RECONFIGURATION AND TOOL PATH PLANNING OF HEXAPOD MACHINE TOOLS

by
Zhenqun Li

Hexapod machine tools have the potential to achieve increased accuracy, speed, acceleration and rigidity over conventional machines, and are regarded by many researchers as the machine tools of the next generation. However, their small and complex workspace often limits the range of tasks they can perform, and their parallel structure raises many new issues preventing the direct use of conventional tool path planning methods. This dissertation presents an investigation of new reconfiguration and tool path planning methods for enhancing the ability of hexapods to adapt to workspace changes and assisting them in being integrated into the current manufacturing environments.

A reconfiguration method which includes the consideration of foot-placement space (FPS) determination and placement parameter identification has been developed. Based on the desired workspace of a hexapod and the motion range of its leg modules, the FPS of a hexapod machine is defined and a construction method of the FPS is presented. An implementation algorithm for the construction method is developed. The equations for identifying the position and orientation of the base joints for the hexapod at a new location are formulated. For the position identification problem, an algorithm based on Dialytic Elimination is derived. Through examples, it is shown that the FPS

determination method can provide feasible locations for the feet of the legs to realize the required workspace. It is also shown that these identification equations can be solved through a numerical approach or through Dialytic Elimination using symbolic manipulation.

Three dissimilarities between hexapods and five-axis machines are identified and studied to enhance the basic understanding of tool path planning for hexapods. The first significant difference is the existence of an extra degree of freedom (γ angle). The second dissimilarity is that a hexapod has a widely varying inverse Jacobian over the workspace. This leads to the result that a hexapod usually has a nonlinear path when following a straight-line segment over two sampled poses. These factors indicate that the traditional path planning methods should not be used for hexapods without modification.

A kinematics-based tool path planning method for hexapod machine tools is proposed to guide the part placement and the determination of γ angle. The algorithms to search for the feasible part locations and γ sets are presented. Three local planning methods for the γ angle are described. It is demonstrated that the method is feasible and is effective in enhancing the performance of the hexapod machine. As the nonlinear error is computationally expensive to evaluate in real time, the measurement of total leg length error is proposed. This measure is proved to be effective in controlling the nonlinear error.

RECONFIGURATION AND TOOL PATH PLANNING OF HEXAPOD MACHINE TOOLS

by
Zhenqun Li

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Mechanical Engineering

January 2000

APPROVAL PAGE

RECONFIGURATION AND TOOL PATH PLANNING OF HEXAPOD MACHINE TOOLS

Zhenqun Li

Dr. Zhiming Ji, Dissertation Advisor — Date
Associate Professor of Mechanical Engineering, NJIT

Dr. Ming C. Leu, Committee Member — Date
New Jersey State Sponsored Chair in Manufacturing/Productivity
Professor of Mechanical Engineering, NJIT

Dr. Ernest S. Geskin, Committee Member — Date
Professor of Mechanical Engineering, NJIT

Dr. Denis Blackmore, Committee Member — Date
Professor of Mathematics, NJIT

Dr. Rajpal S. Sodhi, Committee Member — Date
Associate Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author:	Zhenqun Li
Degree:	Doctor of Philosophy
Date:	January 2000

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering
  New Jersey Institute of Technology, Newark, NJ, USA, 2000

- Master of Science in Mechanical Engineering
  Huazhong University of Science and Technology, Wuhan, P.R. China, 1987

- Bachelor of Science in Mechanical Engineering
  Huazhong University of Science and Technology, Wuhan, P.R. China, 1984

Major: Mechanical Engineering

Publications:

Li, Z., and Ji, Z., 1999, "Tool path planning and optimization o hexapod machine tools," *International Conference on Advanced Manufacturing Technology*, June, Xian, P.R. China.

Ji, Z., and Li, Z., 1999, "Identification of placement parameters for modular platform manipulators," *Journal of Robotics Systems*, 16(4), pp. 227-236.

Ji, Z., and Li, Z., 1998, "Determination of individual foot-placement space for modular platform manipulators," *ASME Design Engineering Technical Conference*, MECH-5966, Sept., Atlanta, USA.

Li, Z., and Ji, Z., 1998, "Symbolic Dialytic elimination for parameter identification of platform manipulators," *Proceedings, the ISCA 13th International Conference*, March, Hawaii, USA.

Li, Z., Liao, X., and Li, G., 1994, "On trajectory control of NC laser processing machines," *Journal of Hubei Insti. of Tech.*, Vol. 9, No. 3, pp. 94-99.

Li, G., Li, Z., Chen, Y., Liao, X., and Li., B., 1988, "The application of the principle of time-sharing interpolation in open-loop CNC systems," *Journal of Huazhong Univ. of Sci. & Tech.*, Vol. 16, No. 2, pp. 39-45.

Li, G., Chen, Y., Li, Z., Li., B., and Liao, X., 1988, "A microcomputer-based control system for laser processors," *Journal of Huazhong Univ. of Sci. & Tech.*, Vol. 16, No. 2, pp. 63-66.

Li, G., and Li, Z., 1988, "A new interpolation algorithm for open-loop CNC systems," *Proceedings of the First Conference on Mechtronics of P.R. China*, 2(2), pp. 753-760.

Z. Li, 1987, "A study on the trajectory control systems of laser machining systems," *Master Thesis*, Huazhong Univ. of Sci. & Tech., Wuhan, P.R. China.

This dissertation is dedicated to
my family for their love and support

ACKNOWLEDGMENT

The author would like to express his sincere thanks and deepest gratitude to his advisor, Dr. Zhiming Ji, for the remarkable guidance, constant support, encouragement and patience throughout his study. The author's deepest gratitude and appreciation also goes to Dr. Ming C. Leu for his valuable insights, continuous fruitful suggestions and support.

The author would also like to express his sincere thanks and appreciation to Drs. Dennis Blackmore, Ernest S. Geskin, and Raj S. Sodhi for their valuable suggestions to this dissertation and for serving as members of the committee.

The work is supported, in part, by the National Science Foundation (grant number CMS 9522840) and by the Department of Mechanical Engineering, NJIT, in the form of an assistantship during the author's doctoral program.

The author would like to acknowledge Dr. W. Zhang, Mr. Y. Pei, Mr. B. Yang, Miss Y. Luo, and the other members of the research group for their help and suggestions to this study.

And finally, the author would like to express his deepest gratitude to his family members, especially his wife Qiaoli Yu and son Kailun Li, for their firm support and encouragement.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF COTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Higher productivity with higher quality is a pertinent issue in the field of manufacturing engineering. Increasing standards of them require that machine tools have the corresponding capability in speed and accuracy. In addition, changing market demands, shorter life cycles of products and greater variation of parts require that work cells have improved characteristics of flexibility, responsiveness and reconfigurability. Conventional machine tools characterized by a serial arrangement of the axes have the limitations of large masses to be moved, accumulation of errors, and a low number of repetition parts. It is becoming evident that they can not provide a satisfactory solution for these requirements. A new class of machine tools based on parallel mechanisms called Parallel Kinematic Machines (PKMs) presents a new and promising alternative to meet these needs.

Hexapod machine tools (HMTs) are one type of parallel kinematic machines. They represent a dramatic departure from traditional machine tools, because their six degrees of freedom (6-DOF) are obtained through parallel kinematic chains. This parallel structure offers superior stiffness, low mass, and high precision and acceleration, and therefore provides a possibility to meet the needs for high speed and high accuracy machining. Other benefits of hexapods over conventional machine tools include: the advantage of 3-D sculpting and reduced number of fixtures and setups because of the enhanced dexterity; simpler maintenance and easy reconfiguration due to six identical axes. Their potential as the next generation machine tools was first proposed by Tindale in 1966 [Tindale, 1966]. This potential is being actively explored by the National Institute of Standards and Technology (NIST) with the octahedral Hexapod made by Ingersoll Milling Machine Company, by Sandia National Lab with the geodetic Hexapod made by Hexel Corp., by the University of Nottingham with Variax made by Giddings and Lewis, and many others [Aronson, 1997; Moriwaki, 1999]. A more detailed discussion on the development of parallel kinematics machines is presented in Chapter 2.

Most of hexapod machine tools are still prototypes under tests and evaluations. Experiments are being carried out on those machines to achieve a basic understanding of performance in terms of speed, accuracy and stiffness compared to those of conventional machines. According to Aronson, the hexapod is five times more rigid, four times faster, and two to three times better in accuracy than a conventional machine tool [Aronson, 1997]. These results showed the promising capabilities and provide strong support for their applications in practical production. However, there are also some technical problems originating from their inherent disadvantages. These problems must be resolved, so that wide production applications can take place.

1.1 Motivation

One of the major limitations of a hexapod is that its workspace is relatively smaller with a complex shape when compared with that of a serial CNC due to its parallel mechanical structure. The range of tasks is often limited due to this.

In a conventional serial kinematic machine tool, the motion range of each axis is the range of the workspace in that direction. All such ranges typically form the overall

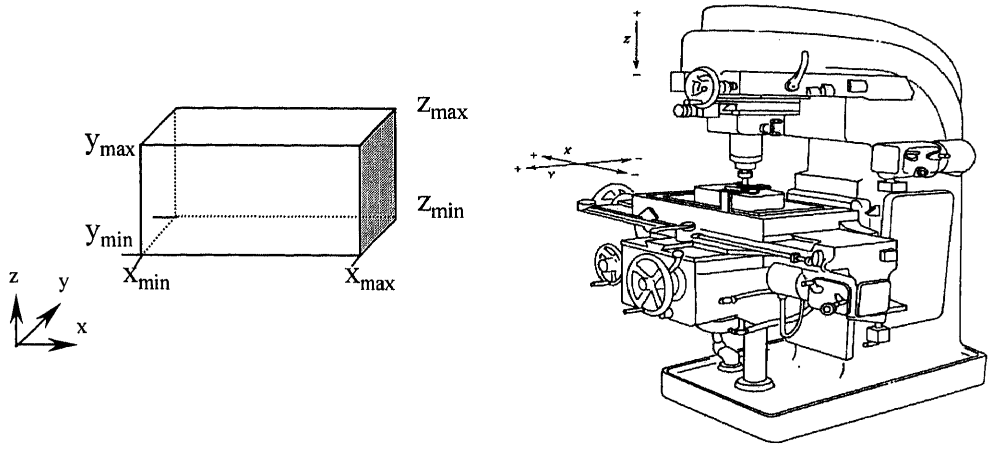


Figure 1.1 A three-axis conventional machine and its workspace

workspace. Therefore large workspaces are easily obtained. For a three-axis conventional machine tool as shown in Figure 1.1, the motion ranges $[x_{\min}, x_{\max}]$, $[y_{\min}, y_{\max}]$, and $[z_{\min}, z_{\max}]$ in the x, y, and z axes determine a cubic workspace directly without coupling. If two rotational degrees of freedom are added to form a five-axis machine, the orientation capability is the same across the position workspace. However, in a parallel kinematic machine, all the leg axes together determine the range of the workspace in any direction. Because of the coupling, the shape of the workspace is highly irregular. Figures 1.2 and 1.3 show respectively the position workspaces of the ACROBAT and Delta hexapods with fixed the spin parameter [Negri, et al., 1999]. The physical limits on the passive joints and the need to avoid interference among the legs put additional constraints on the actual workspace range. In addition, the orientation capability of PKMs varies a great deal within their workspace.

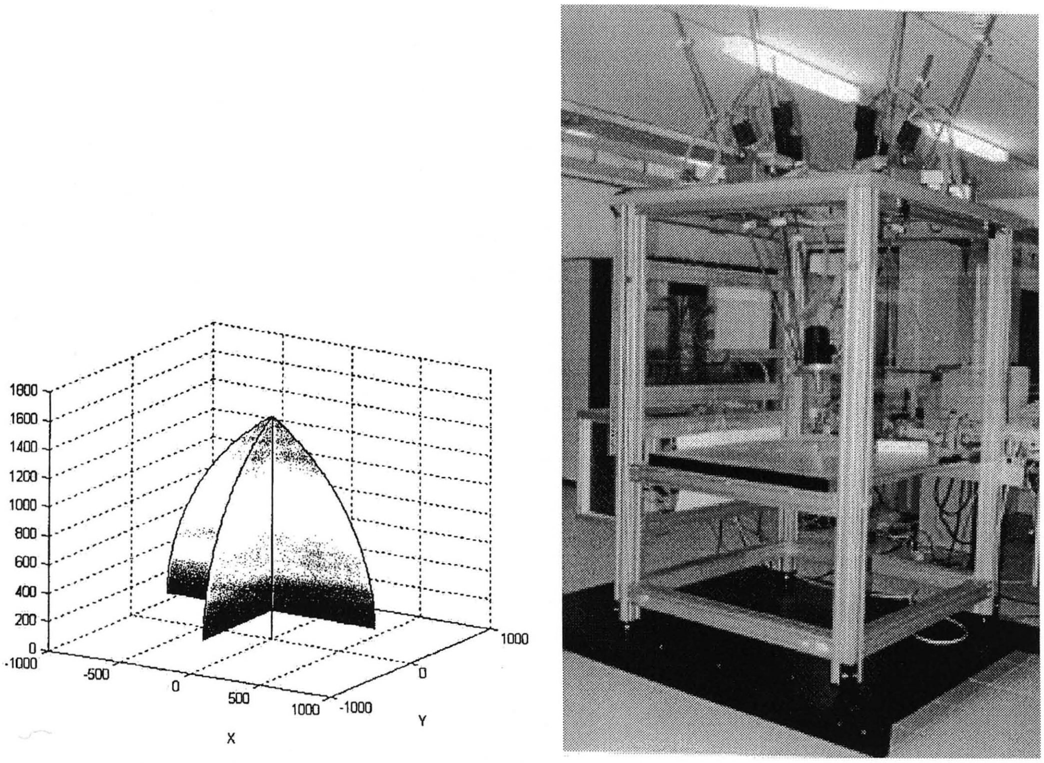


Figure 1.2 ACROBAT hexapod and its workspace [Negri, et al., 1999]

Optimizing the design parameters and increasing the machine size alone may not be sufficient to enlarge the range of tasks for the PKMs. Machines that can be easily reconfigured to adapt to diverse workspace requirements of different tasks provide an attractive alternative. Modular design provides a basis for adaptability of machines and has already been pursued for serial machines. Examples include the Reconfigurable Module Manipulator System RMMS [Schmitz et al., 1988; Paredis et al., 1996], the modular robot [Cohen et al., 1992], the Toshiba Modular Manipulator System (TOMMS) [Matsumaru, 1995], and the modular robot system [Wurst, 1986]. Similar effort has not been seen for parallel kinematic machines. To study the reconfiguration issues, a modular reconfigurable experimental hexapod machine has recently been designed and constructed (Fig. 1.4) in the Department of Mechanical Engineering at New Jersey

Institute of Technology [Ji and Song, 1998]. This hexapod has a base and mobile plate with multiple placement patterns, and six identical leg modules with easy installation and separation design. Different workspace requirement can be met by placing the legs in proper locations. The re-configurable design only provides a basis for reconfiguration. Successful reconfiguration also requires planing and identification methods/algorithms to guide users to find a feasible or most efficient configuration for a desired workspace and to obtain the true placement parameters of the new configuration for correctly controlling the machine. Part of this dissertation study is to develop such methods.

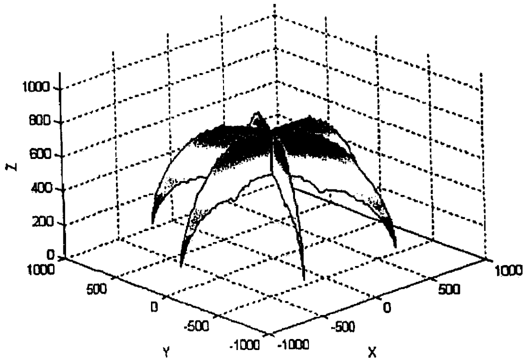


Figure 1.3 Workspace of Delta hexapod [Negri, et al., 1999]

Another important problem related to effective production applications of hexapods is how to plan the tool paths for these new machines. As tool path planning is an essential task in the operation of all NC and CNC machine tools, there is no exception for the hexapod machines. Many tool path planning methods have been used or proposed for conventional CNC machine tools [Lee, 1998; Lin and Koren, 1996]. However, direct use of those planning methods for the hexapod machines may not be sufficient in view of the dramatic difference in their kinematic structures. It is therefore of great importance to have a comprehensive understanding of their motion behavior and their effect on the tool path.

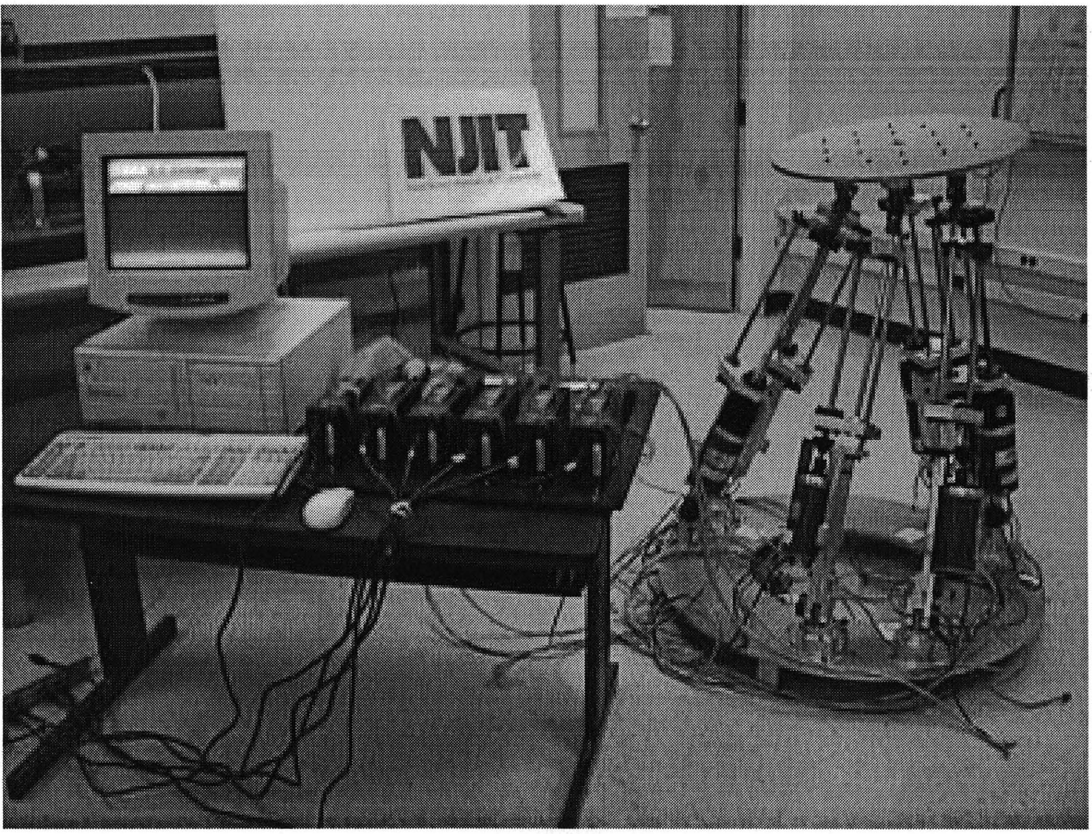


Figure 1.4 A modular reconfigurable experimental hexapod

The first significant difference is the existence of an extra degree of freedom (e-DOF), which is the rotation of a hexapod machine tool about its tool axis. It represents a redundancy for 5-axis machining but must be specified for controlling the motion of the hexapod. If properly used, this e-DOF could improve the geometric and kinematic conditions of the tool paths. During the course of this dissertation study, we have found that the hexapod machine tools have highly non-uniform kinematic conditions across their workspace. There are regions with poor stiffness and accuracy inside their workspace. Sarma and Slocum [1999] also observed that the maximum achievable velocity of their Hexel Hexapod varies greatly within the workspace. In addition, a hexapod machine usually generates a nonlinear segment when following a straight-line with a commonly used motion profile, unlike a conventional CNC, which can follow a straight-line segment very well. This nonlinear segment reduces the accuracy when the sampling length is relatively large in high speed machining. The tradeoff between the accuracy and speed must be carefully considered [Ji and Leu, 1999]. All these factors indicate that the traditional path planning methods should not be used for hexapod machines without modification. The hexapod machines' potential for faster, stiffer, and more accurate operations might not be realized if the tasks are not well planned. Development of tool path planning methods with the consideration of those factors becomes the key to the efficient usage of hexapod machine tools and is another part of this dissertation research.

1.2 Objectives

One objective of the dissertation is to develop methodologies to guide the reconfiguration. The following are the two main tasks to achieve this goal. They will be described in detail in Chapters 3 and 4, respectively.

1. To develop methods and algorithms to guide the placement of leg modules to obtain a feasible or the most effective configuration for the intended operation. As the shapes of workspace of PKMs are complex, it is highly valuable to have such methods to assist the placement.
2. To identify the placement parameters of leg modules on the base.

Another objective of the dissertation is to obtain a fundamental understanding of the tool path planning and path execution of hexapod machine tools, and to develop a new method to plan the tool paths for hexapods. The following two tasks will be performed in order to achieve these goals.

1. To analyze the motion characteristics and dissimilarities between hexapods and five-axis machines in the tool path planning and execution, and identify the problems of using conventional five-axis planning methods for hexapod machine tools.
2. To develop a new tool path planning scheme that accounts for these dissimilarities and implement its main components.

The overall objective of this doctoral dissertation is to enhance the ability of hexapods so that they can be effectively integrated into production systems.

1.3 Organization of this Dissertation

This dissertation is organized as follows. Chapter 2 presents the background knowledge of parallel kinematic machines. The reconfiguration of hexapod machine tools is the subject of Chapters 3, and 4. Chapter 3 defines the foot-placement problem and presents a method for determining individual Foot-Placement Spaces (FPSs) for a given set of leg modules based on a desired workspace. A method to identify position and orientation of base joints is described in Chapter 4, when the six legs of a hexapod are individually installed to a new base or different locations of the installation site. Chapters 5 and 6 are aimed at the tool path planning of hexapod machine tools. Chapter 5 discusses traditional tool path planning methods and new issues associated with their use in hexapods. A kinematics-based tool path planning scheme along with its simulation and examples of its implement for hexapods are described in Chapters 6. Finally in Chapter 7, conclusions and suggested further investigations are presented.

CHAPTER 2

PARALLEL KINEMATIC MACHINES

This chapter looks back into the history of parallel kinematic machines and gives a brief summary of recent advances in their research and development. It is the background knowledge to the dissertation.

2.1 Historical Developments

Parallel kinematic machines (PKMs) are the machines whose mechanisms are formed with two or more in-parallel subchains of links and joints. In a parallel kinematic machine, the first and last links of each subchain are connected together through a joint, and these subchains connect a base plate with a mobile plate. Hexapod machines are one type of PKMs with their six degree-of-freedom (6-DOF) obtained through controlling the lengths of the six legs as shown in Figure 2.1. Compared with their serial counterpart, this parallel structure offers superior stiffness and low mass because of several closed subchains sharing the load. They have high precision due to no cumulative errors, high acceleration, but smaller workspace and lower dexterous maneuverability. Parallel kinematic machines are also known as Stewart Platforms, Gough Platforms, parallel manipulators, platform manipulators, and hexapod machines. Although the terms have often been used interchangeably for any parallel kinematic machines, some of the terms are associated to a particular type of parallel kinematic machines.

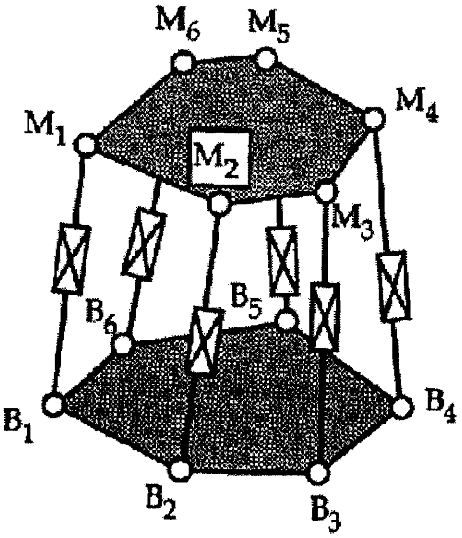


Figure 2.1 A 6-DOF parallel kinematics machine (a hexapod)

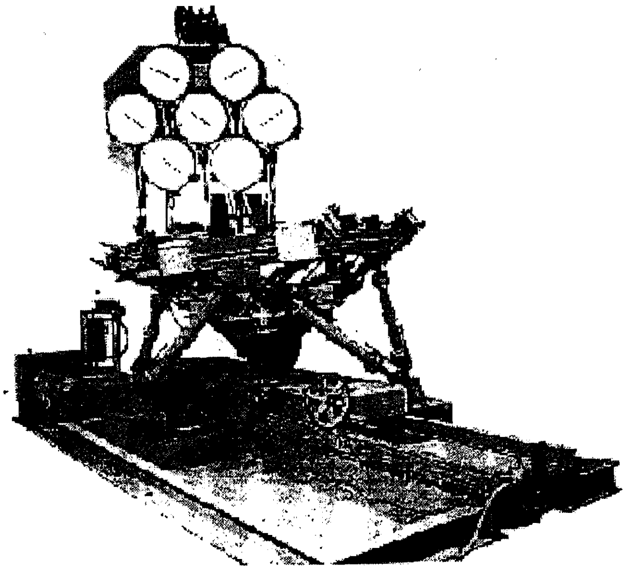


Figure 2.2 Gough's Universal Tire Testing Rig

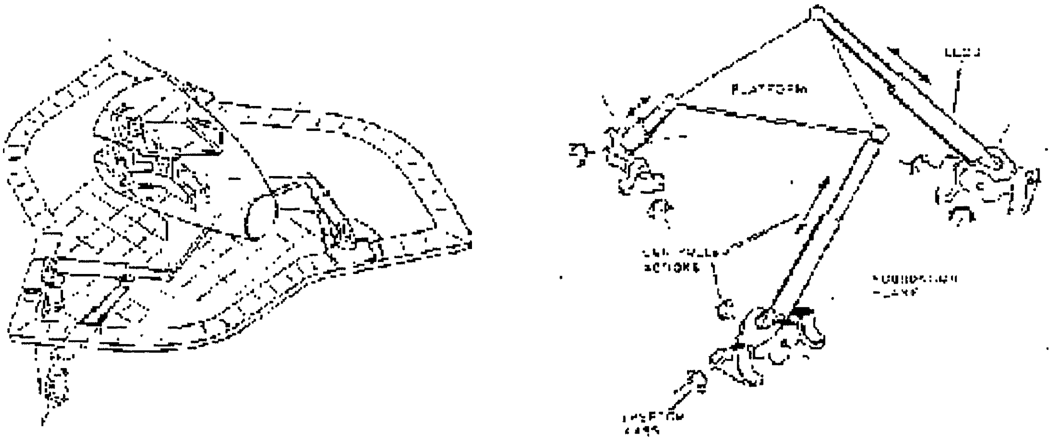


Figure 2.3 Stewart's Aircraft Simulation Platform

Although the history of parallel structure machines can be traced back to a long time ago, they have not attracted worldwide interest until recent years. Cauchy's idea of an "articulated octahedra" in 1813 represents perhaps the earliest interest in the parallel structures [Merlet, 1994]. It is Gough's work on a tire test apparatus in 1947 that started the multi-degree-of-freedom parallel mechanisms. Gough's effort produced the Universal Tire Testing Rig [Gough, 1956] as shown in Figure 2.2, which presses tires on a rolling belt in different postures for studying the wear of the tire. The most well known parallel mechanism with six degrees of freedom was introduced in 1965 by Stewart for use as a flight simulator [Stewart, 1965] (Figure 2.3). Because of this work, now the parallel manipulator is popularly known as Stewart platform. Hunt is the first to suggest its use as a manipulator in 1978 [Hunt, 1978], and he also discussed some alternative mechanical designs of the Stewart platform for robotic applications in 1983 [Hunt, 1983]. Yang and Lee further made a theoretical investigation on the feasibility of using a platform mechanism as a robotic manipulator [Yang & Lee, 1984]. Fichter performed a detailed study on a Stewart platform type manipulator. He theoretically solved the inverse kinematics problems and also addressed a group of practical problems when a real machine was built [Fichter, 1986]. Behi presented the kinematic analysis of a parallel mechanism as a general-purpose spatial manipulator arm [Behi, 1988]. Merlet began comprehensive investigations on the kinematics and workspace of parallel manipulators in his Ph.D. thesis study in 1986, and studied singular configurations of parallel manipulators in 1989 [Merlet, 1989].

2.2 Theoretical Developments

Since the end of 1980s, the Stewart platform has attracted the attention of more and more researchers due to its simplicity in achieving sufficiently many DOFs, large payload capacity, high stiffness and accuracy, potential for wide applications, as well as the advancement in computation and control. The successful development of the applications is closely related to the tremendous amount of work on the fundamentals of parallel kinematic machines.

Early research focused on kinematics. Without kinematics algorithms, we can not control PKMs correctly and don't know the actual locations of the machines. Inverse kinematics is straightforward for parallel manipulators, but forward kinematics is much more difficult. There is a very long list of publications on this subject and various algorithms have been presented, but closed-form solutions have only been discovered for some very special cases [Innocenti and Parenti-Castelli, 1990; Lee and Roth, 1993]. Raghavan showed that even in the most general case there will be no more than 40 solutions [Raghavan, 1993]. One practical way to solve the direct kinematics problem is to add appropriate passive sensors in the links or joints [Merlet, 1993; Parenti-Castelli and Gregorio, 1995; Cheok, Overholt, and Beck, 1993; Ji and Song, 1998].

Another heavily studied topic in parallel platforms is workspace determination. We should know the workspace of a machine in design stage in order to understand the scope of operations. The workspace of a parallel platform is defined as the possible positions and orientations of the mobile plate of the platform. There is no convenient way to represent the workspaces of parallel kinematic machines, since the position and orientation of the mobile plates are closely coupled. Graphic visualization has been used extensively to display workspaces of parallel kinematic machines, particularly those with six-degrees-of-freedom. The results are presented in the form of either the layered boundaries or the volume for a fixed orientation. This descriptive approach is useful for characterizing given designs. Workspace analysis of a 6-DOF manipulator, is often split up into two simpler problems. One is to determine the 3D reachable position space of the mobile plate for a specified orientation. Gosselin, et al. presented an algorithm for solving problems of this type and applied a wire-frame representation, which allows a fast graphical response [Gosselin, et al., 1992]. The other way is to compute the possible rotations around a fixed point. Merlet used a fixed-length link attached to the mobile plate to compute the reachable regions of the extremity of this link, and thus the rotation space can more easily be understood [Merlet, 1995a]. A concept of vertex space has also been introduced by Ji in 1996 to decompose complex workspace problems into simpler subproblems [Ji, 1996].

Singular configurations within the workspace cause uncontrollable degrees of freedom that should be avoided. Ma and Angeles (1991) classified the singularities into three types: architecture singularity, configuration singularity and formulation singularity. Some singular configurations have been identified [Ma and Angeles, 1991; Merlet, 1989; Fitchter, 1986]. Many researchers have conducted analyses to establish the relationships between the position parameters and singular configurations (singular map or loci).

2.3 Machine Tool Applications

With many theoretical barriers being overcome and having controllers capable of computing the leg lengths in real-time available, many application prototypes of parallel manipulators have been developed. Their sizes range from the micro manipulator of a few nanometers in motions to the huge manipulator used for mining purposes, and the intended application areas range from surgical operations, flight simulation, to various manufacturing devices [Merlet, 1994]. The introduction of hexapods, as new types of machine tools, is really an innovation in the machine tool industry for a long time. They have shown great potentials as the next generation machine tools [Aronson, 1997].

In a conventional machine tool, motions are obtained from linear and rotary axes arranged in serially. Each axis provides an independent degree of freedom. In hexapod machines, motions are obtained from six parallel-actuated linear axes. This arrangement presents two major advantages: High accuracy and high rigidity. Unlike conventional machine tools, hexapods do not accumulate errors; the worst single leg error is almost the worst machine error. The price paid for the rigidity of conventional machine tools is the heavy mass, which is unevenly distributed among the axes because of the serial arrangement of axes. Acceleration and deceleration of a heavy mass is frequently a problem in following complex tool paths. Although five-axis machines have the capability to orient tools, the arrangement of their two orientation axes is not very efficient. They often weaken the rigidity of the machine tools. However, for a hexapod machine tool, six legs with only tension and compression stresses share the load; the overall rigidity of the machine is the sum of the contributions from each leg. According to Aronson, a hexapod machine tool is five times more rigid and two to three times better in accuracy than a conventional machine tool [Aronson, 1997]. This makes it possible to cut difficult materials more accurately.

Another advantage of a hexapod machine tool is the true 6-DOF compared with conventional 5-axis machines, which offers advantages in 3D sculpting and reduces the requirement on fixtures and setups in complex parts machining. The reduced mass of moving parts provides higher speeds and accelerations. And six identical axes make the machine easier to maintain, and convenient for modular design and reconfiguration to meet diverse process requirements.

Several companies in the U.S. and U.K. began developing hexapod machine tools in the late 80's, and released their prototypes in the early 90's. They were built to provide a basic understanding of machining performance and to be compared with conventional machine tools. The first few hexapod machine tools were exhibited at the Chicago International Machine Tool Show in the Fall of 1994 (IMTS'94). One (called Variax, Figure 2.4) was built by the U.S. firm, Giddings and Lewis, and the other by an U.K. company, Geodetics. The Variax machine is used to determine the feasibility of manufacturing small complex parts for the aerospace industry. Figure 2.5 shows the structure of the Hexapod made by Ingersoll Milling Machine Co. The main feature of this machine is that a patented octahedral frame is used to hold the hexapod and small spindle platform. Hexel Co. produces the hexapod type machines called Tornado 2000 milling machine. The potential main applications of the machine are mold work with an abundance of surface contouring.

Other countries such as Japan, Russia, Germany, China, and France also show a great interest in developing hexapod machine tools. Russia exhibited a hexapod machine tool in the 1996 Beijing International Machine Tool Show.

