INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Minimum Error Tool Path Generation Method
and
An Interpolator Design Technique
for Ultra-precision Multi-Axis CNC Machining

Hong Liang

A Thesis

in

The Department

of

Mechanical Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

July 1999

ABSTRACT

Minimum Error Tool Path Generation Method
and
An Interpolator Design Technique
for Ultra-precision Multi-axis CNC Machining

Hong Liang, Ph.D.
Concordia University, 1999

This thesis investigates an ultra-precision multi-axis CNC machining problem encountered in machining sculptured surfaces. Conventional multi-axis CNC machining uses straight line segments to connect consecutive data points, and uses linear interpolation technique to generate the command signals for positions between machining data points. However, due to the multi-axis simultaneous and coupled translational and rotational movements, the actual machining motion trajectory is a non-linear curve. The non-linear curve segments deviate from the linearly interpolated straight line segments, resulting in non-linearity errors, which in turn cause obstacles to ensuring high precision machining.

The problem in multi-axis CNC machining is that non-linearity errors result in total machining error which is beyond the range of the machining tolerance. The problem arises from the fact that the linear interpolation technique generates commands for positions along a straight line segment, while rotational movements superimposed onto translational movements cause the cutting point moving along a curved machining motion trajectory. The machining motion trajectory depends on both multi-axis CNC machine tool configuration and the machining rotational movements. The machining rotational movements are kinematically

related to cutter orientation variations. Thus, The factors causing the multi-axis CNC machining error problem are the spatially varying cutter orientations and the utilization of linear interpolation method .

A novel off-line tool path generation methodology for is developed and reported in this thesis in order to solve the non-linearity error problem in ultra-precision multi-axis CNC machining. The new off-line tool path generation method reduces non-linearity errors by modifying cutter orientation changes based on machine kinematics and machining motion trajectory. A software routine for implementing the new tool path generation methodology is developed. A simulation of the process for machining a sculptured surface by applying the novel methodology illustrates that it increases machining precision considerably.

A novel interpolator design technique for solving the non-linearity error problem in ultra-precision multi-axis CNC machining is also presented. A 3D circular interpolation principle is developed which is capable of tracking spherical curves with low position errors and uniform feedrates. On the basis of this (3D circular) interpolator, a combined 3D linear and circular (L&C) interpolator is proposed for five-axis CNC machining. The proposed 3D L&C interpolator is able to drive the pivot of rotational movements along a predesigned 3D curve and conduct the cutting point along a linear spatial path, so that the elimination of non-linearity errors in five-axis CNC machining is achieved. A software interpolation routine of the 3D L&C interpolator is developed, and a computer simulation illustrating the machining of a sculptured surface validates the novel technique.

ACKNOWLEDGEMENT

I would like to extend the sincerest gratitude and appreciation to my thesis supervisors, Dr. J. Svoboda and Dr. H. Hong, for their time, encouragement, guidance and directive supervision, moral and financial support during the course of this research. I feel extremely fortunate to have had the opportunity of working with them and sharing their experience and insight. I am grateful to my former thesis supervisors, Dr. R. Cheng and Dr. C. Wu, for giving me the opportunity of experiencing the current industry issues.

In addition, I would like to thank Mr. T. Luong and Mr. L. Gagnon, of Pratt & Whitney Canada Inc., for many useful discussions, suggestions and comments I have received from them through this research.

Last but not least I owe many thanks to my family and especially my parents for their encouragements and help at all stages of my studies. I am very grateful to my husband and my daughter for their patience, encouragements and support during this work. Without their unwavering support this would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

B	..... rotation angle about the machine B_m axis
B_{fix}	..... assumed fixed rotational axis
C	..... rotation angle about the machine C_m axis
C_{fix}	..... assumed fixed rotational axis
B_m	..... machine rotational variables
C_m	..... machine rotational variables
B_{move}	..... moving rotational axis
C_{move}	..... moving rotational axis
B_0	..... the start coordinate of the B_m axis rotation movement for each move
C_0	..... the start coordinate of the C_m axis rotation movement for each move
B_1	..... the end coordinate of the B_m axis rotation movement for each move
C_1	..... the end coordinate of the C_m axis rotation movement for each move
D_x	..... difference between the interpolated x-point and the end point of the segment
D_y	..... difference between the interpolated y-point and the end point of the segment
D_z	..... difference between the interpolated z-point and the end point of the segment
P	..... the rotation pivot of the B_m axis and the C_m axis
F	..... fixture length (in chapter 4)
F	..... feedrate (in chapter 6&7)
T	..... interpolation period
G	..... workpiece stacking position
GL	..... tool gage length
L	..... segment length
ΔL	..... interpolation increment
L_x	..... segment length in x-axis direction

L_y ..... segment length in y-axis direction

L_z ..... segment length in z-axis direction

O_x ..... the x-coordinate of the workpiece frame origin w. r. t. the machine coordinate system

O_y ..... the y-coordinate of the workpiece frame origin w. r. t. the machine coordinate system

O_z ..... the z-coordinate of the workpiece frame origin w. r. t. the machine coordinate system

O'_x ..... the rotated x-coordinate of the workpiece frame origin w. r. t. the machine coordinate system

O'_y ..... the rotated y-coordinate of the workpiece frame origin w. r. t. the machine coordinate system

O'_z ..... the rotated z-coordinate of the workpiece frame origin w. r. t. the machine coordinate system

PB ..... position of the B_m axis pivot

PC ..... position of the C_m axis pivot

R ..... rotation transformation matrix

ΔX_i ..... the i-th interpolated increment in x-axis

ΔY_i ..... the i-th interpolated increment in y-axis

ΔZ_i ..... the i-th interpolated increment in z-axis

i ..... direction cosine of cutter orientation angle α

j ..... direction cosine of cutter orientation angle β

k ..... direction cosine of cutter orientation angle γ

k_f ..... surface local curvature

l ..... distance between the cutter contact point and the rotation pivot P

n_x ..... the unit normal vector to the y-z coordinate plane in Cartesian coordinate system

n_y ..... the unit normal vector to the x-z coordinate plane in Cartesian coordinate system

n_z ..... the unit normal vector to the x-y coordinate plane in Cartesian coordinate system

p^0 ..... cutter vector

p^0_h ..... cutter vector on horizontal plane

p^1 ..... spindle vector

p^1_h ..... cutter vector parallel to the spindle vector

x ..... cutter position x-coordinate w. r. t. the workpiece coordinate system

y ..... cutter position y-coordinate w. r. t. the workpiece coordinate system

z ..... cutter position z-coordinate w. r. t. the workpiece coordinate system

x_h ..... cutter position x-coordinate on a horizontal plane

y_h ..... cutter position y-coordinate on a horizontal plane

z_h ..... cutter position z-coordinate on a horizontal plane

x^1 ..... rotated cutter position x-coordinate w. r. t. the "fixed" axes

y^1 ..... rotated cutter position y-coordinate w. r. t. the "fixed" axes

z^1 ..... rotated cutter position z-coordinate w. r. t. the "fixed" axes

x_m ..... machine translational variables

y_m ..... machine translational variables

z_m ..... machine translational variables

x_P ..... x-coordinate of the rotation pivot P w. r. t. the machine coordinate system

y_P ..... y-coordinate of the rotation pivot P w. r. t. the machine coordinate system

z_P ..... z-coordinate of the rotation pivot P w. r. t. the machine coordinate system

x_{cc} ..... x-coordinate of the cutter contact point

y_{cc} ..... y-coordinate of the cutter contact point

z_{cc} ..... z-coordinate of the cutter contact point

x_0 ..... x-coordinate of the start point for each segment

y_0 ..... y-coordinate of the start point for each segment

z_0 ..... z-coordinate of the start point for each segment

x_1 ..... x-coordinate of the end point for each segment

y_1 ..... y-coordinate of the end point for each segment

z_1 ..... z-coordinate of the end point for each segment

x_i ..... the linearly interpolated rotation centre x-coordinate

y_i ..... the linearly interpolated rotation centre y-coordinate

z_i ..... the linearly interpolated rotation centre z-coordinate

α ..... cutter axis orientation angle relative to the x axis of the workpiece coordinate system

β ..... cutter axis orientation angle relative to the y axis of the workpiece coordinate system

γ ..... cutter axis orientation angle relative to the z axis of the workpiece coordinate system

Δs ..... segment length between a pair of adjacent points

$\delta_{a,n}$ ..... allowable non-linearity error

δ_{nl} ..... non-linearity error

δ_{max} ..... maximum non-linearity error

δ_t ..... linearity error

λ_l ..... linear interpolation scale factor

λ_c ..... circular interpolation scale factor

θ ..... the latitude angle

ϕ ..... the longitude angle

τ ..... the rotational movements interpolation parameter

LIST OF ABBREVIATIONS

AIGP = automation intelligence generalization postprocessor

APT = automatically programmed tools

BLU = basic length-unit

CAM = computer-aided manufacturing

CC = cutter contact

CLDATA = cutter location data

CNC = computer(-ized) numerical control

DDA = digital differential analyzer

FRN = feedrate number

ipm = inch per minute

L&C = linear and circular

MCP = machine control point

MCU = machine control unit

NC = numerical control

OM-1 = OMINIMILL series-1

rpm = revolution per minute

2D = two-dimensional

3D = three-dimensional

CHAPTER 1

INTRODUCTION

1.1 CNC Machining, NC Programming and Postprocessing

Computer-aided manufacturing (CAM) is the utilization of computers to assist in the process of manufacturing. CAM includes the on-line and the off-line applications of the digital method. Computer Numerical Control (CNC) machining is the on-line application, which uses a computer with a machine control unit (MCU) to generate commands for controlling the machining process. The off-line application is the utilization of computers in production planning and non-real-time assistance in the manufacturing processes. Examples of off-line CAM are the preparation of NC part programs (referred also as tool paths) or the display of the tool paths in machining simulation.

In CNC machining, the MCU plays a key role in the on-line control of machining. The functions of the MCU include (1) reading and decoding the information from the tool path data and distributing the data among the controlled axes; (2) the tool centre control, automatic tool selection, and the various compensation functions; (3) the feedrate calculations, and the preparatory functions, the spindle motions and the miscellaneous functions control; (4) the interpolation to supply velocity commands between successive data points; (5) control of simultaneous multi-axis movements; (6) on-line diagnostics and troubleshooting; (7) display of machining information on the CRT screen and (8) the communication between the MCU itself and the external devices. The overall design of a

CNC system first requires the selection of the appropriate control techniques (reference-pulse or sample-data) and the optimal setting of the control-loop parameters. Subsequently, the appropriate interpolation routines must be written. The function of interpolation is to generate the successive tool positions, called commands, for each segment of the cutting curve based on the initial and the final machining tool positions and the desired temporal parameters such as feedrates. The interpolation method is the core of the CNC system, since the accuracy of calculated intermediate position directly affects the machining precision of the whole system and the time for computing the intermediate positions directly affects the controlled axis velocity, which in turn, affects the quality of the machined surface and the machining time.

The off-line application of CAM includes NC part programming and postprocessing. NC part programming involves the collection of all data required to produce the part, the calculation of a tool path along which the machine operations will be performed, and the arrangement of the given and calculated data in a standard format which could be converted to an acceptable form for a particular CNC machine. Most CAM systems generate one or more types of neutral language files containing instructions for CNC machining. The Automatically Programmed Tools (APT) language is the most comprehensive and popular system for NC part programming. The APT language enables a programmer to provide the MCU of a CNC machine with geometric descriptions of the workpiece surfaces and to specify the tool movements. The output of the APT system, which is called the cutter location data (CLDATA), defines the tool path with machining conditions (the feedrate, depth of cut and the spindle speed). In order to realize the machining, the neutral instructions must be

transformed to the specific instructions required by a particular machine tool. The postprocessors are the interfacing tools between APT systems and CNC machines.

