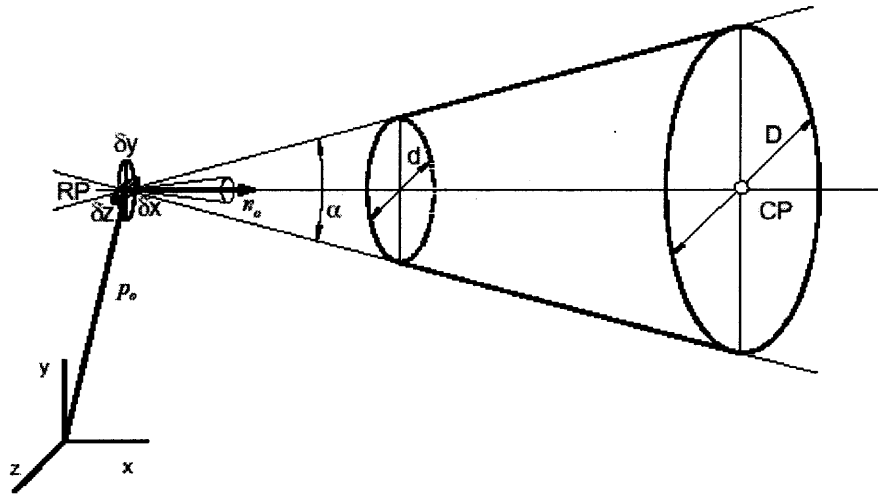
Figure 5.7 Vectorial Tolerancing of a Conical Feature.

A conical feature can be represented in x, y and z Cartesian co-ordinates by using VD and T (vectorial dimensioning and tolerancing). The co-ordinates of the cone's apex (point RP), the cone normal n_0 and the cone angle α are easily accessible in 3-D CAD systems. Topological elements like the circles of the cone and the faces(with diameters D and d) and the connector point CP (center of bigger end face, functionally important in many cases) can be retrieved from the nominal CAD-model too. The cone's deviation position is limited by the vector $(\delta x, \delta y, \delta z)$. The orientation tolerances are represented by the vector $(\rho x, \rho y, \rho z)$. The component ρx equals to zero since rotation about the x-axis will be the only remaining degree of freedom. The cone angle is limited by the value $\delta \alpha$. Using the attribute mechanism of CAD-system the vectorial tolerance parameters can be attached to the nominal cone surface.

One problem of Vectorial dimension and tolerancing (VD and T) is it's completely new drawing indication (proposed by [Henzold95]). An example of this kind of drawing indication which mainly depends on the concept of substitute elements is shown in Figure 5.7.
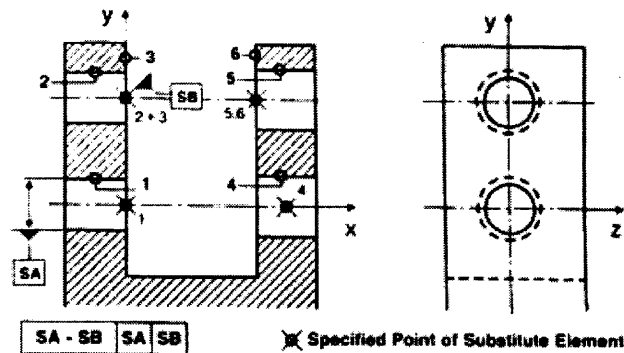

Figure 5.8 Drawing Indication of Vectorial Tolerancing.

Table 5.4 Datum Representation

Primary Datum	1 − (2 + 3)	(Substitute Element and Substitute Intersection Point)
Secondary Datum	1	(Substitute Element)
Tertiary Datum	2 + 3	(Substitute Intersection Point)

Table 5.4   Datum representation

Substitute Element			1	2	3	4	5	6
Location	Nominal	X_0	0	0	0	0	0	0
		Y_0	0	35	35	0	35	35
		Z_0	0	0	0	0	0	0
Permissible		X	-	-	-	-	-	0.1
Deviation		Y	-	0.1	-	0.1	0.1	-
		Z	-	-	-	0.1	0.1	-
Orientation Nominal		E_x	-1	-1	+1	+1	+1	-1
		E_y	0	0	0	0	0	0
		E_z	0	0	0	0	0	0
Permissible Deviation			-	-	-	-	-	-

The permissible deviation in dimension, form and orientation are particularly included in a table (Table 5.4). Trying to transfer this new kind of tolerance symbolic into the shop floor would probably end in confusion. Furthermore, designers would have to throw away their hard gained knowledge of dimensioning and tolerancing.