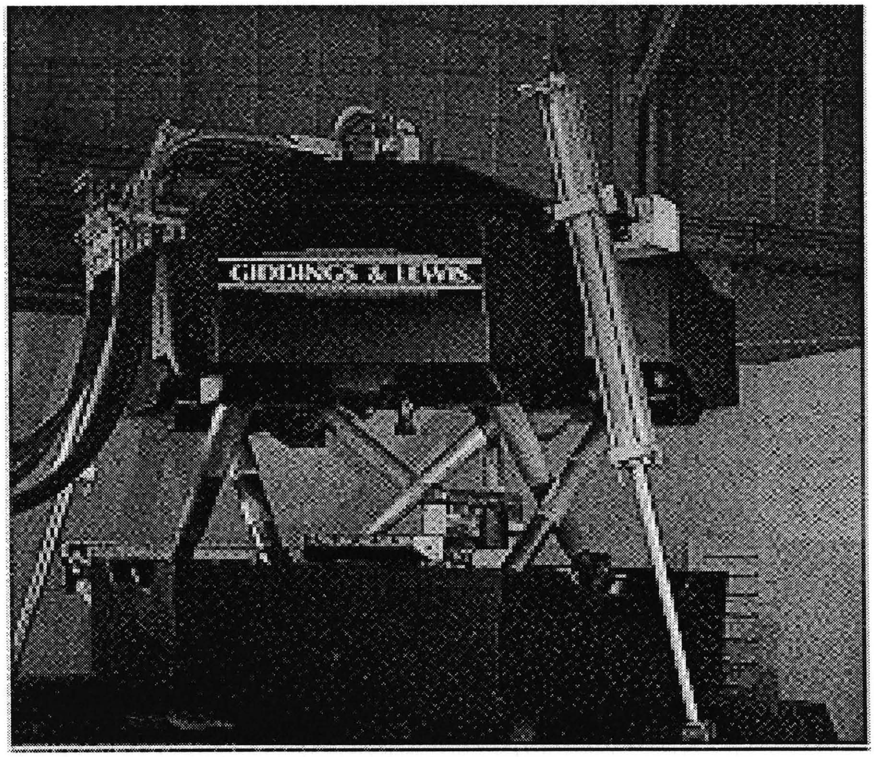


Figure 2.4 Giddings and Lewis - Variax

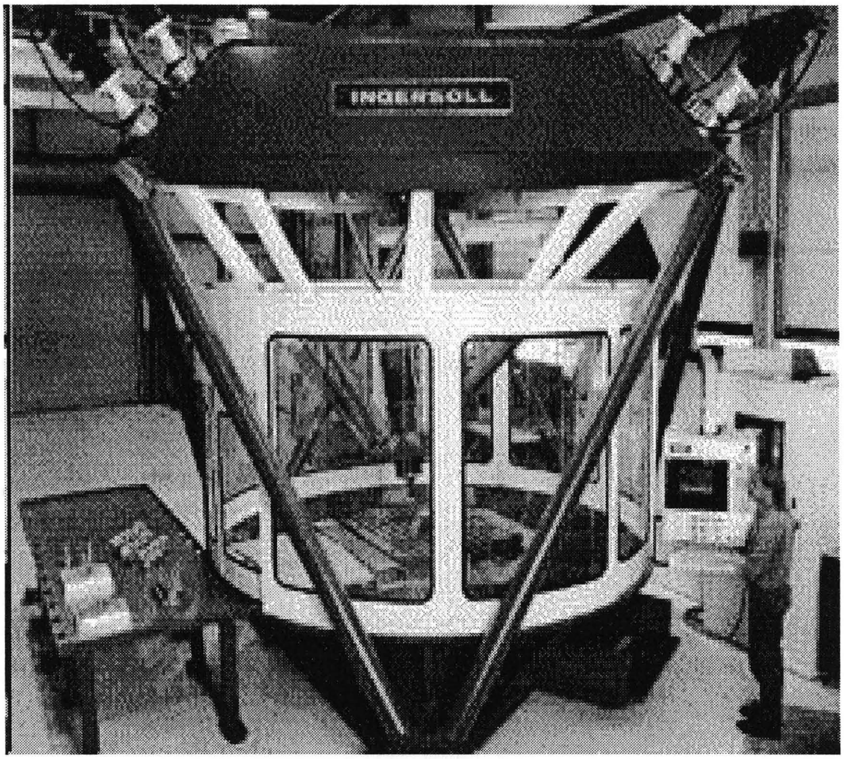


Figure 2.5 Ingersoll Hexapod

2.4 Current Activities

Although the hexapod machine tool is a great innovation for the manufacturing industry, much additional research is necessary to bring hexapods into practical production. The goal of the current research activities is to get a basic understanding of the characteristics of hexapods in terms of accuracy, speed, stiffness, productivity and quality, and to develop the machine utilization methods, which can make full use of their advantages and overcome the limitations.

One active research area is to understand the characteristics of hexapod machines through metrology and compare their performance with that of current production machines. The hexapod builders and owners are the main participants in this activity. Giddings & Lewis tested its Variax machine using ISO and ANSI cutting standards and showed that the machine was able to cut aluminum at 38mm/min normally running at 0.5-0.6m/min, with an acceleration of 1g, and 15micron accuracy. The dynamic stiffness is 613kN/cm, which is five times that of conventional machine tools. A Hexel Tornado 2000 has an accuracy of 25micron and repeatability of 10micron, and the machine can have a feet rate of 300mm/sec with a feet force of 200 kg. To appraise the full capability of the machines properly and effectively, comprehensive and systematic tests are required, as industrial acceptance of hexapods is dependent on the ability to demonstrate the capabilities of these machines in a formalized manner. The Rapid Response Aerospace Manufacturing group at the University of Nottingham, England, did lots of tests on the Variax. These include the standard, nonstandard, static, and dynamic tests. The results showed the Variax is capable of achieving high standards of accuracy. They suggested a new standard of tests for hexapods in order to obtain more pertinent information. NIST and Sandia National Laboratory have extensive programs for evaluating various performance measures. They measure positioning accuracy and repeatability of the machines and determine how these items vary at different positions and directions in the workspace. They plan to perform sensitivity analysis to explore the workspace for the positions and directions of the greatest and least geometric sensitivity. So far, each group only tested a particular machine at a few positions and orientations using limited test methods. Comprehensive comparison can only be made after systematic evaluations are finished.

Another active research area is aimed at helping hexapods become practical machines. Performance enhancement is the main task in this phase. Calibration and error compensation are two commonly used methods to increase the accuracy of a machine.

Because hexapods have highly nonlinear kinematic models, the techniques for conventional serial machines are not readily applicable to them. Zhuang and Roth proposed a calibration method for the hexapod to identify the length of each leg by taking it to be fixed while the other legs stretch or shrink [Zhuang and Roth, 1993]. Models including manufacturing tolerances, installation errors and link offsets were formulated by Wang, et al. [Wang & Masory, 1993; Masory, et al., 1993]. They also demonstrated a method to identify these errors using pose measurements and the effects of these factors on machine accuracy. Ji showed that the traditional roundoff does not always give the best position accuracy, and presented a Jocabian based roundoff procedure for determining the round up or down of each controlled joint in parallel kinematic machines [Ji, 1997]. Other research issues in this area include static and dynamic system modeling and identification, parameter characterization, and predictive model-based control.

The performance of a hexapod depends not only on its machine structure and build (hardware), but also on its planning and control (software). Therefore, how to use them correctly and effectively is another major research area. The goal is to integrate them into the current CAD/CAM production environments. Since hexapods have complex workspaces, their rigidity, accuracy, and static and dynamic properties are not uniform within their workspace. All the existing support functions and systems for conventional machine tools are not necessarily suitable for hexapod machines. The University of Nottingham is currently developing a system that includes workspace analysis, component positioning, setup planning, material removal and tool path simulation with collision detection. Tool path planning is an essential part of integration into a CAD/CAM system and directly affects the quality of machined parts. However, very little work has been done on it so far.

To make hexapod machine tools practical, operation range enhancement through reconfiguration is another very important issue, which seems not to draw enough attention. This kind of technology is highly desirable for hexapods as the range of tasks is often limited by their smaller workspace. Meanwhile, the six almost identical leg modules provide better conditions for modularization and reconfiguration. The difficulty is the lack of reconfiguration methods and algorithms to support this technology.

This dissertation focuses on the reconfiguration and tool path planning of hexapod machine tools. The solutions provided by this research will enable hexapod machine tools to be used more effectively with a wider operation range.

CHAPTER 3

DETERMINATION OF INDIVIDUAL FOOT-PLACEMENT SPACE

The kinematic structure of the NJIT hexapod (Fig. 1.4) is similar to other fully parallel kinematic machines, except that it has features designed for studying issues related to reconfiguration. It consists of a base, a mobile platform on which tools and equipment are mounted, and six parallel-actuated extendible legs between the mobile platform and base. Each of the leg modules is self-contained: it is completed with one actuated prismatic joint for changing its length, a spherical joint on upper end, and a Hook joint on the lower end. The ends of a leg module have been provided with means for easy attachment to and separation from the mobile platform and the base through fasteners. Although a base with multiple placement patterns is provided in the setup, the installation site can be user-defined according to applications. That is, any of the leg modules can be easily replaced by another with a different range of motion, and can be placed on the mobile platform and the base at any desired location and orientation. Our previous study on design parameters in the platform manipulators has shown that the moving range of the legs as well as the placement of the legs has a great effect on the shape and size of the workspace [Ji, 1995 & 1996]. The modular design of hexapods also provides reparability and portability since each module can be individually replaced, or separately transported to a different location and reassembled.

The first step of reconfiguring a modular parallel kinematic machine is to select an actual configuration for task requirement. Placement issues related to the selection of a configuration are discussed in this chapter. Then foot-placement space is defined and a method for determination of individual foot-placement spaces for a given set of leg modules, based on a desired workspace is presented. An implementation algorithm for constructing foot-placement space with examples is also presented.

3.1 Placement Issues of Leg Modules

In choosing a configuration, the parameters to be selected include the position and orientation of the legs' upper joints on the mobile platform, the order of leg modules on the mobile platform if they are available in different moving ranges, and the position and orientation of the feet of the leg modules on the installation site. A good configuration can only be realized after we have methods to deal with the following three issues.

1. How to determine the position and orientation of joints on the mobile plate? The position and orientation of joints on the mobile platform essentially determine the size and shape of the mobile platform. Previous studies on design parameters of the platform manipulators suggests that the size of a mobile platform be kept small for larger orientation space and smaller footprint of the base [Ji, 1996]. Since a mobile platform also needs to accommodate an end-effector or tooling, its shape and size should be selected first.
2. How to select the proper set of leg modules? As the available leg modules usually are of different moving ranges, their use in a hexapod can have different combinations and orders with different workspace performances.

3. How to determine the position and orientation of joints on the base plate? For a given set of leg modules, after placing them onto the mobile plate, the placement of these legs onto the base plate also needs to be defined.

Since the first issue relates mainly to the type of tools or end-effectors used, it is addressed in the design stage. The second issue arises only if there are different leg modules to select from, and can be evaluated based on the solutions to the third issue. Therefore this dissertation concerns only the third issue. The following sections of the chapter will describe the construction of foot-placement space and present a discrete numerical algorithm for the determination of the foot-placement space. Examples are included to illustrate the process.

3.2 Definition of Foot-Placement Space

The Foot-Placement Space (FPS) is defined as follows: For a leg connected to a mobile platform, the leg's foot-placement space (FPS) corresponding to a given desired workspace of the mobile platform is the set of base locations where the foot of the leg can be placed to ensure the prescribed workspace of the mobile platform.

That means that if every leg is placed inside its foot-placement space then the resulting configuration will be able to provide the desired workspace. Note that a FPS is defined for a specific workspace, and a different workspace has a different FPS. Information about the FPS can be very useful for the placement of individual legs and reconfiguration of the parallel kinematic machines.

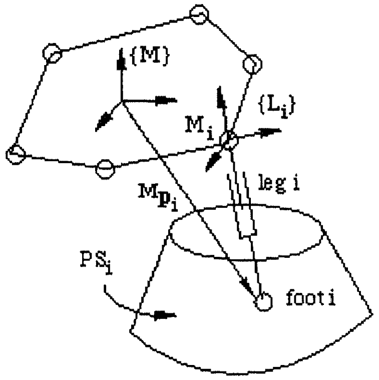


Figure 3.1 Reachable space for the foot of leg *i*

3.3 Construction of Foot-Placement Space

In order to determine the Foot-Placement Space, we should construct this FPS and obtain its expression based on the given required workspace and the mechanical motion limits of the leg modules. Let's first consider the space that is reachable by the foot of a leg connected to a mobile platform. As shown in Fig. 3.1, ${}^{M}\bar{p}_i$ is the position vector of the foot of leg *i* in the coordinate frame $\{M\}$ attached to the mobile platform. $\{L_i\}$ is a coordinate frame of the leg *i* attached at the center of its spherical joint. The point M_i denotes the location at which the origin of the frame $\{L_i\}$ is attached to the mobile platform. Position ${}^{M}\bar{p}_i$ can be expressed in terms of the orientation angles θ_i and ϕ_i and length l_i of the leg. The ranges of θ_i and ϕ_i depend on the motion limits of the upper spherical joint at M_i and the range of l_i is between $l_{i,\min}$ and $l_{i,\max}$. After taking the joint limits into consideration, all the feasible values of θ_i, ϕ_i and l_i generate a volume in $\{M\}$. Let's denote the space occupied by this volume as ${}^{M}PS_i$, which is an invariant space in $\{M\}$.

If the position and orientation of the mobile frame $\{M\}$ in the base frame $\{B\}$ are described by $\bar{x} = (x, y, z, \alpha, \beta, \gamma)^T$, where $(x, y, z)^T$ and (α, β, γ) are respectively the position vector and the set of Euler angles, then ${}^{M}PS_i$ can be mapped into frame $\{B\}$ with transformation matrix ${}^{B}T_M(\bar{x})$ as

$$PS_i(\bar{x}) = {}^{B}T_M(\bar{x})\ {}^{M}PS_i \tag{3.1}$$

Let $\bar{x}_1$ and $\bar{x}_2$ represent any two poses of the mobile platform in $\{B\}$. Then $PS_i(\bar{x}_1)$ and $PS_i(\bar{x}_2)$ each is the respective set of locations reachable by the foot of leg i from one of the two poses. The location that is reachable by the foot from both poses must belong to both $PS_i(\bar{x}_1)$ and $PS_i(\bar{x}_2)$, *i.e.* the intersection $PS_i(\bar{x}_1) \cap PS_i(\bar{x}_2)$. The shaded volume in Fig. 3.2 illustrates the space reachable by the foot from two poses $\bar{x}_1$ and $\bar{x}_2$ of the mobile platform. Generalizing this concept, the foot-placement space for leg i can be expressed as

$$\begin{aligned} FPS_i &= \bigcap\{{}^{B}T_M(\bar{x})^{M}PS_i; \bar{x} \in WS\} \\ &= \bigcap\{PS_i(\bar{x}); \bar{x} \in WS\} \end{aligned} \tag{3.2}$$

where WS denotes the desired workspace.

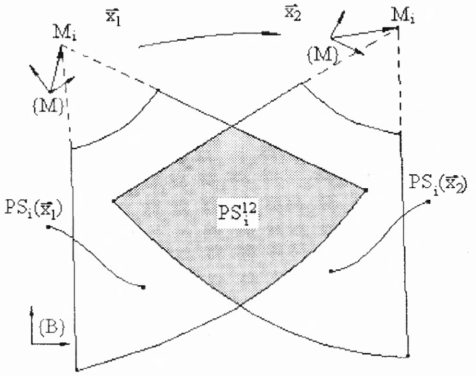


Figure 3.2 Reachable space determined by two poses

Both WS and ${}^{M}PS_i$ represent continuous spaces. While Eq. (3.2) provides the mathematical formation of foot-placement space FPS_i, which is also continuous, a practical method has to be developed to construct this space. Let us discretize WS into a set of poses $\vec{x}_k$ $(k = 1,\cdots,K)$, then Eq. (3.2) can be approximated as

$$\begin{aligned} FPS_i &= \bigcap_k \{{}^{B}T_M(\vec{x}_k)\,{}^{M}PS_i ; \vec{x}_k \in WS\} \\ &= \bigcap_k \{PS_i(\vec{x}_k); \vec{x}_k \in WS\} \end{aligned} \qquad (3.3)$$

When the boundary of ${}^{M}PS_i$ is analytical and simple, Eq. (3.3) can be implemented analytically by taking a sequence of intersections. This process is essentially the constructive geometry used in solid modeling, except that only subtraction is involved here. If the resulting FPS_i is a null space, then the desired workspace can not be obtained no matter where the foot is placed. One has to choose a different location for M_i or use another leg of different range.

To find a more general implementation of Eq. (3.3), we decided to also discretize ${}^{M}PS_i$ into a set of points ${}^{M}\bar{p}_{i,j}\,(j=1,\cdots,J)$. The discretization, however, makes it impossible to use the set operation directly, because the objects are now individual points. Therefore, another approach is used instead. We first map all the points ${}^{M}\bar{p}_{i,j} \in {}^{M}PS_i$ for one of the poses, say $\bar{x}_1$, to their positions in $\{B\}$: $\bar{p}_{i,j}\,(\bar{x}_1)={}^{B}T_M(\bar{x}_1)\,{}^{M}\bar{p}_{i,j}$. Naturally $\bar{p}_{i,j}(\bar{x}_1) \in PS_i(\bar{x}_1)$. Now if $\bar{p}_{i,j}(\bar{x}_1) \in PS_i(\bar{x}_k)$ for any $k \neq 1$, then $\bar{p}_{i,j}(\bar{x}_1) \in PS_i(\bar{x}_1) \cap PS_i(\bar{x}_k)$. That means, if there exists a point $\bar{p}_{i,j}(\bar{x}_1) \in PS_i(\bar{x}_k)$ for all $k \neq 1$, then $\bar{p}_{i,j}(\bar{x}_1) \in FPS_i$. All we need to do is to look for those points in ${}^{M}PS_i$ that can pass this test. This process can be carried out numerically. Several implementation issues are discussed in the next section.

3.4 Implementation of Foot-Placement Space Algorithm

The basic steps in the implementation are: (1) map the entire set of points ${}^{M}\bar{p}_{i,j} \in {}^{M}PS_i$ to a pose $\bar{x}_1$ and get rid of those points that do not belong to $PS_i(\bar{x}_k)$ $(k \neq 1)$. The process will be repeated for a new pose with the remaining points, until all the remaining poses are tested.

One can take further advantage of the subtractive search process by using extreme poses to reduce quickly the points to a much smaller set. For that one needs to choose one extreme pose (i.e., pose at the workspace boundary) such as $\bar{x}_1$ to form the point set $\bar{p}_{i,j}(\bar{x}_1)$ and start the search with those poses that are far apart from $\bar{x}_1$. This tactic can also be employed in the set operation of Eq. (3.3) or before the discretization of ${}^{M}PS_i$, if it is not difficult to implement.

The numerical implementation discussed in the previous section will generate a foot-placement space FPS_i in the form of a set of points. While this result can be used in guiding the placement of the foot of a leg directly, a volumetric representation of this space is often desired for the purpose of visualization. With this in mind, a method based on finite-element meshing is used in our discretization of ${}^{M}PS_i$. In this approach, ${}^{M}PS_i$ is divided into a series of elements (*i.e.*, subspaces) and the nodes of these elements are used to form ${}^{M}\bar{p}_{i,j}\ (j=1,\cdots,J)$. Following this preprocess, the numerical elimination process retains those nodes that are inside FPS_i. A postprocess can then be performed based on the connectivity of the elements: an element will be eliminated if all the nodes of the element are outside FPS_i; an element will be kept if all the nodes of the element are inside FPS_i; an element will be a boundary element if some but not all of its nodes are inside FPS_i. By removing all the edges that are connected to an outside node from all of the boundary elements, the remaining parts of the boundary elements will form the boundary of FPS_i.

Since the finite elements generated in the preprocess are not used for engineering analysis, simple four-node tetrahedral elements are sufficient for our discretization. The size of the elements is, however, directly related to the accuracy of the resulting FPS_i, especially at the boundary. A smaller element size produces better accuracy. To be less affected by the accuracy issue, one may want to place the foot of a leg at a location away from the boundary of its FPS_i. Refinement can also be made by further dividing the boundary elements into finer ones and repeating the process on the new elements and their nodes.

All the discussion so far has been focused on the determination of foot-placement space for an individual leg. The same process has to be applied to all six legs to obtain six foot-placement spaces, one for each leg. It is possible that some of them even overlap at certain portion of their spaces.

3.5 Implementation Examples

For our reconfigurable experimental platform, the spherical joints that connect legs to the mobile platform are actually a combination of a revolute joint and a universal joint. Since there is no restriction on the motion of the revolute joint, ${}^{M}PS_i$ is axis-symmetric, with the centerline being the joint axis of the revolute joint. Due to the limit on the motion of the universal joint, ${}^{M}PS_i$ is bounded by a cone C_i of angle ψ_i with its vertex being M_i. As the length l_i of leg *i* changes inside the range between $l_{i,\min}$ and $l_{i,\max}$, ${}^{M}PS_i$ is also bounded by two spheres, $S_{i,\min}$ and $S_{i,\max}$, centered at M_i and with radii being $l_{i,\min}$ and $l_{i,\max}$, respectively. Thus,

$$ {}^{M}PS_i = (S_{i,\max} - S_{i,\min}) \cap C_i \tag{3.4} $$

Figure 3.3 shows ${}^{M}PS_i$ defined in Eq. (3.4) at two poses $\bar{x}_1$ and $\bar{x}_k$ of $\{M\}$ in $\{B\}$, where superscript 1 or k denotes the corresponding pose. Also shown in the figure is a four-node element r. Let ${}^{L_i}\bar{p}_{r,s}$ be the position vector of node s of the element r in leg frame $\{L_i\}$. Also let ${}^{M}\bar{m}_i$ and ${}^{M}R_{L_i}$ be the position vector and the orientation matrix of frame $\{L_i\}$ in $\{M\}$. Then the position of node s of element r in $\{M\}$ can be expressed as

$$ {}^{M}\bar{p}_{r,s} = {}^{M}\bar{m}_i + {}^{M}R_{L_i}\,{}^{L_i}\bar{p}_{r,s} \tag{3.5} $$

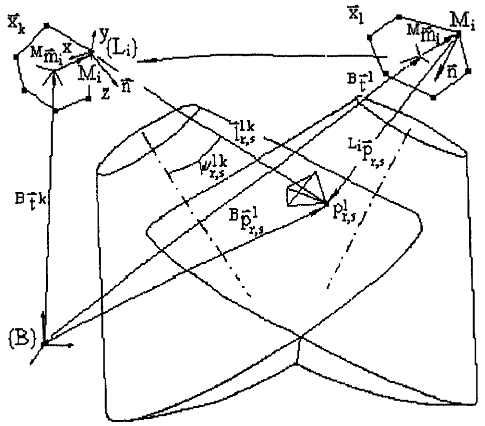


Figure 3.3 ${}^{M}PS_i$ of the example at two poses

When $\{M\}$ is at pose $\bar{x}_1$, the position of node s of element r in $\{B\}$ can be expressed as

$$ {}^{B}\bar{p}_{r,s}(\bar{x}_1) = {}^{B}\bar{t}(\bar{x}_1) + {}^{B}R_{M}(\bar{x}_1)\,{}^{M}\bar{p}_{r,s} \tag{3.6} $$

where ${}^{B}\bar{t}(\bar{x}_1)$ and ${}^{B}R_{M}(\bar{x}_1)$ are respectively the position vector and the orientation matrix of $\{M\}$ in $\{B\}$.

Let $l_{r,s}^{1k}$ and $\psi_{r,s}^{1k}$ be respectively the distance between ${}^{B}\vec{p}_{r,s}(\vec{x}_1)$ and ${}^{B}\vec{m}_i(\vec{x}_k)$ and the angle between the symmetric axis of $PS_i(\vec{x}_k)$ and the line connecting the two points. Both of them can be easily obtained. The conditions for ${}^{B}\vec{p}_{r,s}(\vec{x}_1) \in PS_i(\vec{x}_k)$ can be expressed as $l_{i,\min} \le l_{r,s}^{1k} \le l_{i,\max}$ and $\psi_{r,s}^{1k} < \frac{\psi_i}{2}$.

The parallel manipulator used for the numerical example is defined by the following data: For the leg $i = 1$: $l_{i,\min} = 0.85$, $l_{i,\max} = 1.20$, and $\psi_i = 120°$. The corresponding ${}^{M}PS_i$ is then meshed into 3789 tetrahedral elements with 941 nodes as shown in Fig. 3.4.

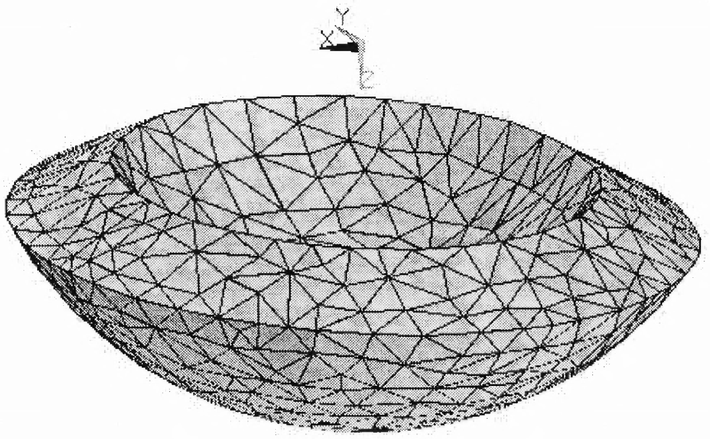


Figure 3.4 Meshed form of ${}^{M}PS_i$

The attachment of the leg to the mobile platform is given in terms of a transformation matrix as

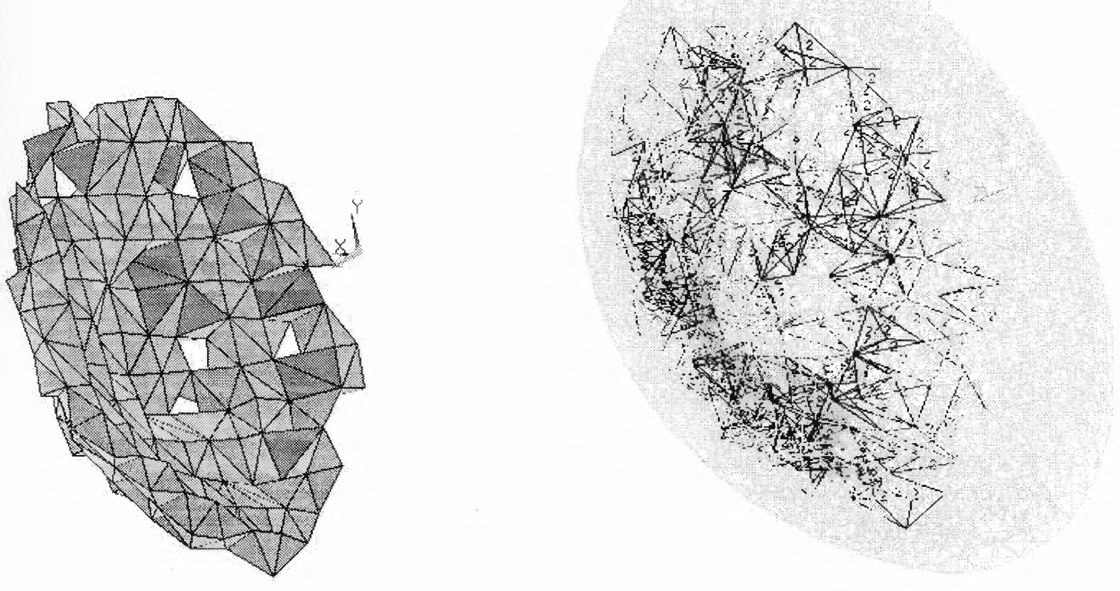
$$^{M}T_{L_i} = \begin{bmatrix} ^{M}R_{L_i} & ^{M}\bar{m}_i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0.2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

The required workspace for one example is as follows. The position space is a rectangular block defined by $-0.1 < x < 0.0$, $-0.1 < y < 0.0$, and $0.9 < z < 0.95$. The orientation space is $\alpha = \beta = \gamma = \pm 2.5^\circ$ (roll, pitch, yaw or fixed angles) for every position within the block. The workspace is then discretized into 512 different poses.

After applying the above data in the program developed for determining foot-placement space, we obtained 458 interior elements as shown in Fig. 3.5(a). Their position in the original $^{M}PS_i$ is shown in Fig. 3.5(b). No further process is conducted on the boundary elements or on the refinement of the result for this example.

The required workspace for the second example is as follows. The position space is a rectangular block defined by $-0.1 < x < 0.0$, $-0.1 < y < 0.0$, and $0.9 < z < 1.0$. The orientation space is $\alpha = \beta = \gamma = -2.5^\circ \sim 0^\circ$ for every position within the block. The workspace is also discretized into 512 different poses.

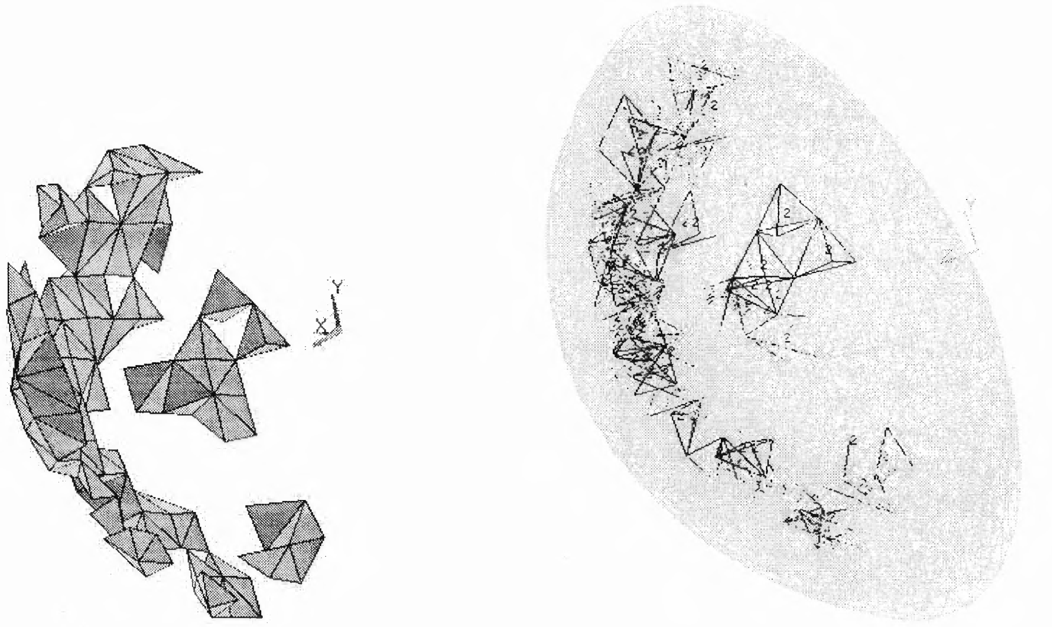
For this example, we obtained 173 interior elements as shown in Fig. 3.6(a). Their position in the original $^{M}PS_i$ is shown in Fig. 3.6(b). No further process is conducted on the boundary elements or on the refinement of the result for this example either.



(a) FPS

(b) FPS mapped into ${}^{M}PS_i$

Figure 3.5 Result of example 1



(a) FPS

(b) FPS mapped into ${}^{M}PS_i$

Figure 3.6 Result of example 2

3.6 Summary of the Chapter

As a utility to guide the reconfiguration of module PKMs, this chapter defined and presented a construction method for Foot-Placement Space (FPS). Based on the desired workspace of the mobile platform and the range of motion of the leg modules, the program will provide information on the feasible locations for the feet of the legs so that the required workspace can be realized. When the placement of the PKM is intended for several applications, one can find either the intersection of the Foot-Placement Spaces for every application or the Foot-Placement Space for their combined workspace. Several implementation issues were discussed for improving the efficiency of computation and for visualizing the result. Two examples were provided to illustrate the process of Foot-Placement Space determination.

CHAPTER 4

IDENTIFICATION OF PLACEMENT PARAMETERS

Chapter 3 provided a method to guide the placement of base joints of leg modules. After the six legs of a hexapod are individually installed to a new base or different locations of the installation site, the true values of the position and orientation of base joints may not be precisely known to the controller and often inconvenient to be measured directly. In this chapter, a method to identify position and orientation of base joints for modular PKMs is described. Through the measurement of the mobile plate pose, orientation change of base joints, and the leg length increment of prismatic joints leads to a set of nonlinear equations for the identification problem. Methods for solving the identification equations are discussed. The method of Dialytic Elimination using symbolic manipulation is presented for a special situation. The identification process is illustrated with an example.

4.1 Introduction

Parameter identification can be applied to find the true values of the position and orientation of base joints. It is a process where the parameters are computed from the kinematic model of a machine based on a set of pose measurements. Mooring et al., and Zhuang and Roth discussed a four-phase calibration including modeling, measurement, identification and correction [Mooring et al., 1991; Zhuang and Roth, 1991]. In this process, the nonlinear relation between the kinematic parameters and the pose of a machine is linearized, and the parameters are estimated by iteratively solving the overdetermined linear least-squares problem. The method requires a sufficiently close enough nominal model of the machine, and the convergence of the estimation depends on the numerical properties and other factors.

Most of the publications on calibration are related to serial machines. Few references address the calibration of parallel kinemic machines [Wang and Masory, 1993; Masory et al., 1993; Zhuang and Roth, 1993]. Zhuang and Roth proposed to identify the length of one leg through measuring the poses obtained by fixing the respective leg while moving other legs [Zhuang and Roth; 1991]. Masory et al. presented an error-model based approach adopted from serial manipulator calibration methodology to identify the kinematic parameters of a Stewart platform [Masory et al., 1993]. This method is capable of identifying more parameters in a larger measurable workspace, but the identification algorithm becomes more complicated. Because of the inherent complexity of the formulations, both of the above approaches use numerical methods to solve for the parameters.

For modular reconfigurable parallel kinematic machines, the calibration process can be performed in steps along with their construction. Either the manufacturer or the user could calibrate individual leg modules. Leg modules can then be attached to the mobile platform and calibrated for their true locations on the mobile platform. Conventional calibration methods can be applied in these steps. Finally each of the feet of the leg modules is installed on the base. Here the base could be user defined, and it is not necessarily planar. The true position and orientation of the leg modules on the base may not be precisely known or are difficult to measure directly. A method for identifying these placement parameters has therefore been developed. Our identification process is as follows. First the mobile plate is moved to several known locations and the corresponding kinematic parameters are measured with internal sensors on both active and passive joints. The measured data are then used in kinematic models to produce nonlinear equations in terms of the unknown position and orientation of the base joints. Methods for solving the identification equations are discussed. Solution with an example to a special situation using Sylvester's Dialytic Elimination [Raghavan and Roth, 1995] is presented. A numerical example of pose identification is also included to verify the method.

4.2 Problem Formulation

4.2.1 Kinematic Model

The kinematic structure of the six-degrees-of-freedom parallel kinematic machine is shown in Fig. 4.1. Its six variable-length legs $l_i (i=1,\ldots,6)$ are connected to a base through the Hook joints, whose locations are denoted as $B_1, B_2, \cdots, B_6$, and to a mobile plate through the spherical joints, whose locations are $M_1, M_2, \cdots, M_6$. Also shown in Fig. 4.1 are two coordinate frames, {B} and {M}, attached respectively to the base and mobile plate. The detailed structure of a Hook joint is shown in Fig. 4.2, which is formed by two mutually perpendicular revolute joints. The stand of the Hook joint is fixed in the base and shaft 2 is connected to the leg through bearings. The unit vector $\vec{n}_i (i=1,\ldots,6)$ in Fig. 4.1 represents the orientation of the Hook joints (axis of shaft 1 in Fig. 4.2) in the reference frame {B}. The position of the joint is represented by the intersection of the axes of shaft 1 and shaft 2. When the machine is relocated, the position of the

$B_i(i=1,\ldots,6)$ and orientation vector $\vec{n}_i(i=1,\ldots,6)$ of the Hook joints are changed. Our focus here is the identification of these position parameters.

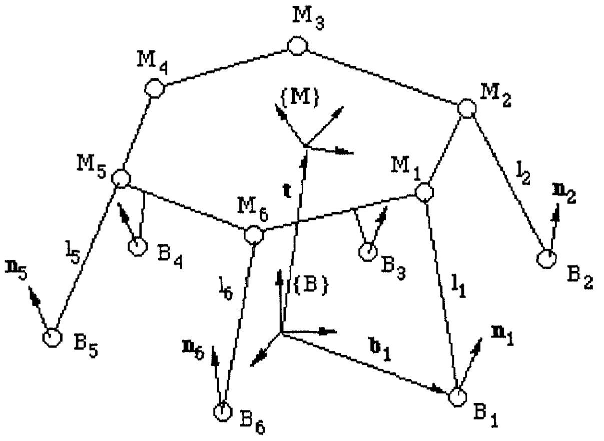


Figure 4.1 Different orientations of Hook joints

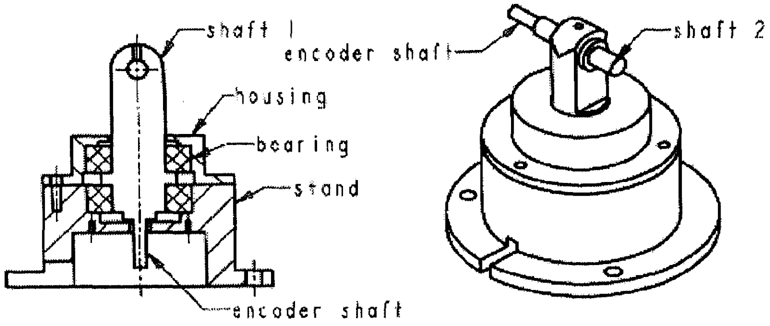


Figure 4.2 Hook joint and its encoders

Solution of the direct kinematics problem in PKMs, that is to find the pose of the platform for a given set of leg lengths, is complicated. Extra sensors have been used to reduce computation time in real time control [Raghavan and Roth, 1995; Merlet, 1993]. In our experimental system, each Hook joint has two rotary encoders (one on shaft 1 and the other on shaft 2, see Fig. 4.2) to measure the orientation of the leg, and each leg has another rotary encoder on the respective motor to measure the leg length. These encoders will not only be used to simplify the forward kinematics of the system, but they will also be used to help our identification process.