A postprocessor is a software which is used for translating neutral instructions from the APT system into the specific instructions required by a CNC machine tool. The CLDATA defines the tool path with cutting conditions in the part coordinate system. Each Machine Control Unit/Machine Tool (MCU/MT) configuration, however, has its own machine coordinate system. Therefore, a postprocessor translates the CLDATA in a part coordinate system to the NC-codes in the machine coordinate system. With the APT part programming standard, the postprocessor writes the instructions as a series of commands in a standard format. Each command contains all required data: the preparatory functions codes (G codes) for preparing the MCU to perform a specific mode of operation; the miscellaneous function codes (M codes) which pertain to the auxiliary information; the geometry descriptions of the workpiece dimensions and cutting conditions, such as the feedrate, the spindle speed and the tool words. A postprocessor usually consists of five elements: input, motion, auxiliary, output and control. The main portion of a postprocessor is the motion element which includes the geometric and the dynamic packages. The geometry package performs the coordinate transformation of CLDATA from the part coordinate system into NC-codes in the particular machine coordinate system. It also checks the tool path and makes corrections where necessary to ensure that the tool path is within the specified machining tolerance. In addition, the geometric package prevents the movement instructions to the MCU from exceeding the machine tool axes limits. The dynamic package modifies the feedrates where necessary and

establishes the distances for acceleration and deceleration to prevent overshoots and undershoots.

1.2 Sculptured Surfaces Machining Methods

A surface that can only be represented as the image of a sufficiently regular mapping of a set of points in a domain into a 3D space is called sculptured surface [1]. A sculptured surface can be represented by a set of curves that connect the design points of the surface. Two main approaches are commonly used for obtaining the curved surfaces: the first approach exploits the parametric curves representation, while the second uses contouring planes (frequently, geometrically equally-spaced parallel planes) to intersect the surface for obtaining a curved surface.

In the first approach, straight lines in a parametric domain are used to define the parametric curves P(u,v) on the actual surface in Cartesian space. By setting one parameter, say v, a set of cutting curve functions $P(u, v_i)$ defines the entire surface. The parametric curves approach includes schemes of the isoparametric curves and the variable parametric curves. With isoparametric curves, tool paths are uniformly distributed across the parametric domain. The step-over interval (which is the distance between tool passes, referred to as cutting curves) is the same in the parametric domain. The step-forward distance (which is the segment length along a cutting curve) is the same in the parametric domain and are independent of the surface geometric properties and of the machining tolerances. This approach is simple and generally efficient because the tool contact curves are easy to retrieve

from the surface definitions. However, because the geometric properties of the machined surface are not taken into account, the relationship between the parametric coordinate and its corresponding Cartesian coordinate is not uniform. Therefore, the accuracy and efficiency of the isoparametric surface representation may vary depending on the geometry of the machined surfaces. With variable parametric curves representation, the tool path is generated by using the local geometric properties of the machined surface (the local surface tangents, normals and curvatures). The step-forward distance and the step-over interval determined on the basis of these geometric properties will vary over the machined surface.

Using the contouring planes approach, the resulting numerically derived non-parametric curves are used to drive the milling tools. With this approach, the tool path generation can be carried out by employing either the offset surface method or the direct intersection curves techniques. The offset surface method involves the computation of the offset surface of the machined surface, after which, the intersection curves of the cutting planes with the offset surface will be the path of the cutter centre of a ball-end mill. In this method, the cutter contact (CC) point path is a space curve. In contrast, using the direct intersection curves technique, the CC point moves along the intersection curves of the cutting plane with the machined surface, so that the CC point path is a plane curve on the cutting plane. The cutter centre in this case, in general, moves along a space curve. For unbounded surface machining, both the direct intersection curve technique and the offset surface method can be used to generate tool paths. Usually, the direct intersection curve technique results in a preferable tool path because the CC points are 'restricted' to the cutting plane. However,

when machining bounded surfaces (which are bounded on one or more sides by surfaces) the offset surface technique is easier. In this case, the cutter centre moves along a plane curve, which is formed by connecting the intersection curves of the cutting plane with the offset surface and the bounded surfaces. With the contouring curves representation method, the variable step-forward distance and step-over interval can be determined by using the local surface geometry and the machining tolerance. The advantage of the contouring plane curves method is that the tool path is a plane curve, so that the distributed tool path is relatively uniform. However, the method requires proper selection of cutting planes, in that the spacing and the direction of the cutting plane must be properly determined.

Common methods for machining sculptured surfaces include point milling, end milling and flank milling techniques. Point milling technique is the traditional machining approach, in which ball-nosed end-mills with either the cylindrical shape or the conical shape are used. In point milling, a curved surface is cut by the ball-end of a cutter following a dense set of parametric curves on the mathematically modelled surface interpolating the surface design curves [2, 3]. The historical reasons for using the ball-end mill are that (1) it is easy to position in relation to curved surfaces, (2) ball-end mills generally require simple and short NC machining programs, (3) ball-end mills often only require two-dimensional cutter compensation [4]. In addition, conical shaped ball-end mills are especially suitable for machining long twisted surfaces with narrow slots between the surfaces because conical shaped cutters provide rigidity and prevent tool chatter. The major advantage of using point milling technique is that almost any smooth surface can be point milled. Ball-end mills,

however, cut along an arc that extends from the cutter axis to a point on the spherical profile of the ball-end. During machining, the cutting point on the spherical surface changes, which results in a variation in cutting speed. When the cutting point is at the portion of the sphere near the axis of rotation, the cutting speed is nearly equal to zero which produces a rough surface. Another disadvantage of point milling is that by its nature, it produces scalloped surface finish. Fig. 1.1 shows a schematic of the point milling process [3]. The height of the scalloped ridges is directly related to the ball-end radius and the number of cuts over the surface.

When machining the flat or low curvature surfaces, end mills with square ends are used. The profile of an end mill can be made to closely match that of a curved surface by inclining it correctly to the surface normal. The effective radius of curvature of the profile of an end mill can vary from infinity to the cutter radius, as the inclination of the cutter to the surface normal changes from zero to 90 degrees [4]. Hence, compared to the ball-end point milling, a better geometric match can be achieved by using end mills. End-milling also produces scalloped surfaces, but the scallop height on the machined surface can be reduced by properly inclining the cutter. Another important factor with end milling is that the material is always cut at the periphery of the cutter at a full and predetermined cutting speed. It is easy to ensure a constant feedrate and thus to obtain better surface finish. However, end mills pose tremendous difficulty in the calculation of collision-free cutter orientations for complicated machining. While the end milling technique is suitable for machining of large low-curvature surfaces, it is not the best choice for machining long and twisted or high-curvature surfaces.