A connection of GD and T and VD and T

This contribution focuses on those tolerances related to conical surfaces; however this can be extended to other features and surfaces. Besides dimensional tolerances like Cone angle, cone diameter, etc. geometrical tolerances are even considered (Table 5.5) may occur in connection with cones and other technical drawings. Some of these tolerances will be related to the cone generator line (like straightness) or a planar section curve (like

roundness). Parallelism may be employed to limit deviation one cone end face related to the second end face or another plane at the same workpiece.

Table 5.5 Symbols for Geometric Tolerances

Symbol	Tolerance	Feature to be toleranced	Datum
—	Straightness	Generator line or axis	No
O	Roundness	Cone section or axis	No
⌒	Profile(form)of a surface	Cone surface	Yes, arbitrary
↗	Run-out	Cone section	Axis
↗↗	Total run-out	Cone surface	Axis
◎	Co-axiality	Cone axis	Axis or common axis
∥	Parallelism	Cone end face	Plane
⊕	Position	Cone end face: reference point of cone	Plane possibble
∠	Angularity	Generator line	Axis or plane
⊥	Perpendicularity	Cone end face	Axis or plane

In the concept of VD and T the profile form tolerance of the surface plays an important role in the concept of converting conventional tolerances into vectorial tolerances. Unlike other form tolerances the profile tolerances of surfaces and lines may be related to datums. This means that the location of a resultant tolerance zone which belongs to special profile tolerances can be fixed in Euclidean space by theoretical exact dimensions. That is the profile tolerances combine characteristics of form tolerances (like the limitation of form deviations) and location tolerances (like tolerance zones fixed by datums).

CONVERSION INTO APPROPRIATE CONVENTIONAL CONE TOLERANCES

The extraction of mathematically sound tolerance zone in euclidean space is the first step towards VD and T. The evaluation of different kinds of conventional cone tolerancing has led to those combinations which serve the best for both, the conversion of task and control of functions which conical parts have to fulfill.

Two different dimensioning and tolerancing schemes for cone are introduced on the basis of ISO 3040 standards. The first scheme is more helpful in VD and T technique, where the offset approach is used. This kind of cone tolerancing is more appropriate for conversion purposes compared to the second scheme which involves cone angle tolerancing techniques.

In Figure 5.9, the offset approach is combined with relative vectorial tolerancing approach to control both form and orientation of the cone surface. The direction L_1 and L_2 must be considered as unidirectional. This means that the dimensional tolerance exists in two form elements: one form element which has to be controlled by tolerance (left cone end face) and the second one that establishes the related datum (left cylinder end face). The measuring dimension of L_1 and L_2 is well defined in contrast to two-point-dimensions.

The tolerance zone shown on left in the figure below is well located and oriented in 3-D space: the position of the zone is fixed by the theoretical exact dimension L referring to datum B. Furthermore, the orientation vector of the tolerance zone has to be coaxial to the datum A.
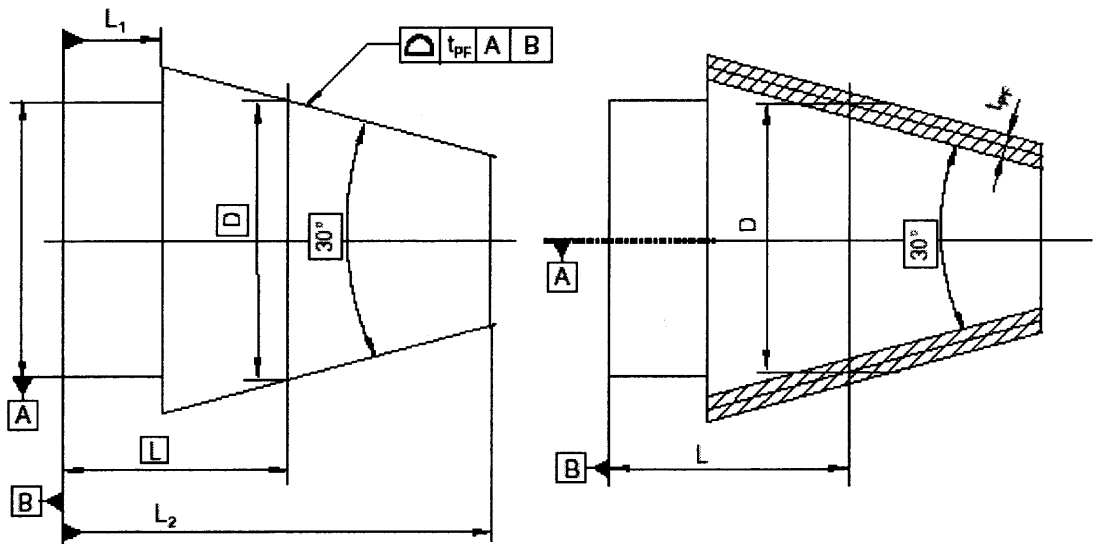
Figure 5.9 Related Profile Tolerance Zone as Converter Input.