In the following process of building the kinematic model, it is assumed that all joints are ideal ones so that the spherical and Hook joints can be treated as points, the axes of the joints pass through the respective joint center, and the prismatic joints can be treated as lines. This assumption is reasonable since the joint inaccuracy is less significant than kinematic errors [Masory et al., 1993]. For more precise applications, the joint inaccuracy can also be separately calibrated and included in the kinematic model.

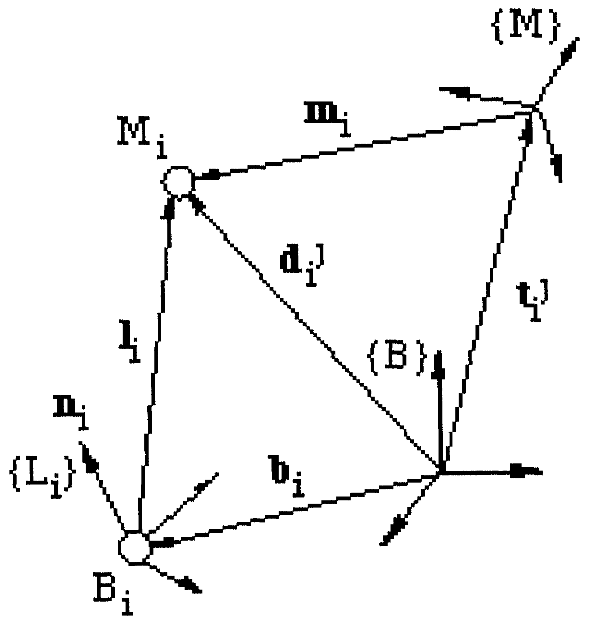


Figure 4.3 Vector representation of a joint-link train

Fig. 4.3 shows the kinematic chain formed by one of the legs, leg *i*, with the base and the mobile plate. In addition to frames {B} and {M}, another coordinate frame is defined in this problem: a leg frame $\{L_i\}$ with its origin at the center of the respective Hook joint and z-axis coincident with the orientation vector $\vec{n}_i$.

When the mobile plate is moved to a position identified by index j during the identification process, its position and orientation at this configuration are denoted by a vector ${}^B\vec{t}^{\,j}$ and a matrix ${}^B R_M^j$. The position and orientation of the leg frame $\{L_i\}$, denoted by a vector ${}^B\vec{b}_i$ and a matrix ${}^B R_{Li}$, are not changed. Let ${}^M\vec{m}_i$ and ${}^{Li}\vec{l}_i^{\,j}$ represent the position vectors of M_i in frames {M} and $\{L_i\}$ respectively. As shown in Fig. 4.3, the position vector of M_i in frames {B}, ${}^B\vec{d}_i^{\,j}$, can be expressed as

$$ {}^B\vec{d}_i^{\,j} = {}^B\vec{t}^{\,j} + {}^B R_M^i \; {}^M\vec{m}_i \quad j = 1,\cdots,m \tag{4.1}$$

or

$$ {}^B\vec{d}_i^{\,j} = {}^B\vec{b}_i + {}^B R_{Li} \; {}^{Li}\vec{l}_i^{\,j} \quad j = 1,\cdots,m \tag{4.2}$$

where m represents the number of the mobile plate configurations used in the process.

Figure 4.4 shows the mobile plate at two different configurations, $j=1$ and $j=k\neq 1$. The vector from the location of M_i at the *k*th pose to the location of M_i at the first pose can be expressed as

$$ {}^{B}\overrightarrow{M_i^k M_i^1} = {}^{B}\vec{d}_i^{1} - {}^{B}\vec{d}_i^{k} = {}^{B}\vec{d}_i^{k1} \qquad k = 2, \cdots, m \tag{4.3} $$

Combining equations (4.2) and (4.3), we have

$$ {}^{B}\vec{d}_i^{k1} = {}^{B}\vec{d}_i^{1} - {}^{B}\vec{d}_i^{k} = {}^{B}R_{Li}\left({}^{Li}\vec{l}_i^{1} - {}^{Li}\vec{l}_i^{k}\right) \qquad k = 2, \cdots, m \tag{4.4} $$

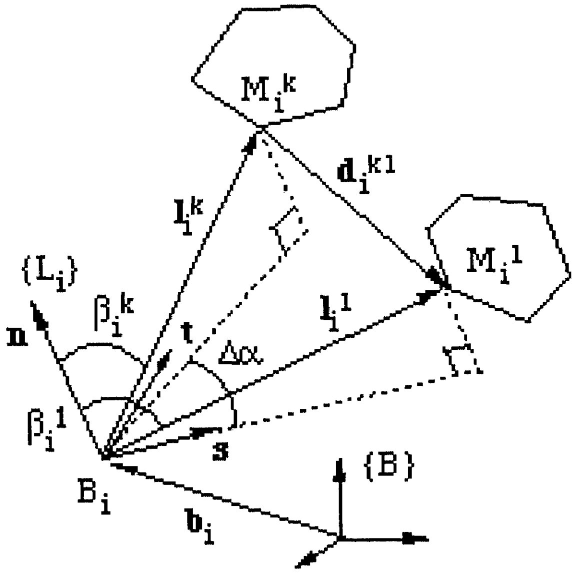


Figure 4.4 Mobile plate at two configurations

Since the vector ${}^{B}\vec{b}_i$ represents the coordinates of B_i and the matrix ${}^{B}R_{Li}$ contains the information of the orientation vector $\vec{n}_i$, the identification problem becomes to find the position vector ${}^{B}\vec{b}_i$ and orientation matrix ${}^{B}R_{Li}$. Because the geometry of the mobile plate does not change during the identification process, ${}^{M}\vec{m}_i$ is known and constant. The position vector ${}^{B}\vec{t}^{\,j}$ and orientation matrix ${}^{B}R_M^j$ of the mobile plate can be obtained either by direct measurement or by placing the mobile plate at known setups. Hence the vectors ${}^{B}\vec{d}_i^1$ and ${}^{B}\vec{d}_i^k$ can be computed from equation (4.1) to give us the vector ${}^{B}\vec{d}_i^{k1}$ in equation (4.4).

If absolute encoders are used in the system then ${}^{Li}\vec{l}_i^{\,j}$ is available directly through the encoders. In this case, only ${}^{B}R_{Li}$ is to be determined. Since an orientation matrix contains three independent parameters, three scalar equations in the form of equation (4.4) are needed to obtain a solution. Therefore, only two configurations (m=2) of the mobile plate are needed for the identification. Since incremental encoders are used in our machine, only the internal encoders can only measure the changes of the leg length and joint angles between any two configurations. That means that both the orientation matrix ${}^{B}R_{Li}$ and initial leg vector ${}^{Li}\vec{l}_i^1$ need to be identified from equation (4.4). Three additional scalar equations are needed for the identification. Thus m=3.

4.2.2 Identification Equations

In order to use equation (4.4) to solve for the rotation matrix ${}^{B}R_{Li}$ and the initial leg vector ${}^{Li}\vec{l}_i^1$, we need to derive their detailed expressions in terms of independent unknowns and the measured parameters. Several angles are defined in frame $\{L_i\}$ for the Hook joint as shown in Fig. 4.5. α_i^1 is the angle made by the projection of ${}^{Li}\vec{l}_i^1$ on the x-y plane with the x-axis; β_i^1 is the angle made by ${}^{Li}\vec{l}_i^1$ with the z-axis; $\Delta\alpha^k$ and $\Delta\beta^k$ are the increments of α_i^1 and β_i^1, respectively, when leg i is moved from the first configuration to the *k*th configuration. That is, $\alpha_i^k = \alpha_i^1 + \Delta\alpha^k$, and $\beta_i^k = \beta_i^1 + \Delta\beta^k$. For convenience, the x-z plane of frame $\{L_i\}$ is selected to be the plane determined by $\vec{n}_i$ and ${}^{Li}\vec{l}_i^{\,1}$. By this convention, $\alpha_i^1 = 0°$, and $0° \le \beta_i^1 \le 180°$.

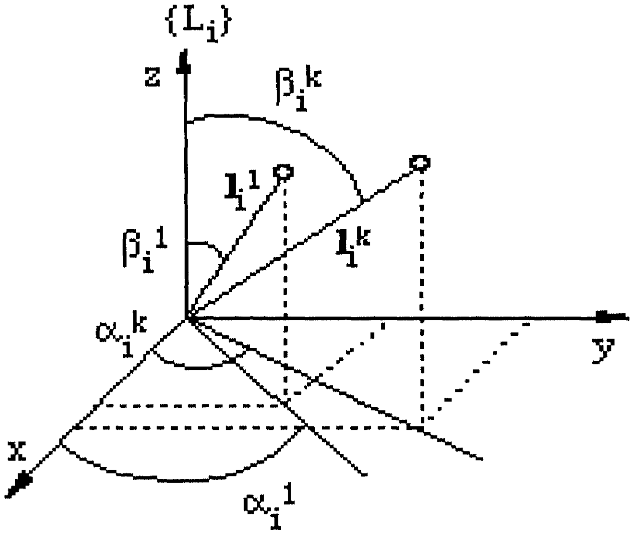


Figure 4.5 Hook joint angles in leg frame $\{L_i\}$

With the angles defined above, we have

$$
{}^{L_i}\vec{l}_i^{\,1} = \begin{bmatrix} l_i s\beta_i \\ 0 \\ l_i c\beta_i \end{bmatrix} \text{ and } {}^{L_i}\vec{l}_i^{\,k} = \begin{bmatrix} (l_i + \Delta l)s(\beta_i + \Delta\beta)c(\Delta\alpha) \\ (l_i + \Delta l)s(\beta_i + \Delta\beta)s(\Delta\alpha) \\ (l_i + \Delta l)c(\beta_i + \Delta\beta) \end{bmatrix} \tag{4.5}
$$

where l_i and Δl are, respectively, the leg length at the first configuration and the leg length increment when the leg is moved from the first configuration to *k*th configuration.

Note that sine and cosine functions have been abbreviated as $s(\cdot)$ and $c(\cdot)$. Also the superscripts 1 and k are omitted.

The rotation matrix ${}^{B}R_{L_i}$ can be expressed as ${}^{B}R_{L_i} = \{\vec{s} \quad \vec{t} \quad \vec{n}\}$, where $\vec{s}$, $\vec{t}$ and $\vec{n}$ are respectively the unit vectors along axes x, y and z of $\{L_i\}$ with respect to the base frame as shown in Fig. 4.4. Let α_s be the angle made by the projection of $\vec{s}$ on the x-y plane of $\{B\}$ with the x-axis of $\{B\}$, and β_s be the angle of $\vec{s}$ to the z-axis of $\{B\}$. Then $\vec{s} = [s\beta_s c\alpha_s \quad s\beta_s s\alpha_s \quad c\beta_s]^T$. Similarly, let α_n and β_n be the respective angles for $\vec{n}$. We have $\vec{n} = [s\beta_n c\alpha_n \quad s\beta_n s\alpha_n \quad c\beta_n]^T$. As $\vec{t} = \vec{n} \times \vec{s}$, ${}^{B}R_{L_i}$ can be expressed as

$$
{}^{B}R_{L_i} = \begin{bmatrix} s\beta_s c\alpha_s & s\beta_n s\alpha_n c\beta_s - c\beta_n s\beta_s s\alpha_s & s\beta_n c\alpha_n \\ s\beta_s s\alpha_s & c\beta_n s\beta_s c\alpha_s - s\beta_n c\alpha_n c\beta_s & s\beta_n s\alpha_n \\ c\beta_s & s\beta_n c\alpha_n s\beta_s s\alpha_s - s\beta_n s\alpha_n s\beta_s c\alpha_s & c\beta_n \end{bmatrix} \tag{4.6}
$$

A constraint equation for $\vec{s}$ and $\vec{n}$ is $\vec{n} \cdot \vec{s} = 0$, i.e.,

$$
s\beta_s c\alpha_s s\beta_n c\alpha_n + s\beta_s s\alpha_s s\beta_n s\alpha_n + c\beta_s c\beta_n = 0 \tag{4.7}
$$

Substituting equations (4.5) and (4.6) into (4.4), we obtain three scalar equations.

$$
\begin{aligned}
& s\beta_s c\alpha_s \{(l + \Delta l) s(\beta + \Delta\beta) c\Delta\alpha - l s\beta\} + \\
& (s\beta_n s\alpha_n c\beta_s - c\beta_n s\beta_s s\alpha_s)(l + \Delta l) s(\beta + \Delta\beta) s\Delta\alpha \\
& + s\beta_n c\alpha_n \{(l + \Delta l) c(\beta + \Delta\beta) - l c\beta\} + dx = 0
\end{aligned} \tag{4.8}
$$

$$\begin{aligned}
&s\beta_s s\alpha_s\{(l+\Delta l)s(\beta+\Delta\beta)c\Delta\alpha - ls\beta\} + \\
&(s\beta_n s\beta_s c\alpha_s - s\beta_n c\alpha_n c\beta_s)(l+\Delta l)s(\beta+\Delta\beta)s\Delta\alpha \\
&+s\beta_n s\alpha_n\{(l+\Delta l)c(\beta+\Delta\beta) - lc\beta\} + dy = 0
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
&c\beta_s\{(l+\Delta l)s(\beta+\Delta\beta)c\Delta\alpha - ls\beta\} + \\
&(s\beta_n c\alpha_n s\beta_s s\alpha_s - s\beta_n s\alpha_n s\beta_s c\alpha_s)(l+\Delta l)s(\beta+\Delta\beta)s\Delta\alpha \\
&+c\beta_n\{(l+\Delta l)c(\beta+\Delta\beta) - lc\beta\} + dz = 0
\end{aligned} \tag{4.10}$$

where dx, dy and dz are components of ${}^{B}\vec{d}_i^{k1}$. Also note that subscript *i* has been omitted in variables α_i, β_i and l_i.

In the identification equations (4.7-4.10), the relative values, $\Delta\alpha$, $\Delta\beta$ and Δl, of each leg between any two configurations are obtained from the three incremental encoders. Values of dx, dy and dz are known from the poses of the mobile plate. There are six unknowns; α_s, β_s, α_n, β_n, l and β, to be solved for. As there are more unknowns than the equations, more equations are needed. Another configuration of the mobile plate will give us three additional equations with another set of $\Delta\alpha$, $\Delta\beta$, Δl, dx, dy and dz. Thus, three configurations (m=3) are needed for our identification problem.

4.3 Solving Identification Equations

The identification problem is now represented by a set of transcendental equations. One way to directly solve this set of equations is the use of a numerical method. An initial guess is needed to start the numerical process can found from our rough knowledge of the position and orientation of the leg placement. Many existing numerical techniques can be used in this approach. An indirect approach can also be used to solve those equations. One approach is to convert the set of transcendental equations into a set of polynomial equations, and then solve the resulting polynomial systems. There are three well-know methods for solving polynomial systems [Raghavan and Roth, 1995]: Dialytic Elimination, Polynomial Continuation, and Grobner bases. The degrees of polynomials obtained from converting the general identification equations (4.7-4.10) to polynomials are respectively 8, 11, 11, and 11. The total degree of the system is 8×11^3. A polynomial system with such a high degree is not easy to solve by any of the three methods. We will show, however, that an univariate polynomial can be obtained through Dialytic Elimination for a special situation. The solution to this univariate polynomial leads to the solution of this type of identification problem. This univariate polynomial makes the identification of parameters simpler for this special situation.

In this special situation, the base has been so prepared that the each Hook joint has a known direction, $\vec{n} = [s\beta_n c\alpha_n \quad s\beta_n s\alpha_n \quad c\beta_n]^T$. That is, α_n and β_n are known. This will reduce the unknowns to four: α_s, β_s, l and β. Hence, only two configurations are needed.

To take full advantage of the available information on $\vec{n}$, we introduce another frame $\{B^*\}$ at the same location of frame $\{B\}$ but with its z-axis parallel to $\vec{n}$. Since {B*} is defined by the user, a rotation matrix R^* can be easily obtained for coordinate transformation between $\{B\}$ and $\{B^*\}$. We can now carry out the identification process in $\{B^*\}$ instead of $\{B\}$. In fact, if all the Hook joints have the same orientation, then we can simply make {B} to be the one with its z-axis parallel to $\vec{n}$. Parameters dx, dy and

dz are now components of ${}^{B*}\vec{d}_i^{k1}$. With $\{B*\}$ as reference frame, $\beta_n = 0^\circ$, $\beta_s = 90^\circ$, and α_n is undefined and no longer needed. Equation (4.6) becomes

$$ {}^{B*}R_{L_i} = \begin{bmatrix} c\alpha_s & -s\alpha_s & 0 \\ s\alpha_s & c\alpha_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6*}$$

and equation (4.7) holds automatically.

The identification equations (4.8-4.10) are now reduced to

$$(l+\Delta l)s(\beta+\Delta\beta)c(\alpha_s+\Delta\alpha) - ls\beta c\alpha_s + dx = 0 \tag{4.11}$$

$$(l+\Delta l)s(\beta+\Delta\beta)s(\alpha_s+\Delta\alpha) - ls\beta s\alpha_s + dy = 0 \tag{4.12}$$

$$(l+\Delta l)c(\beta+\Delta\beta) - lc\beta + dz = 0 \tag{4.13}$$

with three unknowns β , α_s, and l .

4.3.1 Transforming Transcendental Equations to Polynomial Equations

Let $x = \tan\alpha_s/2$, $y = \tan\beta/2$ and $z = l + \Delta l$. Hence,

$$s\alpha_s = \frac{2x}{1+x^2}, \qquad c\alpha_s = \frac{1-x^2}{1+x^2}, \qquad s\beta = \frac{2y}{1+y^2}, \qquad c\beta = \frac{1-y^2}{1+y^2}$$

Substituting them into equations (4.11-4.13), we have three polynomial equations:

$$\begin{aligned}&(c_4 + 2c_2x - c_4x^2 + 2c_3y + 4c_1xy - 2c_3x^2y - c_4y^2 - 2c_2xy^2 + c_4x^2y^2)z \\ &+d_1 + d_1x^2 + 2c_5y - 2c_5x^2y + d_1y^2 + d_1x^2 = 0\end{aligned} \tag{4.14}$$

$$\begin{aligned}&(e_4 + 2e_2x - e_4x^2 + 2e_3y + 4e_1xy - 2e_3x^2y - e_4y^2 - 2e_2xy^2 + e_4x^2y^2)z \\ &+d_2 + d_2x^2 + 4e_5xy + d_2y^2 + d_2x^2y^2 = 0\end{aligned} \tag{4.15}$$

$$(f_2 + 2f_1y - f_2y^2)z + d_3 + f_3 + d_3y^2 - f_3y^2 = 0 \tag{4.16}$$

where

$$\begin{array}{lll}
c_1 = -s(\Delta\alpha_s)c(\Delta\beta) & e_1 = -1 + c(\Delta\alpha_s)c(\Delta\beta) & d_1 = dx \\
c_2 = -s(\Delta\alpha_s)s(\Delta\beta) & e_2 = c(\Delta\alpha_s)s(\Delta\beta) & d_2 = dy \\
c_3 = -(1 - c(\Delta\alpha_s)c(\Delta\beta)) & e_3 = s(\Delta\alpha_s)c(\Delta\beta) & d_3 = dz \\
c_4 = c(\Delta\alpha_s)s(\Delta\beta) & e_4 = s(\Delta\alpha_s)s(\Delta\beta) & f_1 = -s(\Delta\beta) \\
c_5 = \Delta l & e_5 = \Delta l & f_2 = -1 + c(\Delta\beta) \\
 & & f_3 = \Delta l
\end{array}$$

Both equations (4.14) and (4.15) are of degree 5 and equation (4.16) is of degree 3. The total degree of the polynomial system is therefore 75.

4.3.2 Dialytic Elimination

In order to solve the system of polynomial equations (4.14-4.16) for x, y and z, we use Sylvester's Dialytic Elimination to obtain an univariate polynomial. The sequence of elimination has great influence on the result. Therefore great care should be taken to avoid extraneous solutions. Note that the highest powers of variables x and y are two while the highest power of z is 1 in all three equations. Thus, variable z is selected as the first one to be eliminated. By suppressing variables x and y,

$$a_{11}z + a_{10} = 0 \tag{4.17}$$

$$a_{21}z + a_{20} = 0 \tag{4.18}$$

$$a_{31}z + a_{30} = 0 \tag{4.19}$$

where

$$a_{11} = c_4 + 2c_2x - c_4x^2 + 2c_3y + 4c_1xy - 2c_3x^2y - c_4y^2 - 2c_2xy^2 + c_4x^2y^2$$

$$a_{10} = d_1 + d_1x^2 + 2c_5y - 2c_5x^2y + d_1y^2 + d_1x^2y^2$$

$$a_{21} = e_4 + 2e_2x - e_4x^2 + 2e_3y + 4e_1xy - 2e_3x^2y - e_4y^2 - 2e_2xy^2 + e_4x^2y^2$$

$$a_{20} = d_2 + d_2x^2 + 4e_5xy + d_2y^2 + d_2x^2y^2$$

$$a_{31} = f_2 + 2f_1y - f_2y^2$$

$$a_{30} = d_3 + f_3 + d_3y^2 - f_3y^2$$

The determinants of coefficient matrices for any two equations should be zero. This leads to two independent equations containing only x and y:

$$a_{11}a_{30} - a_{10}a_{31} = 0 \tag{4.20}$$

$$a_{21}a_{30} - a_{20}a_{31} = 0 \tag{4.21}$$

The highest orders of y are four while those of x are two in both equations (4.20) and (4.21). The detailed expressions of the two equations are not included here due to their length. Selecting x as the second variable to eliminate and suppressing y in equations (4.20) and (4.21), we have

$$b_{12}x^2 + b_{11}x + b_{10} = 0 \tag{4.22}$$

$$b_{22}x^2 + b_{21}x + b_{20} = 0 \tag{4.23}$$

where b_{12}, b_{11}, b_{10}, b_{22}, b_{21}, and b_{20} are coefficients containing y. Multiplying equations (4.22) and (4.23) by x, yields two additional equations needed for Dialytic elimination:

$$b_{12}x^3 + b_{11}x^2 + b_{10}x = 0 \tag{4.24}$$

$$b_{22}x^3 + b_{21}x^2 + b_{20}x = 0 \tag{4.25}$$

Solution to equations (4.22-4.25) requires the determinant of the coefficient matrix to be zero, that is,

$$\begin{vmatrix} 0 & b_{12} & b_{11} & b_{10} \\ 0 & b_{22} & b_{21} & b_{20} \\ b_{12} & b_{11} & b_{10} & 0 \\ b_{22} & b_{21} & b_{20} & 0 \end{vmatrix} = 0 \tag{4.26}$$

The result is a polynomial equation of degree 16 in one variable y:

$$p_{16}y^{16} + p_{15}y^{15} + \cdots + p_1 y + p_0 = 0 \tag{4.27}$$

where $p_0, p_{1,} \cdots, p_{16}$ are the coefficients of the polynomial in terms of the known parameters.

Sixteen solutions can be obtained from equation (4.27) with a root-finding routine. Substituting each real root of y into equations (4.22) and (4.23) and treating every power product of x as an independent unknown, we get a unique x corresponding to each y. Then, each pair of x and y is substituted into any one of equations (4.17-4.19) to solve for z. Consequently, β , α_s, and l are calculated for each set of x, y and z. Finally equation (4.3) can be evaluated to obtain the placement parameters ${}^B\vec{b}_i$ in $\{B^*\}$ and $\{B\}$.

4.4 A Numerical Example and Discussion

Consider a platform moving from configuration 1 to 2. Here only the placement of one of the legs is discussed. After measuring the position and orientation of each configuration, the coordinates of the spherical joint M_1 at two locations in frame $\{B\}$ are computed, say, $M_1^1 = (1000 \quad 0 \quad 1000)$ and $M_1^2 = (1000 \quad 500 \quad 500)$. Therefore $\vec{d}_1^{21} = \begin{bmatrix}0 & -500 & 500\end{bmatrix}^{\mathrm{T}}$. The true position and orientation of the base joint B_1 are

$${}^B\vec{b}_1 = \begin{pmatrix}2000 & 0 & 0\end{pmatrix}^T, \text{and } {}^B R_{L_1} = \begin{vmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix}, \text{respectively.}$$

The increments of the Hook joint angles at B$_1$ and the length of leg 1 between configurations 1 and 2 are obtained from the three encoders. Their values are respectively $\Delta l = -189.4687$, $\Delta\alpha = -26.5651°$, and $\Delta\beta = 20.9052°$. With this set of parameters, the polynomial of equation (4.27) becomes

$$\begin{aligned}
&1.43907\times10^{10}y^{16} - 1.25216\times10^{10}y^{15} + 1.75354\times10^{11}y^{14} - 2.26006\times10^{9}y^{13} + \\
&5.68998\times10^{11}y^{12} + 4.1917\times10^{11}y^{11} + 5.6278\times10^{11}y^{10} + 4.43135\times10^{11}y^{9} + \\
&1.39848\times10^{11}y^{8} + 6.58389\times10^{10}y^{7} - 2.6355\times10^{10}y^{6} - 2.44243\times10^{10}y^{5} - \\
&6.15476\times10^{9}y^{4} + 1.3265\times10^{9}y^{3} - 6.83559\times10^{8}y^{2} - 1.10417\times10^{9}y \;\; + 5.49758 = 0
\end{aligned}$$

Solving this equation, we obtain 4 real and 12 complex solutions for y. Since only real solutions have physical meaning, only four sets of solutions are obtained. The coefficients b$_{12}$, b$_{11}$, b$_{10}$, b$_{22}$, b$_{21}$, and b$_{20}$ are then evaluated based on the y values. The corresponding values of x can now be found from equation (4.22) or (4.23). Finally the corresponding values of z are available from any of the equations (4.17-4.19). We can now calculate the unknowns β , α_s, and l as

$$\alpha_s = 2\tan^{-1} x, \quad \beta = 2\tan^{-1} y, \quad \text{and} \quad l = z - \Delta l$$

Applying the results in equations (4.5), (4.6*) and (4.2), we obtain the joint position ${}^{B}\bar{b}_1$. The solutions are summarized in Table 4.1.

Since $0 \le \beta \le 180°$, $y = \tan\beta/2 \ge 0$. Solutions 1 and 2 are not the desired results. In the process of mounting the six legs to a new base or in a new environment, we usually have the information on the rough locations of the six Hook joints. These values can be used to determine which location is the desired one among the possible locations obtained from the solutions. We know the center position B_1 of the Hook joint is placed somewhere near (2000, 0, 0) in the base frame. Therefore, solution 4 is the identified location of the Hook joint of leg 1.

Table 4.1 Solution for position identification example

	Solution 1	Solution 2	Solution 3	solution 4
x	-4.236067	-3.184782	12.784030	1.445302×10^{7}
y	-0.648231	-0.569796	0.348662	0.414213
z	-1414.213636	-1475.468881	1286.000190	1224.744945
α_s	-153.434941°	-145.136161°	171.701182°	180°
β	-65.905150°	-59.348609°	38.443452°	45°
l	-1224.744945	-1286.000190	1475.468881	1414.213636
b_{1x}	1999.99994	1907.75498	1907.75499	2000.00005
b_{1y}	500.00014	632.40801	-132.40801	0
b_{1z}	1500.00018	1655.61994	-155.61994	-0.00005

The symbolic derivation of equations (4.17-4.27) and the solutions of the numerical example have been performed on a SUN Sparc20 workstation with Mathmatica 3.0. The same should be applicable for the general identification problem represented by equations (4.7-4.10). That is, one can solve for six unknowns,

$\alpha_s, \beta_s, \alpha_n, \beta_n, l, \beta$, with 3 configurations using Sylvester's Dialytic Elimination method. Six polynomial equations in symbolic form have been derived from the six identification equations. However, our current computing is not capable of performing the required symbolic manipulation in the elimination process since the high degree of the polynomial system. Certainly actual values of the known parameters can be used to obtain six polynomials with numerical coefficients. This can dramatically reduce the terms in the elimination process. Another approach is to numerically solve the general case identification equations.

4.5 A Numerical Example for Pose Identification

As the use of Dialytic Elimination for general placement demands great computation power for symbolic manipulation, Newton's numerical approach is used to solve the pose identification equations. For a machine moving from configuration 1 to 2 and then to 3, after measuring the position and orientation of each configuration, joint M₁ has the following three locations in frame {B}: $M_1^1 = (1000 \quad 0 \quad 1000)$, $M_1^2 = (1000 \quad 500 \quad 500)$ and $M_1^2 = (1000 \quad 1000 \quad 750)$. Therefore $\vec{d}_1^{21} = [0 \quad -500 \quad 500]^{\mathrm{T}}$ and $\vec{d}_1^{31} = [0 \quad -1000 \quad 250]^{\mathrm{T}}$. The increments of the Hook joint angles at B₁ and the length of leg 1 from configuration 1 to 2, and 1 to 3, are obtained from the three encoders. Their values are respectively

$$\begin{array}{ll} \Delta l_1^{12} = -189.469 & \Delta l_1^{13} = -186.567 \\ \Delta \alpha_1^{12} = -24.536^\circ & \Delta \alpha_1^{13} = -40.763^\circ \\ \Delta \beta_1^{12} = 19.455^\circ & \Delta \beta_1^{13} = 13.089^\circ \end{array}$$

After substituting this set of parameters into equations (4.7)-(4.10), we get seven equations. The first six equations for unknowns of $\alpha_s, \beta_s, \alpha_n, \beta_n, l, \beta$ are solved using Newton's numerical approach. The solutions are summarized in Table 4.2, where K is the iteration number. The true values of $\alpha_s, \beta_s, \alpha_n, \beta_n, l, \beta$ are $180^\circ \quad 75^\circ \quad 0^\circ \quad 15^\circ \quad 1414.214 \quad 60^\circ$, respectively. From the table, at the fourth iteration (k=4), the values of the unknowns approach their true values with very small errors. That means the equations can correctly model the pose identification problem.

Table 4.2 Solution Summary for Pose Identification

real value	180°	75°	0°	15°	60°	1414.214
k	α_s	β_s	α_n	β_n	β	l
0	170°	70°	5°	10°	54°	1370
1	180.717°	77.398°	-1.015°	12.403°	56.964°	1403.530
2	180.034°	75.276°	0.554°	14.716°	59.744°	1414.587
3	180.000°	74.999°	-0.011°	15.000°	60.001°	1414.192
4	180.000°	74.998°	-0.002°	15.002°	60.002°	1414.214
5	180.000°	74.998°	-0.002°	15.002°	60.002°	1414.214
6	180.000°	74.998°	-0.002°	15.002°	60.002°	1414.214
Error	0.000°	-0.002°	-0.002°	0.002°	0.002°	0.000

4.6 Summary of Identification of Placement Parameters

A modular parallel kinematic machine can be easily reconfigured to a new location for the most effective configuration. An algorithm for identifying the position and orientation of the base joints of the platform in the new location is presented. This algorithm uses the information on the mobile platform configuration and internal encoders to establish identification equations. For the platforms whose axes of the base joints are precisely defined, the minimum number of configurations for identifying the position is two. The general identification will require at least three configurations of the mobile platforms. The identification equation can be solved through a numerical approach or through Dialytic Elimination using symbolic manipulation for some special situations. The methods with Dialytic elimination and Newton's numerical approach are illustrated with examples.

CHAPTER 5

TOOL PATH PLANNING FOR ORTHOGONAL AND NONORTHOGONAL MACHINE TOOLS

This chapter provides first an overview of the current tool path planning process for conventional machine tools and then a detailed analysis of the dissimilarities between tool path planning and execution in orthogonal and nonorthogonal machine tools. It explains why traditional tool path planning methods should not be used for hexapod machine tools without modification.

5.1 Tool Path Planning for Conventional Machine Tools

Shorter life cycles, greater part variation, lower costs and higher quality of products require higher degrees of automation and flexibility at all stages of a design and manufacturing processes. Computer aided design (CAD) allows engineers to design, synthesize, and analyze their products on a computer interactively, and therefore greatly increases the design automation and flexibility. Computer aided manufacturing (CAM) extracts geometric models from a CAD database and generates the tool paths for computer numerical control (CNC) machine tools so that the machines can produce the parts. The introduction of numerical control (NC) and CNC technology is crucial to manufacturing automation and flexibility. NC and CNC technology also enables the integration of CAD and CAM to achieve still higher degrees of automation in design and manufacturing. NC or CNC machines can execute operators' instructions and generate multi-axis coordinated motions with a high degree of accuracy. This makes the machining of complex surfaces more economical and accurate than machines operated by humans. However, programmable NC or CNC machines only follow the planned tool paths. Planning of tool paths directly affects the quality and efficiency of machining operations, and is one of the key issues in CAM, especially when machining complex sculptured surfaces. Tool path planning has been playing an essential part that will allow CAD/CAM users to integrate their machines into their manufacturing environment.

5.1.1 Tool Path Planning Methods

The purpose of tool path planning is to generate a set of ordered points, the cutter contact (CC) points, to approximate a surface. The more the points, the higher the accuracy of the resulting surface. However, increasing the number of points increases the file sizes of the path data and reduces overall moving speed. Specified error tolerance is used to determine the location of the approximation points so that accuracy and speed can be balanced.

Complex part surfaces are usually defined in parametric forms $\Sigma : \mathbf{r} = \mathbf{r}(u, v)$. There are basically two kinds of methods to plan the tool paths for such surfaces. In an isoparametric method as shown in Fig. 5.1, the tool moves along curves on the surface on which one parameter is constant (for example v). Along each such curve, the cutter contact (CC) points are generated by incrementing the u parameter. The distance between two adjacent CC points on these lines is defined as the step-forward distance, and is determined by the approximation tolerance ($\delta 1$) of an isoparameter curve to the line interpolated by two adjacent CC points. The distance between two adjacent isoparameter curves is defined as the step-over distance, and is determined by the allowed

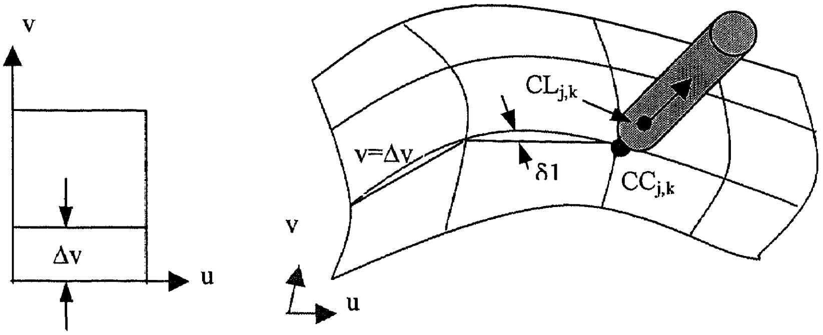


Figure 5.1 Isoparametric tool path planning

scallop height, local surface shape and cutter size. For a given allowable scallop height, the step-over distance changes along an isoparameter curve. The smallest offset distance is then used to determine the constant v parameter for the next isoparameter curve. According to Lee, one problem of this method is inefficient machining as the next tool path is generated by the smallest path interval and hence redundant machining overlap occurs between two adjacent tool paths [Lee, 1998].

A non-isoparametric method can plan an efficient tool-path for machining free-form surfaces without redundant machining. Lin and Koren presented an analytical non-isoparametric method for 3-axis milling operations with a ball-end cutter [Lin and Koren, 1996]. For a 5-axis machine, Lee used an iterative method to find the adjacent non-isoparametric tool path by evaluating the machining strip as shown in Fig. 5.2 [Lee, 1998]. In this method, the adjacent CC point on the next path is found by calculating the machining strip, which is determined by the local geometry, scallop height, and tool size. This CC point is then checked for its step-forward error to determine whether to add an intermediate CC point between the two adjacent CC points on the next path.