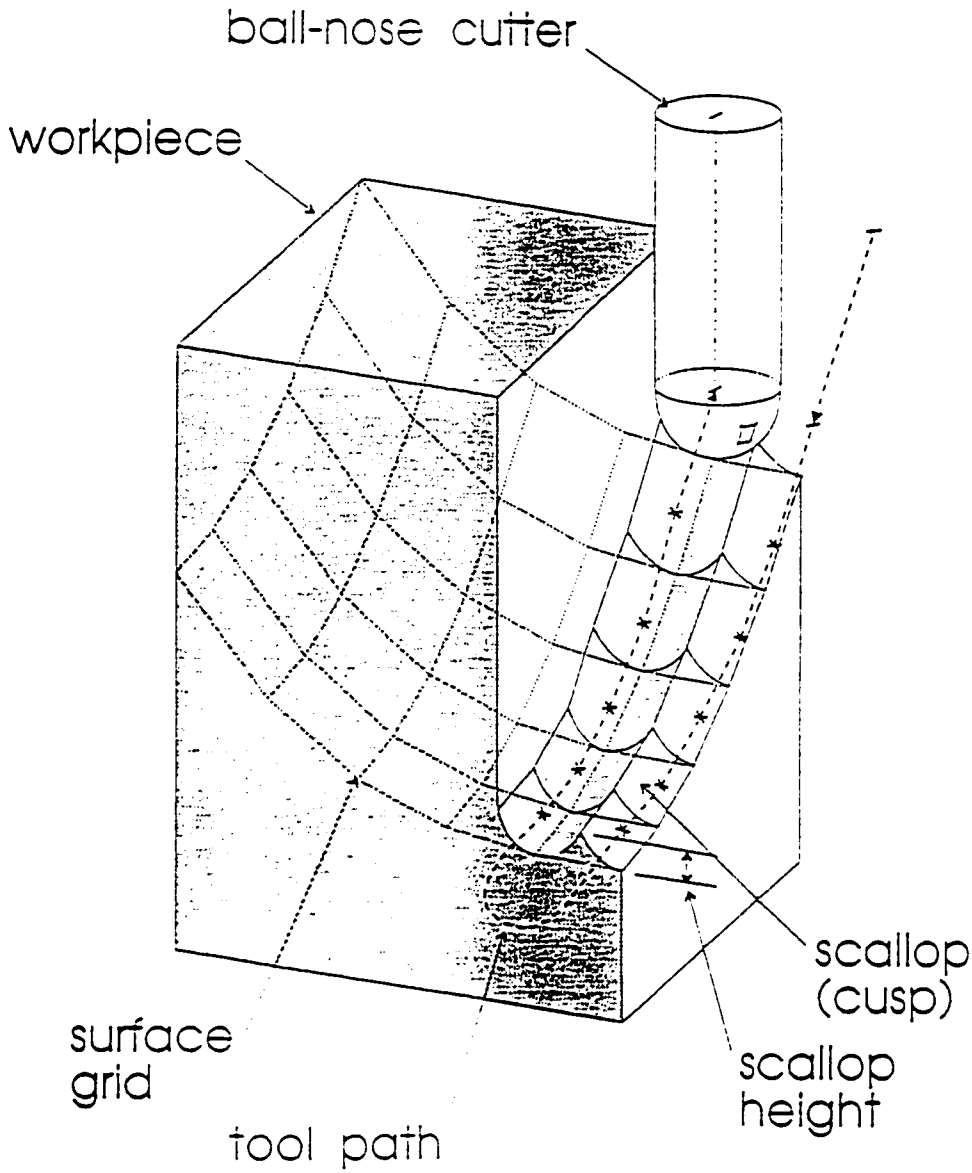


Figure 1.1 A Schematic of the Point Milling Process

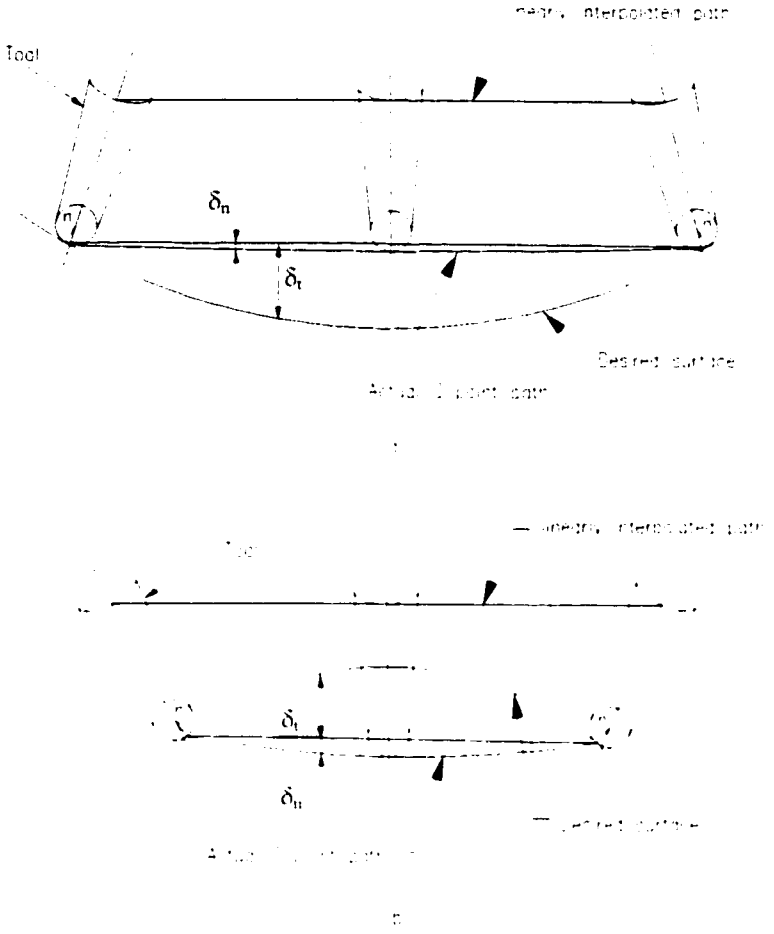
The flank milling technique enables a tremendous increase in productivity and improvement on surface finish for machining sculptured surfaces. In flank milling machining, ball-nosed end-mills with either cylindrical shape or conical shape are commonly used. By using flank milling, the curved surfaces are cut by both the cutting edges on the ball-nose and on the side surface (either conical or cylindrical) of a cutter. In conventional flank milling the entire surface is obtained after one single pass of the cutter through the blank material [2, 5, 6, 7]. The flank milling technique tends to give a good, clean surface finish. This is a productivity improvement factor because it reduces the time required for surface polishing. However, tool path generation for flank milling machining is very complex. It is generally held that a curved surface is flank millable if it can be closely approximated by a ruled surface [8]. Furthermore, the milled surface may deviate from the ruled surface (sometimes quite significantly) owing to the twist of the surface along a straight line element. In fact, Wu [2] demonstrated that the ruled surface criterion for flank milling is neither necessary nor sufficient. Many complex arbitrary surfaces are closely flank millable and can be rendered exactly flank millable with one or more passes per surface.

1.3 Multi-Axis CNC Machining Characteristics and Machining Errors

For machining sculptured surfaces, a tool path is required to contain the spatially varying cutter positions and its axis orientations (referred to as the cutter locations) in order to achieve better cutter accessibility for machining non-single-valued surfaces. Five-axis CNC machining provides more flexibility for the realization of these cutter location spatial changes. In fact, rotational movements themselves in five-axis CNC machining provide better cutter

accessibility and also result in better surface finish. However, five-axis CNC machining involves complex kinematic issues. The coordinated translational and rotational movements are non-linear functions of the cutter locations. These coordinated motion functions not only depend upon the machine configuration, but also upon the machining set-up information, such as the relative positions of the fixture and the part mounted on the machine. A different set of functions is required for each CNC machine tool configuration. In addition, the simultaneous translational and rotational movements are involved, because each new cutter axis orientation requires the motion of at least one other (usually more) axis. There are also coupling effects of the rotary movements on the translational movements, because changing the orientation of the cutter axis will affect the position of the cutter. These simultaneous and coupled movements cause the cutter contact point (CC point) moving in a non-linear manner. As a result, the total machining error in each motion step is made up from two sources.

Many factors contribute to CNC machining errors. One of the factors is due to the MCU interpolation method. Conventional CNC machines support the functions of 2D or 3D linear and 2D circular interpolations. The most common method in multi-axis CNC machining is the 'position contouring' technique. Essentially, this method connects a straight line between each consecutive machining data point and then linear interpolation is used to generate the required commands for positions along each straight line segment. As shown in Fig.1.2, a line segment is used to connect two consecutive machining data points (the spindle chuck is the machine control point, MCP) either for the machining of a concave desired surface (a) or for the machining of a convex desired surface (b). Linear interpolation generates intermediate



δ_t --- linearity error

δ_n --- non-linearity error

Figure 1.2 The Multi-Axis CNC Machining Errors

position points along the line segment. The desired surface is the design cutting curve (either concave or convex). The linear segment approximates the design cutting curve resulting in the linearity error, δ_l. Apart from the linearity error, there is an additional machining error in five-axis machining. Due to cutter orientation changes, the actual cutter contact point trajectory is a non-linear segment (since the cutter gage length is constant and MCP is interpolated along the line segment), rather than a line segment. The CC point's non-linear trajectory deviates from the linearly interpolated line segment resulting in the additional machining error, referred to as the non-linearity error, δ_n. Thus, the total machining error for each machining step includes the linearity and the non-linearity errors. In the case that the desired surface is concave cutting curve (see Fig. 1.2a), the total machining error equal to the difference of the non-linearity error from the linearity error: $\delta_{total} = \delta_l - \delta_n$ (since the non-linearity error is usually smaller than the linearity error). That is, the non-linearity error compensates the total machining error. Therefore, it is not required to reduce the non-linearity error. In other words, the 'position contouring' technique with linear interpolation is desired for machining of concave surfaces, since the total machining error is reduced by the non-linearity error. On the contrary, for the machining of convex surfaces as shown in Fig.1.2b, the total machining error for each machining step is the sum of the linearity error and the non-linearity error: $\delta_{total} = \delta_l + \delta_n$. That is, non-linearity errors add onto linearity errors resulting in bigger total machining errors, which commonly cause difficulties for ensuring ultra-precision machining requirement. Therefore, it is desired to treat the non-linearity errors in order to meet high precision machining requirement. Non-linearity errors depend upon the five-axis machining motion trajectory, which is a function of a particular CNC machine

configuration and machine rotational movements. Because rotational movements are kinematically related to the cutter orientation changes, non-linearity error depend upon the cutter orientation change.

1.4 The Non-linearity Errors Problem in Ultra-precision Five-Axis CNC Machining

In conventional five-axis CNC machining, linear interpolation method is used with the 'position contouring' technique to generate command signals for driving controlled multi-axis motions. The actual machining motion trajectory for each step, however, is a non-linear path segment which deviates from the linearly interpolated straight line segment resulting in a non-linearity error. In ultra-precision five-axis CNC machining of convex sculptured surfaces (hereafter it is referred to as sculptured surfaces), such non-linearity errors commonly cause the total machining error out of the range of the specified machining tolerance. As a result, these non-linearity errors prevent the assurance of high precision machining. The nature of the problem is that the spatially varying cutter orientations require the motion of at least one other (usually more) rotational axes. The rotational movements are superimposed onto the translational movements, causing the actual cutter contact point to move along a curved segment. While linear interpolation method cannot tracking along the curved paths, non-linearity errors are resulted and cause difficulties in ultra-precision five-axis CNC machining.

The ultra-precision multi-axis CNC machining error problem is an important problem in current industry. The problem is that for machining sculptured surfaces using 'linearization process' existing in the current postprocessors, the machine translational axes movements (at

some points) are very small or even do not move while the rotational axes move randomly and rapidly. As a result, the machining tool trajectories are random curves and this damages the workpieces. When a looser machining tolerance is specified, the problem appears to be reduced but the machining precision requirements are lost. Without knowing the cause and nature of the problem, it was normally called the 'linearization problem' in workshops. In order to discover the nature of the problem and define it clearly, investigation and analysis based on the phenomenon in the actual machining process were carried out by the author at Pratt & Whitney Canada, Inc. Through the investigation of the actual machining process, the nature of the problem was revealed as described above. For the purpose of this thesis, the problem is defined as 'the non-linearity errors problem in ultra-precision multi-axis CNC machining'.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Theories and algorithms which relate shapes and geometries of CAD models to the path and motion controls of CNC machine tools constitute a subject area called 'motion intelligence' [9]. This area of concerns, consists mainly of three categories: (1) CAD models to tool path conversion, (2) Tool path to motion trajectory conversion, and (3) Motion trajectory realization (which deals with control theory and controller design of CNC machine tools).

The category of CAD models to tool path conversion deals with the issues such as the surface representation methods and the generation of corresponding tool path. In machining sculptured surfaces, the off-line part programming approaches are utilized, in which the CAM systems divide the design surface into a set of line segments that approximate the design surface with the desired tolerance. The end points of each segment and the geometric properties of the machined surface are then used to generate the cutter locations data (CLDATA). These CLDATA are further processed by postprocessors to produce NC-codes for machining realization. The generation of CLDATA and NC-codes are the issues of the tool paths generation in category (1). The present CLDATA generation approaches consider only the geometry of the machined surfaces, and disregard the machine-dependent machining kinematics. As a result, the generated tool paths (the machining NC-codes transformed from these CLDATA) commonly cause obstacles to meeting the machining precision requirements,

particularly for the cutter orientation generations in five-axis CNC machining. Cutter orientation variations in five-axis CNC machining are kinematically related to the machining rotational movements, which in turn are functions of the machining motion trajectory as well as machining errors. Therefore, the problem with present off-line tool path generation approaches is that the real machining kinematics are not directly incorporated. To ensure machining precision, cutter orientation generations must be based not only on the geometry of the machined surfaces but also on the machine-dependent machining kinematics. Although there are procedures in postprocessors to remedy the machining errors problem caused from the off-line tool path generation approaches, other undesired consequences raise additional problems as will be explained below. In this chapter, the existing CLDATA generation approaches and the existing methods for treating non-linearity errors are reviewed (in section 2.2). Furthermore, the deficiencies of the present tool path generation approaches for actual multi-axis CNC machining are shown.

The off-line part programming produces NC-codes which are fed into the MCU of a CNC system. The interpolator in the MCU processes the NC-codes to generate the reference commands for control loops that drive the machine axes motion. The conversion of tool paths into motion trajectories is the issue of category (2), referred to also as the 'command generation'. The command generation involves kinematics of coordinated motion, machine dynamics, and interpolator design. The study of machining kinematics requires the machine kinematic models which reveal the machining geometry and time-based properties. In section 2.3, the kinematic modelling techniques used in practice are reviewed first. Then, the issues

of interpolator design are discussed. Present five-axis CNC machines utilize linear interpolator to generate and convert data positions into machining trajectories since most conventional CNC machine tools provide only linear (2D and 3D) and circular (2D) interpolators. Although the linear interpolation is the simplest approximation, it generates intermediate positions along a straight line which results in inherent machining errors, and the applications of linear interpolations have the drawback of velocity discontinuity occurring at the end points of each segment. Thus, acceleration and deceleration at each line segment is required, which produces less smooth curves while substantially increasing machining time. To adapt the practical demands required from interpolation schemes, research has been carried out, aimed mainly on 2D curved interpolation techniques that will result in less interpolation position error and maintain velocity continuity at the segment end points for three-axis machining. Five-axis CNC machining involves 3D simultaneous rotational and translational movements, and current linear interpolation techniques are not able to trace the 3D non-linear machining motion trajectory. Finally, the current status of interpolation techniques from the literature are reviewed. From which, it is concluded that there exists insufficient research work on interpolator designs for multi-axis CNC machining, in particular with respect to new designs of 3D curved interpolators.

2.2 Tool Path Generation Approaches

The goal for tool path generation is to create a machined surface which closely approximates the CAD designed surface within a certain prescribed tolerance. The concept of tolerance is central to manufacturing, and its importance cannot be over-emphasized. Therefore, the

errors introduced by tool path generation algorithms, namely, the errors caused by using the linear segment to approximate the desired machining curve must be bounded. Machined surfaces must be gouge free, and the scallop height between tool paths must be controlled. Below, the CLDATA generation approaches and its related research works are discussed first. Afterward, the current methods for generating NC-codes and issues related to these methods are outlined.