The described tolerance zone is the starting point for the examination of possible vectorial positions and orientations that the nominal conical feature can occupy within the tolerance zone. Then the vectorial tolerance parameters and interferences are obtained by mathematical functions, examined from the derived worst case deviations.

WORST DEVIATIONS WITHIN THE CONVENTIONAL CONE TOLERANCE ZONE

In this section two kinds of deviations within the cone tolerance zone explained above are examined. The worst case orientation δ according to figure 5.10 will be determined first. The pivot of the rotation coincides with the connection point CP. From the functional point of view CP will be the most important reference element of the conical face. The vectorial tolerance scheme applies the apex of the cone as the main location of the item of the face, too. If the orientation vector N of the tapered face will be altered during the

visualization of a deviation, the orientation position of RP also has to be altered to keep the connection point CP invariant.

$$\delta = -\gamma' + \gamma \quad \textit{with} \ \gamma = \arctan\frac{d/2}{L_2 - L_1} \quad \gamma' = \arctan\frac{d/2+b}{L_2 - L_1} \quad b = \frac{t_{PF}}{2\cos(\alpha/2)} \quad (1)$$

The radial distances of the offset faces b in (1) is determined by the tolerance value t_{pf} of the profile tolerance. The orientation vector n may deviatefrom the nominal orientation vector n_0 within the limits n plus/minus δ.
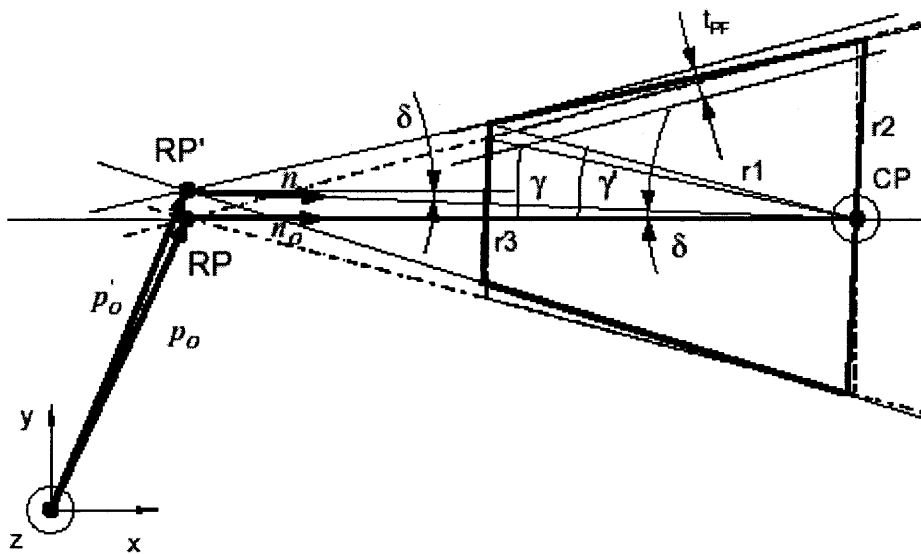

Figure 5.10 Worst Case Orientation Deviation Related to Connection Point CP.