Arc fitting enables one to increase the feed rate of your NC machines and improve cutting accuracy by replacing point-to-point tool paths with arcs. Since one arc represents curved surfaces more accurately than many linear segments, arc fitting can reduce NC data file size while improving overall machining tolerances. Tangential arc fitting takes this process a step further by ensuring that the arcs fit together so seamlessly that the NC machine "sees" one continuous tool path and is able to cut parts more accurately. The resulting smoother tool paths will also help to extend the life of NC machines and cutting tools.

Fitting arcs to the tool path will eliminate almost all the choral deviation caused by the straight line approximation of the original surface, resulting in smoother, more accurate tool paths. Larger interpolation blocks allow one to increase feed rates while virtually eliminating data starvation and stutter step. Smoother tool paths put less strain on machine tools resulting in extended tool life. NURBS curve fitting (polynomial, spline, Bezier and NURBS) is now available as an option in commercial products to produce both small and accurate files for use with controls which support true NURBS machining. The benefits of this technology include very smooth finishes similar to arc fitting, but with absolutely no dependence on planes as there is on most controls with arcs. In addition the control will prevent over travel of the machine tool when using NURBS machining. The NURBS machining also produces good results with true 3D tool paths such as "flowline" type tool paths.

The actual CC points are achieved by controlling a cutter to its corresponding cutter configurations known as Cutter-location (CL) points. The relationship between a pair of CC and CL points is determined by the inclination angle, tilt angle, and radius of the cutter as shown in Fig. 5.3.

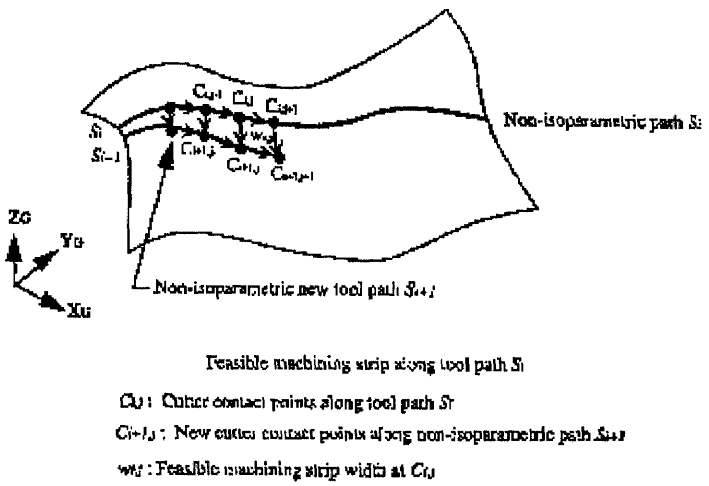

$C_{i,j-1}$
$C_{i,j}$
$C_{i,j+1}$
Non-isoparametric path S_i
S_i
S_{i-1}
$C_{i+1,j}$
$w_{i,j}$
$C_{i+1,j}$
$C_{i+1,j+1}$
ZG
YG
XG
Non-isoparametric new tool path S_{i+1}
Feasible machining strip along tool path S_i
$C_{i,j}$: Cutter contact points along tool path S_i
$C_{i+1,j}$: New cutter contact points along non-isoparametric path S_{i+1}
$w_{i,j}$: Feasible machining strip width at $C_{i,j}$

Figure 5.2 A non-isoparametric tool path planning method [Lee, 1998]

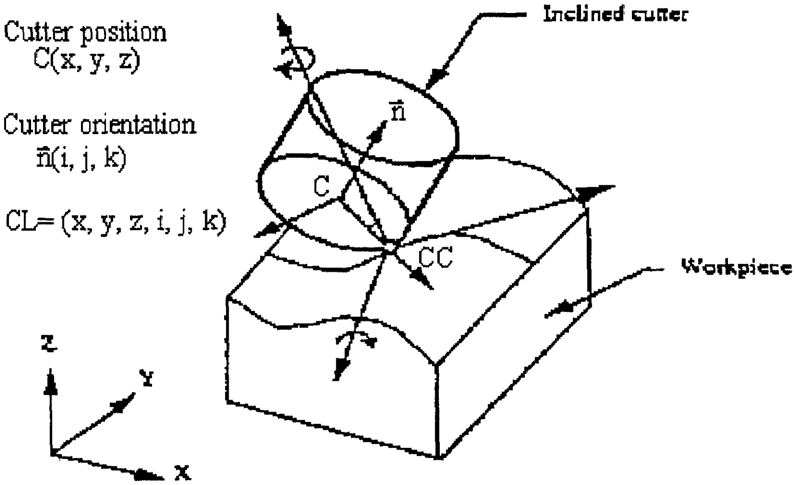


Figure 5.3 Relationship between CC and CL points

Each CC position is used to calculate its corresponding CL position. The final tool path is now represented by CL data (CL paths). For a 5-axis machine, CL data is defined as

$$CL = (x_C, y_C, z_C, i, j, k) \qquad (5.1)$$

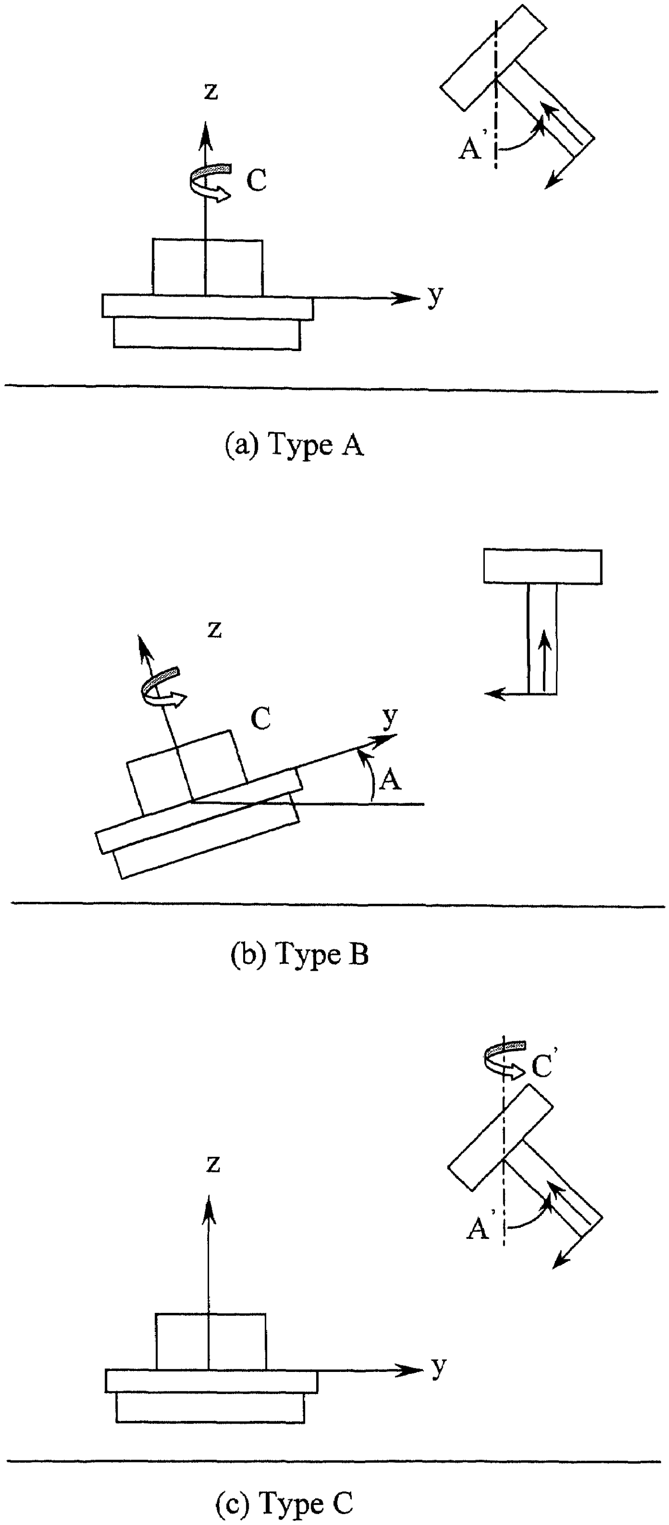
where (x_C, y_C, z_C) are the coordinates of the origin of the cutter frame in a reference frame, and (i, j, k) are the normalized directional cosines of the cutter orientation vector **n**. As the cutter is axisymmetric, CL data completely define its position and orientation. The CL path as well as the specified feedrate will be postprocessed into Machine Control Data (MCD), which is used to control a machine to achieve the required trajectory, speed and other functionalities.

5.1.2 Five-Axis Machines

Because the two rotational axes for cutter orientation in a five-axis machine can be implemented through the spindle head, table or both, common five-axis machines are classified into three main families as illustrated in Fig. 5.4.

Type A: Machines with a rotary table and a tilting spindle head, rotation of the table and head being restricted to one plane.

Type B: Machines with a fixed spindle head and a table capable of rotation in two perpendicular planes.



(a) Type A

(b) Type B

(c) Type C

Figure 5.4 Three types of five-axis NC machines

Type C: Machines with a fixed table and a spindle head capable of rotation in two perpendicular planes.

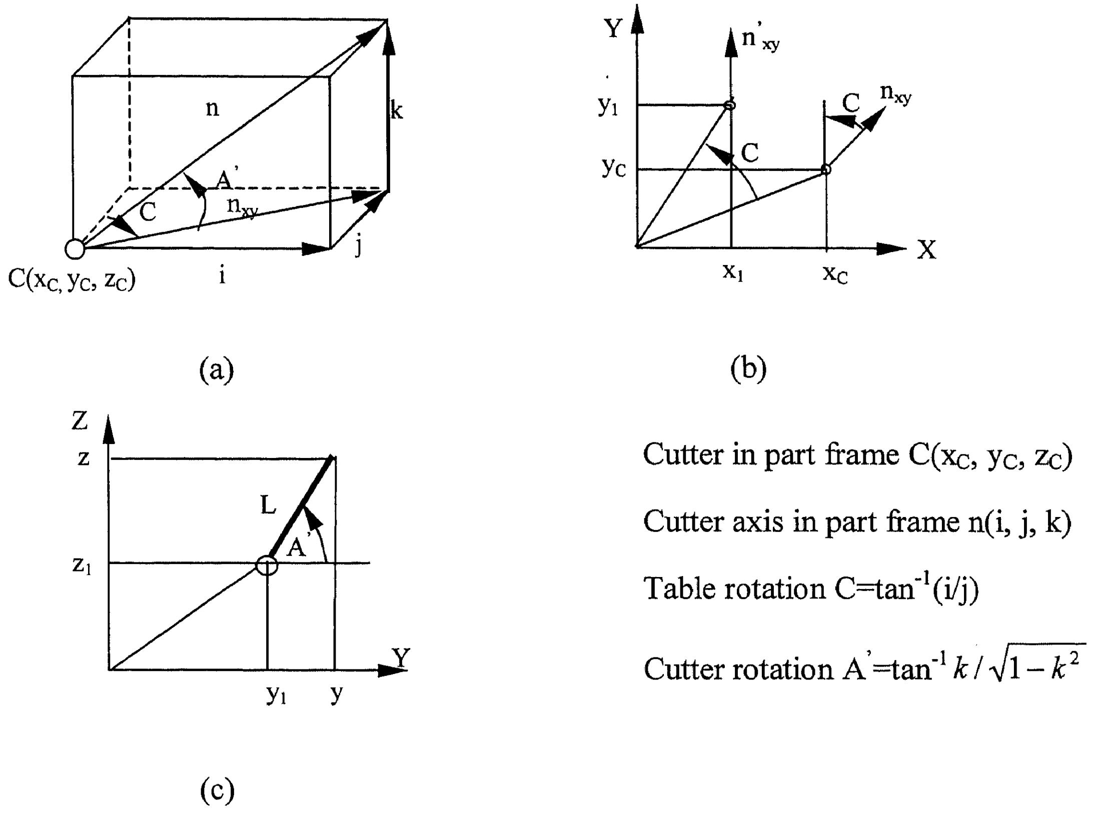
Compared with machines of Type C, machines of Type B usually have larger motion ranges in two rotation axes, higher stiffness, but require higher driving power. Machines of type A have properties between those of Types B and C.

For a 5-axis CNC machine, The transformation of CL data to Machine Control Data (MCD) is a mapping from $CL = (x_C, y_C, z_C, i, j, k)$ to $MCD = (q_1, q_2, q_3, q_4, q_5)$, where q_i (i=1,…,5) are the coordinates of the machine axes corresponding to the CL data. The transformation depends on the machine type. For example, MCD in five-axis machines of Type A is defined as

$$MCD = (X, Y, Z^{'}, A^{'}, C) \tag{5.2}$$

where X and Y are coordinates along the two translational axes of the table, $Z^{'}$ is the coordinates along the translational axis of the cutter, $A^{'}$ is a rotation angle about the x-axis of the cutter, and C is a rotation angle about the z-axis of the table.

The purpose of this transformation is to get the displacements of each axis of a machine for reaching the planned cutter location. If $CL = (x_C, y_C, z_C, i, j, k)$, where (x_C, y_C, z_C) are the coordinates of the cutter center C and (i, j, k) are the components of the cutter unit axis vector **n**, the cutter of a Type A machine can reach this CL through rotating the table to the angular position of C, rotating the spindle head to the angular position of $A^{'}$, and moving the three translational axes to X, Y, $Z^{'}$, respectively.



(a)

(b)

(c)

Cutter in part frame $C(x_C, y_C, z_C)$

Cutter axis in part frame n(i, j, k)

Table rotation $C=\tan^{-1}(i/j)$

Cutter rotation $A'=\tan^{-1} k/\sqrt{1-k^2}$

Figure 5.5 Transformation from CL to MCD

Figure 5.5 shows the transformation relationship of this process. From this figure, the coordinates of the axes of the machine in the machine reference frame {B} can be derived as

$$
\begin{aligned}
X &= x_C \cos C - y_C \sin C \\
Y &= x_C \sin C + y_C \cos C + L\sqrt{1-k^2} \\
Z' &= z_C + Lk \\
A' &= \tan^{-1}(k/\sqrt{1-k^2}) \\
C &= \tan^{-1}(i/j)
\end{aligned}
\qquad (5.3)
$$

where L represents the distance of the cutter center C to the rotation pivot of the spindle head.

5.1.3 Tool Path Execution

After CL files are postprocessed into Machine Control Data (MCD) or a CNC program, the program can be executed by the specified CNC machine to achieve the planned tool path with a desired feedrate V. A typical CNC interpolator can process only straight line and circular arc motion commands; any two adjacent CL points in the path are connected by either a straight line or circular arc segment. In a CNC, the interpolator generates intermediate reference points by sampling from the segment at sampling intervals of T. During each sampling period, the next reference point is obtained and converted into axis coordinates; the intermediate axis coordinates are computed by linear interpolation of axis coordinates of two neighboring reference points. In the next interval, these coordinates are used as references to control each axis, and therefore the tool position can proceed by a distance of VT.

If cutter orientation does not change between two consecutive CL points, there are no approximation errors for linear interpolation in the execution stage, but an approximation error exists for circular interpolation as shown in Fig 5.6. The following formulae can be used to compute the error.

$$\delta 2 = R - \sqrt{R^2 - V^2 T^2 / 4} \tag{5.4}$$

where R is the radius of a circular arc, V is the feedrate, and T is the sampling interval.

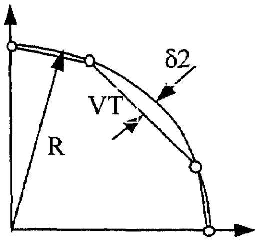


Figure 5.6 Approximation error

5.1.4 Direct Interpolators

The approximation of tool path by linear/circular segments and the execution of the motion through linear/circular interpolation in a CNC have the following problems:

1. An approximation error occurs in the planning stage. In order to have better approximation and smoothness, the number of segments should be increased.
2. The increased number of segments will reduce the average tool feedrate because of the acceleration and deceleration applied at the beginning and end of each segment by the controller.
3. The increased segments require a large memory space on a CNC machine and increase the communication load between the CAD/CAM system and the CNC machine as well as between the machine controller and the motion interpolator.

The ideal situation for machining sculptured surfaces is not to approximate the original surface geometry at all. While direct surface machining based on original surface geometry may seem a remote possibility, some manufacturers of machine controllers are quickly moving down that path. Siemens already has a CNC that represents all geometry internally in the form of NURBS (non-uniform rational B-spline) surfaces and curves, and Fanuc recently has announced similar capabilities as well. Of course, having such a capability in a CNC does not necessarily mean that it is easy to import surfaces from a wide range of commercially available CAD/CAM systems. The memory space requirement could be smaller as there is no need to specify all the linear/circular segments. The major requirement is that the controller must handle all the necessary computation in real-time. Wallace reported a Makino A66 3-axis machining center which could do NURBS interpolation and produce the NURBS surfaces directly from solid models with the machine cycle times reduced by 15~20% [Wallace, 1997]. For 5-axis machines, Koren proposed a scheme of a real-time interpolator in 1997 as shown in Figure 5.7 [Koren, 1997]. According to the scheme, the input of the CNC could be G-code, the curve or even the surface, and the controller can generate reference commands directly from these inputs.

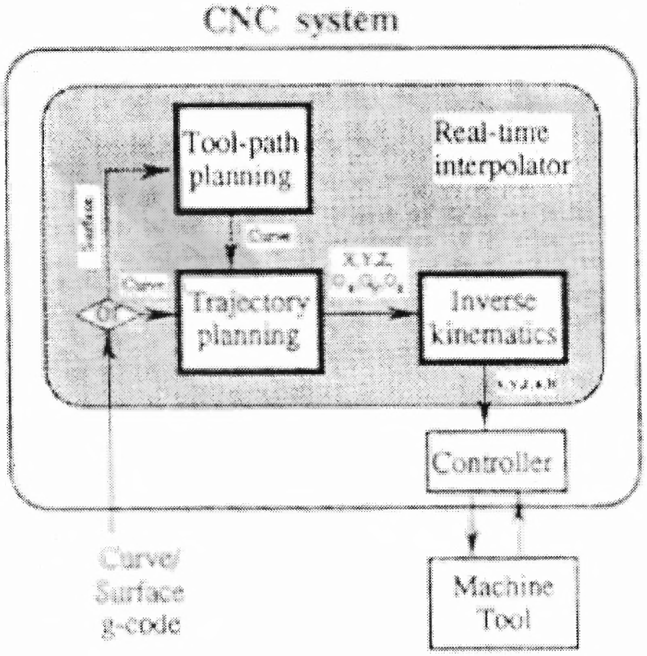


Figure 5.7 Scheme of a real-time interpolator [Koren, 1997]

5.2 Tool Path Planning Dissimilarities

For hexapod machine tools to be integrated into the current manufacturing environment, planning tool paths for them is an essential task. This section discusses whether the existing tool path planning methods developed for five-axis machines can be used directly for hexapod machines. The dramatic difference in their kinematic structure makes it necessary to have a good understanding about the dissimilarities between path planning in five-axis machines and hexapods.

As shown in Fig. 5.8, a hexapod machine tool has a six-degree of freedom mechanism composed of six variable-length legs l_i $(i = 1,\cdots,6)$ connected to a base plate,

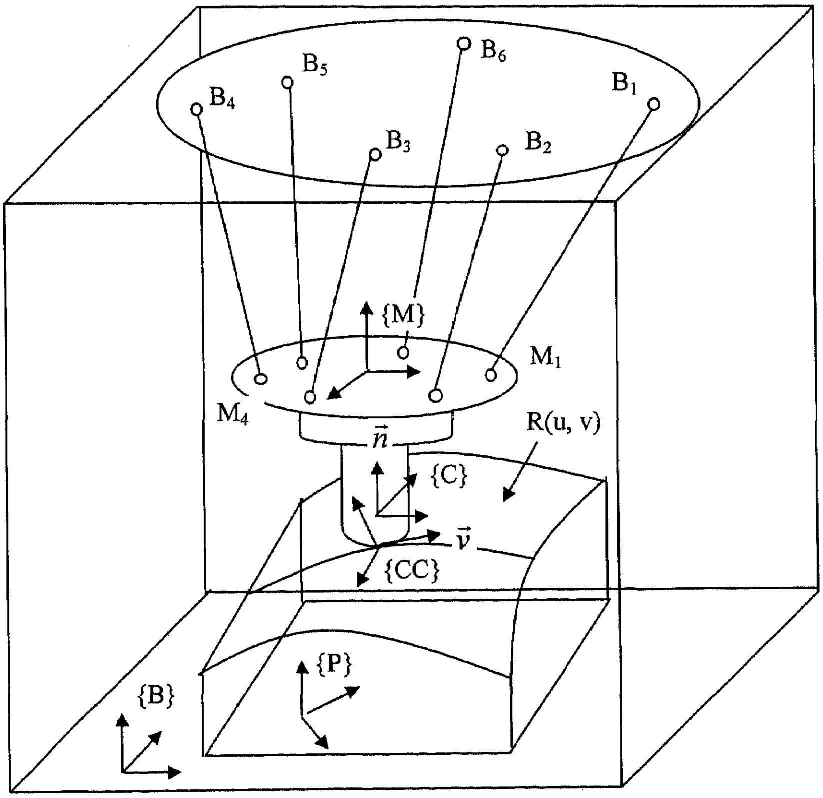


Figure 5.8 Schematics of a hexapod machine

whose joint locations are denoted as B_1, B_2, ..., B_6, and to a mobile plate, whose joint locations are M_1, M_2, ..., M_6. A base frame $\{B\}$ is attached to the base plate at an arbitrary location and used as the reference. A mobile frame $\{M\}$ is attached to the mobile plate and used to define the motion of the mobile plate and its six mobile joints. A cutter frame {C} is attached to the cutter at the center of the cutter with the z-axis parallel to the cutter axis $\vec{n}$ and an arbitrary x-axis. The part geometry for machining operations is described in a part frame {P} which is defined in the base frame {B}. The six-degrees-of-freedom (6-DOF) of mobile frame {C} is obtained by controlling the six leg lengths.

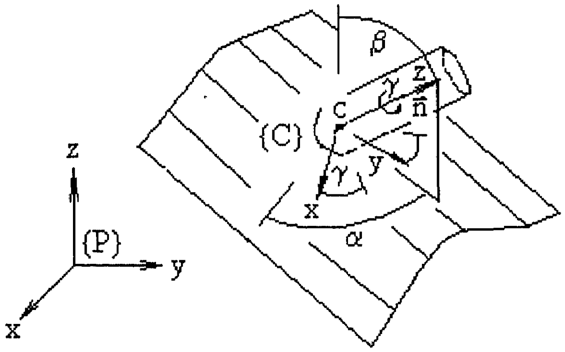


Figure 5.9 Cutter frame in frame {P}

5.2.1 Introduction of Extra Degree of Freedom (e-DOF)

One obvious difference for the tool path planning of a hexapod is the introduction of an extra degree of freedom. When a 6-DOF hexapod is used for five-axis machining, the motion is specified only by the 5-DOF cutter pose $CL(x_C, y_C, z_C, i, j, k)$. Since the vector $\vec{n}$ can also be defined as $\vec{n}(\alpha, \beta)$ with angles α and β as shown in Figure 5.9, another form of CL data is

$$CL^{*} = (x_C, y_C, z_C, \alpha, \beta) \tag{5.5}$$

where α is the angle of the x-y plane projection of $\vec{n}$ to the x-axis, and β is the angle between $\vec{n}$ and the z-axis. During the actual machining, the cutter is driven by a spindle motor with respect to its axis at high speed and the mobile plate that carries the spindle is also allowed to rotate about the cutter axis. The same cutter pose, $CL^{*} = (x_C, y_C, z_C, \alpha, \beta)$, can be maintained while rotating the cutter frame {C} about its z-axis. This rotation (angle γ) about the tool axis by the mobile plate does not affect the machining motion (5-DOF) of the cutter, but must be defined to control the motion (6-DOF) of the hexapod. We call this rotation as the 'Extra Degree of Freedom (e-DOF)' for machining.

In this dissertation, the value of e-DOF is measured based on the Z-Y-Z Euler angles as shown in Figure 5.10. To reach a CL^{*} with a tool orientation of $\vec{n}(\alpha, \beta)$ in the part frame {P} defined by O-XYZ, a frame of O-X''Y''Z'' is uniquely defined. The steps to obtain this frame are as follows. Start with frame O-X''Y''Z'' coincident with {P}. First rotate O-X''Y''Z'' about Z'' by an angle α to frame O-X'Y'Z', and then rotate about Y'' to frame O-X''Y''Z''. If a value is assigned to e-DOF, the orientation of the cutter frame {C} is the same as frame O-X'''Y'''Z''' which is obtained by rotating about Z'' by an angle γ. Therefore, the rotation matrix from frame {C} to {P} is

$$^{P}R_C(\gamma) = \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix} \tag{5.6}$$

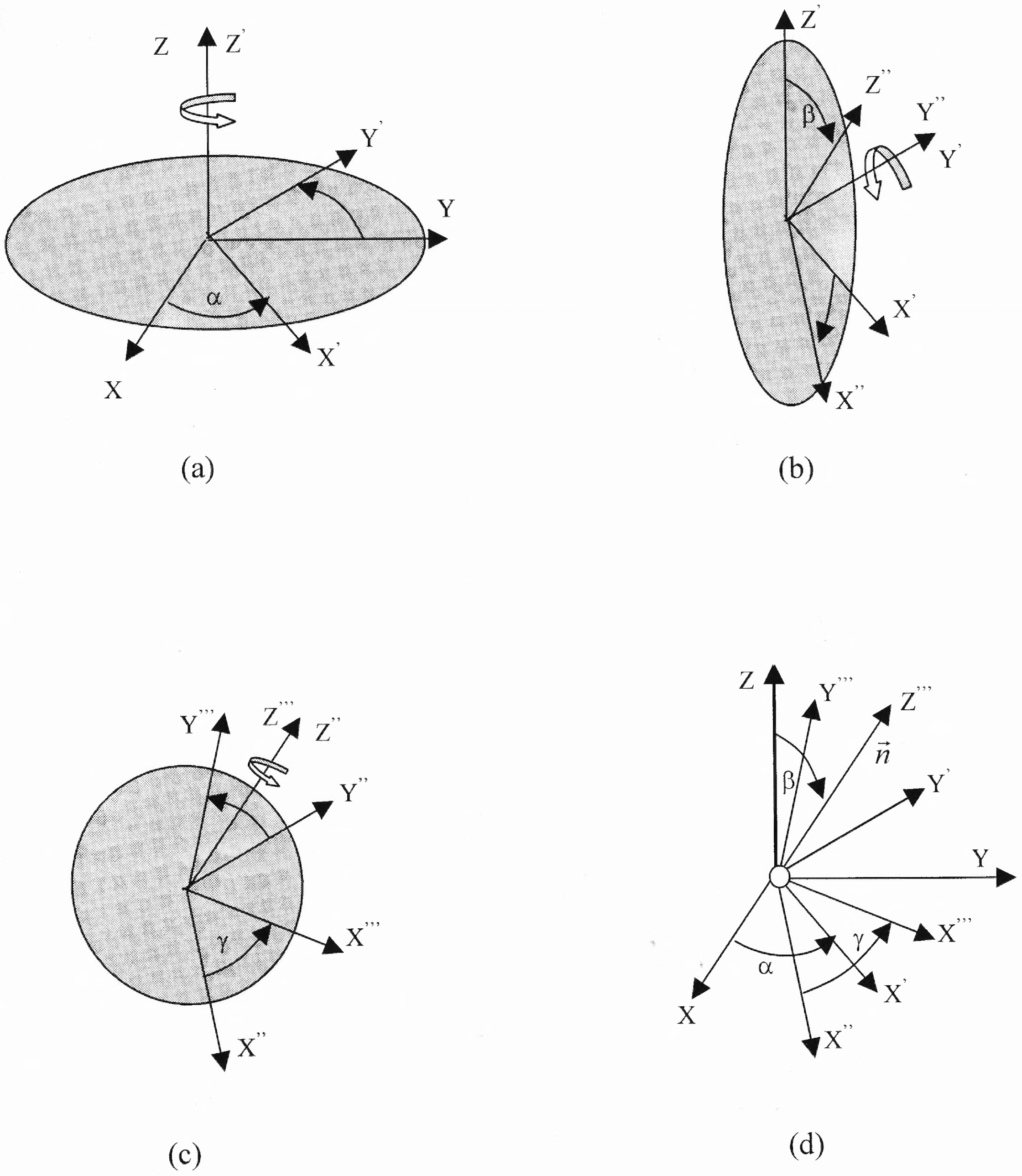


Figure 5.10 Z-Y-Z Euler angles

At a 5-DOF tool location, this transformation requires the value of γ to be completely defined. Therefore the e-DOF has direct influence on the kinematics, dynamics and motion of a hexapod.

5.2.2 Highly Variable Inverse Jacobian of Hexapods

For a five-axis machine, a closed form solution of the forward kinematics can be obtained by $\vec{x} = f(\vec{q})$, where $\vec{x}$ is a vector of the five Cartesian coordinates and $\vec{q}$ is a vector of the five joint coordinates. The mapping of joint velocity $\dot{\vec{q}}$ to the Cartesian velocity $\vec{v}$ ($\dot{\vec{x}}$) is described by $\vec{v} = J\dot{\vec{q}}$, where J is called the Jacobian matrix, while the mapping of $\vec{v}$ to $\dot{\vec{q}}$ is $\dot{\vec{q}} = J^{-1}\vec{v}$. Here J^{-1} is the inverse Jacobian matrix. For a machine of type A, the forward kinematics can be derived from Equation (5.3) as

$$\begin{aligned} x_C &= X\cos C + (Y - L\cos A')\sin C \\ y_C &= -X\sin C + (Y - L\cos A')\cos C \\ z_C &= Z' - L\sin A' \\ \alpha &= 90^\circ - C \\ \beta &= 90^\circ - A' \end{aligned} \tag{5.7}$$

with $\vec{x} = (x_C, y_C, z_C, \alpha, \beta)^T$ and $\vec{q} = (X, Y, Z', C, A')^T$. By taking partial derivatives with respect to (X, Y, Z', C, A') based on Equation (5.7), the Jacobian matrix is obtained as

$$J = \begin{bmatrix} cC & sC & 0 & LsA'sC & -XsC + (Y - LcA')cC \\ -sC & cC & 0 & LsA'cC & -XcC - (Y - LcA')sC \\ 0 & 0 & 1 & -LcA' & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{5.8}$$

Similarly, J^{-1} can be obtained from Equation (5.3),

$$J^{-1} = \begin{bmatrix} s\alpha & -c\alpha & 0 & 0 & -x_C c\alpha - y_C s\alpha \\ c\alpha & s\alpha & 0 & -Lc\beta & 0 \\ 0 & 0 & 1 & Ls\beta & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{5.9}$$

Note that sine and cosine functions have been shorthanded as $s(*)$ and $c(*)$.

For a 6-DOF hexapod, the explicit expressions, $x_i = f_i(l_1,\cdots l_6)$ (i=1,...,6), for the forward kinematics are not available. It is therefore impossible to derive the Jacobian matrix directly. On the other hand, the explicit expressions, $l_i = f_i(x_1,\cdots x_6)$ (i=1,...,6), which describe its inverse kinematics, can be conveniently obtained. Therefore, the inverse Jacobian can be easily obtained from the inverse kinematics expressions. This matrix can then be inverted to produce the Jacobian matrix. Let's first find expressions for leg lengths. In Figure 5.8, the 6-DOF location of cutter frame {C} expressed in the part frame {P} is $\vec{x}(x,y,z,\alpha,\beta,\gamma)$ which includes the 5-DOF required by a machining task and the extra DOF, i.e., $\vec{x}(x,y,z,\alpha,\beta,\gamma) = \vec{x}(CL^*,\gamma)$. Note that the subscript 'C' has been omitted in $CL^{*}=(x_C, y_C, z_C, \alpha, \beta)$. For one cutter frame pose $\vec{x}(CL^{*},\gamma)$, the position of joint M$_i$ in part frame {P}, ${}^{P}\vec{M}_i$, is

$$ {}^{P}\vec{M}_i = {}^{P}\vec{C} + {}^{P}R_C\,{}^{C}\vec{M}_i \qquad (5.10)$$

where ${}^{P}\vec{C}$ is the cutter center position vector $\{x, y, z\}^T$, and ${}^{C}\vec{M}_i = {}^{C}t_M + {}^{C}R_M\,{}^{M}\vec{M}_i$. The joint position in the base frame {B} is

$$ {}^{B}\vec{M}_i = {}^{B}t_P + {}^{B}R_P\,{}^{P}\vec{M}_i \qquad (5.11)$$

Therefore, the leg length of leg i is

$$ l_i(x, y, z, \alpha, \beta, \gamma) = \left| {}^{B}\vec{B}_i - {}^{B}\vec{M}_i(\gamma) \right| = \left| {}^{B}\vec{B}_i - {}^{B}\vec{t}_P - {}^{B}R_P\left({}^{P}\vec{C}(x, y, z) + {}^{P}R_C(\alpha, \beta, \gamma)\,{}^{C}\vec{M}_i \right) \right| \qquad (5.12)$$

If ${}^{B}B_i = \begin{pmatrix} x_{Bi} & y_{Bi} & z_{Bi} \end{pmatrix}$, ${}^{C}M_i = \begin{pmatrix} x_{Mi} & y_{Mi} & z_{Mi} \end{pmatrix}$, ${}^{B}\vec{t}_P = \begin{pmatrix} x_P & y_P & z_P \end{pmatrix}^T$

and ${}^{B}R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & & 1 \end{bmatrix}$, then the leg length for leg i is

$$ l_i = \sqrt{\begin{array}{l} \left(\left(x - x_{Bi} - x_P\right) + \left(c\alpha c\beta c\gamma - s\alpha s\gamma\right)x_{Mi} + \left(-c\alpha c\beta s\gamma - s\alpha c\gamma\right)y_{Mi} + \left(c\alpha s\beta\right)z_{Mi}\right)^2 \\ + \left(\left(y - y_{Bi} - y_P\right) + \left(s\alpha c\beta c\gamma + c\alpha s\gamma\right)x_{Mi} + \left(-s\alpha c\beta s\gamma + c\alpha c\gamma\right)y_{Mi} + \left(s\alpha s\beta\right)z_{Mi}\right)^2 \\ + \left(\left(z - z_{Bi} - z_P\right) + \left(-s\beta c\gamma\right)x_{Mi} + \left(s\beta s\gamma\right)y_{Mi} + \left(c\beta\right)z_{Mi}\right)^2 \end{array}} $$

$$ (i=1,\ldots,6) \qquad (5.13)$$

From equation (5.13), each leg length is a function of the six variables of x, y, z, α, β, and γ. By obtaining the part derivatives of each leg with respect to x, y, z, α, β, and γ from (5.13), the inverse Jacobian J^{-1} mapping Cartesian velocity $\vec{v}$ to the joint velocity $\dot{\vec{l}}$ becomes a 6x6 matrix

$$J^{-1} = \begin{bmatrix} \partial l_1/\partial x & \partial l_1/\partial y & \cdots & \partial l_1/\partial \gamma \\ \partial l_2/\partial x & \partial l_2/\partial y & \cdots & \partial l_2/\partial \gamma \\ \vdots & \vdots & \ddots & \vdots \\ \partial l_6/\partial x & \partial l_6/\partial y & \cdots & \partial l_6/\partial \gamma \end{bmatrix} \tag{5.14}$$

The expression for each entry of the Inverse Jocabian is very complicated and lengthy. For example, entry $\partial l_1/\partial \alpha$ is shown in Equation (5.15). Note as the formula is directly retrieved from Mathematica 3.0, parameters like xB1, xM1 and xP actually represent x_{B1}, x_{M1}, and x_P, respectively. The detailed expression for the whole inverse Jacobian matrix is presented in Appendix A. For such a complex symbolic matrix, the explicit expression for the Jacobian matrix is not available now, as it is beyond our computation power (Mathematica 3.0 on SUN SPARK 20) to get its inversion. The only way to get the Jacobian is from the numerical inversion of the inverse Jacobian for a given cutter frame pose.