The generation of CLDATA (hereafter referred to as tool path generation), may be carried out through two different approaches: direct tool path generation, and generation of CLDATA from the cutter contact data (CC data). In direct tool path generation, a tool is dropped onto the surface with the following constraints: the cutter axis must be in a vertical plane, and the intersection point of the cutter axis and the surface normal is the position data of the CLDATA. The CC data are then calculated from the cutter radius and the surface normal. Three-axis machining uses this method. For five-axis machining, this approach is complicated by difficulties in finding the contact points between the tool and the surface, as well as in finding the best cutter axis orientation.

Using the second approach, tool paths are obtained on the basis of CC data. The techniques for generating CC data are related to the surface representation methods. Sculptured surfaces are usually represented either by parametric curves or by contouring plane curves as mentioned in section 1.2 . Using the parametric curves approach, a sculptured surface can be generally characterized by a bivariate parametric vector function p (u,v), which

represents the spatial coordinates of surface points. By keeping one parameter (v for instance) constant, the surface definition p(u,v) is reduced to a three-dimensional space curve dependent only on one parameter (u). One possible straight forward approach to generating CC data is based on incrementing the parameter u along the constant parametric curve. The step-forward distance can be set at uniform parametric steps, i.e., the isoparametric approach. This approach to surface representation and tool path generation have been used by most CAD/CAM package producers and researchers. Advanced commercial CAD/CAM packages, such as CATIA [10, 11] and SmartCAM [12], generate tool paths by using the isoparametric curves. Elber and Cohen [13] introduced an adaptive sub-isocurve extraction algorithm to develop a series of isoparametric sub-paths with uniform separation. The isoparametric curves approach is simple and generally efficient because the cutter contact curves are easy to retrieve from the surface definitions. However, the geometric properties of the machined surface are not taken into account, and the relationship between the parametric coordinate and the corresponding Cartesian coordinate is not uniform. Also, large CLDATA files, while potentially more accurate dimensionally, result in unacceptably long processing, verification, and milling times. To address this trade-off between milling accuracy and CLDATA file size, non-uniformly spaced parametric distribution of points were explored [14, 15] by utilizing the geometric properties of the machined surface, thus reducing the number of CLDATA points while maintaining milling accuracy.

By using the contouring planes method, the CC data can be generated from the cutting curves which are defined by the intersections of a group of parallel planes and the part

surface. In this case, the cutting curves are plane curves and the CC points are restricted on the cutting planes, which results in preferable tool paths. The Unigraphics [16] CAD/CAM package uses this method to generate tool paths by finding the intersection curves of the part surface and the parallel contouring planes. The step-forward distance generated using contouring planes method can be determined on the basis of the machining tolerance and the geometric properties of the machined surface. The advantage of the contouring planes method is that the CC data is a plane curve, so that the distributed tool path is relatively uniform and the machined surfaces have uniform surface smoothness that meet the specified scallop height tolerance without sacrificing machining efficiency.

For three-axis and five-axis machining, a great deal of research work have been done concerning the CC data generation, the CLDATA calculation, the step-forward distance and step-over interval setting, and the analysis of gouging errors. As reviewed in the following, the studies are all based exclusively on the geometric properties of the machined surfaces and the cutter.

For machining sculptured surfaces on three-axis CNC machine tools, Wysocki [14], Loney and Ozsoy [15] investigated the variable parametric curves tool path generation techniques. The basis of the research method was to calculate the step-forward distance based on the chordal deviations. By assuming the furthest point on the curve that deviates from the chord as having half of the total parametric variation, Wysocki approximated the deviations between the machined surface and the chord. This technique is generally sufficient for

surfaces with uniform parametric distribution and is simple to implement, but only approximates the actual chordal deviation. If the underlying surface is defined by the nonuniform parametric distribution, this approximation method could yield inaccurate results. Loney and Ozsoy determined step-forward distances by subdividing the isoparametric curve into variable parameter segments that yielded the maximum chordal deviation. This variable cutter step distance algorithm is more robust and provides a numerical method to solve for the parameter value that yields the maximum chordal deviation. However, both of these chordal deviation methods suffer from limited accuracy and can produce unacceptable gouging because they are based only on surface points and do not consider the surface normal and the geometry of the cutter. Further, the chordal deviation between adjacent CC points is assumed as the machining error. This is only true if the surface normal vectors at the adjacent CC points are parallel, and both are perpendicular to the chord. In general, the true machining error should be determined by considering not only the chordal deviation but also the distance between the tool tip trajectory and the corresponding chord. Oliver et al. [17] presented a procedure to determine the chordal deviation by considering the local surface normal and the geometry of the cutter. This procedure offers improved overall accuracy for characterizing the chordal deviation at a relatively small computational cost as compared with the nominal chordal deviation methods.

Huang and Oliver [18] developed an algorithm for three-axis tool path generation in which the cutting curves are defined by using the contouring planes technique. The step-forward distance is determined by calculating the true machining errors, which employs the

orthogonal projection method to calculate the exact distance between a cutter motion trajectory and the surface. Since this true machining error calculation method is based on the physical interference between the cutter and the surface, it is more accurate than those methods based on nominal chordal deviation. Furthermore, by finding the longest linear motion that yields the specified machining tolerance, this algorithm effectively minimizes the total number of tool motions. This technique also provides a higher degree of flexibility in planning the tool path direction, because the cutting curves are defined by using the contouring planes method. However, it requires more computational effort in locating cutting curves as compared to the isoparametric method.

The research work mentioned above deals with tool path generation in three-axis machining. Five-axis machining offers many advantages over the three-axis machining. Vickers and Quan [4] compared the three-axis machining with ball-end mills and five-axis machining with flat-end mills. The effective radius of curvature of a tilted flat-end mill was introduced. The effective radius of curvature of an end-mill can vary from infinity down to the cutter radius as the inclination of the cutter to the surface normal changes from 0 to 90 degrees, allowing the cutter to accommodate a wide variation in local curvature. This property enables five-axis end milling to achieve acceptable surface quality with fewer tool passes. In comparison, the effective radius of curvature of a ball-mill is restricted to the spherical radius of the cutter. The research concluded that the five-axis end milling of sculptured surfaces can reduce the overall cutting time when compared to three-axis point milling.

Research work in the area of CC data generation for five-axis machining has been carried out extensively. Marciniak [8,19] analyzed the relationship between the machining strip width and the CC data generation for five-axis end-milling. The geometric foundations for CC data generation were presented and the possibility of obtaining the maximum width of cut of a machined strip by fitting the cutter motion trajectory to the surface shape was explored. The study concluded that, for machining surfaces which have smoothly changing curvatures, the broadest machined strip can be obtained when the CC point moves along the minimum curvature line of the surface. The maximum width of the machining strip depends mainly on the difference of the surface main curvatures at the CC point. This result presented the possibility for reducing the cutting time and promoting the machining efficiency for five-axis machining.

Li and Jerard [20] used the contouring plane method to determine the CC data by representing the part surface as a set of parametric triangles. The CLDATA are then generated by using the CC data and the local surface geometry properties, and through an interference checking procedure. It is concluded that this distinct CC data determination procedure with the CLDATA calculation algorithm can be used to avoid gouging the surface in five-axis end milling.

Choi et al. [21] presented a method for optimizing CLDATA in five-axis end milling to minimize the scallop height. The CLDATA optimization problem was formulated as a 2D constrained minimization problem in terms of the cutter orientation angles. The cutter location

data were initialized based on the local geometry analysis. Then, the final CLDATA were obtained by solving the 2D constrained minimization formula using the scallop height as a measure of optimality. The method was successfully applied in the five-axis end milling of large marine propellers. This method revealed one way to determine five-axis end milling cutter orientations to produce minimum scallop height errors.

Cho, et al. [22] presented a method for determining cutter orientation angles for five-axis end milling to produce minimum scallop height surfaces. The cutter orientations were determined by using a z-map method based on the fact that the bottom plane of the flat-end mill must not interfere with the machined surfaces. This method is another way to determine five-axis end milling cutter orientations based on the geometry of the machined surface.

Jensen and Anderson [23] presented an algorithm for generating the tool path in five-axis end milling by applying differential geometry techniques. The cutter positions and its axis orientations are generated by considering both the tangent plane and the local surface curvature. By matching the curvatures of a silhouette of the cutter to the curvatures of the surface at a CC point, excess or gouging amounts of materials in the vicinity of the cutter contact point can be mathematically determined and eliminated. Therefore, this algorithm eliminates the gouging errors in five-axis end milling. However, this algorithm assumes that the surface must have at least first-order continuity at a given CC point, and global interference is not prevented.

Lee and Chang [24] presented an error analysis method for five-axis end milling which also applies differential geometry techniques to evaluate the scallop height between adjacent tool paths. This error analysis method can also be used to generate appropriate tool path distribution.

Kruth and Klewais [25] used a two-step procedure for generating CLDATA. First, the cutter inclination was initialized based on the principal surface curvature at the CC point. Then, the cutter orientation was determined by calculating the distance between the surface and the cutter, and the distance from the CC point to the cutter axis. This procedure was aimed to achieve the best combination of scallop height, machined surface accuracy and machining time.

Lee and Chang [26] proposed a two-phase approach to global tool interference avoidance in five-axis machining. First, the convex hull of the control mesh was used to detect potential interference. Then, if the first check fails, the second detailed feasibility checking calculates the tool interference on the basis of the physical constraints. Methods for correcting global tool interference were also presented.

Bedi, et al. [3] presented a principal curvature alignment technique for five-axis machining using a toroidal shaped tool. It was proved that the best fit at a CC point can be achieved by aligning the maximum principal curvature of the cutter with the minimum principal curvature of the surface to reduce the scallop height on the machined surface. Rao

et al.[27] presented the experimental verification of the Bedi's principal axis method described above. The use of the toroidal shaped end-mill with the presented technique gave a new approach to increase the material removal in five-axis end milling.

Liu [28] presented two algorithms for five-axis flank milling tool path generation based on differential geometry and analytical geometry, which include the single point offset (SPO) algorithm and the double point offset (DPO) algorithm. These algorithms can be used in different situations in flank milling. The SPO algorithm can be used to determine the flank milling CLDATA when the overcut at the middle part of the machined surface is not permitted. The DPO algorithm can be used to calculate the flank milling CLDATA if the middle part of the machined surface cannot be undercut.

Liu, et al. [29] summarised the tool path generation techniques for three-axis and five-axis CNC machining. The procedure of CLDATA generation for five-axis machining including techniques for point milling, end milling and flank milling is outlined in detail.

Morishige, et al. [30] presented a tool path generation method for five-axis CNC machining. The method applies the C-space (a 3D configuration space) to determine collision-free cutter positions and its orientations. The determination of the C-space is based on the geometric properties of the machined surface and surrounding collision surface, thus, the method ensures collision free operation, but without considering gouging and 'overcut' (non-linearity error) problems.

The tool path generation approaches reviewed above are all based on the pure geometry analysis of the machined surfaces without considering the CNC machine tools that will be used to realize machining. Therefore, the generated CLDATA are further processed by postprocessors into NC-codes which constitute the commands needed to control the axes motions of machining. Among its numerous functions, a postprocessor checks the tool path precision for each path segment during the generation of NC-codes, in other words, the postprocessor checks if the total machining error between the desired cutting curve and the actual CC point's travelled path is within the machining tolerance. Upon testing, a process, referred to as the 'linearization process' of NC-codes, is normally used to treat out-of-tolerance errors along the tool path where it is required. The detailed procedures of the 'linearization processes' in the existing postprocessors and in the literature are reviewed in the following paragraphs.