To keep the connection point CP at it's correct position a movement RP along the y-(ΔRP_y) x-axis (ΔRP_x) must be guaranteed during CAD-visualization.

$$\Delta RP_y = r_3 \sin(\gamma' - \gamma) \qquad \Delta RP_x = r_3 (1 - \cos(\gamma' - \gamma)) \qquad \textit{with} \quad r_3 = \frac{D/2}{\tan(\alpha/2)} \qquad (2)$$

The variable r_3 in (2) means the virtual height of the non-truncated cone. The second example of deviation within the cone tolerance zone according to figure 5.9 is shown in figure 5.11. The greatest cone angle α' can be calculated as given in (3)
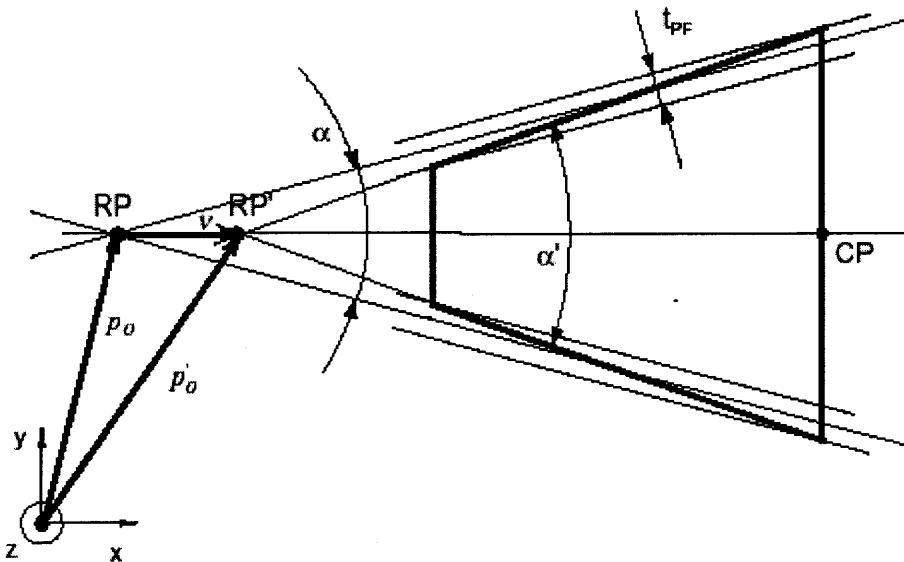

Figure 5.11 Worst Case Cone Angle Within the Tolerance Zone.

$$\alpha' = 2 \arctan \frac{(D-d)/2 + 2b}{l} \qquad \Delta RP_x = v = \frac{D/2}{\tan(\alpha/2)} - \frac{D/2+b}{\tan(\alpha'/2)} \qquad (3)$$

If there is no variation of the connection point CP allowed the reference point RP has to be moved along the x-axis (ΔRP_x) during the deviation simulation. The upper limit of the target vectorial tolerance parameter 'cone angle' will be set to $\alpha^{max} = \alpha'$. The lower limit and the other vectorial tolerance parameter (like position tolerance) can be calculated in a similar way as demonstrated.
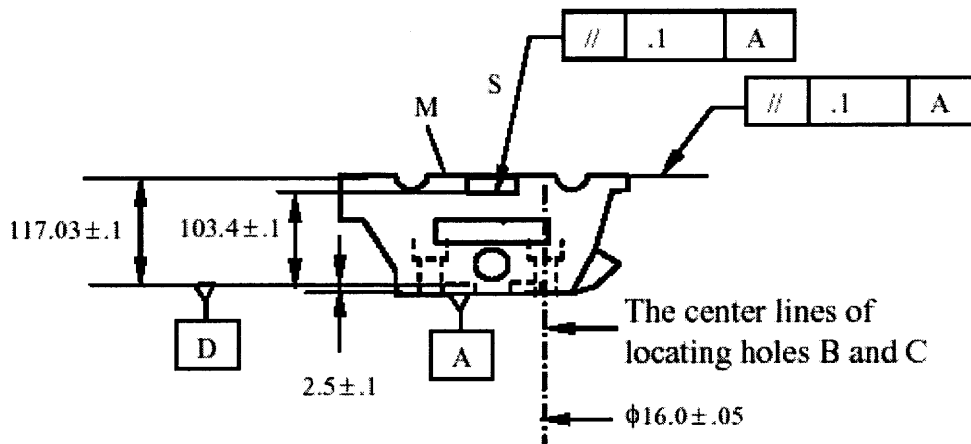
CHAPTER 6

CASE STUDY

Fig 6.1 Major Design specification.