Comparison between the inverse Jacobian matrices of these two types of machines could give us some useful information about their difference. In the inverse Jacobian of the five-axis machine in Equation (5.9), there are 13 zeros and 3 constants. A

$\partial l_1 / \partial \alpha =$

```
   (2 (-zM1 Cos[α] Sin[β] -
         yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
         xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ]))
       (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
         xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
         yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) +
      2 (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
         yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
         xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ]))
       (zM1 Sin[α] Sin[β] -
         xM1 (-Cos[β] Cos[γ] Sin[α] - Cos[α] Sin[γ]) -
         yM1 (-Cos[α] Cos[γ] + Cos[β] Sin[α] Sin[γ]))) /
 (2 √ ((-z + zB1 - zp - zM1 Cos[β] +
            xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
         (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
            yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
            xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
         (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
            xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
            yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
        2))
```

(5.15)

zero entry means that there is no coupling between the corresponding Cartesian and joint components, and a constant indicates that the contribution of the Cartesian component to the corresponding joint velocity does not change. Also the two rotations α and β are totally decoupled. As each axis of the machine can individually realize its Cartesian component without coupling, the nonuniformity of the inverse Jacobian is due to the nonlinear transformation from CL to MCD. However, for the inverse Jacobian of the hexapod shown in Equation (5.14), every entry of J^{-1} is a nonlinear function of pose $\vec{x}(x,y,z,\alpha,\beta,\gamma)$. This nonlinearity is mainly from the coupling of the six legs and it can not be decoupled at any time and any location. As a result, the inverse Jacobian varies with $\vec{x}$ moving across the workspace.

A varying inverse Jacobian within a hexapod's workspace is a significant dissimilarity between a hexapod and a five-axis machine. As a result, the accuracy, rigidity, dexterity, and manipulability also vary across the workspace. Various measures of the inverse Jacobian may be used to evaluate the kinematics performance of a hexapod. Among them, the condition number is defined as the ratio of the maximum to minimum singular values of the inverse Jacobian [Salisbury and Craig, 1982]. With the singular value decomposition method, this inverse Jocabian matrix can be decomposed as

$$J^{-1} = \mathbf{U}\Sigma\mathbf{V} \tag{5.16}$$

where **U** and **V** are 6x6 orthogonal matrices, and $\Sigma \overset{\Delta}{=} diag(\sigma_1, \cdots, \sigma_6) \in R^{6x6}$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_6 \geq 0$. The $\sigma_i (i = 1, \cdots, 6)$ are called the singular values (here singular values are the eigenvalues) of J^{-1}. Particularly, σ_1 and σ_6 are referred to as the largest and smallest singular values respectively. Note that the singular values are uniquely determined, although **U** and **V** may not be. Therefore the condition number is $w_1 = \sigma_1 / \sigma_6$, and is used to measure the dexterity, which is the most dominant factor in the kinematic performance of parallel machine tools [Huang et al, 1998]. An isotropy configuration, which is regarded as a criterion to find the optimal configuration, can be found by minimization of the condition number [Gosselin and Angeles, 1988]. The smallest singular value is used as the second measure [Gosselin and Angeles, 1988],

$w_1 = \sigma_6$. The third measure was proposed by Yoshikawa [Yoshikawa, 1985], which uses the absolute value of the inverse Jacobian determinant ($w_3 = \left|\det J^{-1}\right|$) to judge the manipulability. According to Huang et al. [Huang et. al, 1998], any one of the above three measures can be used as the kinematic performance index as they have the same identical extreme value conditions.

We choose the manipulability as our kinematic performance index. The kinematic performance index of a hexapod varies within its workspace. Let us use the NJIT Hexapod [Ji and Song, 1998] as an example to illustrate this. In this machine, the positions of joint B$_i$ in base frame {B} and joint M$_i$ in {M} are:

$$\begin{aligned}
{}^{B}B_1 &= (1636200 \quad 107500 \quad 428100) \\
{}^{B}B_2 &= (1470700 \quad 725000 \quad 428100) \\
{}^{B}B_3 &= (-725000 \quad 1470700 \quad 428100) \\
{}^{B}B_4 &= (-1363200 \quad 911200 \quad 428100) \\
{}^{B}B_5 &= (-725000 \quad -1470700 \quad 428100) \\
{}^{B}B_6 &= (-107500 \quad -1636200 \quad 428100)
\end{aligned}$$

$$\begin{aligned}
{}^{M}M_1 &= (576800 \quad -100000 \quad -450000) \\
{}^{M}M_2 &= (375000 \quad 449500 \quad -450000) \\
{}^{M}M_3 &= (-201800 \quad 549500 \quad -450000) \\
{}^{M}M_4 &= (-576800 \quad 100000 \quad -450000) \\
{}^{M}M_5 &= (-375000 \quad -449500 \quad -450000) \\
{}^{M}M_6 &= (201800 \quad -549500 \quad -450000)
\end{aligned}$$

where the length unit is the basic length unit (BLU) which is the resolution (displacement per pulse) of the leg drive (all lengths are in BLU and angles in degree unless specified).

In this example, one BLU is 0.00001 inch/pulse and frame {C} coincides with frame {M}. Figure 5.11 shows that the index varies over a very big range across one x-β plane of frame {C} with y=300000, z=4900000, α=0° and γ=1°. This means that the kinematic performance is different from one position to another position. This variation will directly affect workspace quality such as accuracy and stiffness.

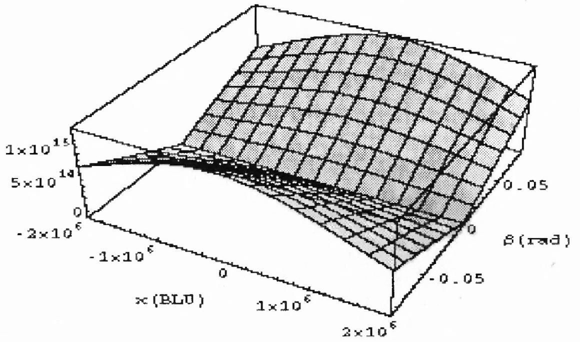


Figure 5.11 Kinematic performance index of the hexapod

One effect of the volatile inverse Jacobian is that the same amount of joint errors will cause different Cartesian errors at different locations $\vec{x}$. Suppose each leg has an error of one BLU, $\delta\vec{l} = (1,1,1\,1,1,1)^T$; Cartesian errors caused by the joint errors have a form $\delta\vec{x} = J\delta\vec{l}$. Consider two different locations of Figure 5.11. At the location with x=100000 and β=1°, the kinematic index is w=2.699×10^{14}. The corresponding position error is $\sqrt{\delta x^2 + \delta y^2 + \delta z^2}$ =1.036 and the corresponding orientation error, $\sqrt{\delta\alpha^2 + \delta\beta^2 + \delta\gamma^2}$, is 1.652°×10^{-7}. At the location with x=100000 and β=0.005°, the kinematic index decreases to w=1.347×10^{12}. The position error is unchanged, but the orientation error increases to

$2.763^{\circ}\times 10^{-5}$. For the same joint errors, the orientation error at location two is more than 160 times larger than that at location one.

5.2.3 Nonlinear Errors of Hexapods

If $CL^{*,k}$ and $CL^{*,k+1}$ represent two 5-DOF CL cutter poses with a constant orientation for a five-axis machine, and $\vec{q}^{k}$ and $\vec{q}^{k+1}$ are the corresponding joint coordinates (MCD data), then the five-axis machine will follow a linear line segment connecting $CL^{*,k}$ and $CL^{*,k+1}$ by computing the intermediate joint coordinates with linear interpolation of $\vec{q}^{k}$ and $\vec{q}^{k+1}$, since its joint coordinates are collinear functions of its Cartesian coordinates in this case. Therefore there is no error caused by this linear interpolation. If the orientation changes between two neighboring CL locations, the two rotation axes can realize this change in the two orthogonal angles of α and β. To maintain contact between the workpiece and cutter tip, the x, y, and z axes will have to perform the corresponding changes with the change of orientation. However the joint coordinates (leg lengths) are nonlinear functions of the Cartesian coordinates in a hexapod in most cases. Figure 5.12 shows the length change in one of the legs when the hexapod moves linearly from $\begin{pmatrix}9000 & 0 & 4900000 & 9^{\circ} & 9^{\circ} & 0^{\circ}\end{pmatrix}$ to $\begin{pmatrix}11000 & 0 & 4900000 & 11^{\circ} & 11^{\circ} & 0^{\circ}\end{pmatrix}$, which is far from linear. Therefore, driving the legs with linear interpolation in joint space will result in Cartesian errors as shown in Fig. 5.13.

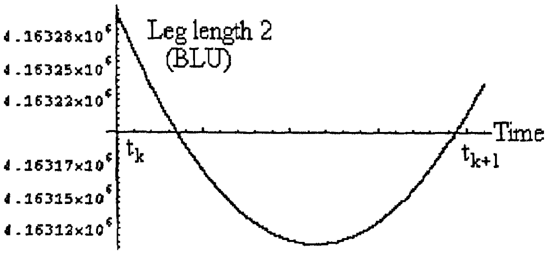


Figure 5.12 Hexapod leg length change for a linear path

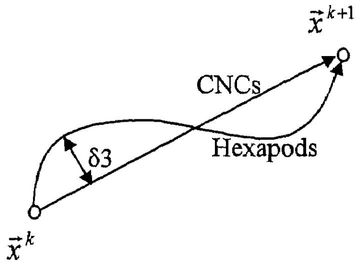


Figure 5.13 Nonlinear error of a hexapod

To find the actual path generated by the linear interpolation in joint space will require difficult forward kinematics. Newton's numerical approach is therefore used to evaluate the resulting Cartesian errors caused by the linear interpolation in joint space. In this method, the six leg length expressions, $l_i = f_i(x, y, z, \alpha, \beta, \gamma)$ (i=1,...,6), form an equation vector $\vec{F} = (l_1 - f_1, \cdots, l_6 - f_6)^T$. If $\vec{X} = (x, y, z, \alpha, \beta, \gamma)^T$, we have a system of equations of $\vec{F}(\vec{X}) = 0$. Define the recursive sequence by

$$\vec{X}^{k+1} = \vec{X}^{K} - J(\vec{X}^{k})F(\vec{X}^{k}) \qquad \text{(k=0, 1, 2, ...)} \qquad (5.17)$$

If the sequence converges to a vector $\vec{X}$ such that $\left|\vec{F}(\vec{X})\right| \approx 0$, the solution for the system is found. As an example, let's look at a linear path between pose $(200000 \quad 200000 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ)$ and pose $(202500 \quad 202500 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ)$ of our hexapod; its sampling length is 3356 BLU with a fixed orientation. Figure 5.14 shows the nonlinear errors of the cutter frame under joint interpolation. In this calculation, the position error is defined by $\sqrt{\delta x^2 + \delta y^2 + \delta z^2}$, and the orientation by the $\sqrt{\delta\alpha^2 + \delta\beta^2 + \delta\gamma^2}$. The position error reaches 2.76BLU at the middle of the path. This is about 0.7μm for this machine (1BLU=0.254μm).

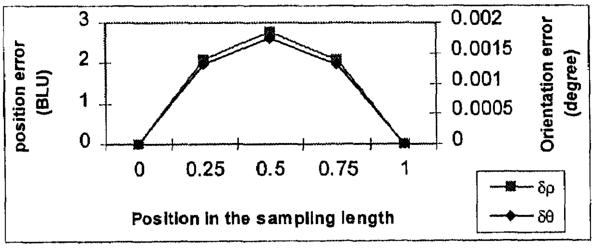


Figure 5.14 Nonlinear errors within one sampling length

When the sampling length increases, the nonlinear error will also increase as shown in Fig. 5.15. Therefore, this error will become an important source of the total approximation error for a hexapod in high speed machining. In order to reduce the nonlinear error, the sampling length should be controlled within an allowable range. This will limit the maximum feedrate of a hexapod when sampling frequency can not be increased. Linear interpolation in Cartesian space or direct interpolation may be used to reduce this nonlinear error. However the computation overhead may put some limitation on the sampling frequency and then limits the sampling length to be smaller for a desired feedrate. Therefore, the high speed machining applications of this type of machines will require a method to provide high feedrate with low nonlinear errors.

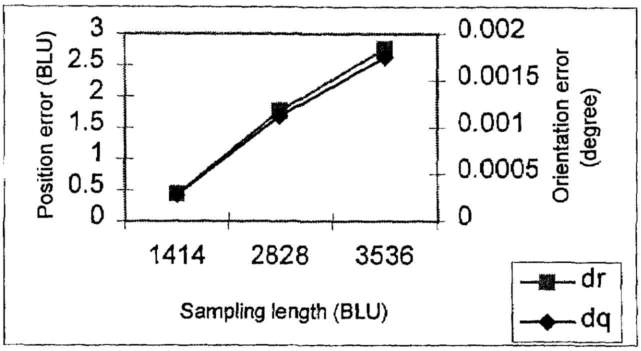


Figure 5.15 Nonlinear errors vs. sampling lengths

5.2.4 Effects of Extra Degree of Freedom (e-DOF)

The existence of an extra-degree-of-freedom makes it possible to modify the machine motion for the same tool motion. Figure 5.16 shows the change in the determinant of the inverse Jacobian of our hexapod for different γ values at location $(x, y, z, \alpha, \beta) = \left(700000 \quad 700000 \quad 4900000 \quad 5^\circ \quad 5^\circ\right)$. Although the value of γ does not vary in such a big range in practice, it clearly shows that the kinematic performance can be affected by the value of the e-DOF without changing the required 5-DOF location.

There is a lack of understanding on the effect of the e-DOF because of the complexity of the kinematics involved. As a result, there exists no really effective method to plan the e-DOF. A fixed value is often assigned to the angle γ arbitrarily in practice.

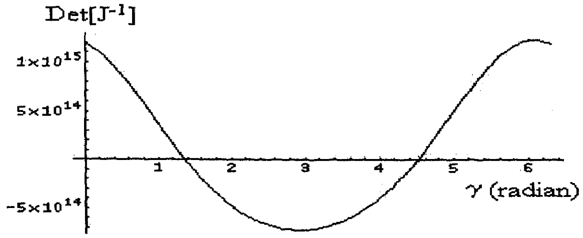


Figure 5.16 Determinant vs. e-DOF γ

Determination of e-DOF is an essential task for tool path planning. If properly planned, this e-DOF can be used to avoid joint limits, leg interference and singularities, to reduce the nonlinear error and peak velocity, acceleration and torque, to improve the geometric and kinematic conditions of the tool paths, motion and dynamics performances, and finally to recover from any leg failure. In order to generate successful tool paths for hexapods, we should fully explore the benefits of the e-DOF. Good planning based on the e-DOF will certainly provide better usage of a hexapod than that with an arbitrarily chosen γ value.

All these factors indicate that the traditional path planning methods do not have the ability to handle the problems associated with these new machines, and should not be used for hexapods without modification. New methods that account for these factors are highly desired to generate the tool paths for hexapod machine tools.

5.3 Summary

In this chapter, we studied and analyzed the three dissimilarities between tool path planning of hexapods and five-axis machines. The first significant difference is the existence of an extra degree of freedom (e-DOF), which is the rotation of a hexapod machine tool about its tool axis. It represents a redundancy for 5-axis machining but must be specified for controlling the motion of the hexapod. If properly used, this e-DOF could improve the geometric and kinematic conditions of the tool paths. Second, a hexapod has a widely varying inverse Jacobian. As a result, the accuracy, rigidity and other characteristics also vary across the workspace, and some regions have poor accuracy and stiffness. The stiffer and more accurate operations might not be realized if the tasks are not well planned. Third, a hexapod machine usually generates a nonlinear segment when following a straight-line segment over two sampled poses. This nonlinear segment reduces the accuracy when the sampling length must be large in high speed machining. All these factors indicate that the traditional path planning methods should not be used for hexapod machines without modification.

CHAPTER 6

KINEMATICS-BASED TOOL PATH PLANNING

Chapter 5 analyzed the major differences in tool path planning between five-axis CNC machines and hexapods. It shows that traditional tool path planning methods developed for conventional CNC machines should not be used directly for hexapods. In this chapter, a kinematics-based planning scheme for hexapod tool path generation is proposed. First, it illustrates the overall design and major components of this method. Then detailed algorithms for the two major new components (part placement and e-DOF planning) are described. Finally, a system which implements this method is presented to show the process and results of the method. It will be shown how this new approach can produce better results than conventional ones.

6.1 Methodology

When planning the tool paths for a hexapod to machine a surface, a set of 5-DOF tool paths can be planned first based on the differential geometry of the surface and the type of cutter used. Referring to Figure 5.8, both the surface and the 5-DOF tool paths are defined in a frame {P} attached to the part. The actual machining tool paths in the reference frame of the machine depend on the placement of the part within the workspace of the hexapod. The first problem is therefore how to determine a part location (represented by frame {P}) in the base frame {B} for a given machining task. Once the part is mounted for machining, another problem is how to determine the value of the e-

DOF to completely define the transformation from cutter frame {C} to base frame {B}. Since the conventional geometry-based tool path planning methods do not address these problems, a kinematics-based method is proposed to guide the part placement and the determination of γ angle. The proposed method is actually a combination of a traditional five-axis tool path planning method with two major new modules for the kinematics-based enhancement. This method is referred to as the kinematics-based method because only the physical constraints and kinematic performance are considered in the determination of the γ angle. The stiffness of a machine is another one of the important characteristics that affects the machine's performance. There are several types of machining operations that do not require direct contact between the tool, such as machining with laser or water-jet. The kinematics-based planning is when a hexapod is used for these operations since cutting force is not present. The planning of the e-DOF based on stiffness is currently under investigation for machining operations that involve significant machining forces and is not part of this study.

Figure 6.1 is the block diagram of the proposed kinematics-based tool path planning module. This interpolator takes the input from design models created by CAD programs, and produces the reference pulses for controlling each axis of a hexapod. It has the following major components.

CAD model interpreter: The purpose of the CAD model interpreter is to prepare the geometries for planning their tool paths. The input to this scheme can be G-code, curves, surfaces or part models.

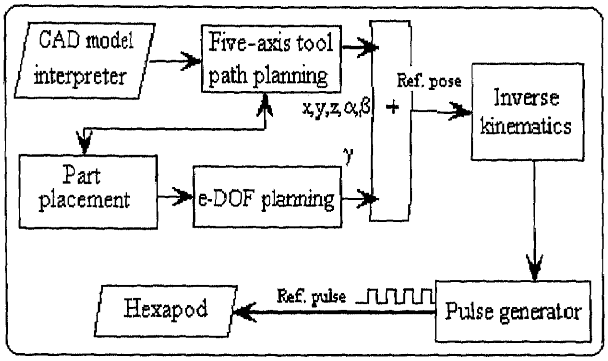


Figure 6.1 A kinematics-based tool path planning scheme

Five-axis tool path planner: If the input is G-code, this unit does not need to do anything. When the input is curves, the interpolator starts directly from the trajectory planning, which generates the discrete 5-DOF CL path of $CL^*(x, y, z, \alpha, \beta)$. For surface machining, the interpolator begins with tool path planning, which generates curves as the tool paths, and subsequently executes the trajectory planning portion.

Part placement: This unit will search for a feasible location of the part frame {P} such that the part is placed in a location which meets all the constraints. A search algorithm for part placement is introduced in section 6.2.

E-DOF planner: This planner extends the set of 5-DOF tool paths to a set of 6-DOF tool paths by adding values of γ thereby enhancing machines' performance. A detailed discussion of the planner is presented in Section 6.3.

Also shown in Figure 6.1 are the components of the inverse kinematics, pulse generator and a hexapod where the planned 6-DOF tool paths are sampled and converted into the leg lengths for controlling the machine.

Gouging is particularly pernicious in sculptured-surface machining, and it is often encountered when the tool size is too large relative to the concave radius of curvature. The machining of objects composed of multiple surfaces can also cause gouging. This problem is well investigated and there are many methods to deal with it [Li and Jerard, 1993; Huang and Oliver, 1995]. Most commercial CAD/CAM software can generate gouge free tool paths. Since the proposed method is based on traditional five-axis tool path planning, gouging avoidance and other 5-DOF tool path issues are not part of this research; it is assumed that the 5-DOF tool paths used in this dissertation are gouge free.

6.2 Part Placement

In CAD systems, a surface is usually defined in parametric forms as **r**(u,v)=$(x(u,v), y(u,v), z(u,v))^T$. After a set of 5-DOF tool paths is planned for this surface, the following two conditions must be satisfied for a placement of {P} in {B} to be considered as feasible.

1. The surface (i.e. the planned cutter paths) should be completely accessible by the cutter mounted on the hexapod from the specified orientations.
2. The kinematics performance index at each location of the surface should be above a threshold value to guarantee the required performance.

The accessibility of a cutter location deals with the physical limitations in the structure of a hexapod. It checks whether the leg lengths are within their ranges of motion, whether the passive joints are within their limits, and whether there is interference among legs and plates. For one location of {P} defined by translation ${}^{B}\bar{t}_{P}$ and rotation ${}^{B}R_{P}$ in {B}, the leg lengths, joint angles, and kinematics performance corresponding to a 5-DOF cutter pose $CL^{*}_{j,k}(x,y,z,\alpha,\beta)$ are functions of γ, where j and k are respectively the two indexes for the parametric variables u_j and v_k, which correspond to CC point $CC_{j,\,k}$. Condition 1 checks the kinematic constraints of the hexapod. All the values of γ satisfying condition 1 form an accessible γ set, $\gamma K_{j,k}$. A value of γ in the set $\gamma K_{j,k}$ means that the hexapod can freely access the cutter frame pose $\vec{x}_{j,k}(x,y,z,\alpha,\beta,\gamma)$. If at least one value of γ exists in the corresponding set for every $CL^{*}_{j,k}(x,y,z,\alpha,\beta)$ of the surface, then the surface is accessible from the location represented by ${}^{B}\bar{t}_{P}$ and ${}^{B}R_{P}$. There are many singular points within the workspace. The measure of manipulability becomes zero at these points. In the neighborhood of singular points, the index is still low and the region is regarded as a poor kinematics performance region. The surface, which is accessible from the location, may contain these points. Condition 2 is designed to avoid these singularities by keeping a certain "distance" from the singular point. The values of γ satisfying both conditions 1 and 2 form a feasible γ set. Let's denote it as $s_{j,k}$. If any one of $s_{j,k}$ is a null set, the location is not feasible. The search process will be rejected with a new part location. If none of $s_{j,k}$ is a null set, the location is considered as feasible. Whether this location will become a practical location depends on if we can find a smooth γ-path in the stage of e-DOF planning. For example, the intersection of all the $S_{j,k}$ form another set $S_\gamma = \cap S_{j,k}$, which contains the entire constant feasible γ for that placement location. A null set of S_γ means we can not find a constant γ-path across the surface to simplify the planning at this part location. A γ-path with a smoothly varying γ must be used.

Since the accessible γ set $\gamma K_{j,k}$ and feasible γ set $S_{j,k}$ are the basis of planning, their determination is an essential task. Sections 6.2.1 and 6.2.2 will describe the proposed methods for their determination.

6.2.1 Accessible γ Sets

In determining the range of γ which is accessible, there are three types of constraints to be considered. They are leg length limits, joint angle limits, and leg interference.

6.2.1.1 Leg Length Constraint: Referring to Fig. 5.8, for each planned $CC_{j,k} = \vec{R}(u_j, v_k)$, the five-axis tool path planner outputs a corresponding 5-DOF CL data of $CL^*_{j,k}(x, y, z, \alpha, \beta)$, where (x, y, z) are the coordinates of the cutter center C in the part frame {P}, and $\vec{n}(\alpha, \beta)$ is the cutter axis vector. At $CL^*_{j,k}(x, y, z, \alpha, \beta)$, the cutter center position vector ${}^{P}\vec{C}$ is constant and the rotation matrix ${}^{P}R_C$ is changing with γ. The position of the joint M_i in the part frame {P}, ${}^{C}\vec{M}_i$, is

$${}^{P}\vec{M}_i(\gamma)={}^{P}\vec{C}+{}^{P}R_C(\gamma)\,{}^{C}\vec{M}_i \tag{6.1}$$

Its position in the base frame {B} is

$${}^{B}\vec{M}_i(\gamma)={}^{B}t_P+{}^{B}R_P\,{}^{P}\vec{M}_i(\gamma) \tag{6.2}$$

Therefore, the leg length of leg i is also changing with γ angle.

$$l_i(\gamma)=\left|{}^{B}\vec{B}_i-{}^{B}\vec{M}_i(\gamma)\right|=\left|{}^{B}\vec{B}_i-{}^{B}\vec{t}_P-{}^{B}R_P\left({}^{P}\vec{C}+{}^{P}R_C(\gamma)\,{}^{C}\vec{M}_i\right)\right| \tag{6.3}$$

The leg length of each leg must be inside the movement range. That is

$$l_{i,\min}\le l_i(\gamma)\le l_{i,\max} \tag{6.4}$$

6.2.1.2 Joint Rotation Angle Constraints: The joints connecting the legs to the base and mobile plate have a fixed range of rotation. It is necessary to impose a rotation angle constraint for each rotation joint.

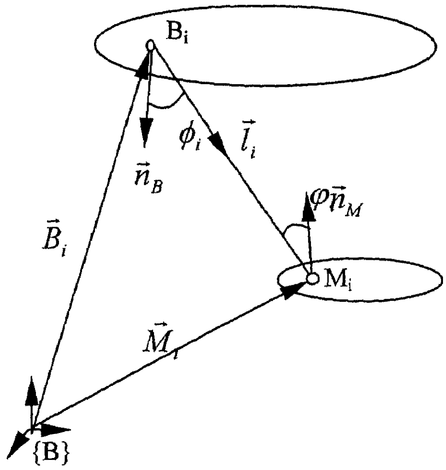


Figure 6.2 Joint angles

As shown in Fig 6.2, the leg vector for leg i is $\vec{l}_i = \vec{M}_i - \vec{B}_i$. If $\vec{n}_M$ and $\vec{n}_B$ respectively represent the vectors bisecting the rotational ranges of the mobile plate joint and base joint of leg i, the half rotational angles and their constraints can be described as

$$|\varphi_i(\gamma)| = \left|\cos^{-1}\left(\frac{\vec{n}_M \bullet (-\vec{l}_i)}{|\vec{n}_M||-\vec{l}_i|}\right)\right| \le \varphi_{i,\max} \text{ for mobile joint} \tag{6.5}$$

$$|\phi_i(\gamma)| = \left|\cos^{-1}\left(\frac{\vec{n}_B \bullet \vec{l}_i}{|\vec{n}_B||\vec{l}_i|}\right)\right| \le \phi_{i,\max} \text{ for base joint} \tag{6.6}$$

6.2.1.3 Leg Interference: When a hexapod moves, interference between any of two legs may occur. The most commonly used method to check interference is by solid modeling techniques, but it needs a lot of computation. Here we can simplify each leg module as a cylinder. If the shortest distance between any two legs, d_s, is smaller than the allowed

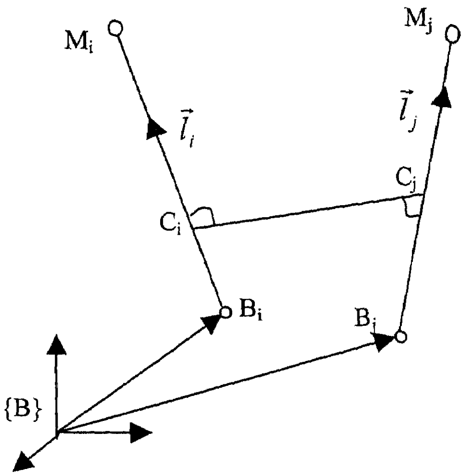


Figure 6.3 Common normal line

distance, d_a, they will be considered as interfering with each other. Therefore the constraint of no interference should be

$$d_s \geq d_a \tag{6.7}$$

The allowed distance d_a is chosen as the radius summation of the two leg cylinders with a clearance.

Procedure for Computing d_s:

The shortest distance between legs, l_i and l_j, is the distance between the intersecting points c_i and c_j of the legs in their common normal plane as shown in Fig. 6.3. From the figure, there is the following vector loop.

$$\overrightarrow{B_iC_i} + \overrightarrow{C_iC_j} - \overrightarrow{B_jC_j} = \vec{B}_j - \vec{B}_i \tag{6.8}$$

The unit vector of the common normal is $\vec{C} = \frac{\vec{l}_i \times \vec{l}_j}{|\vec{l}_i||\vec{l}_j|}$. Taking the dot product of both sides by $\vec{C}$, we have

$$\overrightarrow{C_iC_j} \bullet \vec{C} = (\vec{B}_j - \vec{B}_i) \bullet \vec{C}$$

As $\overrightarrow{C_iC_j} \bullet \vec{C} = \pm d_s$, the shortest distance is

$$d_s = |(\vec{B}_j - \vec{B}_i) \bullet \vec{C}| \tag{6.9}$$

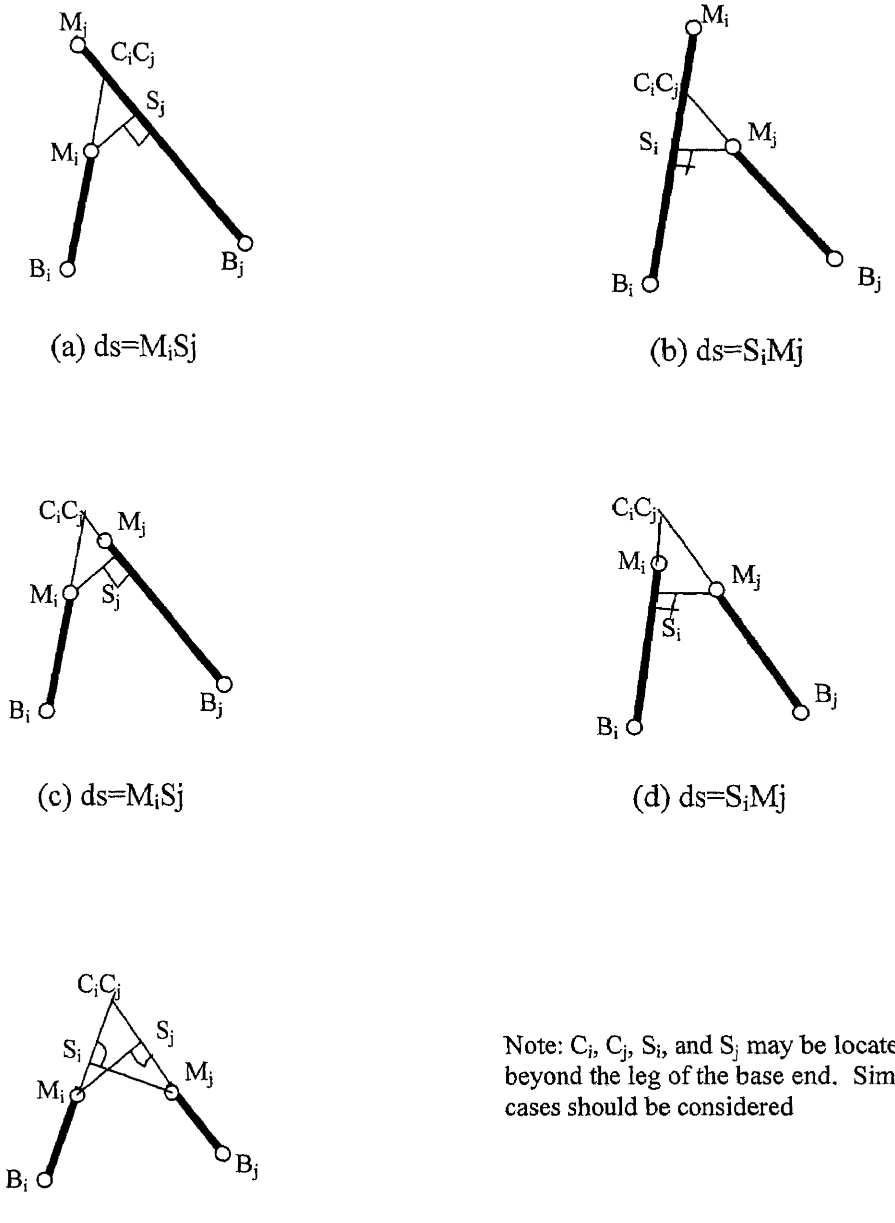
Equation (6.9) is used for the case where the two intersection points C_i and C_j are located within the leg line segments B_iM_i and B_jM_j, respectively. For the other cases, the actual shortest distance is larger than that from equation (6.9). The calculation depends on the relation of the two legs as shown in Fig. 6.4.

Case 1: If C_i is beyond B_iM_i, and C_j is between B_jM_j,

$$d_s = \overline{M_iS_j}$$

Case 2: If C_i is between B_iM_i, and C_j is beyond B_jM_j,

$$d_s = \overline{S_iM_j}$$



(a) ds=M_iSj

(b) ds=S_iMj

(c) ds=M_iSj

(d) ds=S_iMj

(e) ds=M_iMj

Note: C_i, C_j, S_i, and S_j may be located beyond the leg of the base end. Similar cases should be considered

Figure 6.4 Actual shortest distance

Case 3: If C_i is beyond B_iM_i, C_j is beyond B_jM_j,

S_i is beyond B_iM_i, and S_j is between B_jM_j,

$$d_s = \overline{M_i S_j}$$

Case 4: If C_i is beyond B_iM_i, C_j is beyond B_jM_j,

S_i is between B_iM_i, and S_j is beyond B_jM_j,

$$d_s = \overline{S_i M_j}$$

Case 5: If C_i is beyond B_iM_i, C_j is beyond B_jM_j,

S_i is beyond B_iM_i, and S_j is beyond B_jM_j,

$$d_s = \overline{M_i M_j}$$

where S_i and S_j are the intersection points of legs i and j with the normal lines from joint position M_j and M_i, respectively.

The points C_i, C_j, S_i, and S_j may be located beyond the leg ranges of the base side. If so, similar formulas can be obtained to compute their d_s.

Solving for $\gamma K_{j,k}$

Solve four equations (6.4)~(6.7) simultaneously for γ angle. The results form a set $\gamma K_{j,k}$. If the set is not null for all CL data, the surface is accessible from the location ${}^B\vec{t}_P$ with ${}^BR_P$. Whether or not this location will become a practical location depends on the evaluation of the kinematics performance index.

6.2.2 Feasible γ Sets

If γ angle changes within $\gamma K_{j,k}$, the inverse Jacobian will vary with this angle, i.e., $J^{-1}_{j,k} = J^{-1}_{j,k}(\gamma)$. By choosing manipulability $|\det J^{-1}|$ as the performance index w and determining a required value of this index, w_a, the following performance constraint should be satisfied in order to obtain a good kinematics performance for the CL point.

$$w_{j,k} = |J^{-1}_{j,k}(\gamma)| \geq w_a, \qquad \gamma \in \gamma K_{j,k} \tag{6.10}$$

The measure of manipulability can be considered as a kind of distance from singular points. It is nonnegative and becomes zero only at singular points. In a neighborhood of singular points ($w_{j,k} < w_a$), the kinematic ability is poor. The value of w_a should be set to not only avoid singular points, but also maintain the kinematic ability above a required level. The determination of w_a is based on what level of accuracy in Cartesian space can be afforded by a machine when it has an error in joint space. For a given Cartesian error $\delta\vec{x}$ and joint error $\delta\vec{l}$, w_a can be estimated by calculating the Cartesian error $\delta\vec{x}$ at the locations around singular points. All the possible solutions of inequality (6.10) construct a set $S_{j,k}$. If the set is not a null set for every CL data of the surface, the current location of the part in the base frame {B} is a feasible location. Otherwise, any null set will make the location unfeasible and a new location of the part will be sent to the above evaluations.