In the Automation Intelligence Generalization Postprocessor (AIGP)[31], a 'linearization process' was designed to reduce the non-linearity machining errors. The method relies upon testing the amount of deviations of the actual non-linear tool path from the linear segment of the NC-codes. This method inserts bisectionally additional data points between adjacent CLDATA, which in turn, are transformed into NC-codes to ensure that the deviations (non-linearity errors) do not exceed the allowable machining tolerance. The insertion can be performed until either all points are within the machining tolerance or until a maximum of 63 points are inserted between each two consecutive data points. The Vanguard Custom Postprocessor [32,33], the Omnimill Custom Postprocessor [34], the

Bosto Custom Post [35], the AIX Numerical Control Post Generator [36] and the ICAM Post Generator [37] all use the same 'linearization process' to treat the non-linearity errors problem.

Cho et al. [22] analyzed the non-linearity error in five-axis CNC machining problem and presented a 'linearization process' procedure for generating the NC-codes in a five-axis end milling process. The five-axis CNC machining errors were analyzed as two parts: one portion of the machining errors is the linearity error which is due to the linear line segment approximation to the desired cutting curve. Another portion known as the 'overcuts', is the result of the rotational movements from the current position to the command position with different cutter axis orientations. These 'overcuts' are actually non-linearity errors. The machining error for each move was the summation of the linearity and the non-linearity errors. The function relating non-linearity errors to the cutter orientation changes for the considered five-axis swivel head type CNC machine were developed. Based on this function, actual non-linearity errors were determined for the original cutter orientations. The allowable non-linearity errors were calculated on the basis of the specified machining tolerance and linearity errors determined using the tool tip position change. Upon testing whether the actual non-linearity errors exceeded the allowable non-linearity error ranges, a set of intermediate cutter position data were inserted where the test was true. The cutter orientations of the inserted data were set to vary linearly in successive positions. In this way, the resulting NC-codes include not only the data transformed from the CLDATA but also the additional inserted data points. The final NC-codes contain a dense set of unequally spaced machining data points.

The algorithm of linearly varying the cutter orientation is simple, but it interpolates the orientations inaccurately between end points, which in turn causes surface errors.

Takeuchi et. al. [38] presented a 'linearization process' procedure to modify NC-codes in a multi-axes CNC machining process. The function of the 'linearization process' was to insert additional data points between the adjacent NC-codes where the total machining error exceeds the specified tolerance range. The inserted points were calculated by subdividing the straight line segments into equally spaced intervals and the cutter orientations were set to vary linearly in the successive insertion positions. Although the final cutter positions were equally spaced, a rather dense set of machining data resulted. The cutter orientations suffer from the same problem as in Cho's procedure described above.

The 'linearization processes' discussed above manipulates NC-codes by inserting extra machining data position points. Although the produced NC-codes satisfy the machining requirement, they may contain dense sets of non-equally spaced data position points with constant or linearly varying cutter orientations. The constant cutter orientation algorithm causes severe roughness around the end points along the surface since the cutter orientation changes abruptly at these points. Linearly varying cutter orientations produce a better surface, but still insert the orientations inaccurately since the change in orientation is not necessarily linear. As a consequence, the dense sets of machining data cause an non-constant feedrate along the curve, which in turn causes an non-smooth surface finish. In addition, the total machining time is increased because the mean feedrate is less than the desired value.

2.3 Command Generation Techniques

Coordinated machining kinematics describe the geometric and time-based properties of multi-axis movements. The functional relationships between multi-axis movements and cutter locations, known as the machine kinematic models, depict the kinematic geometric properties. These kinematic models depend not only upon a multi-axis CNC machine tool configuration, but also upon the machining setup data (such as the fixture length and the workpiece mounting positions). A different set of transformation functions is required for each type of machine configuration. Machining kinematics include the forward and the inverse kinematics: forward kinematics deals with the problem of determining the cutter locations by knowing the machine axes movements, while inverse kinematics involves the computing of machining movements which are used to attain the given position and orientation of the cutting tool.

Two methods are generally used to derive the kinematic models. Paul [39] proposed the homogeneous transformation method that first derives the forward kinematic model, which is then used to determine the inverse kinematic model. Lee and Ziegler [40] proposed a geometric approach which uses the geometric configuration of the mechanism and the directly perceived geometric senses (geometry intuition) of the machining movements to determine the inverse kinematic model. The forward kinematic model is then determined from the inverse kinematic model. The geometrical approaches that have been utilized by researchers in modelling of machine kinematic models are reviewed below.

Using a geometric approach, Chou and Yang [41] formulated the inverse kinematic

model for a fixed-bed type of machine tool with Euler angle structure (the machine table moves translationaly only and the spindle rotates to approach the cutter orientations). The formulation of the translational motion and the rotational motion were carried out separately. First, the translational movements were formulated on the basis of the machining setup data. Then, the rotational movements were formulated on the basis of the geometric intuition of the rotary motions. Next, by considering the coupling effects of the rotational movements on the translational movements, the variations on the translational coordinates due to rotational movements were superimposed on the translational movements coordinates obtained in the first step.

Cho et al. [22] formulated the inverse kinematic model for a swivel-head type five-axis CNC machine tool by using a geometric approach. The machine movements of the swivel-head type five-axis machine tool consist of the spindle rotation movements about the pivot which translates in space simultaneously. Based on the machine configuration and the geometric intuition of the machining movements, the inverse kinematic model was formulated. The machine translational movements thus obtained were the functions of the cutter positions and orientations. The rotational movements were the functions of the cutter orientation angles, which shows that the cutter orientation changes result in rotational movements that are coupled with the translational movements.

Using a geometric approach, Liu [6] formulated inverse kinematic models for five-axis CNC machines which have the configurations where only the swivel-head moves (the machine

table does not move), and for those in which only the machine table moves (the five-axis motions are the machine table's movements). The results showed that the machine rotational movements are related to the cutter orientation changes, and that the machine translational movements are functions of both the cutter position and the orientation changes.

Lin and Koren [42] formulated the inverse kinematic model for five-axis CNC machines which have the structure of one tilt table and one rotary table placed on top of a three-axis machine. The kinematic modelling was based on a geometric approach and was termed the 'decoupling approach'.

Based on the machine kinematic models, the time-based characteristics of machining movements can be determined and machining motion dynamics analysis can be carried out. Chou and Yang [9, 41] presented a procedure for relating the time-based properties of machining to the geometrical properties of a tool path based on the derived kinematic model. Further, they established the relationship between the machining kinematics and machining dynamics.

Interpolators are essential components in CNC machines which generates commands for tool motion between adjacent tool path data points as per accuracy requirements. Interpolation methods can be divided into reference-pulse and reference-word techniques [43, 44]. In reference-pulse systems, an interpolator produces a sequence of reference-pulses for each axis of motion, where each pulse generates a motion of one basic length-unit (BLU).

With the reference-word scheme in the sampled-data system, the control loop of each axis is closed by software through the computer itself (which generates reference binary words). The most widely used interpolation method in both of the reference-pulse and reference-word systems is the digital differential analyzer (DDA) interpolator. The DDA techniques can be used to perform interpolation of integral, exponential, trigonometric, and polynomial functions. Mayorov [45] and Sizer [46] detailed how the interpolation of these functions can be implemented using actual hardware. DDA techniques can be used for both parametric and nonparametric curve generation as explained by Danielsson [47]. However, some degeneration errors may occur with losses of interpolation accuracy as shown by Danielsson [47] and Milner [48]. The hardware DDA interpolation techniques are well known [44, 49, 50], which are capable to perform 2D and 3D linear interpolations and 2D circular interpolations. Simulating the hardware DDA technique, Koren [51] introduced a software DDA interpolator for CNC system applications. Koren and Masory [52] discussed and compared four reference-pulse interpolation methods: the software DDA interpolation, the stair approximation interpolation, the direct search interpolation and the improved direct search interpolation. It was concluded that only the DDA interpolator produces a constant feedrate along a circular path. With the other methods, considerable variations can occur along the circular path. However, the problems of register overflow and integer round-off limit the adoption of the DDA interpolators in some applications. Gan and Woo [53] discussed the DDA's register overflow and the integer round-off error problems. It was pointed out that the overflow has nothing to do with the register size, unlike the round-off error which is a function of the register size. They applied the DDA technique to parametric

curves and proposed solutions to the problems of overflow and round-off errors.

There are three major requirements for the interpolated curve in the command generation stage. The first requires the fitted curve to have second order continuity, because this results in better machining quality, less vibration, and a longer tool life. The second requirement is that the interpolated curve should be easily convertible from a position parameter to the time domain. Thus the required machining conditions, such as the speed, acceleration, actuating torques, and jerk can be calculated and computer control can be incorporated. The third requirement is that a fast algorithm is required for an on-line implementation of this space/time conversion.

To satisfy the second requirement above, linear interpolation is commonly used in machining sculptured surfaces on five-axis machine tools. The tool path data are interpolated by the point to point type interpolator using straight lines from one point to another. This interpolation method, however, has an inherent position error and has the drawback of velocity discontinuity at the tool path data points. Sata et al. [54] presented an analytical interpolation method, which used an incremental method for generating the Bézier curves to connect a series of discrete tool path data points. With this improvement, the number of interpolation segments were reduced as compared to linear interpolation. Stadelmann [55] developed a speed interpolator which can be used when the surface topology is available as a series of Bézier curves. By using parametric discretization, the method significantly reduced the computation time as compared with Sata's interpolator. Both of these interpolators

ensured velocity-continuity from one segment to the next based on the assumption of constant acceleration over the entire segment. Makino [56] presented a trajectory control method using planar Clothoid or Cornu Spiral curves to interpolate two lines or a series of points. By using the Cornu Spiral curve, the direction and the curvature of the interpolated curve is kept continuous, so that a higher speed continuous path control of a robot could be achieved. Papaionnou and Kiritsis [57] presented an extrapolated algorithm in which the next interpolation point is determined by solving a constrained optimization problem. Liu et al.[27] presented a summary on the analytical surface interpolation techniques used in practice.

To satisfy the second order continuity requirement, the cubic spline interpolation technique has been applied by researchers. This interpolation technique entails the fitting of a composite third order parametric curve to the set of tool path points. Chou and Yang [9, 41] proposed an analytical off-line interpolator for command generation, in which the cubic spline interpolation technique was used to generate a parametric cubic spline curve tool path, instead of a straight line segment of the linear interpolation. From the parametric model, the integrated relationship between the position parameter and time was also formulated and the dynamic information needed to control the machine motion, the velocity and the acceleration were determined. On the basis of Chou and Yang's [9] proposition, Huang and Yang [58] presented a real-time version of the interpolator which determines the intermediate points of parametric curves under the machining precision, the velocity and the acceleration constraints. The interpolation position parameter formulated as a function of time, i.e., $u = f(t)$, and the feedrate function were solved by using the Euler method. This interpolator is capable of

generating position commands with variable speed control and better speed accuracy for parametrically represented tool paths. This is the most advanced interpolator in existence; the design of this interpolator, however, is based on a three-axis machine and therefore it can be applied only to three-axis machining.