The experimental set up is shown in figure 6.1 and table 6.1. The above mentioned methodology was applied and the results are explained. First, the tolerances are fixed with respect to datums [8], the machining operations are described in table 6.2, and secondly, the geometric tolerances are represented in terms of a vector for the ANN input. The process capability [9] is calculated using the control charts (refer ANN 1). The various machine control parameters are represented by ANN 2 and the output is discussed. The output of the ANN (refer ANN 3) creates a vectorial space for geometrictolerance and the propagation of error in multi stage machining using state space modeling [10].After feeding the inputs in the neural networks the following results (graphs) were generated:

Refer to the appendix for inputs, outputs and learning rules.

Table 6.1 Description of Tolerance Requirement (refer figure 6.1)

Characteristics	Specifications(mm)	Stage
1. Distance btw M and rough datum surface D	117.03 ± 0.1	1st
2. Distance btw A and rough datum surface D	2.5 ± 0.1	2nd
3. Parallelism btw M and A	0.1	
4. Distance btw slot S and A	103.4 ± 0.1	3rd
5. Parallelism btw A and S	0.1	

Table 6.2 Description of the Operations

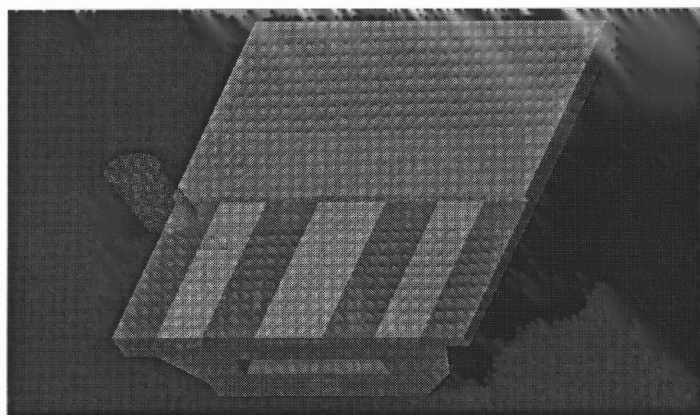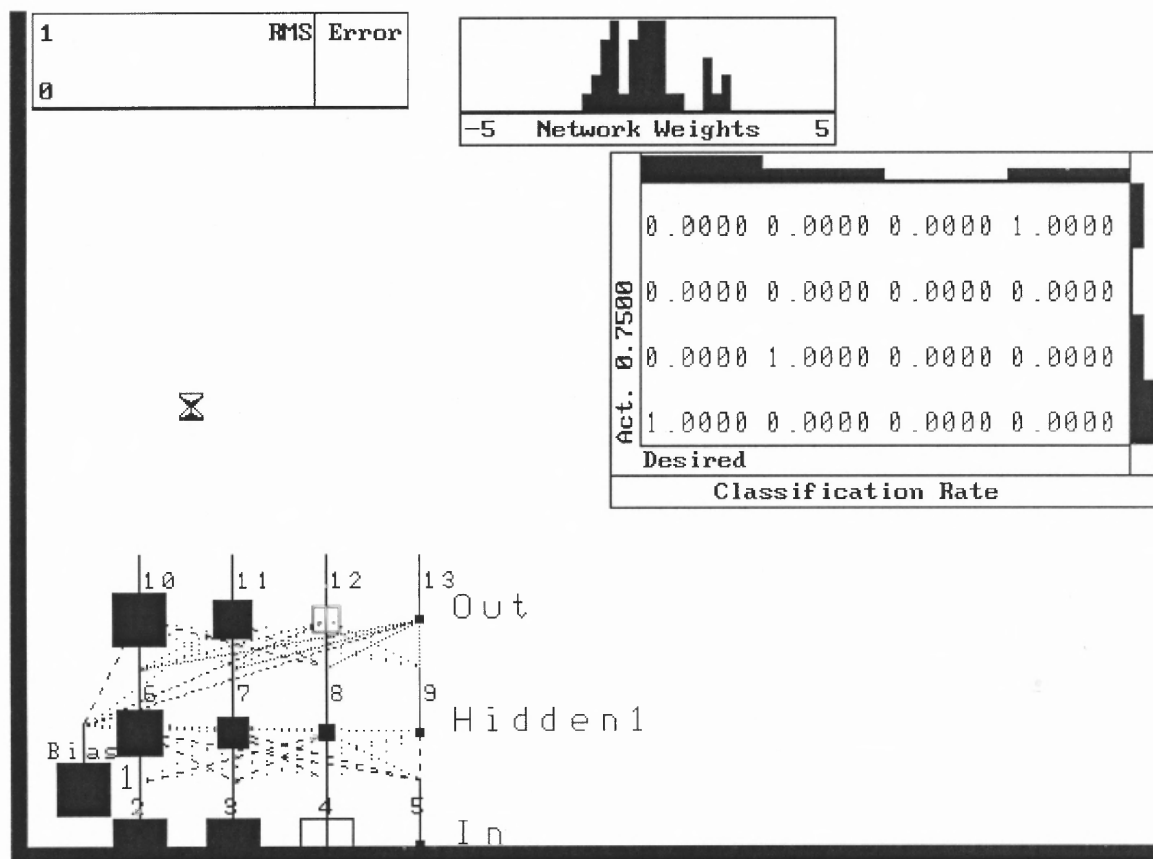| Operation #	Locating Datum (Primary +secondary +Tertiary)	Operation Descriptions
1	$(X_1, X_2, X_3) + (Y_1, Y_2) + Z$	• Mill cover face M • Spot drill H451~H458 • Spot face H451~H458
2	$M + (Y_1, Y_2) + Z$	• Mill joint face A • Spot drill H101~H108 • Drill, chamfer, counterboring, and reaming hole H101(B) and H104(C) • Drill H102, 103, and H105~H108
3	$A + B + C$	• Mill slot S • Drill and chamfer twelve screw holes S1~S12.