Relationship between the Measure of Manipulability and Stiffness

The structural stiffness of a hexapod has a great influence on the accuracy of the machine, and it is another important performance criterion of the machine. Although the measure of manipulability is not a direct measure of stiffness, it can be shown that an enhanced value of the measure will improve the stiffness.

For a hexapod we can relate a force, $\vec{F}$, applied at the tool to the resulting axial force in the legs, $\vec{f}$, as

$$\vec{f} = J^T \vec{F} \tag{6.11}$$

Here force $\vec{F} = [f_x, f_y, f_z, m_\alpha, m_\beta, m_\gamma]^T$, f_x, f_y, and f_z represent the forces in x, y, and z axes, and m_α, m_β, and m_γ are the moments in the three rotation axes, respectively. The leg deformation, $\delta\vec{l}$ caused by force $\vec{f}$ is $k^{-1}\vec{f}$, where k is the joint space stiffness and it is a 6x6 diagonal matrix of the form

$$k = EA\begin{bmatrix} 1/l_1 & 0 & 0 & 0 \\ 0 & 1/l_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1/l_6 \end{bmatrix} \tag{6.12}$$

In Equation (6.12) E is the elastic modulus of the leg material and A is the cross-sectional area of the legs (the cross-sectional area of the six legs is assumed to be the same). As the relationship of Cartesian deformation and leg deformation is $\vec{\delta l} = J^{-1}\vec{\delta x}$, we can relate force $\vec{F}$ applied at tool to the resulting Cartesian deformation, $\vec{\delta x}$, as

$$\vec{F} = J^{-T}\vec{f} = J^{-T}k\vec{\delta l} = J^{-T}kJ^{-1}\vec{\delta x} \tag{6.12}$$

where J^{-T} is the transpose matrix of the inverse Jacobian. Therefore the Cartesian space stiffness matrix is

$$K = J^{-T}kJ^{-1} \tag{6.13}$$

The envelope of the Cartesian stiffness in a 6-dimensional space becomes a hyperellipsoid. If $\sigma_{K,i}(i = 1,\cdots,6)$ are the six singular values of the stiffness matrix K, they are precisely the lengths of the semi-axes of the Cartesian stiffness ellipsoid. The volume of the ellipsoid, V_K, can be a measure of the magnitude of the stiffness [Nakamura, 1991] in the form of

$$V_K = \frac{\pi^3}{\Gamma(4)}\prod_{i=1}^{6}\sigma_{K,i} = \frac{\pi^3}{\Gamma(4)}\det K \tag{6.14}$$

where $\Gamma(*)$ is the gamma function. Since the determinant of joint space stiffness, $\det K$, is $EA\Big/\prod_{i=1}^{6} l_i$, Equation (6.14) becomes

$$V_K = \frac{\pi^3}{\Gamma(4)} \det J^{-1} * EA \Big/ \prod_{i=1}^{6} l_i * \det J^{-1} = \frac{EA\pi^3 w^2}{\Gamma(4)} \prod_{i=1}^{6} l_i \tag{6.15}$$

From Equation (6.15), it is evident that the stiffness will directly benefit from the enhancement of manipulability w. Therefore, we can use the measure of manipulability to evaluate the quality of the planned paths in terms of stiffness.

6.2.3 Part Placement Procedure

For a given part surface and its 5-DOF tool paths, $CL^*_{j,k}(x, y, z, \alpha, \beta)$, the following search procedure can be used to evaluate part placement and determine feasible γ sets which meet the kinematics requirements.

Procedure:

```
Generate a part location grid and a search sequence (usually starts from a location
corresponding to the workspace center);
Give the starting location and a set of 5-DOF tool paths;
Set ‘a location is not found’;

While (a location is not found), do begin
    For (each $CL^*_{j,k}(x, y, z, \alpha, \beta)$), do begin
        Compute accessible set $\gamma K_{j,k}$;
        If $\gamma K_{j,k}$ is null, change to next location and back to while loop;
        For (each $\gamma \in \gamma K_{j,k}$), compute feasible set $S_{j,k}$;
```

```
            If $S_{j,k}$ is null, change to next location and back to while loop;
        End;
        A location is found;
    End;
    Return {the location, $S_{j,k}$ };
```

In addition to evaluate placement location for its feasibility, the kinematic performance index can also be used to compare different feasible locations in searching for better placement. We can search for the maximum performance index $w_{j,k}$ for all $\gamma \in S_{j,k}$ and the minimum value of all $w_{j,k}$, that is

$$w = \min\{w_{j,k}\} = \min\{\max_{\gamma}\left|\det\left(J_{j,k}^{-1}(\gamma)\right)\right|\}, \quad \gamma \in S_{j,k}, \text{for all } j, k \qquad (6.16)$$

Since the MinMax value w represents the lowest performce index for the particular placement, it can then be used to compare with other placement locations.

6.3 Planning of e-DOF

The part placement produces a part placement location and the corresponding $S_{j,k}$ for each of the 5-DOF tool locations $CL^*_{j,k}(x,y,z,\alpha,\beta)$. One value of γ must be selected from set $S_{j,k}$ to form a 6-DOF tool frame pose at each $CL^*_{j,k}$. We can plan the final γ for the maximum performance index. In this method, the performance index is maximized by changing γ within the set $S_{j,k}$. The angle $\gamma_{j,k}$, associated with $w_{j,k}$ from equation

(6.16), can be used as the final planned value for each 5-DOF cutter pose $CL^*_{j,k}(x,y,z,\alpha,\beta)$. However, other values in the sets $s_{j,k}$ may also be selected based on considerations such as the continuity or smooth transition of γ along the tool paths.

6.4 Local γ-Path Planning

The planned 6-DOF tool paths are expressed as the ordered discrete cutter frame poses $\vec{x}_{j,k}(CL^*,\gamma)$. With the specified feedrate V, these paths are sent to the controller of a hexapod to generate the reference points for the machine to follow the specified intermediate paths between two 6-DOF locations of $\vec{x}_{j,k}(CL^*,\gamma_j)$ and $\vec{x}_{j+1,k}(CL^*,\gamma_{j+1})$.

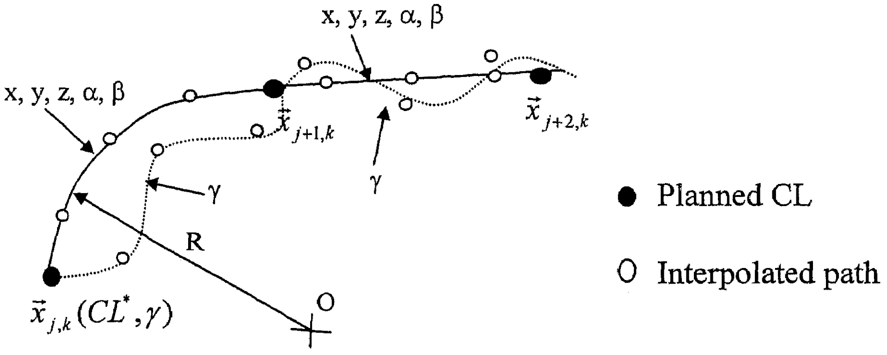


Figure 6.5 Path interpolation

Fig. 6.5 shows a planed tool path consisting of two segments formed by three poses $\vec{x}_{j,k}(CL^*,\gamma)$, $\vec{x}_{j+1,k}(CL^*,\gamma)$, and $\vec{x}_{j+2,k}(CL^*,\gamma)$. One is a circular segment from $\vec{x}_{j,k}$ to $\vec{x}_{j+1,k}$ centered at point O with a radius of R, and the other is a line segment from $\vec{x}_{j+1,k}$ to $\vec{x}_{j+2,k}$. The reference for position interpolation is the circular segment followed by the linear line segment. The references for orientation interpolation of components α and β are derived from the circle segment between $\vec{x}_{j,k}(CL^*,\gamma)$ and $\vec{x}_{j+1,k}(CL^*,\gamma)$, and the linear connections of $\alpha_{j+1,k}$, $\alpha_{j+2,k}$, and of $\beta_{j+1,k}$, $\beta_{j+2,k}$, respectively, between $\vec{x}_{j+1,k}(CL^*,\gamma)$ and $\vec{x}_{j+2,k}(CL^*,\gamma)$. These references for x, y, z, α and β are designed to guarantee the intermediate position and cutter axis orientation accuracy, and they are sampled by the interpolation algorithm of a conventional five-axis machine to generate an interpolated path. However, as there is no such a requirement for γ, its actual trajectory could be any curve as long as it passes through each of the $\gamma_{j,k}$. We can still use its intermediate values to improve the machining operations. Here are three methods to plan the intermediate γ-path.

Planning γ by Linear Interpolation of $\gamma_{j,k}$ and $\gamma_{j+1,k}$

We can simply use linear interpolation of $\gamma_{j,k}$ and $\gamma_{j+1,k}$ to generate the intermediate reference for γ. That is

$$\gamma(s) = \gamma_{j,k} + (\gamma_{j+1,k} - \gamma_{j,k})\frac{s}{S} \tag{6.17}$$

where S is the total length of the segment and s is the length from pose $\vec{x}_{j,k}$ to the current sampling point. This method can easily generate the local γ-path and guarantee the smooth transition within the segment.

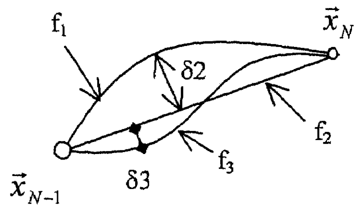


Figure 6.6 Path between two interpolation points

Planning γ by Minimizing the Total Leg Length Error

As it will be shown in Section 6.6.2, the nonlinear error changes locally with the change of γ angle, and this error can be reduced by selecting a γ with a reduced leg length error or increased performance index. The above γ-path planning method may not yield a cutter trajectory with the best accuracy because its planned γ is usually not the one with the minimum leg length error or the maximum index. In many cases, high speed and accuracy machining is sought. High speed means large sampling lengths for a machine. But when the sampling length increases, the approximation error δ2 and nonlinear error δ3 will also increase, and the leg velocity and acceleration may be above the limitations. So it remains to find the best trajectory accuracy for a given feedrate V and motion limitations. A method to find the value of e-DOF with the best trajectory accuracy within one segment can be described as follows.

Figure 6.6 shows two interpolation points of a circular segment and their intermediate path. Between the two points, the reference path is the circular arc f_1. A five-axis machine uses a straight-line segment, f_2, to approximate the circle, and the approximation error is δ2. But for a hexapod, the path is a nonlinear curve segment f_3, and the actual error in following the line segment is δ3. The worst total error in the execution stage is

$$\delta_e = \delta 2 + \delta 3 \qquad (6.18)$$

Note δ2=0 for a straight-line segment. The equation of line f_2 connecting the two cutter centers C_{N-1} and C_N is

$$\vec{C}_2(t) = \vec{C}_{N-1} + (\vec{C}_N - \vec{C}_{N-1})\frac{t}{T}, \qquad (0 \le t \le T) \qquad (6.19)$$

And the expression for f_3 can be written as

$$\vec{C}_3(t) = \vec{C}_{N-1} + J^{-1}(\xi(t))\vec{\Delta l}\frac{t}{T}, \qquad (0 \le t \le T) \qquad (6.20)$$

where only the first three rows of the inverse Jacobian are used, and $\vec{\Delta l}$ is the leg displacements between these two cutter frame poses. Error δ2 is computed by Equation (5.4), and error δ3 should have a form of

$$\begin{aligned}\delta 3 &= \vec{C}_2(t) - \vec{C}_3(t) = (\vec{C}_N - \vec{C}_{N-1} - J^{-1}(\xi(t))\vec{\Delta l})\frac{t}{T} \\ &= J^{-1}(\xi(t))(\vec{\Delta l}(\xi(t)) - \vec{\Delta l}\frac{t}{T})\end{aligned} \qquad (6.21)$$

The difference between the ideal leg length and the one generated by interpolation causes the nonlinear trajectory error. If the leg length error is denoted by $\vec{\delta l}(t)$, it becomes

$$\delta_3 = J^{-1}(\xi(t))\,\overrightarrow{\delta l(t)} \qquad (6.22)$$

It is difficult to compute the error for each point between two cutter centers. Based on the simulation results shown in Chapter 5, the maximum error usually appears at the middle of the two interpolation cutter points. We install an error indicator at the middle of the two points. The equations for calculating the error at this point can be derived as

$$\delta_{0.5VT} = R - \sqrt{R^2 - \frac{V^2T^2}{4}} + \frac{J^{-1}{}_{N-1} + J^{-1}{}_{N}}{2}\left(\vec{l}\,(\vec{x}_{N-1} + \frac{\Delta\vec{x}}{2}) - \frac{1}{2}(\vec{l}\,(\vec{x}_{N-1}) + \vec{l}\,(\vec{x}_N))\right) \qquad (6.23)$$

The solution of the above equation requires the calculation of the inverse Jacobian matrices. This is computationally expensive and almost impossible in real time. Instead, we calculate the total leg length error, $E(\gamma)$, which is defined as

$$E(\gamma) = \sum_{i=1}^{6}\left| l_i(\vec{x}_{N-1} + \frac{\Delta\vec{x}(\gamma)}{2}) - \frac{1}{2}(l_i(\vec{x}_{N-1}) + l_i(\vec{x}_N(\gamma)))\right| \qquad (6.24)$$

to indicate the magnitude of the nonlinear error, and the value of γ with the minimum leg length error is chosen. That is

$$\min_{\gamma} E(\gamma), \qquad \gamma \in [\gamma_l - \Delta\gamma, \gamma_l + \Delta\gamma] \qquad (6.25)$$

where γ_l is the value of the linear interpolation, and $\Delta\gamma$ is the tolerance of γ_l. Whether or not this planning method can succeed depends on the correlation of the nonlinear error and the total leg length error. This relationship will be analyzed and demonstrated by simulations in Section 6.6.2.

The velocity and acceleration of the legs should also be controlled to stay within their maximum values. That is

$$\dot{l}_i = \frac{\Delta l_i}{T} \leq \dot{l}_{i,a} \tag{6.26}$$

$$a_i = \frac{\Delta \dot{l}_i}{T} \leq a_{i,a} \tag{6.27}$$

Therefore to plan γ with the best trajectory accuracy is to minimize the total leg length error under the constraints of equations (6.26) and (6.27).

Planning γ-path by Minimizing the Maximum Leg Displacement

The above two planning methods have not considered axis motion performance. We can use γ to alter the dynamic characteristics of motion. For example, let legs i and j respectively have the largest and second largest displacements between two interpolation poses $\vec{x}_{N-1}$ and $\vec{x}_N$. The displacement of leg i can be reduced to a value close to the second largest by changing γ within the interval $[\gamma_l - \Delta\gamma, \gamma_l + \Delta\gamma]$. This is to minimize the term

$$\min_{\gamma}\left|l_{i,N}(\gamma)-l_{i,N-1}\right|, \qquad \gamma\in[\gamma_l-\Delta\gamma,\gamma_l+\Delta\gamma] \tag{6.28}$$

subject to the constraint of

$$\left|l_{i,N}(\gamma)-l_{i,N-1}\right|\geq\left|l_{i,N}(\gamma)-l_{i,N-1}\right| \tag{6.29}$$

In a small range of γ, as leg length is proportional to γ angle, the magnitude and direction of adjusting γ to obtain the required leg displacement can be easily determined.

6.5 A Planning and Control System

As tool path planning study is one part of the NJIT hexapod research project, the proposed kinematics-based tool path planning method will be implemented into the host computer of the system to form an integrated CAD/CAM system, so that the machine can be operated directly from a CAD file. Figures 6.7 shows the block diagram of its control system. In this system, the host computer serves as the off-line tool path planner, real-

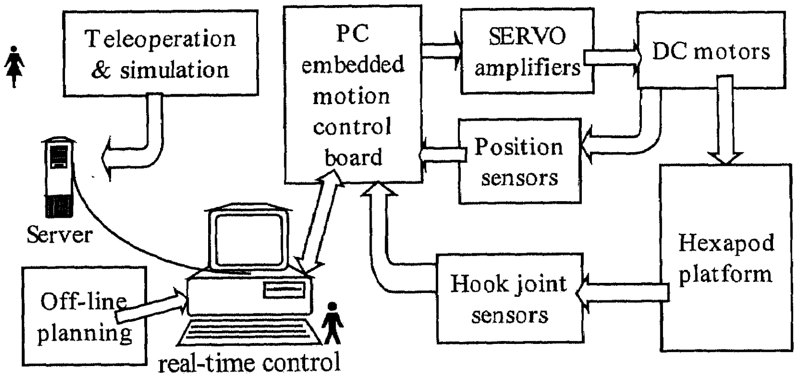


Figure 6.7 Block diagram of the system

time controller, and communicator between the system and Internet. The off-line tool path planner is a Visual C++ program implementing the proposed kinematics-based tool path planning method. The CAD files and 5-DOF CL files for planning are either generated by this program or imported from Pro/Engineer^{TM}. The planner outputs a feasible part location and a set of 6-DOF tool paths. Then the planned 6-DOF tool paths are executed in real-time. They are sampled into discrete reference poses, and then transferred into leg length references, which are used by the embedded motion control board to control the legs and cause the mobile plate to reach the reference poses. In the following section, several examples will be presented to demonstrate the feasibility, implementation process, potential and effectiveness of the method.

6.6 Simulation Examples

As the proposed kinematics-based tool path planning method is built up on the base of a traditional five-axis tool path planning method which has considered the gouging problem, the 5-DOF tool paths in the following examples are assumed gouging free.

6.6.1 Example for Feasibility Study

The first example is to illustrate the possibility and potential of the proposed method for improving machining operations. In this example, We define a machine with the same base and mobile plates as the NJIT hexapod without considering its mechanical constraints, and a cubic NURBS surface is also defined. A set of 5-DOF tool paths is then generated for this surface, and the performance index across the surface for different values of γ and different part locations is finally evaluated.

For this machine, the positions of joint B$_i$ in base frame {B} and joint M$_i$ in {M} are:

$$
\begin{aligned}
{}^{B}B_1 &= (1636200 \quad -107500 \quad 7571900) \\
{}^{B}B_2 &= (1470700 \quad -725000 \quad 7571900) \\
{}^{B}B_3 &= (-725000 \quad -1470700 \quad 7571900) \\
{}^{B}B_4 &= (-1363200 \quad -911200 \quad 7571900) \\
{}^{B}B_5 &= (-725000 \quad 1470700 \quad 7571900) \\
{}^{B}B_6 &= (-107500 \quad 1636200 \quad 7571900)
\end{aligned}
$$

$$
\begin{aligned}
{}^{M}M_1 &= (576800 \quad -100000 \quad -450000) \\
{}^{M}M_2 &= (375000 \quad 449500 \quad -450000) \\
{}^{M}M_3 &= (-201800 \quad 549500 \quad -450000) \\
{}^{M}M_4 &= (-576800 \quad 100000 \quad -450000) \\
{}^{M}M_5 &= (-375000 \quad -449500 \quad -450000) \\
{}^{M}M_6 &= (201800 \quad -549500 \quad -450000)
\end{aligned}
$$

where the length unit is the basic length unit (BLU).

The surface defined in part fame {P} has the following parametric equation:

$$\vec{r}(u,v) = \vec{v}^T D^T P D \vec{u}$$

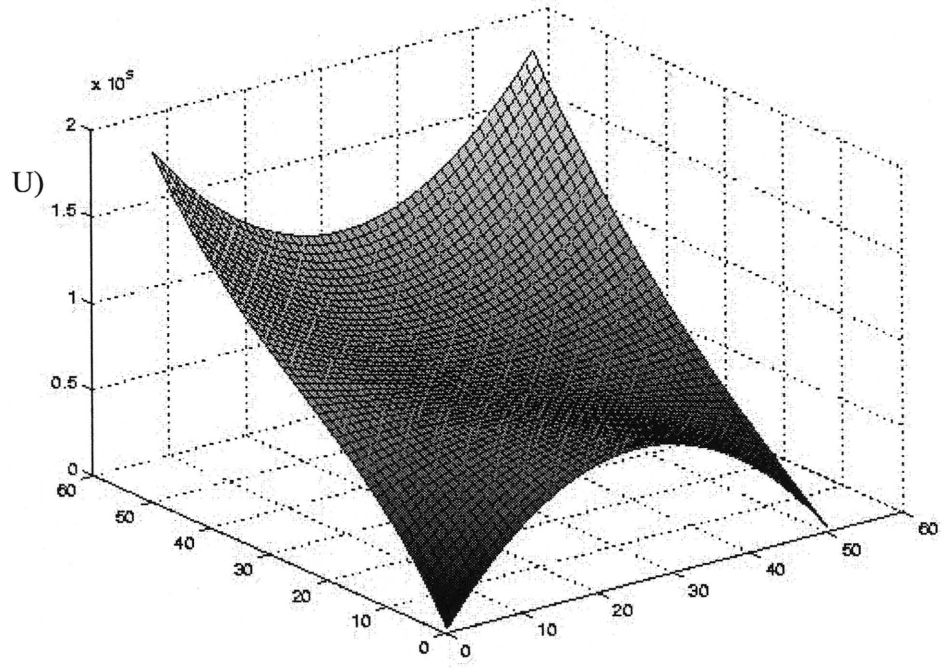
where $\vec{u} = \left(u^3, u^2, u^1, 1\right)^T$, $\vec{v} = \left(v^3, v^2, v^1, 1\right)^T$, D is the coefficient matrix and P is the control point matrix. If the u and v knot vectors are $[0,0,0,0,1,1,1,1]$, the D matrix will have the value of

$$D = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The P matrix is composed of the control points of the surface and has the following structure:

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}$$

Figure 6.8 shows the surface defined with the following sixteen control points:



P_{00}= [0,0,0], P_{01}= [1,0,1], P_{02}= [2,0,1], P_{03}= [3,0,0],

P_{10}= [0,1,1], P_{11}= [1,1,1], P_{12}= [2,1,0.5], P_{13}= [3,1,0.5],

P_{20}= [0,2,1], P_{21}= [1,2,0.5], P_{22}= [2,2,0.5], P_{23}= [3,2,1],

P_{30}= [0,3,2], P_{31}= [1,3,2], P_{32}= [2,3,1], P_{33}= [3,3,2].

Their dimensions are in inches (2.54 cm).

Suppose a ball-end cutter is used and the relation between frame {M} and cutter frame {C} is

$$ {}^{C}\vec{t}_{M} = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} \text{ (in inch)}, \quad {}^{C}R_{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} $$

By discretizing the surface with parameters $u_j = j/49$ $(j = 0,\cdots,49)$ and $v_k = k/49$ $(k = 0,\cdots 49)$, we generate 2500 CC points $\vec{r}(u_j, v_k)$ $(j = 0,\cdots,49; k = 0,\cdots,49)$. Assume the radius of the cutter is 0.125 inch. For convenience, a fixed tool orientation mode is used with both the inclination angle λ and the tilt angle ω fixed at zero (nonzero values should be used in actual machining). Thus each cutter center position $C(x, y, z)$ is a 0.125 inch offset of the CC data along the surface normal and the cutter axis orientation $\vec{n}(\alpha, \beta)$ is the surface normal. This gives us a set of 2500 5-DOF CL data of $CL^{*}_{j,k}(x, y, z, \alpha, \beta)$.

Part Placement Effects

Add $\gamma = 0^{\circ}$ to the 5-DOF CL data. Now, the performance index across the surface for different part locations can be evaluated. For a required index w_a of 10^{13} (see Section 6.6.3 for the determination of w_a), when the part is placed at the following location,

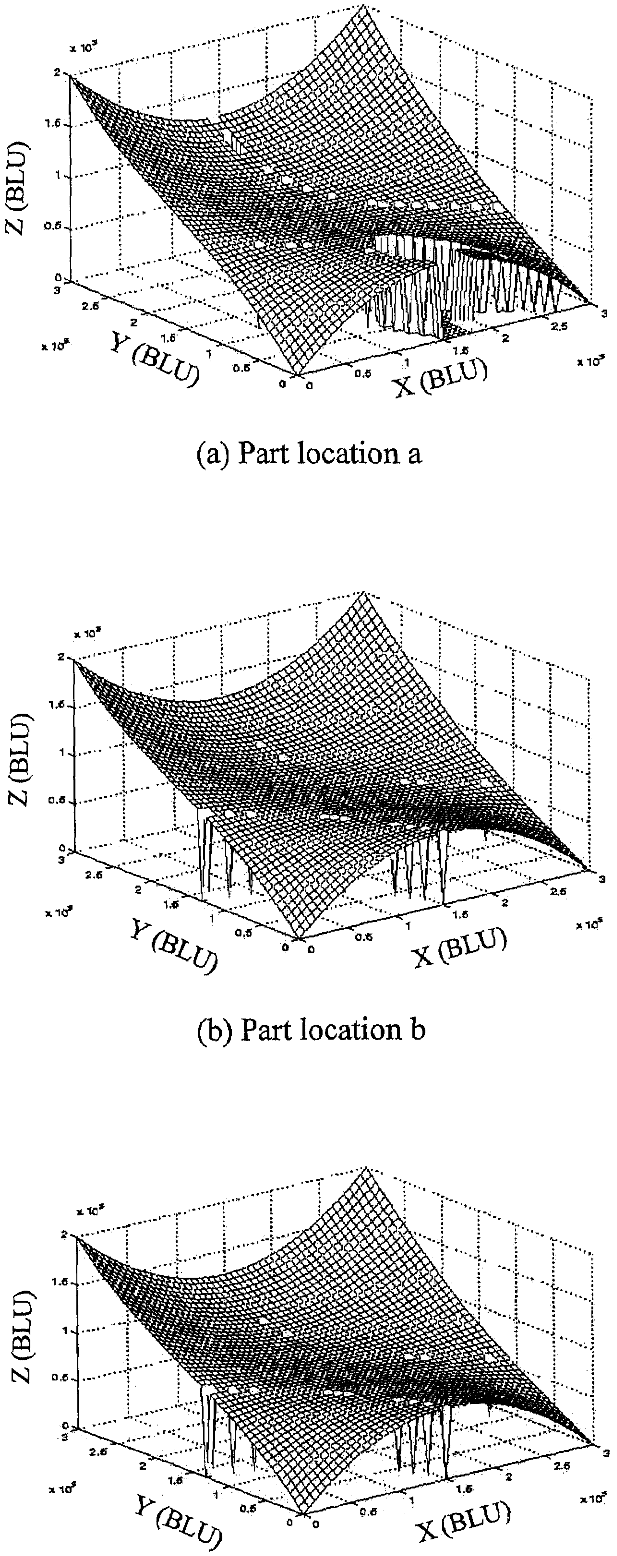
$$\text{Location (a):} \quad {}^{B}\vec{t}_{P} = \begin{bmatrix} -2.5 \\ -1.5 \\ 23 \end{bmatrix} \text{(inch)}, \quad {}^{B}R_{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & 0.5 \\ 0 & -0.5 & 0.866 \end{bmatrix}$$

the index is shown in Figure 6.9(a). The regions where the surface breaks mean that the performance index is below the requirement. From the figure, there is a significant percentage of the surface area where the index can not meet the requirement. If we change part frame {P} (only the orientation) to

$$\text{Location (b):} \quad {}^{B}\vec{t}_{P} = \begin{bmatrix} -2.5 \\ -1.5 \\ 23 \end{bmatrix} \text{(inch)}, \quad {}^{B}R_{P} = \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0.25 & 0.866 & -0.433 \\ -0.433 & 0.5 & 0.75 \end{bmatrix}$$

and

$$\text{Location (c):} \quad {}^{B}\vec{t}_{P} = \begin{bmatrix} -2.5 \\ -1.5 \\ 23 \end{bmatrix} \text{(inch)}, \quad {}^{B}R_{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



(a) Part location a

(b) Part location b

(c) Part location c

Figure 6.9 Part placement effects

Then the area that does not meet the specified index requirement decreases sharply as shown in Figures 6.9(b) and 6.9(c). From these three pictures, one can see that the area with satisfactory index is very sensitive to part locations. This means that a carefully planned part location usually has better machining performance than an arbitrarily chosen part location.

γ Angle Effects

The part is now placed at the following location:

$$ {}^{B}\vec{t}_{p} = \begin{bmatrix} -1 \\ -1.3 \\ 23 \end{bmatrix} \text{(inch)}, \quad {}^{B}R_{p} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

Let us first fix the γ angle to 0°. The performance index for each of the cutter poses is plotted across the surface's x-y projection plane as shown in Figure 6.10. We can see that the performance index varies dramatically across the surface and there are two regions

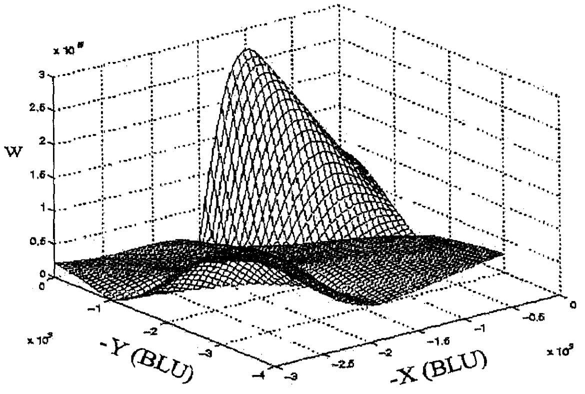


Figure 6.10 Performance index for γ=0°

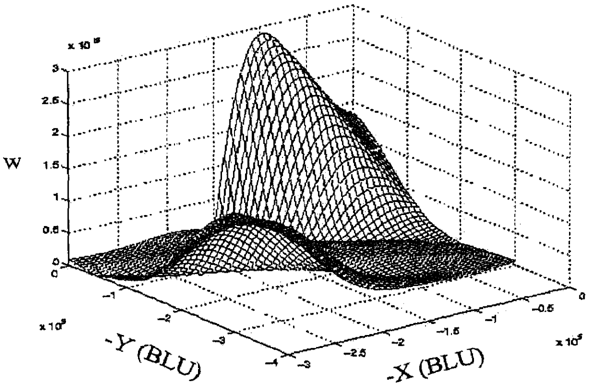


Figure 6.11 Performance index for γ=-20°

where the indexes are close to zero. The low index in these regions indicate that the machine will have large errors and small Cartesian stiffness when it reaches the regions with γ=0°. Clearly γ=0° should not be used in these regions. For comparison, we record the location of one of the regions. It approximately locates along the surface curve whose x and y coordinates of the two end points are (0, 10^5) and (1.5x10^5, 0). Now let us change γ to -20° and 20°, respectively. Figure 6.11 and Figure 6.12 are the plots of the performance index associated with the new γ values. Now the region mentioned is along the surface curve whose x and y coordinates of the two end points are (0, 0.5x10^5) and (1.x10^5, 0) in figure 6.11 and (0, 1.5x10^5) and (2.x10^5, 0) in Figure 6.12, respectively. Keeping a constant γ of 0° is not good enough for the low index regions in Figure 6.10. We may use γ of 20° or -20° in these regions and switch back to 0° when the machine passes the low index regions. This means that γ angle can be used as a tool to avoid the regions with poor performance indexes.

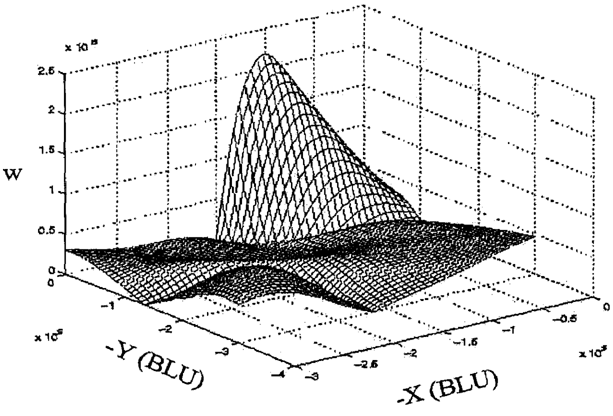


Figure 6.12 Performance index for $\gamma=20^{\circ}$

From the above simulations, it is obvious that part locations globally affect the performance index and the value of γ also affects this index. It is not good to keep the γ value constant in general. Therefore there is a great possibility and potential of better machining performance through planning the γ angle. If we combine part placement planning and e-DOF planning together, it is possible to generate a set of 6-DOF tool paths without low performance indexes.

6.6.2 Simulations of Nonlinear Error

The above example shows that part placement planning and global γ planning can generate the tool paths with a high performance index. In this section, we will show that a high index will benefit the trajectory accuracy. It will also show that local γ planning for the minimum leg length error can be a simple and effective way to reduce the nonlinear error, after finding a feasible part location and γ set $S_{j,k}$ to guarantee a high index.

Between two interpolation poses $\vec{x}_{N-1}$ and $\vec{x}_N(\gamma)$, the relationship between a small Cartesian error, $\delta\vec{x}(\gamma)$, and its corresponding joint error (leg length error), $\delta\vec{l}(\gamma)$, is $\delta\vec{l}(\gamma) = J^{-1}(\gamma)\delta\vec{x}(\gamma)$. Then the Cartesian error can be calculated as

$$\delta\vec{x}(\gamma) = \frac{A(\gamma)}{\left|J^{-1}(\gamma)\right|}\delta\vec{l}(\gamma) = \frac{A(\gamma)}{w}\delta\vec{l}(\gamma)$$

where $A(\gamma)$ is the adjoint matrix of the inverse Jacobian, and w is the index defined in this dissertation. From the above equation, we can use the following two methods to reduce Cartesian error. In the following simulations, we use the machine parameters and setup defined in Section 5.2.1.

Improving Accuracy by Minimizing the Leg length Error

The nonlinear error is basically caused by the leg length error. When the other conditions do not change, a reduced leg length error will result in a smaller nonlinear error. For example, consider the case when machine moves through the following poses (the length unit is BLU):

$$\vec{x}_1\left(200000 \quad 200000 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ\right)$$

$$\vec{x}_2\left(201000 \quad 201000 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ\right),$$

$$\vec{x}_3\left(202000 \quad 202000 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ\right),$$

$$\vec{x}_4\left(202500 \quad 202500 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ\right).$$

For an intermediate pose $(200500 \quad 200500 \quad 2000000 \quad 5^\circ \quad 5^\circ \quad 190^\circ)$, the leg length error can be obtained from the difference between the actual leg length and ideal leg length. Define $\|\vec{\delta l}\|$ as $\sqrt{\delta l_1^2 \quad + \cdots + \quad \delta l_6^2}$. The relationships of $\|\vec{\delta l}\|$ with the sampling length, and then with the nonlinear error are shown in Figures 6.13 and 6.14, respectively. From Figure 6.13, we see that the leg length error will increase with increasing sampling length. As the same Jacobian is used for this simulation, a larger leg length error will result in a larger Cartesian error as shown in Figure 6.14.

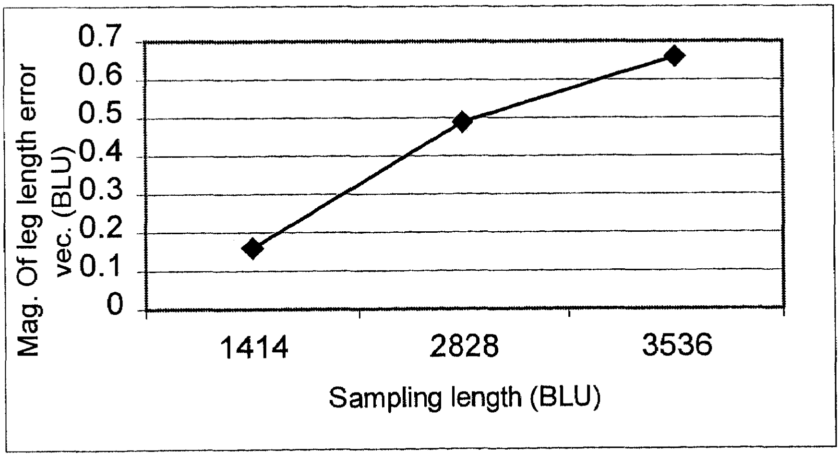


Figure 6.13 Sampling length vs. leg length error

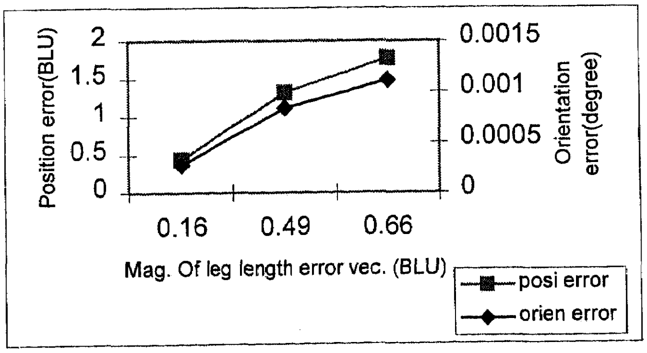


Figure 6.14 Leg length error vs. nonlinear error

Improving Accuracy by Increasing the Performance Index

For the same leg length error, the larger the performance index, the smaller Cartesian error. At $CL^*(200000 \quad 200000 \quad 2000000 \quad 5^\circ \quad 5^\circ)$ (BLU in length), five different values of e-DOF are chosen to form five poses. From these starting poses, keeping the orientation and z-axis unchanged, the machine moves to the second end poses with x=200500 and y=200500, respectively. At their middle point, the indexes and nonlinear errors are shown in Table 6.1.