To satisfy the requirement of easy conversion of the position from a tool path to the machining trajectory and the requirement of fast execution of interpolation, a tool path ideally should be parameterized in arc-length of the spline segment. However, the arc-length for each segment is unknown and therefore the method of cubic spline interpolation must approximate its arc length. Usually, the chordal length is used as an approximation. Because of this approximation, relatively large undesired speed fluctuations will occur in the execution of commands. Renner and Pochop [59] and Renner [60] developed methods of fitting a composite cubic spline with closely being arc-length parameterization. Wang and Yang [61] presented a composite quintic spline interpolation method. The resultant composite quintic spline, in comparison with cubic spline, are nearly arc-length parameterized. Since, for a fifth-order spline, each coordinate has two more coefficients that can be used to force a better unit tangent property along the cubic spline curve, the algorithm generates the quintic spline with nearly arc-length parameterization based on a cubic spline of a set of tool path data. The nearly arc-length parameter was estimated by minimizing the deviation of the rate of change of arc length to the path parameter.

Kiritsis [62] presented an incremental step interpolation algorithm which permits

interpolation of any kind of 3D parametric curves. The incremental step algorithm uses two principles. First, each selected step must follow a given direction of the curve. Secondly, each selected step must be at closest distance from the curve. Based on these two criteria, the algorithm was used to generate a 3D helix. The method shows that a non point-to-point interpolation scheme is another possible direction for accurate interpolator design.

Lo and Hsiao [63] presented an interpolator with a contour error compensation procedure which is based on previous machining results. Applying the proposed interpolator, in the initial machining process, the contour-error at every sampling instant is calculated and a data extracted procedure is conducted. Then, in the repeated machining process, the contour error is interpolated based on the previous extracted data and is added to the reference position commands. Thus, the previous contour machining result can be used to compensate the machining contour errors. The proposed interpolator improves the accuracy of the subsequent repeated machining, but it is designed on the basis of analysis of 2D contouring curves.

The off-line part programming approaches decompose the design surface into line segments. These segments are then interpolated and converted into machining trajectory. The drawbacks of this off-line procedure are: (1) the acceleration and deceleration at each line segment is required, which produces less smooth curves and substantially increases machining time, (2) cutter orientations in five-axis machining are interpolated inaccurately, which causes position errors and unsmooth surfaces, and (3) the size of the tool path file could be very

large for complicated parts and could cause memory shortage problems and data transmission errors [64]. To overcome the drawbacks, real-time interpolators have been designed. Shpitalni, et al. [65] presented a real-time reference-word interpolator for the implicitly defined 2D curves and the parametrically represented curves. By using the real-time interpolator, the CAD system transfers only the information about the curves to the CNC machine. The curves are broken into segments by the MCU and executed by the interpolator. The real-time interpolator then calculates new commands for the control loop during the execution time of the current commands. This approach produces smoother surfaces and requires less machining time. The real-time interpolator is of the point-to-point type, and therefore it can be executed by the linear interpolators available on conventional CNC machine tools. For the parametrically represented curves, the determination of successive values of the path parameter was based on the curve segments of equal arc-length. The curve position parameter as a function of time was solved through a recursive procedure based on Taylor's expansion.

Koren [7] developed a real-time interpolator for five-axis CNC machining. The interpolator calculates cutter positions and orientations during the same time period needed for sampling the control-loop feedback devices. The cutter axis for five-axis end milling was oriented in the direction of the surface normal. For flank milling, the cutter axis was oriented parallel to the ruled surface. These cutter orientation interpolation methods are not adequate. In fact, in end milling the cutter orientation cannot be set always in the surface normal directions, and in flank milling, the cutter orientations may not be always parallel to the ruled

surface, especially for the non-ruled flank millable surfaces. While this real-time five-axis interpolator is the most advanced interpolator for five-axis CNC machining, the cutter orientation interpolation methods are not absolutely correct.

Lo [66] presented a real-time surface interpolator which is capable of maintaining machining feedrate and condensing the cutter location file. Using the proposed surface interpolator, the surface parameters in both of the tool path (step-forward) direction and the tool interval (step-over) direction, and the cutting conditions (such as feedrate, scallop height limit, etc.) are fed into the MCU. Thus, the off-line tool path generation procedure is moved into the real-time surface interpolator. Since the CC point path is generated in real-time according to the present feedrate, the desired feedrate can be maintained. However, the proposed interpolator focuses on the feedrate problem, and it may not provide solution to the machining error problem in the current five-axis CNC machining process.

CHAPTER 3

THE OBJECTIVES

3.1 Thesis Objectives

The non-linearity error problem in five-axis CNC machining causes difficulties when high precision in the machining of sculptured surfaces is required. The problem arises from the fact that five-axis CNC machining motion trajectories are non-linear curve segments due to the rotational movements superimposed on the simultaneous translational movements and linear interpolation technique is not able to generate command signals for positions along curved paths. The five-axis machining movements are kinematically related to the cutter locations data, in other words, non-linearity errors are related to the tool path generation. Thus, the solution can be approached from tool path (CLDATA) generation with the requirement that the machining errors are minimized. Another route to the solution is from the machining motion trajectory planning and control viewpoint (i.e. an interpolator design that drives the machine axes along a predesigned curved path in order to eliminate the non-linearity errors). This thesis sets two objectives:

The first objective is the development of an off-line tool path generation method that minimizes the multi-axis CNC machining errors. Tool path generation involves generating cutter positions and its axis orientations (CLDATA) and subsequent conversion of CLDATA into machining NC-codes. The review of the literature revealed that the existing methods for generating CLDATA are all based solely on the pure geometry of the machined surfaces. The

'linearization processes' treat non-linearity errors by inserting additional cutter positions in generating NC-codes, which results in the unsmooth tool path, the undesired variations in machining feedrate and machined surface finish problems. The cause of non-linearity errors is the superposition of multi-axis rotational movements on the simultaneous translational movements. Rotational movements are kinematically related to cutter orientation changes, and non-linearity errors are the functions of cutter orientation changes. Therefore, a methodology that generates tool path with optimum cutter orientations and reduced non-linearity errors is required. The research program leading to the development of a minimum errors tool path generation methodology for the ultraprecision multi-axis CNC machining can be described as follows. The cutter orientation generation should consider not only the geometry properties of the machined surfaces, but also the machining kinematics and motion trajectory characteristics. Specifically, the cutter locations data may be initially generated based on the geometric properties of the machined surface, and then modification of cutter orientations based on the particular machine kinematics and motion trajectory should be carried out to achieve minimum machining non-linearity errors. The proposed new methodology in this thesis is based on these analysis and is verified through a case study.

The second objective of this thesis is to develop an on-line 3D interpolation technique that generates the command signals for the control loops to drive the machine axes moving along a 3D curved trajectory such that the multi-axis machining non-linearity errors can be eliminated. Linear interpolation techniques generate command signals for positions along a straight line path, not along non-linear motion trajectories in multi-axis machining. From the

review of the literature, it was demonstrated that the current interpolators are capable to perform 2D linear, 2D circular, 2D curved paths and 3D linear interpolation functions. The multi-axis machining motion trajectories are spatial non-linear paths, and therefore 3D non-linear interpolation techniques are desired. Since the multi-axis CNC machining motion trajectory is constructed by 3D circular movements superimposed on 3D linear movements, the required 3D non-linear interpolator should be capable of tracing 3D circular curves and also the combined 3D linear and circular loci. The 3D non-linear interpolator should perform in a way of point-to-point command generation so that the available linear interpolation function can be utilized as the base to drive the axes motions. The 3D non-linear interpolator should have accurate position tracking ability and feedrate uniformity along a curved path. The methodologies presented in research studies in this thesis are innovative and results encouraging. These studies contribute solutions to some present day concerns in the manufacturing industry. The motion trajectory functions of the combined 3D linear and circular movements on the common five-axis CNC machine tools are utilized to develop the 3D non-linear interpolation formula. The 2D DDA circular interpolation principle has the advantage of the feedrate uniformity and therefore are used to develop the 3D DDA circular interpolation principle. Finally, by combining the existing point-to-point 3D linear interpolation function with the new 3D DDA circular interpolation principle, a combined 3D linear and circular interpolation technique is developed.

3.2 Thesis Outline and Methodology

The research program of investigating the off-line tool path generation methodology in the

thesis will concentrate on the following items:

1) Development of the kinematic models for a common five-axis CNC machine tool. The machining kinematics deal with geometry and time-based properties. Therefore, the inverse and the forward kinematic models are required in tool path generation.

2) Development of the machining motion trajectory model for five-axis machining on a common five-axis CNC machine tool. The locus of the cutting point is depicted by the machining trajectory functions. Hence, a machine-dependent machining motion trajectory model is required in the determination of non-linearity errors.

3) Development of a representation approach for the analytic description of the tool path with which the geometric properties of the machined surface curves can be calculated. To determine the allowable non-linearity errors for a certain specified machining tolerance, linearity errors must be pre-determined based on the geometry properties of the machined surfaces. Thus, an analytic tool path representation of the machined surface is desired to calculate the curvatures along the curve, which in turn are used to determine the linearity errors.

4) Development of an off-line software program to implement the minimum errors tool path generation method for solving the multi-axis CNC machining errors problem.

5) Verification of the minimum error tool path generation methodology through a simulation of the process for machining a typical sculptured surface on a five-axis machine tool.

The research program of developing a 3D interpolation technique for multi-axis CNC machining in this thesis will concentrate on the following items:

1) Development of a 3D circular interpolation principle. The multi-axis rotational movements are 3D circular motions. In order to trace a 3D curved path in multi-axis machining, 3D linear interpolation or 2D interpolation techniques are not adequate. A 3D circular interpolation function is required for both the machining of 3D circular curves and the machining of combined 3D linear and non-linear curves.

2) Development of a combined 3D linear and circular interpolation principle for multi-axis CNC machining systems. The multi-axis machining motion trajectories are constructed by the simultaneous and coupled 3D rotational and 3D translational movements, therefore it is necessary to develop an on-line 3D interpolation command generator that can be applied to combined 3D linear and circular trajectory planning and control.

3) Development of a real-time interpolation routine to implement the new designed 'combined 3D linear and circular interpolation technique' for five-axis CNC machine tool systems.

4) Verification by computer simulation of the machining process using the new 3D interpolator in a five-axis machine tool.

CHAPTER 4

MINIMUM ERROR TOOL PATH GENERATION METHOD

4.1 Introduction

In the machining of sculptured surfaces, five-axis CNC machining offers many advantages over three-axis machining which include faster material-removal rates, improved surface finish, and the elimination of the necessity for hand finish. In general, a single surface which can be machined by three-axis milling can also be machined in five-axis machining with a flat-end mill. But, in some cases, for example, as in the machining of an integral impeller or an integral turbo-wheel, it is impossible to use three-axis machining because the machined surface, the drive surface and the check surface [43] are not single-valued surfaces. The cutter axis orientation must be changed in order to avoid the non-tool-edge interference between the cutter and the workpiece. Five-axis machining exploits a machine tool with more flexibility to realize the cutter position and its axis orientation spatial changes, employing the simultaneous rotational and translational movements. The actual cutter contact point in five-axis machining moves along a non-linear trajectory, and the machining errors comprise not only linearity errors but also non-linearity errors. In ultra-precision multi-axis CNC machining, non-linearity errors are the main cause of obstacles that make it difficult to ensure the machining precision. A new methodology for solving the non-linearity error problem in ultra-precision multi-axis CNC machining is presented in this chapter. The methodology is based on machine-specific kinematics and the machining motion trajectory analysis. The non-

linearity errors are reduced by modifying the cutter orientations to ensure the machining precision, provided that there is no interference between the cutter tip region and the workpiece. In section 4.2, the basic concepts and the detailed description of the common techniques for tool path generation in five-axis CNC machining are discussed, and the need for optimal cutter orientation generation based on machining kinematics and motion trajectory characteristics is explained. Sections 4.3 and 4.4 present the developments of the kinematic models and the machine motion trajectory model for a five-axis CNC machining centre. Next, the proposed methodology is presented in section 4.5. Finally, a software program for implementing the new methodology is presented in section 4.6.