ANN OUTPUT 3

Figure 12. Vericut using Pro/engineer for Workpiece Model.

ANN OUTPUT 2

ANN OUTPUT 3

1	RMS	Error
0		

-5	Network Weights	5

0.0000	0.0000	0.0000	1.0000
0.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000

Act. 0.7500

Desired

Classification Rate

Out

Hidden1

Bias

In

Figure 12. Vericut using Pro/engineer for Workpiece Model.

Output before ANN:

N5 G71
N10(/ EH7)
N15 G0 G17 G99
N20 G90 G94
N25 G0 G49
N30 T1 M06
N35 S1170 M03
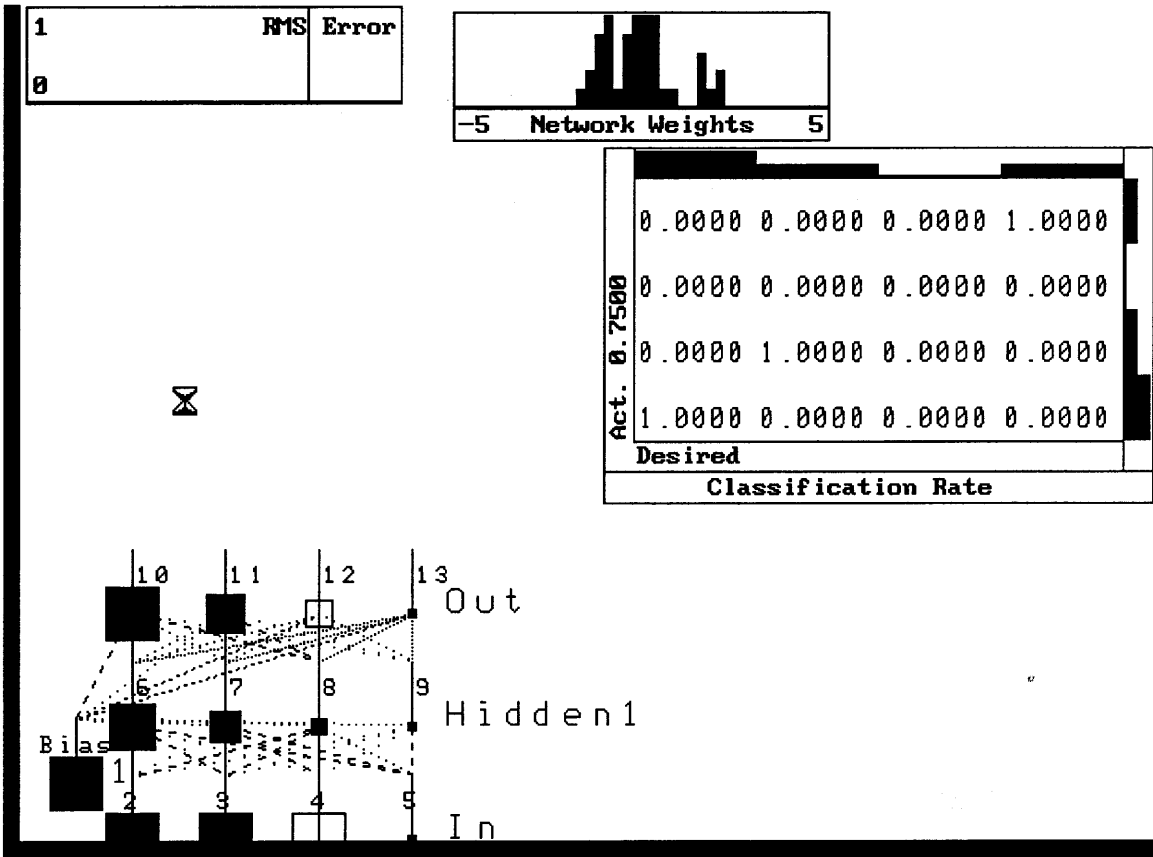*N40 G0 G43 Z137.03 M08 H1*
...
...
...
N5045 Y-.4
N5050 X315.
N5055 Y-.2
N5060 X0.
N5065 Y0.
N5070 X315.
N5075 Z137.03
N5080 M30

Output 2 (after ANN) :

N1 T1 M6
N2 S1170 M3
N3 G0 X0. Y-500.
N4 G43 Z137.03 H1 M7
N5 Z117.13
N6 G1 Z117.03 F5.
N7 X315.
N8 Y-499.8
N9 X0.
N10 Y-499.6
N11 X315
...
...
N5002 Y-.4
N5003 X315.
N5004 Y-.2
N5005 X0.
N5006 Y0.
N5007 X315.
N5008 Z137.03
N5009 M5
N5010 M30

CHAPTER 7

ANALYSIS OF RESULTS AND SCOPE FOR FUTURE WORK

The ANN output 1 represents UTL, LTL, UTL-μ, μ-LTL. Observe how the UTL-μ is within limits where desired is 0.5 and actual is 0.1250 and μ-LTL is 1.00 (above 0.5). The ANN was trained for 50,000 sets of data with one pass/all.

The ANN output 2 represents the empirical formula for f, a, κ,υ (these are the inputs) and F_c, P_c, R_a , t (these are the outputs). Observe that "f" (node 1) influences the output most and depending on the input see the corresponding output. Hence, the feed rate, spindle speed and the depth of cut with respect to tolerances after the ANN has been trained for 50,000 sets of data for machining (milling, turning and drilling). Three of the hidden-layer 1 neurons (6, 7 and 8) reacted strongly to the output.

The ANN output 3 just gives the vectorial space for parallelism of surface M and A (input 1), relative distance between S, M with respect to rough datum surface D (input 2), distance between slot S and A (input 3) and parallelism between S and A (input 4). See node 10 and 11 which is the output for trained data set, these are the outputs for vectorial space for parallelism of surface M and A (input 1) and relative distance between S, M with respect to rough datum surface D (input 2). Both of them were linearly separable in the data space and are most influential so care has to be taken while considering the rough datum surfaces especially for multi-stage machining.

Previous comparisons of neural networks and linear predictors have shown that neural networks sometimes can give better results. However, the data sets used were not particularly long, so the statistical significance of these comparisons may be questionable. In the studies a feed-forward neural network, with a single hidden layer was

79

used. Quickprop was used for training as it has faster convergence than standard back-prop. For the neural network, trial and error showed that four hidden units and a linear output unit gave best results. This architecture was fixed prior to training.