Table 6.1 Nonlinear errors for different indexes

	$\gamma = 5^\circ$	$\gamma = 35^\circ$	$\gamma = 70^\circ$	$\gamma = 75^\circ$	$\gamma = 78^\circ$
w	9.3×10^{15}	3.96×10^{15}	3.5×10^{14}	1.2×10^{14}	4.1×10^{12}
$\|\delta\vec{l}\|$	1.24	1.13	0.93	0.91	0.89
δr	0.8	0.99	1.6	2.2	30.92
δθ	0.00012°	0.00026°	0.00017°	0.00028°	0.00175°
Length unit: BLU; $\delta r = \sqrt{\delta x^2 + \delta y^2 + \delta z^2}$; $\delta\theta = \sqrt{\delta\alpha^2 + \delta\beta^2 + \delta\gamma^2}$					

According to Figure 6.15, when the value of the e-DOF increases, the index decreases, and the leg length error also decreases. If we have a constant inverse Jacobian matrix, a smaller leg length error will result in a smaller Cartesian error. However, as the index reduces, a smaller leg length error has a larger Cartesian position error as shown in Figure 6.16. This means that large indexes can directly reduce the nonlinear error.

Therefore, planing tool paths to a machine's high index zone is an effective way to improve the machining accuracy.

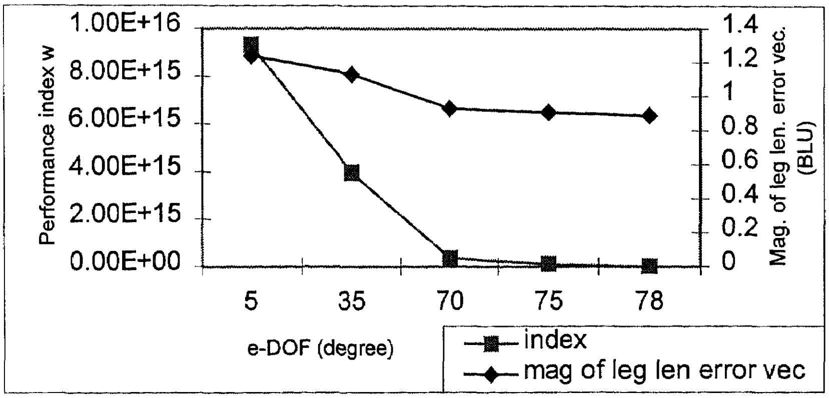


Figure 6.15 E-DOF vs. index & leg length error

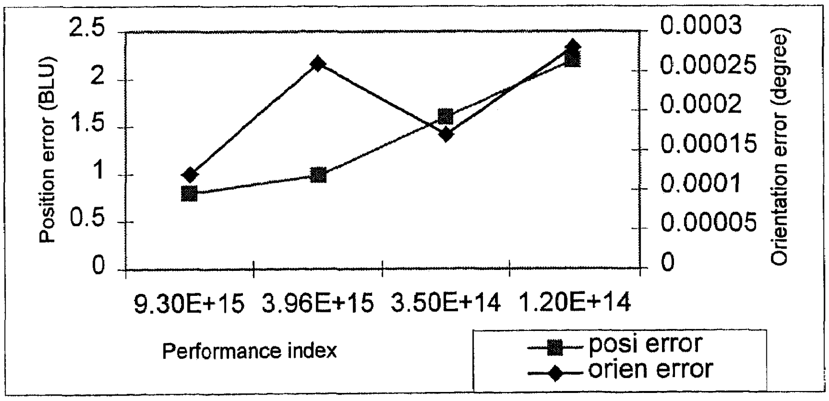


Figure 6.16 The index vs. nonlinear error

In summary, the leg length error is the root of the nonlinear error. For the same inverse Jacobian, smaller leg length errors will result in smaller nonlinear errors. The performance index is the amplifier applied to the leg length error. A larger index will reduce each component of the nonlinear error and hence the overall nonlinear error.

After finding a feasible part location and γ set $S_{j,k}$ to guarantee a high index, a simple and effective way to improve the nonlinear error is to minimize the leg length error in local γ planning.

6.6.3 Illustrative Example I

In this section, we extend the example in Section 6.6.1 by considering all mechanical constraints. The purpose is to demonstrate the overall process of using the kinematics-based tool path planning method.

Finding Feasible Part Locations and Set $S_{j,k}$

Suppose the following mechanical constraints are applied on the machine: each leg has a movement range from 18.5 to 66.5 inches; the extreme half rotation angle of each joint is 90°; and the minimum leg interference distance is 1.5 inches. The determination of the minimum kinematics performance index w_a is based on what level of accuracy in Cartesian space can be afforded by a machine when it has an error in joint space. For a given Cartesian error $\delta\vec{x}$ and joint error $\delta\vec{l}$, w_a can be estimated by calculating the Cartesian error $\delta\vec{x}$ at locations around singular points. For example, if each leg has an error of one BLU, $\delta\vec{l} = (0.5,0.5,0.5,0.5,0.5,0.5)^T$, we calculate the index and Cartesian error caused by this joint error at 21 locations around singular points within the workspace. Figure 6.17 shows the relationship between the Cartesian positional error ($\delta r = \sqrt{\delta x^2 + \delta y^2 + \delta z^2}$) and the index based on this calculation. It is clear that with the increment of the index, the positional error will decrease. If one BLU of positional error

δr is allowed under this joint error, the allowable minimum index w_a estimated from Figure 6.17 is 10^{13}. After preparing all the needed parameters, the first step of planning is to find a feasible location of the part to meet all these constraints. Let us try a location expressed as

$$^{B}\vec{t}_P = \begin{bmatrix} -2.5 \\ -1.5 \\ 23 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & 0.5 \\ 0 & -0.5 & 0.866 \end{bmatrix}$$

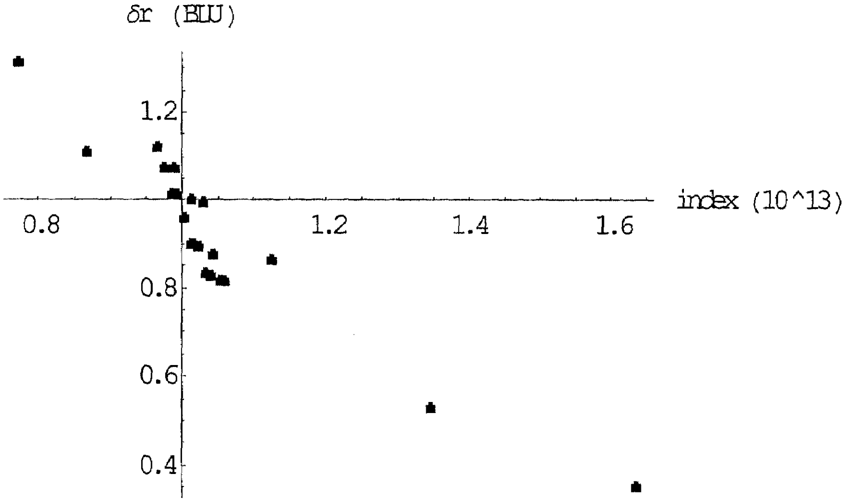


Figure 6.17 The relationship between Cartesian positional error and the index

Figure 6.18 shows the region of the surface satisfying these mechanical and performance constraints. The missing part of the surface indicates that the surface can not be fully accessed by the cutter from this location or it is not acceptable in its performance index. Therefore this location is not a feasible location. Let us locate the part to

$$ {}^{B}\vec{t}_{P} = \begin{bmatrix} -1.5 \\ -1.5 \\ 23 \end{bmatrix} \text{(inch)}, \quad {}^{B}R_{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

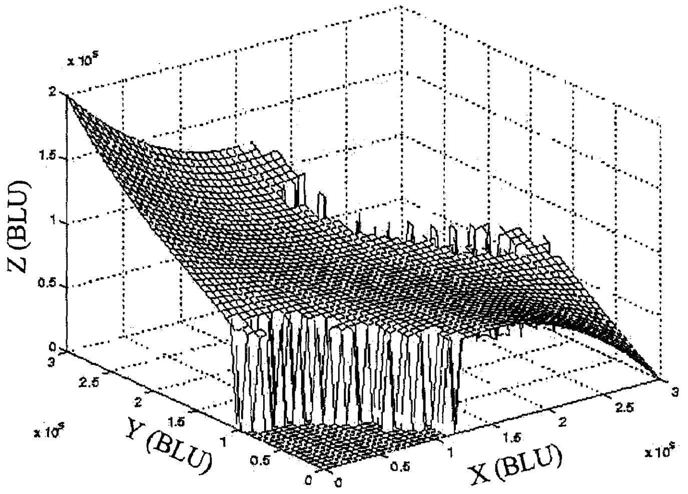


Figure 6.18 Feasible region

Figure 6.19 shows the feasible points of γ for the isoparametric path 0 defined by $\vec{x}_{j,k}$ $(j = 0; k = 0,\cdots,49)$ at this part location. For each pose of the path, if a value of γ is feasible, we use a black dot to relate the pose and the γ. It is clear that there is at least one feasible value of γ for each cutter frame pose. Figures for the other 49 paths are not presented here. When we check these 49 paths, each of their poses also has at least one feasible γ angle. Therefore this part location is feasible. All the γ values belong to the feasible γ set for this pose. Together with the feasible part location, they are the output of the planning in the part placement stage.

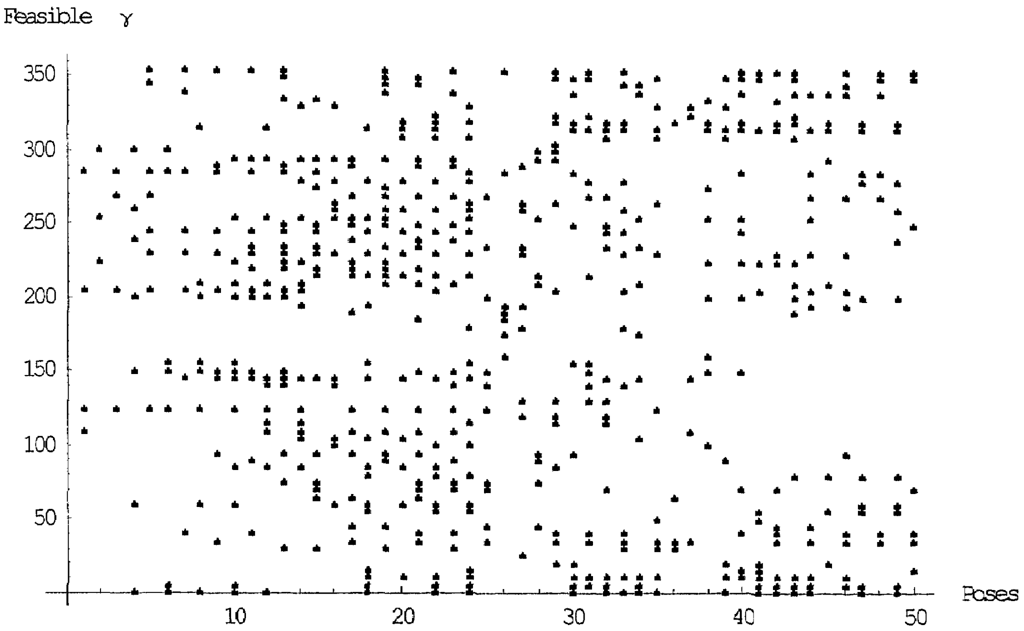


Figure 6.19 Feasible γ angle set $S_{j,k}$ (γ in degree)

Global γ-Path

The information provided by the feasible γ set $S_{j,k}$ is the basis for planning the final γ-path. In the global γ-path planning stage, our final γ for each pose is chosen from the set with the maximum index. After calculating the performance index associated with each of feasible γ in the set $S_{j,k}$, the maximum index for each pose of the surface is presented in Figure 6.20. By examining values of the index, we find that they are well above the chosen allowable index value of 10^{13}. This means when machining this surface the machine avoids the possible singular points by large distances. Figure 6.21 shows the corresponding γ angle with the maximum performance index. Each pose of the tool paths is found with its x and y coordinates, and the third axis is the value of γ for the pose. Together with the 5-DOF tool paths, they form the 6-DOF tool paths for machining the surface. More clearly determined γ values can be seen from the 2D γ-path plot for each path. One of them (path 0) is given in Figure 6.22.

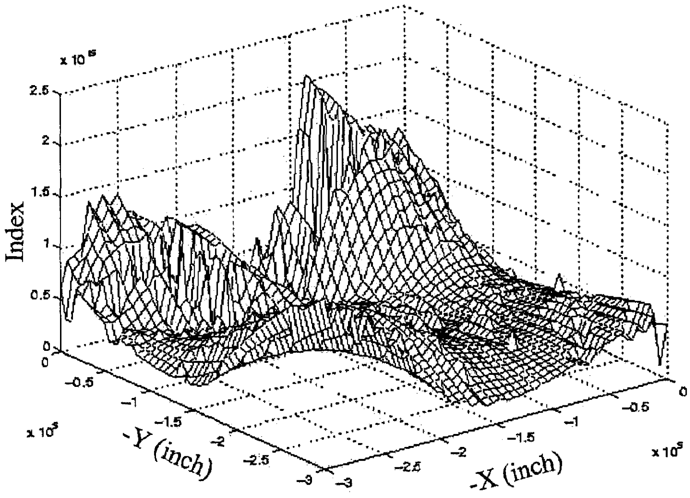


Figure 6.20 The maximum index for illustrative example I

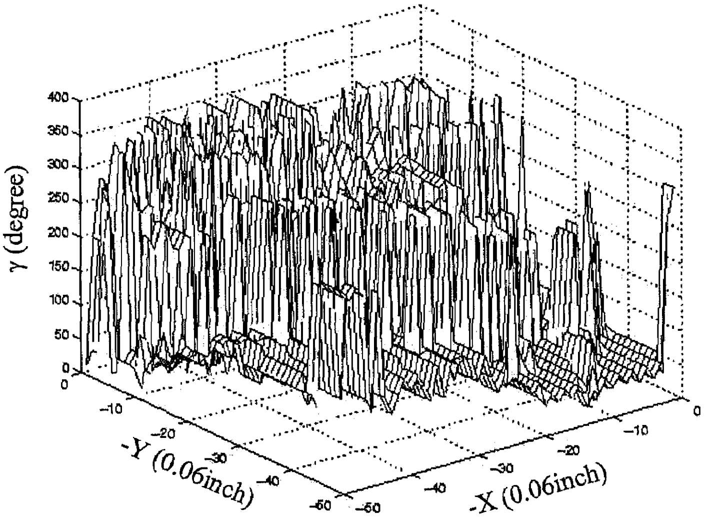


Figure 6.21 γ with the maximum index

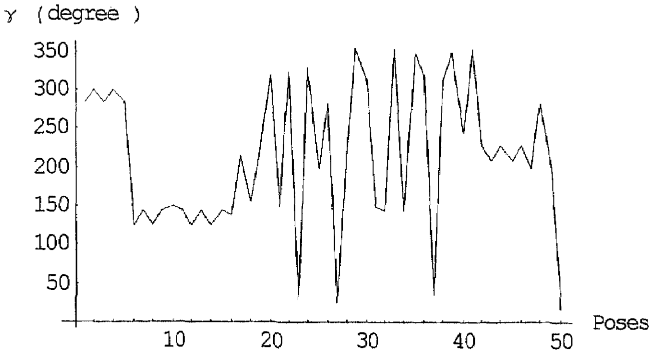


Figure 6.22 γ with the maximum index for path 0

Local γ-Path

For a straight-line segment consisting of two poses of $\vec{x}_{8,0}(48980,0,40983,219.468°,41.101°,150°)$ and $\vec{x}_{9,0}(55102,0,44981,218.641°,39.007°,150.75°)$, (BLU for x, y, z), suppose the sampling length L_s is 2437BLU for a given moving feedrate V. We need three increments formed by two intermediate sampling poses to machine this segment.

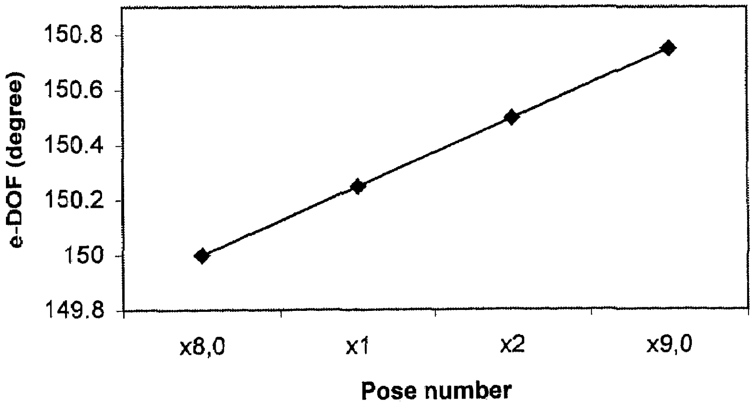


Figure 6.23 Local γ-path by linear interpolation

Table 6.2 Actual middle poses and nonlinear errors by method 1

	x	y	Z	α	β	γ	Δr	Δθ
$X_{m,0}$	50021.3	-20.8	41532.8	219.42°	40.748°	150.125°	120.3	0.004°
$X_{m,1}$	52043.1	-6.6	42872.5	219.055°	40.05°	150.375°	109.8	0.004°
$x_{m,2}$	54083.7	-6.1	44209.1	219.779°	39.352°	150.625°	105.6	0.004°
x, y, z: BLU; $\delta r = \sqrt{\delta x^2 + \delta y^2 + \delta z^2}$; $\delta\theta = \sqrt{\delta\alpha^2 + \delta\beta^2 + \delta\gamma^2}$								

Method 1: Planning γ-Path by Linear Interpolation

Figure 6.23 shows the local γ-path obtained by linear interpolation. Along the distance of the two end poses, γ changes at a constant rate. The two intermediate interpolation poses are

$$\vec{x}_1(51021,0,42316,219.192^\circ,40.403^\circ,150.25^\circ)$$

$$\vec{x}_2(53061,0,43648,218.917^\circ,39.705^\circ,150.5^\circ)$$

Calculate the leg lengths of each pose using inverse kinematics. The intermediate leg lengths are obtained by linear interpolation. By using a numerical method to compute the real poses corresponding to the middle joint locations, the actual middle poses and their errors compared to the nominal ones are presented in Table 6.2 and Figure 6.24. Suppose the sampling frequency of this machine is 500Hz. The sampling length of 2437BLU gives a feedrate of 37m/min. This feedrate is quite possible in practical use as the maximum feedrate of a hexapod machine tool ranges from 13m/min to 100m/min

[Tonshoff, 1999]. However, at this feedrate, the positional nonlinear error reaches 30um in the middle point of the two interpolation poses. This scale of error is really an important source to machines' overall accuracy. Actions must be taken to either increase the sampling rate of the system or decrease the feedrate if no better methods appear.

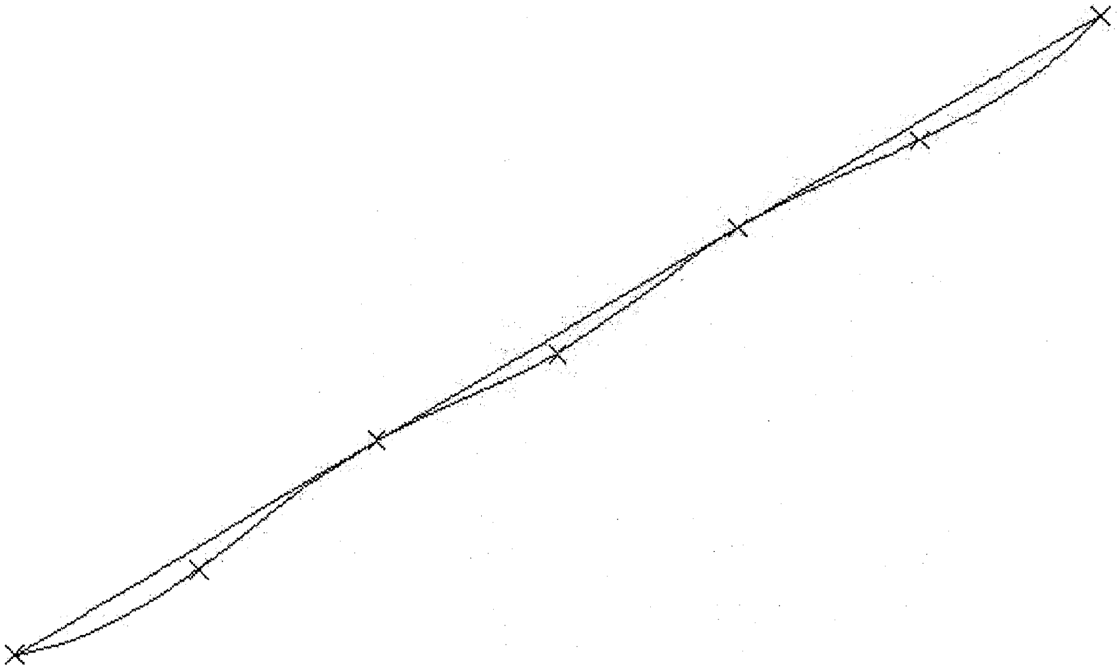


Figure 6.24 Actual trajectory between one straight line segment

Method 2: Planning γ by Minimizing the Total Leg Length Error

The γ-path determined by method 2 usually result in a trajectory with better accuracy. Suppose the γ ranges of poses $\vec{x}_1$ and $\vec{x}_2$ are 0.2° and 0.125°, respectively, that is

$$150.05^\circ \le \gamma_1 \le 150.45^\circ \text{ and } 150.375^\circ \le \gamma_2 \le 150.625^\circ$$

For pose 1, the relationship between the total leg length error and γ through the range are shown in Figure 6.25. At γ=150.258° (2.6225rad), the total leg length error reaches its minimum of 115.2BLU. Therefore, 150.258° is chosen as the value of γ at this pose.

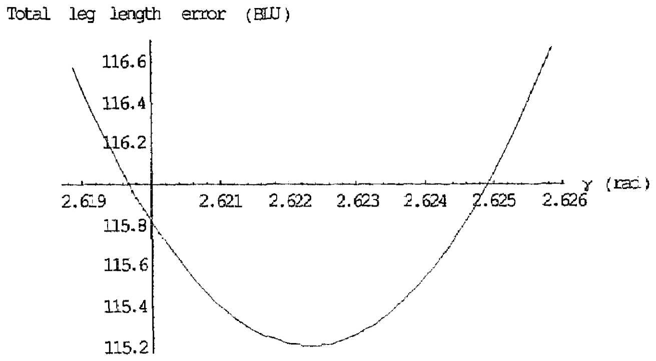


Figure 6.25 Total leg length error vs. γ for pose 1

Table 6.3 Actual middle poses and nonlinear errors by method 2

	X	Y	Z	α	β	γ	Δr	Δθ
$x_{m,0}$	50001.9	-6.7	41535.2	219.33°	40.748°	150.129°	114.5	0.004°
$x_{m,1}$	52044.9	-8.1	42872.1	219.055°	40.05°	150.373°	109.5	0.004°
$x_{m,2}$	54082.5	-5.2	44208.8	219.779°	39.352°	150.619°	105.8	0.004°
x, y, z: BLU; $\delta r = \sqrt{\delta x^2 + \delta y^2 + \delta z^2}$; $\delta\theta = \sqrt{\delta\alpha^2 + \delta\beta^2 + \delta\gamma^2}$								

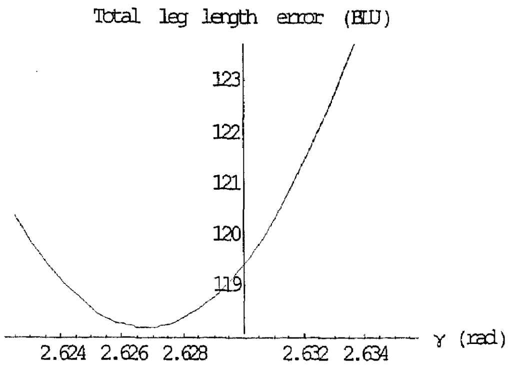


Figure 6.26 Total leg length error vs. γ for pose 2

Figure 6.26 shows the variation of the total leg length error with γ for pose 2. At γ=150.487° (2.6265rad), the total leg length error has its minimum of 118.2BLU. We plan γ =150.487° for pose 2. The γ-path planned by this method is shown in Figure 6.27. The values of γ are only slightly different from those in method 1 (0.008° for pose 1 and 0.003° for pose 2). If we use them to replace those from linear interpolation, the actual middle poses and the errors compared to the nominal ones are listed in Table 6.3. When we compare the results with those from linear interpolation, the position error at the middle of $\vec{x}_{8,0}$ and $\vec{x}_1$ has a significant reduction, and the error at the middle of $\vec{x}_1$ and $\vec{x}_2$ also reduces, as shown in Figure 6.28. This means that the total leg length error is effective to control the nonlinear error, and a smaller total leg length error usually has higher trajectory accuracy. As the last γ has already been determined at the global planning stage, this method has no capability to affect the accuracy of the last small segment. In order to avoid a large error in the last small segment, the γ range $\Delta\gamma$ should be reduced with the interpolation point approaching the last pose. In this example, as γ_2

is close to its linear interpolation location, the accuracy of the last small segment is almost the same as that from the linear interpolation.

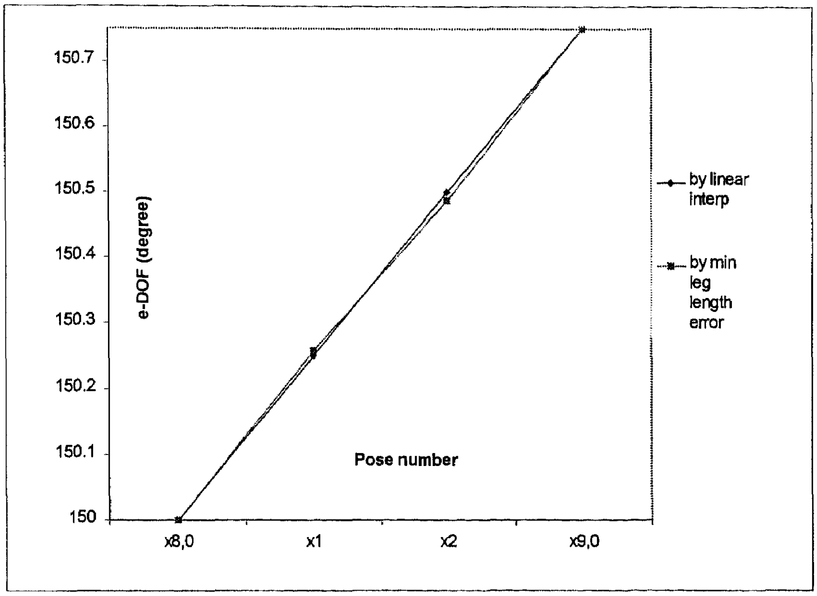


Figure 6.27 γ-path planned by minimizing total leg length error

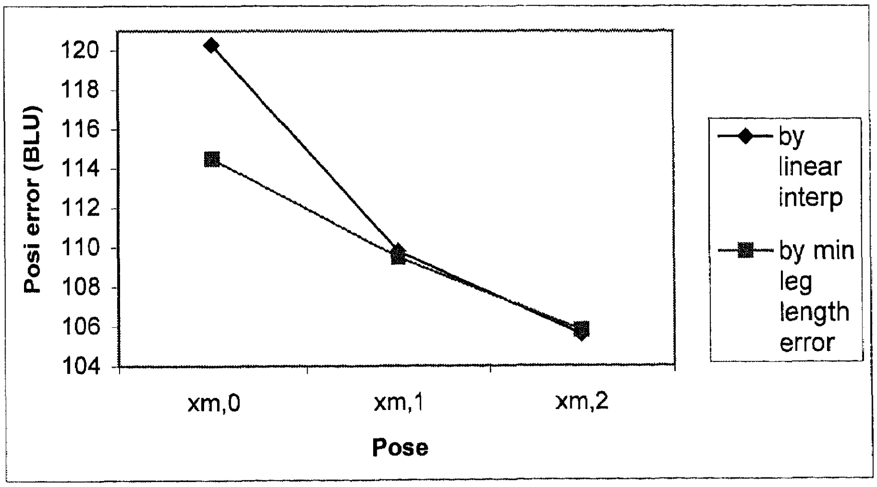


Figure 6.28 Position error comparison

Method 3: Planning γ-path by Minimizing the Maximum Leg Displacement

If the linear interpolation method is used, the maximum leg displacement from pose $\vec{x}_{8,0}$ to $\vec{x}_1$ is 17818.9BLU , which occurs at leg 3. In order to reduce the velocity of this leg, a γ of 150.22° is used for pose $\vec{x}_1$. The displacement of this leg then decreases to 17679.9BLU. Further reduction of this displacement is impossible, as its difference with the second largest leg displacement is very small. Similarly, γ_2 of pose $\vec{x}_2$ is planned as 150.475°. It is shown in Figure 6.29 that the range of displacement by this method is much smaller than that from the linear interpolation. This means that the maximum velocity and acceleration among the six leg axes are reduced, and therefore the motor motion performance is improved.

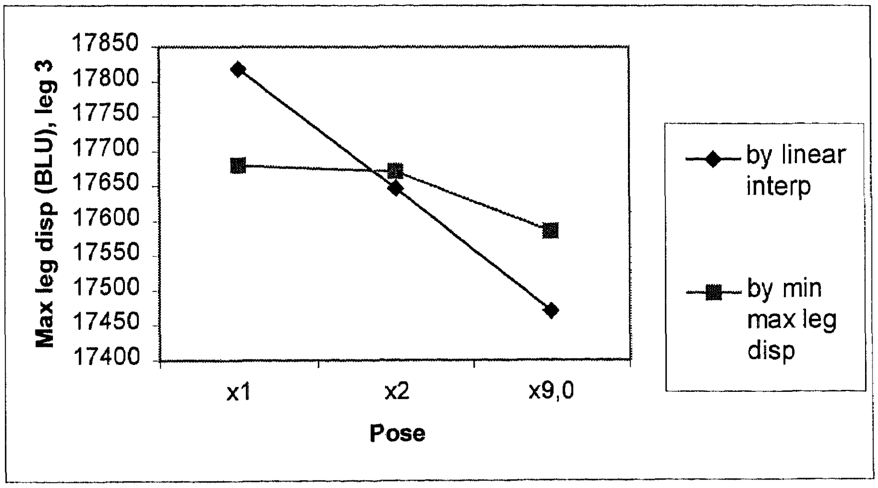
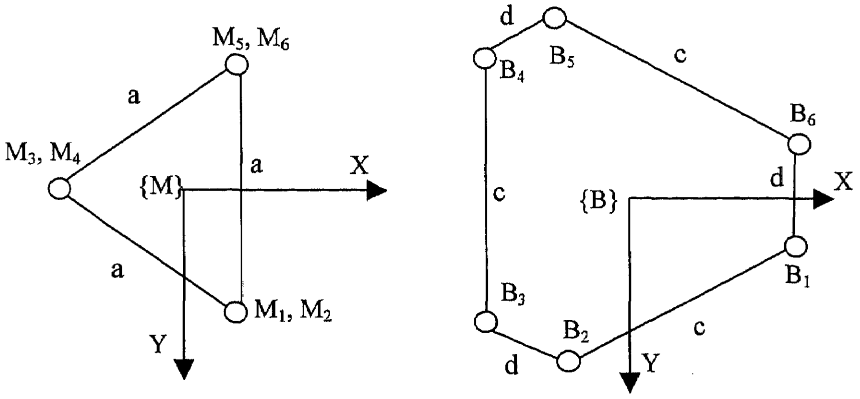


Figure 6.29 Velocity difference of the two methods

6.6.4 Illustrative Example II

The machine selected in this example is a Hexel hexapod machine tool with considering all its mechanical constraints and the performance index constraint. Figure 6.30 shows the Hexel hexapod geometry of the base and mobile plates. The positions of joint B$_i$ in base frame {B} and joint M$_i$ in {M} are:



a=300mm, c=572mm, d=244mm

Figure 6.30 Hexel hexapod geometry of the base and mobile plates [Sapio, 1998]

$$
\begin{aligned}
{}^{B}B_1 &= (1577560 \quad -480314 \quad 8000000) \\
{}^{B}B_2 &= (-372834 \quad -1606300 \quad 800000) \\
{}^{B}B_3 &= (-1204720 \quad -1125980 \quad 800000) \\
{}^{B}B_4 &= (-1204720 \quad 1125980 \quad 800000) \\
{}^{B}B_5 &= (-372834 \quad 1606300 \quad 800000) \\
{}^{B}B_6 &= (1577560 \quad 480314 \quad 800000)
\end{aligned}
$$

$$
\begin{aligned}
{}^{M}M_1 &= (340944 \quad 590550 \quad 0)\\
{}^{M}M_2 &= (340944 \quad 590550 \quad 0)\\
{}^{M}M_3 &= (-681888 \quad 0 \quad 0)\\
{}^{M}M_4 &= (-681888 \quad 0 \quad 0)\\
{}^{M}M_5 &= (340944 \quad 590550 \quad 0)\\
{}^{M}M_6 &= (340944 \quad -590550 \quad 0)
\end{aligned}
$$

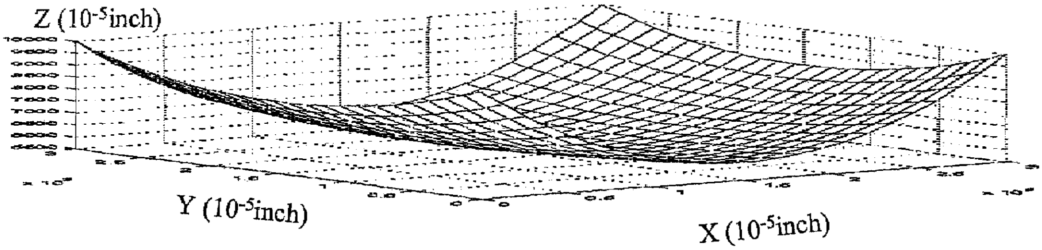


Figure 6.31 The surface for illustative example II

where the length unit is 10^{-5}inch. The NURBS surface defined in part fame {P} for this example has changed to one defined with the following sixteen control points:

P_{00}= [0, 0, 0.1], P_{01}= [1, 0, 0.05], P_{02}= [2, 0, 0.05], P_{03}= [3, 0, 0.1],

P_{10}= [0, 1, 0.08], P_{11}= [1, 1, 0.05], P_{12}= [2, 1, 0.05], P_{13}= [3, 1, 0.08],

P_{20}= [0, 2, 0.08], P_{21}= [1, 2, 0.05], P_{22}= [2, 2, 0.05], P_{23}= [3, 2, 0.08],

P_{30}= [0, 3, 0.1], P_{31}= [1, 3, 0.05], P_{32}= [2, 3, 0.05], P_{33}= [3, 3, 0.01].

Their dimensions are in inches (2.54 cm). It is a concave surface with a large radii of curvature as shown in Figure 6.31.