4.2 Five-Axis CNC Machining Tool Path Generation Methods

In CNC machining, the tool path is the cutting tool's motion locus which is known as the cutter location data (CLDATA). In multi-axis CNC machining, a set of CLDATA is specified in terms of the coordinated cutter positions and its axis orientations with reference to the workpiece coordinate system. The cutter position data are the motion coordinates of the cutter centre which is the spherical centre of a ball-end mill or the flat-end disc centre of a flat-end mill. The cutter orientation data are the direction cosines of the cutter axis vector relative to the workpiece coordinate system. The generation of tool path in multi-axis CNC machining includes the generation of the cutter position data and the cutter orientation data.

The goal for tool path generation is to create a machined surface which approximates the design surface within a certain prescribed tolerance. Ideally, the generated tool path

should contain a set of uniformly distributed cutting curves, equally spread machining errors and smoothly spaced cutting steps. In sculptured surface machining, the cutter position data are generated on the basis of the cutter contact point data (CC data) on the machined surface. The common methods for generating CC data are the parametric curves approach and the cutting plane contouring curves approach.

The parametric curves approach uses straight lines in a parametric domain to define the corresponding parametric curves on the actual surface in Cartesian space. A sculptured surface is described as a two parameter function, P(u, v). Holding one parameter constant such as P (u, v = constant) defines a parametric space curve (hereafter referred to as the 'cutting curve'). By setting the parameter v, a set of curve function $P(u, v_i)$ define the entire surface. The distance between two adjacent cutting curves (i.e., for v = i and v = i+1) is referred to as the tool pass interval. The methods for determining the tool pass interval can be classified into the constant parameter method and the non-constant parameter method.

With the constant parameter method, the parametric lines are equally spaced across the parametric domain. Depending on the machining surface for which it is used, this method may not lead to equally spaced cutting curves on the machining surface because a constant line interval in parametric domain does not generally yield a constant tool pass interval in Cartesian space. If the surface is flat, the tool pass interval is constant for all tool passes, and the method is accurate and efficient. However, for the machining of sculptured surfaces, the tool pass interval is generally different from one pass to another, depending on the tool radius

and the local surface curvature. Therefore, this method may not attain the machining accuracy and its efficiency may vary depending on the geometry of the surface. The constant parameter method is simple, but the relationship between the parametric coordinate and the corresponding Cartesian coordinate is not uniform as it does not consider the geometric properties of the machined surface. For instance, for machining a fan blade surface with the constant parameter method [18], the cutting curves are close to one another at one end but further apart at the other. To ensure the machining accuracy, more cutting curves may be generated. As a result, the scallop heights on the machined surface vary from one end to the other: the roughest portion meets the specified scallop height tolerance while the finest portion is unnecessarily too smooth.

With the non-constant parameter method, the tool pass interval is determined by calculating the machined strip width on the surface and by controlling the scallop height between the cutting curves. The calculation of the machining strip width and the scallop height on the machined surface are based on the geometric properties of the surface. Thus, the cutting curves can be distributed to ensure surface smoothness that meets the specified scallop height tolerance without sacrificing machining efficiency.

The determination of the tool pass interval is one aspect of tool path generation. Another difficult problem is the correct generation of the cutting point step distances along each cutting curve which resolves the distances between cutter positions. In multi-axis CNC machining, each cutting curve is approximated by piece-wise linear segments joined together.

The length of the linear segments is referred to as the step-forward distance. The algorithms for calculating the step-forward distance include the isoparametric algorithm and the varying parametric algorithm. In the isoparametric algorithm, the cutting points are a function of somewhat arbitrary choice of parameterization. The step-forward distance calculated from this algorithm is not related to the machining errors and the surface local curvature. Alternatively, the varying-parameter algorithms calculate the step-forward distance by using the surface local curvature and the specified machining tolerance. The isoparametric algorithm is simple and generally efficient. In practice, a hybrid approach is used. The isoparametric algorithm is usually used to obtain a dense set of discrete cutting points, then the varying-parameter algorithm is used to refine the step-forward distance by sifting the discrete cutting points. These cutting points are considered as those at which the cutter will contact with the machined surface. The cutter position data are actually the offset points of the machined surface. Based on the cutting points, the surface normals to these cutting points and the cutter radius, the cutter position data can be easily generated. Thus, the cutter position changes depend upon the cutting points step-forward distance.

In the cutting plane contouring curves approach, the cutting curves usually are defined by the intersections of a group of parallel planes or cylindrical surfaces with the workpiece surface. For point milling, the cutter position data of a ball-end mill can be generated by constructing the offset surface of the machined surface such that the cutter centre moves on the intersecting curves of the cutting planes and the offset surface of the machined surface. To construct the offset surface in sculptured surface machining, the common method [6]

includes the following steps:

1) Divide each cutting curve to obtain a set of discrete cutting points such that the line segment and the curve are within certain prescribed tolerance;
2) Calculate the equal offset surface points of the discrete cutting points;
3) Use mathematical spline techniques, such as the B-spline technique, to obtain the offset surface from the offset points.

The intersection curves of the cutting plane with the offset surface will be the path of the cutter centre of ball-end mills. The cutter position changes can then be determined by employing the varying parametric calculation methods. With this method, the cutter position data are generally on the space curves while the cutting points are on the plane cutting curves. Therefore, the tool path and the resulting scallop height are uniformly distributed, and the machining efficiency is higher than in the parametric curves approach. This method, however, requires proper selection of the cutting planes, in terms of determining the spacing and the direction.

The location of a cutter is completely specified by the cutter centre position and the cutter axis vector. In five-axis machining, the cutter axis vector cannot be orientated in the surface normal direction for the following reasons: (i) the cutting velocity at the tool tip of a ball-end mill is zero; (ii) there is usually no cutting edge on the flat-end surface centre of flat-end mills; (iii) the cutter axis vector of a flat-end mill must be tilted to avoid the cutter flat-end contacts with the machined surface and to avoid the cutting edges interference with the machining surface. Therefore, five-axis contour milling uses a process called 'sturz'

milling. The German term 'sturz' refers to the forward tilt of the milling cutter in the direction of the feed. The 'sturz' angle is the angle between the cutter axis vector and the surface normal to the cutting point. The 'sturz angle' can be divided into a tilting angle and a rotation angle [22]. The values of the tilting and the rotation angle determine the cutter axis vector at the cutter position with no interference between the cutter top region and the workpiece. The cutter orientation data is the representation of the cutter axis vector in terms of the direction cosine of the vector in reference to the workpiece coordinate system.

The generation of cutter orientation data must consider the constraints of machining. One of the constraints comes from the revolute joints limit on the CNC machine tool. A five-axis CNC machine tool has at least two revolute joints. With a particular cutter position, a considerable portion of the space of the cutter orientation is constrained by the machine revolute joints limit. Another source of infeasible cutter orientations is the collisions between the machine structure and the workpiece surface. Apart from these structural limits, the third constraint is the 'cutter overcuts' on the machined surface. The 'cutter overcuts', i.e., the non-linearity machining errors, are caused by the cutter orientation changes which depend upon the geometric properties of the machined surface and also relate to the machining kinematics.

The present methods for generating the cutter orientations consider only the geometry of the machined surface. For machining of surfaces with a narrow passage between them, for instance the airfoil surfaces of a typical impeller shown in Fig.4.1, five-axis point milling is commonly used.

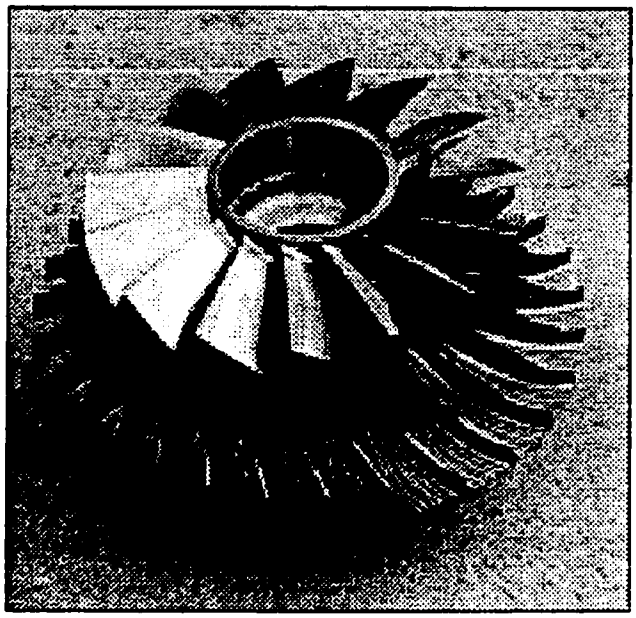


Figure 4.1 Blade Surfaces of An Impeller

To determine the cutter orientations for five-axis point milling, the cutter axis Marginal Point Technique [6] can be utilized as shown in Fig.4.2. In this technique, the normal plane of the surface tangent in the feed direction of the cutting point is used as the cutter axis oscillating plane. In Fig. 4.2, the oscillating plane for the CC point is parallel to the x-z plane and is perpendicular to the surface tangent of the CC point (the feed direction of the machining). The cutter axis can vary its orientation in the oscillating plane within a certain limit. Using the cutter axis Marginal Point Technique, the steps for determining the cutter orientation are as follows:

1) Determine the marginal point:

   In the cutter axis oscillating plane, rotate the cutter axis about the cutter position point C_o to the position at which the cutter contacts the surface at a marginal point Q;

2) Determine the projection point of the marginal point on the cutter axis:

   Based on the cutter geometry data (cutter radius and conic angle) and the distance between the cutter position point C_o and the marginal point Q , a projection point Q' of the marginal point Q on the cutter axis can be determined using the cutter conic angle, α, and the points of C_o and Q through a simple triangle geometry calculation;

3) Determine the cutter axis vector:

   From the coordinates of the cutter position point C_o and the projection point Q', the cutter axis vector, T_a , is uniquely orientated.

Using such a technique, a set of cutter orientation data can be generated with collision-free conditions and maximum tilting angle. The generated cutter orientations vary at each cutter position.