Finally, all the outputs can be used to generate G and M codes using a post-processor.

The spikes in the ANN OUTPUT 2 (objective function) mean that it is possible to do better than use a linear predictor for this data set. The general nonlinear methods of neural networks and local approximation did well and can be expected to be near optimal as the data set size increases. Some progress in understanding the role of each hidden unit in the neural network predictor was obtained. Larger data sets for ANN 3 will be necessary for determining errors in multi-stage machining. Data sets for machine control parameters shows the best output with interaction at each level, but could get better if the number of PE's in the hidden layers are optimized. Data points can be entered into the trained neural networks to get the expected output.

Looking at the conclusions of the case study it is clear that future work must continue in order to optimize the neural network performance by means of reducing the number of hidden layers for the network representing the cutting speed parameters. A more efficient algorithm could be developed for obtaining the target values for the networks.

The data sets for obtaining rough datum lines and vectorial representations of geometrical tolerances could be increased for better results.

REFERENCES

[1]   H. B. Kief : NC/CNC Handbook '97/98, Carl Hanser Verlag München Wien, 1997.

[2]   L. Pryzybylski; B. Slodki: The Influence of Local Operating Features on the Cutting Parameters in Turning, The International Journal of Advanced Manufacturing Technology 13:233-236, Springer-Verlag London, 1997.

[3]   A. Mischo: Model Based Acquisition and Implementation of Technological Data in CNC Data Planning, University of Saarland, Schriftenreihe Produktionstechnik Band 3, 1993.

[4]   C. Weber, W. Britten, O. Thome: Conversion of Geometrical Tolerances into Vectorial Tolerance Representations - A Major Step towards Computer Aided Tolerancing, International Design Conference - Design '98, Dubrovnik, May 19-22, 1998.

[5]   H. Dagli, Cihan.: Artificial Neural Networks for Intelligent Manufacturing, London; New York: Chapman & Hall, 1994.

[6]   J. A Gregory,.: The Institute of Mathematics and Its Applications Conference Series; Oxford Science Publications.

[7]   Neuralworks Professional II/plus: Neural Computing-Neural ware Inc. 1993.

[8]   Neuralworks Professional II/plus: Advanced Reference Guide-Neural ware Inc. 1993.

[9]   Neuralworks Professional II/plus: Handbook for Professional Neuralwork Explorer-Neural ware Inc. 1993.

[10]   Neuralworks Professional II/plus: System Guide for Personal Computers-Neural ware Inc. 1993.

[11]   J. Peklenik, A. Jerele: Some Basic Relationships for Identification of the Machining Processes, Annals of the CIRP Vol. 41/1/92.

[12]   F. C. Moon: Dynamics and Chaos in Manufacturing Processes, Wiley Series in Nonlinear Science, New York, 1998.

[13]   R. Teti, S. Kumara: Intelligent Computing Methods for Manufacturing Systems, Annals of the CIRP Vol. 6/2/1997.

[14]   T. Warren, J. Chen: A Neural Network Approach for Grinding Processes: Modelling and Optimization, International J. Machining Tools Manufacture, Vol. 34 No. 7.

[15] L. Monostori and Cs. Egresits: Modeling and Monitoring of Milling through Neuro-Fuzzy Techniques, 2nd World Congress on Intelligent Manufacturing Processes and Systems 1994. 13-15 Jun 1994.

[16] E. Westkämker: Supervision of Quality in Process Chains by Means of Learning Process Models, Proceedings of Second International Workshop on Learning in Intelligent Manufacturing System, Monostori, L. (Ed.), pp. 565-590, 9633113431, Budapest, March 1995.

[17] W.H Greenwood, and Chase, K.W. 1987: A New Tolerance Analysis Method for Designers and Manufactures, ASME, J. Eng. Ind. 109, pp. 112-116.

[18] J. Gilson, 1951, A new Approach to Engineering Tolerances. The Machinery Publishing Co., London, UK.

[19] G. Abdou and Jujiang Bao: Building Oriented Ruled Surface Generation Through IGES Specification, October 1994, Thesis submitted to the faculty of New Jersey Institute of Technology.

APPENDIX

OUTPUTS

Please find the attached copy of the CD which includes all the ANN outputs and the post NC-processor outputs for G&M codes.