The relation between frame {M} and cutter frame {C} for the Hexel hexapod is

$$^{C}\vec{t}_{M} = \begin{bmatrix} 0 \\ 0 \\ 12 \end{bmatrix} \text{(inch)}, \quad ^{C}R_{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

A ball end cutter with a radius of 0.25 inch is chosen. By discretizing the surface with parameters $u_j = j/23$ $(j = 0,\cdots,23)$ and $v_k = k/23$ $(k = 0,\cdots 23)$, we generate 576 CC points $\vec{r}(u_j, v_k)$ $(j = 0,\cdots,23; k = 0,\cdots,23)$. This gives us a set of 576 5-DOF CL data of $CL^{*}_{j,k}(x, y, z, \alpha, \beta)$. All the other parameters are the same as those of Illustrative Example I.

The following mechanical constraints are applied on the machine: each leg has a movement range from 11.8 to 39.3 inches; the extreme half rotation angles of base joints and mobile plate joints are respectively 70° and 80°; and the minimum leg interference distance is 3 inches. The determination of the minimum kinematics performance index w_a is based on what level of accuracy in Cartesian space can be afforded by a machine when it has an error in joint space. If each leg has an error of, $\delta\vec{l} = (0.5,0.5,0.5,0.5,0.5,0.5)^{T}$ (unit in 10^{-5}inch), we calculate the index and Cartesian error caused by this joint error at the locations around singular points within the workspace. Figure 6.32 shows the results about the Cartesian positional error ($\delta r = \sqrt{\delta x^2 + \delta y^2 + \delta z^2}$) and the index. With an increment of the index, the positional error decreases. Once the user has an error requirement, the corresponding index can be found using this figure. Similar relationship exists between the orientation and the index.

In some applications, both the position and orientation may be considered in the determination of the allowable index. In this example, if 10^{-4}inch of positional error δr is allowed under this joint error, the allowable minimum index w_a estimated from Figure 6.32 is 10^{14}.

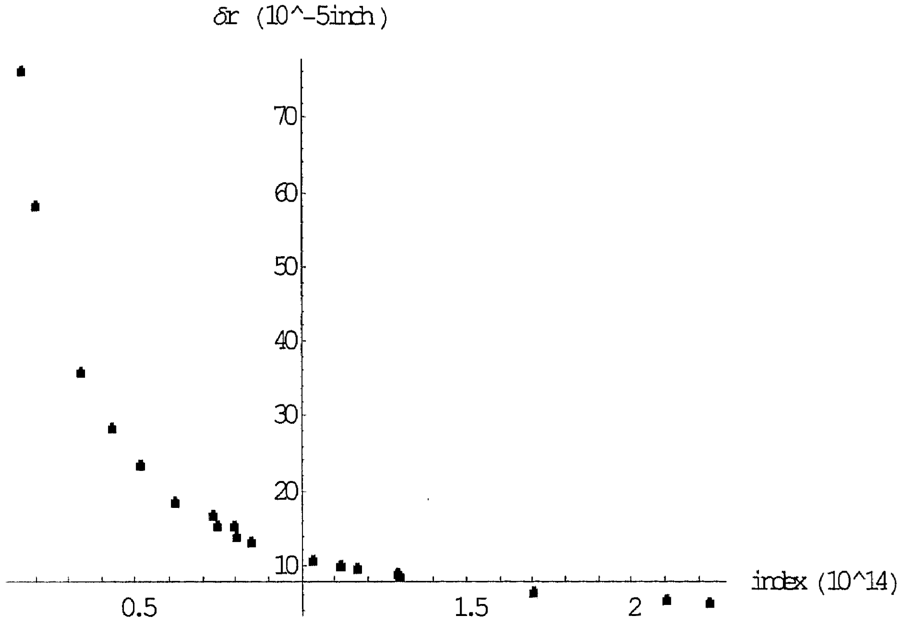


Figure 6.32 The relation of the position error and index for Hexel hexapod

The first step of planning is to find a feasible location of the part to satisfy all these constraints. Six locations of the part around the center of the workspace are selected for this placement study. Their locations are

$$(1)\ {}^{B}\vec{t}_P = \begin{bmatrix} 5 \\ -1.5 \\ 48.5 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0.25 & 0.866 & -0.433 \\ -0.433 & 0.5 & 0.75 \end{bmatrix}$$

$$(2)\ {}^{B}\vec{t}_P = \begin{bmatrix} 1 \\ -1.5 \\ 48.5 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0.25 & 0.866 & -0.433 \\ -0.433 & 0.5 & 0.75 \end{bmatrix}$$

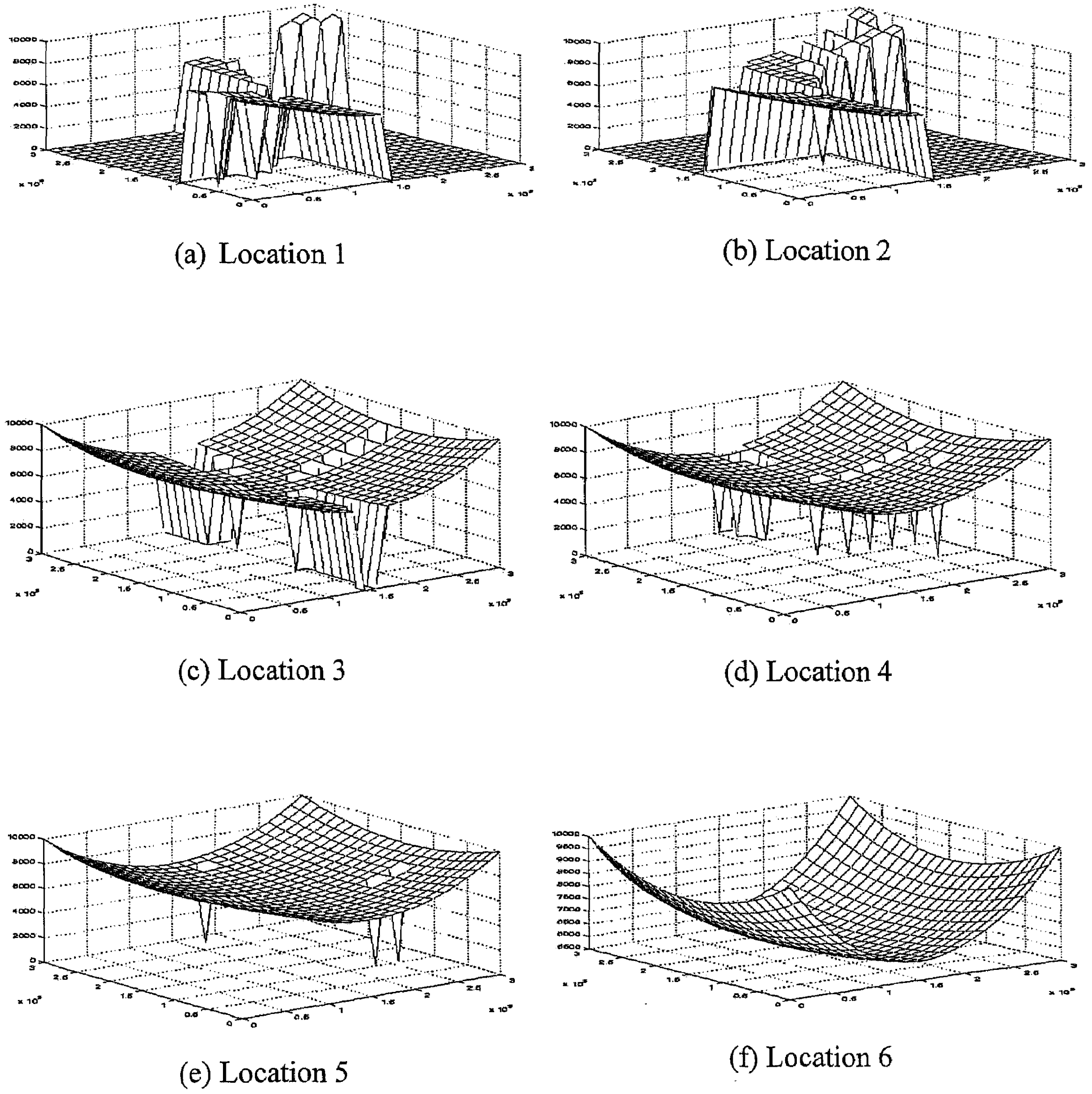
$$(3)\ {}^{B}\vec{t}_P = \begin{bmatrix} -1.5 \\ -1.5 \\ 48.5 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & -0.5 \\ 0 & 0.5 & 0.866 \end{bmatrix}$$

$$(4)\ {}^{B}\vec{t}_P = \begin{bmatrix} -2 \\ -1.5 \\ 48.5 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(5)\ {}^{B}\vec{t}_P = \begin{bmatrix} -1.5 \\ -1.5 \\ 46.5 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(6)\ {}^{B}\vec{t}_P = \begin{bmatrix} -1.5 \\ -1.5 \\ 45.5 \end{bmatrix} \text{(inch)},\ {}^{B}R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 6.33 shows the surface regions where all the mechanical and performance index constraints are satisfy by the machine at these six locations. In the figure, the regions of the surface remaining at their original positions means that the associated tool paths are feasible, while an area with unfeasible tool paths is removed from the surface and has spikes touching the horizontal plane. The figure shows that part locations have a great effect on the feasible area of the surface. At locations 1 and 2, where the part frames have an orientation different from the base frame, only a small area of the surface is feasible. For the same orientation (locations 4, 5, & 6), most of the surface is feasible. Among these six locations, only location 6 has an unbroken surface. This means that the tool paths placed at this location are totally accessible by the cutter with the required



(a) Location 1

(b) Location 2

(c) Location 3

(d) Location 4

(e) Location 5

(f) Location 6

Figure 6.33 Feasible surface regions of six part locations

performance index. When we check the data file containing the feasible γ angle for the 24 paths at location 6, each pose of them also has at least one feasible γ angle. Therefore this part location is feasible. The feasible γ sets for one of the isoparametric paths (path 0 defined by $\vec{x}_{j,k}$ $(j = 0; k = 0,\cdots,23)$) are shown in Figure 6.34. A black point in the figure indicates that γ with the value is feasible for that pose. All the points corresponding to one pose are the elements of the γ set for that pose. Figures for the other 23 paths are not presented here.

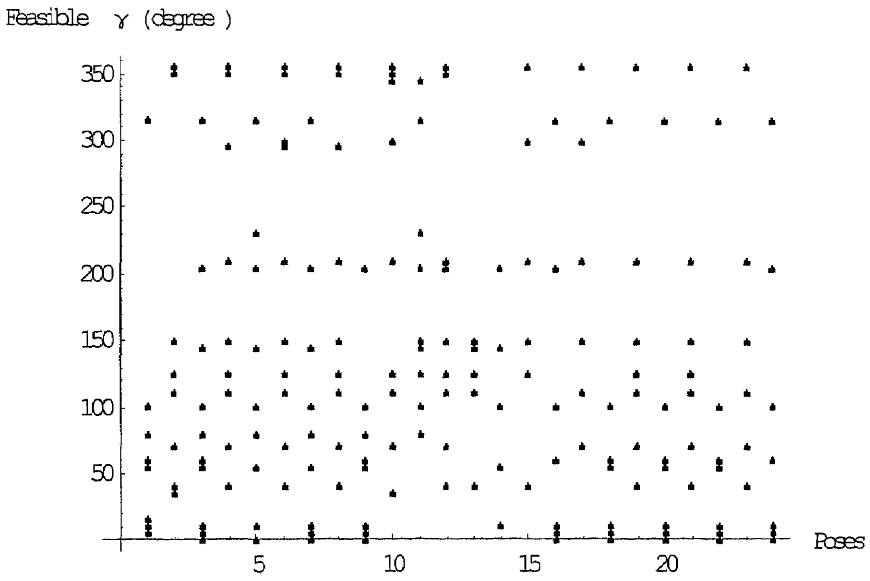
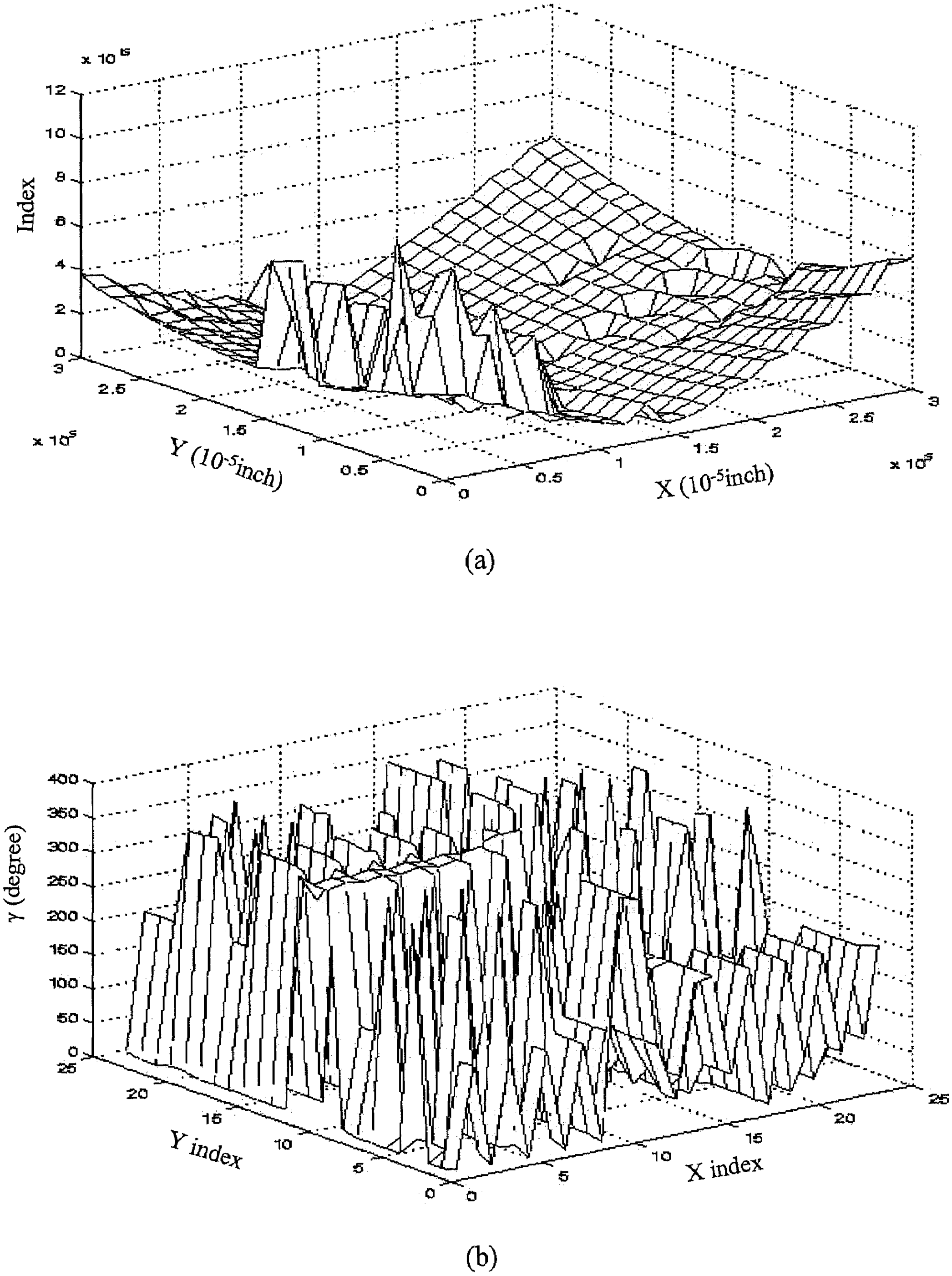


Figure 6.34 Feasible γ sets for path 0 of illustative example II

The feasible γ set $S_{j,k}$ is the basis for planning the final γ-path. For each pose, calculate the performance index associated with each of the feasible γ in set $S_{j,k}$, then take the γ with the maximum index as the final γ. Figure 6.35(a) shows the maximum



(a)

(b)

Figure 6.35 Illustrative example II, (a) The maximum index, (b) Corresponding γ values.

index for the surface. The values of the index are well above the chosen allowable index value of 10^{14}. This means that when machining this surface the machine avoids the possible singular points by large distances. Figure 6.35(b) is the corresponding γ angle with the maximum performance index. Each pose of the tool paths is referred to by its x and y index values, and shown in the third axis is the value of γ for the pose. Together with the 5-DOF tool paths, they form the 6-DOF tool paths for machining the surface.

6.7 Summary

A kinematics-based tool path planning method for hexapod machine tools was proposed to guide the part placement and the determination of the extra degree of freedom. The algorithms to evaluate feasible part locations and find γ set $S_{j,k}$ were described. Through simulations, it was shown that the part placement locations and extra degree-of-freedom have great effects on the kinematic performance index. It was also shown that maintaining a constant value for the extra degree-of-freedom may not be a good approach in planning the machining motion. The illustrative examples demonstrated the feasibility of the method and its effectiveness in avoiding inaccessible and low-accuracy tool locations. The feasible part locations and γ set $S_{j,k}$ obtained from this method can provide the key information for performing practical machining operations.

Three local γ planning methods were described and proved to be feasible and effective. In real applications, a combination of them could be applied. Among the three, the linear interpolation method is the basic one. When the nonlinear error becomes unacceptable, γ should be determined by minimizing the total leg length error. Moreover, if the velocity and acceleration are too high, the method of changing γ to improve the motor motion performance could be used. In addition, as the nonlinear error is computationally expensive to evaluate in real time, the total leg length error was proposed to measure its effect. It has been shown that this measure is very effective in controlling the nonlinear error.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the major contributions and results of this thesis research. It also discusses the future work needed for continuation of this study.

7.1 Major Contributions

The Foot-Placement Space (FPS) based on the desired workspace of a hexapod and the range of motion of its leg modules is defined and a method of constructing the FPS is presented. An implementation algorithm for determining individual Foot-Placement Spaces (FPSs) is developed. The results provide information on the feasible locations for the feet so that the required workspace can be realized.

The equations and an algorithm for identifying the true position and orientation of the base joints of hexapods are developed for newly installed hexapods. For hexapods whose axes of the base joints are precisely defined, the minimum number of configurations for identifying the position is two. The general identification will require at least three configurations of the hexapods. The identification equations can be solved through a numerical approach or by Dialytic Elimination using symbolic manipulation for some special situations. The identified parameters can improve the accuracy of kinematic models of a hexapod at the new location.

Three major dissimilarities of tool path planning between hexapods and five-axis machines are studied. The first significant difference is the existence of an extra degree of freedom (e-DOF) in a hexapod. If properly used, the extra degree of freedom can improve the geometric and kinematic conditions of the tool paths. Second, a hexapod has a complex workspace and a varying inverse Jacobian within the workspace. Faster, stiffer and more accurate operations can not be realized if tasks are not well planned. The third dissimilarity is that a hexapod usually generates a nonlinear segment in joint-interpolated motion when following a straight-line segment between two consecutive poses. This reduces the accuracy when the sampling length must be large in high speed machining. All these factors indicate that traditional path planning methods should not be used for hexapods without modification.

A kinematics-based tool path planning method for hexapod machine tools is proposed to guide the part placement and the use of the e-DOF. Methods of searching for feasible part locations and γ sets are presented. Through the simulations, it is demonstrated that methods are feasible and effective in enhancing the performance of the tool paths. Moreover, it is shown that maintaining a constant value for the e-DOF may not be a good approach in planning the machining motion. The feasible part locations and γ sets obtained are valuable information for machining operations. Three local γ planning methods are discussed and shown to be feasible and effective. Since the assessment of nonlinear error is computationally expensive in real time, the total leg length error is proposed as an indirect measure. It is shown that this measure is effective in controlling the nonlinear error.

7.2 Future Work

The method for determining individual foot-placement spaces developed in this research can only be used to obtain feasible locations for placing the feet of the legs. It is desirable to develop methods to obtain optimum locations for given operation conditions.

The Dialytic Elimination method can only be used to solve the position identification equations. The accuracy and efficiency of identification will be improved if it can be extended to the general identification.

The described search method for a feasible part location is a passive process. Active or predictive part placement methods are desired, as they will increase the searching efficiency. A more complete implementation of the method, especially with computer graphics based simulations, is needed to show its full capabilities.

Although the kinematics-based planning has been shown to be beneficial to improvement of stiffness, a stiffness specific planning module should be investigated and integrated for e-DOF planning.

APPENDIX A

INVERSE JACOBIAN MATRIX ENTRIES FOR HEXAPODS

The inverse Jacobian matrix is directly derived from the leg length expressions in Equation (5.13) by taking partial derivatives with respect to x, y, z, α, β, and γ individually. Appendix A presents the six entries in the first row of the inverse Jacobian. The following is a list of the parameters in the expressions,

$x, y, z, \alpha, \beta, \gamma$:	Six coordinates of cutter frame {C} in part frame {P}
xP, yP, zP:	Position coordinates of part frame origin in frame {B}
xB1, yB1, zB1:	Position coordinates of base joint B1 in frame {B}
xM1, yM1, zM1:	Position coordinates of mobile joint M1 in frame {C}

Any entry in the other rows has the same expression as the one in the first row with the same column number except for xB1, yB1, zB1, xM1, yM1, and zM1. For these parameters, the number '1' is replaced by a corresponding joint number, for example, in the entries of the second row, xB1, yB1, zB1, xM1, yM1, and zM1 respectively change to xB2, yB2, zB2, xM2, yM2, and zM2.

$${}^{\partial l_1}\!/_{\partial x} =$$

```
-(-x + xB1 - xp - zM1 Cos[α] Sin[β] -
    yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
    xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) /
  (√ ((-z + zB1 - zp - zM1 Cos[β] +
          xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
      (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
          yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
          xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
      (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
          xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
          yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
       2))
```

$${}^{\partial l_1}\!/_{\partial y} =$$

```
-(-y + yB1 - yp - zM1 Sin[α] Sin[β] -
    xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
    yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) /
  (√ ((-z + zB1 - zp - zM1 Cos[β] +
          xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
      (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
          yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
          xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
      (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
          xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
          yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
       2))
```

$$\partial l_1 / \partial z =$$

```
-(-z + zB1 - zp -
    zM1 Cos[β] + xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) /
  (√ ((-z + zB1 - zp - zM1 Cos[β] +
          xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
      (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
          yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
          xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
      (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
          xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
          yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
       2))
```

$$\partial l_1 / \partial \alpha =$$

```
(2 (-zM1 Cos[α] Sin[β] -
        yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
        xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ]))
     (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
        xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
        yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) +
   2 (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
        yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
        xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ]))
     (zM1 Sin[α] Sin[β] -
        xM1 (-Cos[β] Cos[γ] Sin[α] - Cos[α] Sin[γ]) -
        yM1 (-Cos[α] Cos[γ] + Cos[β] Sin[α] Sin[γ]))) /
  (2 √ ((-z + zB1 - zp - zM1 Cos[β] +
           xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
       (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
           yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
           xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
       (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
           xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
           yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
        2))
```

$$\partial l_1 / \partial\beta =$$

```
(2 (xM1 Cos[β] Cos[γ] + zM1 Sin[β] - yM1 Cos[β] Sin[γ]) (-z + zB1 -
        zp - zM1 Cos[β] + xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) +
    2 (-zM1 Cos[α] Cos[β] +
        xM1 Cos[α] Cos[γ] Sin[β] - yM1 Cos[α] Sin[β] Sin[γ])
      (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
        yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
        xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) +
    2 (-zM1 Cos[β] Sin[α] +
        xM1 Cos[γ] Sin[α] Sin[β] - yM1 Sin[α] Sin[β] Sin[γ])
      (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
        xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
        yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ]))) /
 (2 √ ((-z + zB1 - zp - zM1 Cos[β] +
            xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
         (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
            yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
            xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
         (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
            xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
            yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
          2))
```

$\partial l_1 / \partial\gamma =$

```
(2 (-yM1 Cos[γ] Sin[β] - xM1 Sin[β] Sin[γ]) (-z + zB1 - zp -
        zM1 Cos[β] + xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) +
    2 (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
        yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
        xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ]))
      (-xM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
        yM1 (-Cos[α] Cos[β] Cos[γ] + Sin[α] Sin[γ])) +
    2 (-yM1 (-Cos[β] Cos[γ] Sin[α] - Cos[α] Sin[γ]) -
        xM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ]))
      (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
        xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
        yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ]))) /
 (2 √ ((-z + zB1 - zp - zM1 Cos[β] +
            xM1 Cos[γ] Sin[β] - yM1 Sin[β] Sin[γ]) ^2 +
        (-x + xB1 - xp - zM1 Cos[α] Sin[β] -
            yM1 (-Cos[γ] Sin[α] - Cos[α] Cos[β] Sin[γ]) -
            xM1 (Cos[α] Cos[β] Cos[γ] - Sin[α] Sin[γ])) ^2 +
        (-y + yB1 - yp - zM1 Sin[α] Sin[β] -
            xM1 (Cos[β] Cos[γ] Sin[α] + Cos[α] Sin[γ]) -
            yM1 (Cos[α] Cos[γ] - Cos[β] Sin[α] Sin[γ])) ^
          2))
```

REFERENCES

Aronson, R. B., 1997, "Hexapods: Hot or Ho Hum," *Manufacturing Engineering*, October, pp. 60-67.

Behi, F, 1988, "Kinematic analysis of a six-degree-of freedom 3-PRPS parallel mechanism," *IEEE Journal of Robotics and Automation,* Vol. 4, No. 5, pp. 561-565.

Cheok, K. C., Overholt, J. L., and Beck, R., 1993, "Exact Methods for Determining the Kinematics of a Stewart Platform Using Additional Displacement Sensors," *Journal of Robotic Systems*, 10(5), pp. 689-708.

Cohen, R., Lipton, M. G., Dai, M. Q., and Benhabib, B., 1992, "Conceptual Design of a Modular Robot," *ASME Journal of Mechanical Design*, Vol. 114, pp. 117-125.

Fichter, E. F., 1986, "A Stewart Platform-Based Manipulator: General Theory and Practical Construction," *International Journal of Robotics Research*, Vol. 5, No. 2, pp. 157-182.

Gosselin, C., and Angeles, J., 1988, "Optimum Kinematic Design of a Planar Three-Degree-Of-Freedom Parallel Manipulator," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 110, No. 1, pp. 35-41.

Gosselin, C., Lavoie, E., and Toutant, P,. 1992, "An Efficient Algorithm for the Graphical Representation of the Three-Dimensional Workspace of Parallel Manipulators," *Robotics, Spatial Mechanisms, and Mechanical Systems*, ASME DE-Vol. 45, pp. 323-328.

Gough, V. E., and Whitehall, S. G., 1956-57, "Universal Tyre Testing Machine," *Proceedings, 9th International Technical Congress FISIA*, vol. 117, pp. 117-135.

Huang, T., Whitehouse, D. J. and Wang, J. S., 1998, "Local Dexterity, Optimal Architecture and Design Criteria of Parallel Machine Tools," *CIRP Annuals*, 47(1), pp. 347-351.

Huang, Y., and J. Oliver, 1995, "Integrated simulation, error assessment, and tool path correcton for five-axis NC milling," *J. of Manufacturing Systems*, Vol. 14, No. 5, pp. 331-344.

Hunt, K. H., 1978, *Kinematic Geometry of Mechanisms*, Oxford University Press, Oxford.

Hunt, K. H., 1983, "Structural Kinematics of In-Parallel-Actuated Robot-Arms," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, pp. 705-712.

REFERENCES (Continued)

Innocenti, C., and Parenti-Castelli, V., 1990, "Direct position analysis of the Stewart platform mechanism," *Mech. Mach. Theory*, Vol. 25, No. 6, pp. 611-621.

Ji, Z., 1994a, "Workspace Analysis of Stewart Platforms Via Vertex Space," *Journal of Robotic Systems*, Vol. 11, No. 7, pp. 631-639.

Ji, Z., 1994b, "Dynamic Decomposition for Stewart Platforms," *ASME Journal of Mechanical Design*, Vol. 116, No. 1, pp. 67-69.

Ji, Z., 1995, "Placement Analysis for a Class of Platform Manipulators," *Proceedings, ASME 1995 Design Engineering Conferences*, Vol. 1 (DE-Vol. 82), Boston, MA, pp. 773-779.

Ji, Z., 1996, "Analysis of Design Parameters in Platform Manipulators," *ASME Journal of Mechanical Design*, Vol. 118, No. 4, pp. 526-531.

Ji, Z., 1997, "A roundoff method for improving position accuracy of parallel manipulators," *8th International Conference on Advanced Robotics*, California, U.S.A., July 7-9, pp.557-562.

Ji, Z., and Leu, M., 1999, "Design, reconfiguration and control of parallel kinematic machines," *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements* (Advanced Manufacturing Series), (ed.) C. R. Boer, L. Molinari-Tosatti, K. S. Smith, Springer-Verlag, London, October 1999, pp.111-129.

Ji, Z., and Song, P., 1998, "Design of a Reconfigurable Platform Manipulator," *Journal of Robotic Systems*, Vol. 15, No. 6, 1998, pp. 343-346.

Ji, Z., and Li, Z., 1998, "Determination of Individual Foot-Placement Space for Modular Platform Manipulators," *Proceedings of 1998 ASME Design Engineering Technical Conferences,* Mech-5966 (CD-ROM), Atlanta, GA, September 13-16, 1998.

Koren Y., "Control of Machine Tools," *Journal of Manufacturing Science and Engineering,* Vol. 119, Nov. 1997, pp. 749-755.

Lee, H.-Y., and Roth, B., 1993, "A closed-form solution of the forward displacement analysis of a class of in-parallel mechanisms," *Proceedings of the 1993 IEEE Robotics and Automation Conference*," Vol. 1, pp. 720-724.

Lee, Y., 1998, "Non-isoparametric tool path planning by machining strip evaluation for 5-axis sculptured surface machining," *Computer-Aided Design,* Vol. 30, No. 7, pp. 559-570.

REFERENCES
(Continued)

Lee, Y., and Chang, T., 1996, "Machined surface error analysis for 5-axis machining," *Int. J. of Prod. Res.*, Vol. 34, No. 1, pp. 111-135.

Li, S, and R. Jerard, 1993, "5-axis machining of sculptured surfaces with a flat-end cutter," *Computer-Aided Design*, Vol. 26, No. 3, March, pp. 165-178.

Lin, R., and Koren, Y., 1996, "Efficient tool-path planning for machining free-form surfaces," *J. of Engineering for Industry, Transactions of ASME*, Vol.118, Feb., pp. 20-28.

Ma, O. and Angeles, J., 1991, "Architecture Singularities of Parallel kinematic machines," *Proceedings, IEEE International Conference on Robotics and Automation*, Sacramento, CA., pp. 1542-1547.

Matsumaru, T., 1995, "Design and Control of the Modular Robot System: TOMMS," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2125-2131.

Merlet, J.-P., 1989, "Singular Configurations of Parallel Manipulators and Grassmann Geometry," *International Journal of Robotics Research*, Vol. 8, No. 5, pp. 45-56.

Merlet, J.-P., 1993, "Closed-form resolution of the direct kinematics of Parallel Manipulators using Extra sensor Data," *Proceedings IEEE International Conference on Robotics and Automation*, Atlanta, GA, pp. 200-204.

Merlet, J.-P., 1994, "Parallel manipulators: state of the art and perspectives," *Advanced Robotics*, Vol. 8, No. 6, pp. 589-596.

Merlet, J.-P., 1995a, "Determination of the Orientation Workspace of Parallel Manipulators," *Journal of Intelligent and Robotic Systems*, Vol. 13, pp. 143-160.

Merlet, J.-P., 1995b, "Designing a Parallel Robot for a Specific Workspace," *Research Report 2527*, INRIA.

Mooring, B. W., Roth, Z. S. and Driels, M., 1991, *Fundamentals of Manipulator Calibration*, John Wiley & Sons, New York.

Moriwaki T., 1999, "Survey of R&D activities related to parallel mechanism in Japan," *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements* (Advanced Manufacturing Series), (ed.) C. R. Boer, L. Molinari-Tosatti, K. S. Smith, Springer-Verlag, October 1999, London, pp.431-440.

REFERENCES
(Continued)

Nakamura, Y., 1991, *Advanced Robotics Redundancy and Optimization*, Addison-Wesley Publishing Company, New York.

Negri, S., and et al., 1999, "Kinematic analysis of parallel manipulators," *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements* (Advanced Manufacturing Series), (ed.) C. R. Boer, L. Molinari-Tosatti, K. S. Smith, Springer-Verlag, October 1999, London, pp.69-84.

Paredis, C. J. J., Brown, H. B., and Khosla, P. K., 1996, "A Rapidly Deployable Manipulator System," *Proceedings, IEEE International Conference on Robotics and Automation*, Minneaplolis, Minnesota, pp. 1434-1439.

Parenti-Castelli, V. and Gregorio, R., 1995, "Determination of the actual configuration of the general Stewart platform using only one additional displacement sensor," *Proceedings of the 1995 ASME International Mechanical Engineering Congress & Exposition*, Nov. 12-17, San Francisco, CA.

Raghavan, M., 1993, "The Stewart Platform of General Geometry Has 40 Configurations," *Journal of Mechanical Design*, Vol. 115, No. 2, pp. 277-282.

Raghavan, M., and Roth, B., 1995, "Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators," *Special 50th Anniversary Design Issue, Transactions of the ASME,* Vol. 117, June, pp.71-79.

Salisbury, J. K., and Craig, J. J., "Articulated Hands: Force Control and Kinematic Issues," *International Journal of Robotics Research*, Vol. 1, No. 1, 1982, pp. 4-17.

Sapio, V., 1998, "Some approaches for modeling and analysis of a parallel mechanism with Stewart platform architecture," *Proceedings of the ASME Manufacturing Science and Engineering Division*, MED-Vol. 8, pp. 637-649.

Sarma, S., and Slocum, A., 1999, "On the use and augmentation of hexapod machine tools," *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements* (Advanced Manufacturing Series), (ed.) C. R. Boer, L. Molinari-Tosatti, K. S. Smith, Springer-Verlag, October 1999,London, pp. 257-268.

Schmitz, D, Khosla, P. K., and Kanade, T., 1998, "The CMU Reconfigurable Modular Manipulator Systems," *Technical Report*, Carnegie Mellon University, CMU-RI-TR-88-7, Pittsburgh, PA..

Stewart, D., 1965-66, "A Platform with Six Degrees of Freedom," *Proc. Instn. Mech. Engrs*, Vol. 180, Pt. 1, No.15, pp. 371-386.

REFERENCES (Continued)

Tindale, J., 1965-66, Discussion on the Stewart Paper, *Proc. Instn. Mech. Engrs*, Vol. 180, Pt. 1, No.15, pp. 383-384.

Tonshoff, K., 1999, "A systematic comparison of parallel kinematics," *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements* (Advanced Manufacturing Series), (ed.) C. R. Boer, L. Molinari-Tosatti, K. S. Smith, Springer-Verlag, October 1999, London, pp. 295-312.

Wang, J. and Masory, O., 1993, "On the accuracy of a Stewart platform - part I: the effect of manufacturing tolerances," *Proceedings of the 1993 IEEE Robotics and Automation Conference*, Vol. 1, May, pp. 114-120.

Wurst, K. H., 1986, "The Conception and Construction of a Modular Robot System," *Proc. International Symposium on Industrial Robots*, pp. 37-44.

Yang, D. C. H. and Lee, T. W., 1984, "Feasibility Study of a Platform Type of Robotic Manipulators from a Kinematic Viewpoint," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 106, pp. 191-198.

Yoshikawa, T., "Manipulability of Robotic Mechanisms," *International Journal of Robotics Research*, Vol. 4, No. 2, 1985, pp. 3-9.

Zhuang, H. and Roth, Z. S., 1991, "A closed form solution to the kinematic parameter identification of robot manipulators," *Proceedings of the 1991 IEEE Robotics and Automation Conference*," Vol. 3, April, pp. 2682-2688.

Zhuang H. and Roth Z. S., 1993, "Method for kinematic calibration of Stewart platforms," *Journal of Robotic Systems*, 10(3), pp. 391-405.