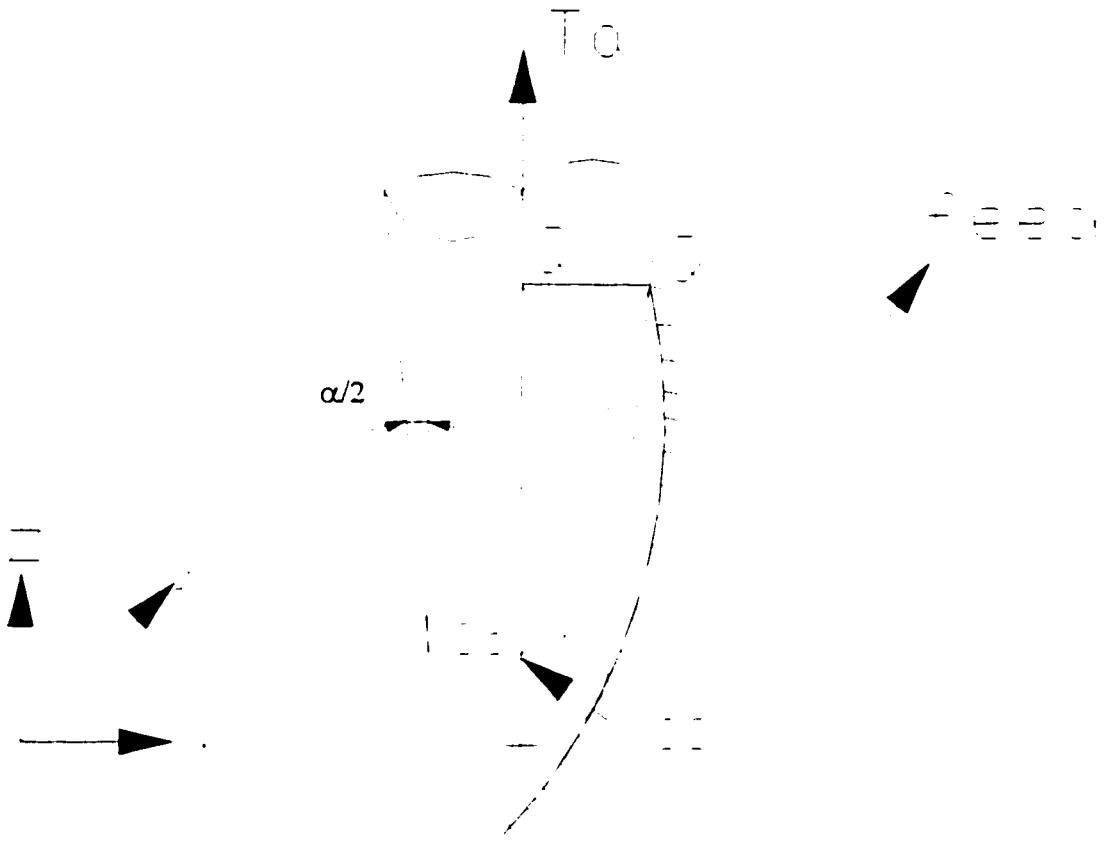


Figure 4.2 The Marginal Point Technique For Determining Point Milling Cutter Orientation

For the machining of large sculptured surfaces, such as the machining of the blades of a hydraulic turbine, five-axis CNC machining with flat-end mills is commonly used. The cutter axis vector for end milling can be determined by estimating the cutter tilt angle from the surface curvatures for different machining surfaces, and using the surface normals and tangents in the feed direction [6]. Li and Jerard [20] and Choi, et al. [22] also discussed techniques of generating cutter orientations for end milling.

Flank milling has wide applications for the milling of small- and medium- dimensional surfaces, such as the milling of the integral impellers. For machining of impeller airfoil surfaces, point milling with conic shaped ball-mills can be used. Although the determination of cutter orientations is more difficult in flank milling, this method is more favourable because it does not require hand finishing. The generation of the CLDATA for milling the airfoils of an impeller by flank milling can be determined by applying the Bind Inclining Technique [6] as shown in Fig.4.3. The Bind Inclining Technique uses the cutter calculation centre and the equivalent cutter axis vector. The intersection point between the cutter axis of a cylindrical shaped ball-end mill and the normal to the CC point is the cutter calculation centre. The C_p is the cutter calculation centre for the side CC point as shown in Fig.4.3. The cutter axis of an equivalent cylindrical shaped ball-end mill is the equivalent cutter axis vector, T_a^*, to the conical shaped ball-end mill as shown in Fig.4.3. With a conical shaped ball-end mill (the ball-end radius is r and the conic angle is α), the Bind Inclining Technique determines the cutter locations using the following steps:

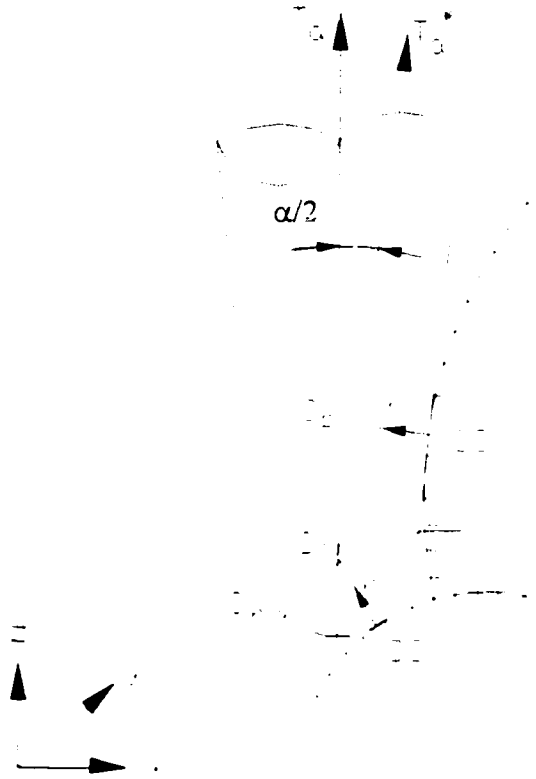


Figure 4.3 The Bind Inclining Technique For Determining Flank Milling Cutter Locations

1) Determine the cutter calculation centre and the equivalent cutter axis vector:

Consider a cylindrical shaped ball-end cutter with the ball-end in contact with the hub surface at the bottom CC point as shown in Fig.4.3. By inclining the cylindrical shaped cutter on the bind surface (i.e., the airfoil surface), a side CC point can be determined. The vector of the axis of the cylindrical shaped cutter is the equivalent cutter axis vector T_a*. The intersection point of the airfoil surface normal to the side CC point and the equivalent cutter axis vector is the cutter calculation centre C_p;

2) Determine an offset surface of the hub:

Based on the cutter ball-end radius, r, and the surface normals from each of the bottom CC points, an offset surface can be determined by using mathematical spline techniques as mentioned above;

3) Determine the cutter position data:

The intersection point of the offset surface with the equivalent cutter axis vector is the cutter position C_o as shown in Fig.4.3.

4) Determine the cutter orientation data:

By rotating the equivalent cutter axis vector T_a * about the cutter position C_o by an angle of $\alpha/2$, the resulting orientation of the vector T_a * is the desired cutter axis vector T_a of the conical shaped ball-end mills.

Using this technique, under collision-free conditions, a set of CLDATA can be generated with varying cutter orientations at each cutter position.

As mentioned above, the cutter orientations are not only dependent upon the pure geometry of the machined surface, but are also related to the machining kinematics. In fact, the machine translational movements are functions of position and axis orientation of the cutter, and the machine rotational movements are functions of the cutter orientation data. Therefore, the determination of cutter orientation data must also consider the machine kinematics. Machine kinematic models describe the geometric properties of machining kinematics, and reveal the relational functions between the cutter locations (CLDATA) and the machining movement coordinates (NC-codes).

4.3 Development of Machine Kinematic Models

The cutter variables are the cutter's positions and axis orientations in reference to the workpiece coordinate system. The machine variables are NC-codes which, for five-axis CNC machining, consist of the translational axes motion parameters and the rotational axes motion parameters. The functions that relate the cutter variables and the machine variables are the kinematic models. Machine kinematics describe the geometrical and time-based properties of multi-axis CNC machining movements which include the forward and the inverse kinematic models. The forward kinematics deals with the problem of determining the cutter variables by knowing the machine motion variables. The inverse kinematics is used to compute the machine motion variables which are used to attain the given position and orientation of the cutter. The functional relationship between the machine motion parameters and the cutter variables portrays the kinematic characteristics of a multi-axis CNC machine tool. This functional relationship is dependent not only upon a multi-axis machine tool configuration,

but also upon the machining set up data, i.e., the position of the workpiece coordinate system in the machine coordinate system which may consist of the fixture thickness and the position of the workpiece. A different set of transformations is required for each type of machine configuration. The five-axis CNC machine tools with two rotary tables, one is on top of the other (hereafter referred to as rotary-table-type), are the common configuration. Therefore, a five-axis CNC milling centre, OMINIMILL series-1 (OM-1) which is the rotary-table-type, has been chosen to illustrate the work in this research.

The schematic configuration of the OM-1 five-axis CNC milling centre is as shown in Fig. 4.4. The machine spindle is horizontal, therefore, the machine z_m axis is horizontal and parallel to the machine spindle based on the convention of machine coordinate system specification [67]. The machine x_m axis is horizontal and perpendicular to the z_m axis. The machine y_m axis is perpendicular to both the x_m axis and the z_m axis, which abides by the right hand rule of the Cartesian coordinate system. The machine coordinate system (x_m , y_m , z_m, O_m) is fixed on the machine table top centre. The rotational B_m axis is perpendicular to the machine table and coincides with the y_m axis. The rotational C_m axis is parallel to the machine table top surface and is offset from the table top by y_m = 298.4 mm (11.75"). The five-axis motions of the OM-1 milling centre are rotations about the B_m axis and the C_m axis, and translates along the z_m axis and the x_m axis. The spindle slides vertically relative to the y_m axis. The given machine configuration data include (1) the home position of the spindle: (x_m , y_m, z_m) = (381.0 mm, 615.9 mm, 0.0 mm)= (15.0" , 24.25", 0.0"), (2) the pivot of the B_m axis (B_{pivot}) which is at the machine table top centre, and (3) the pivot of C_m axis (C_{pivot}) as shown

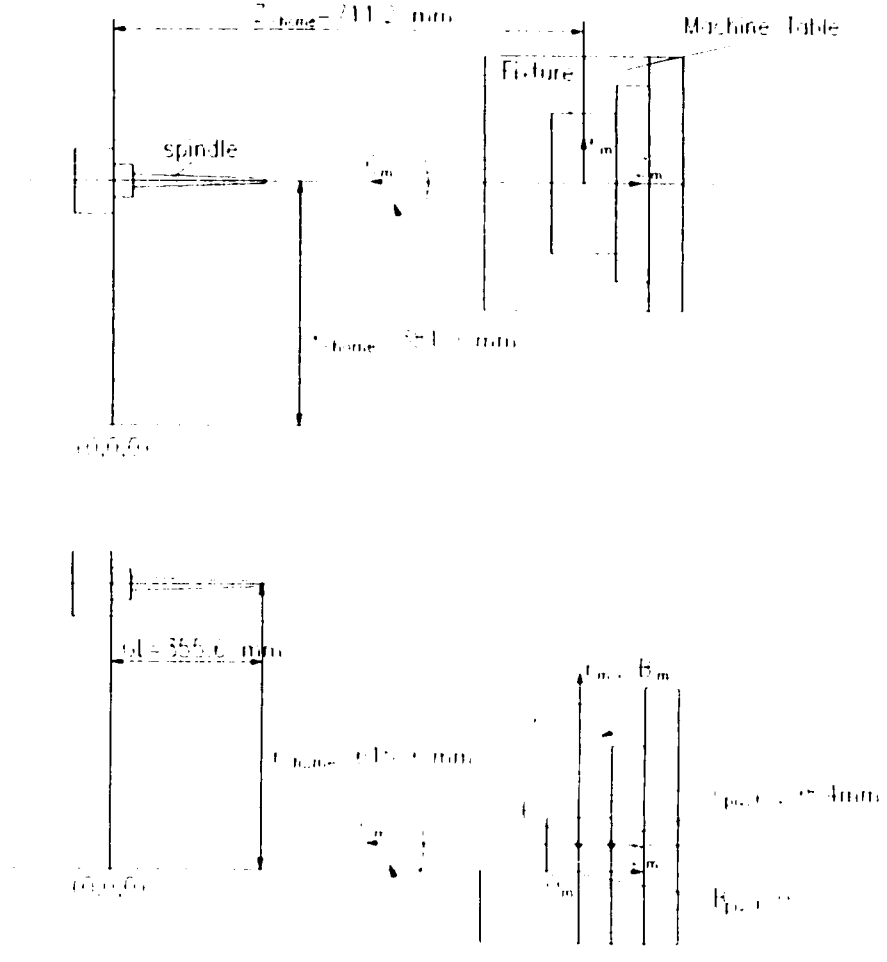


Figure 4.4 The Home Position of the OM-1 CNC Milling Centre

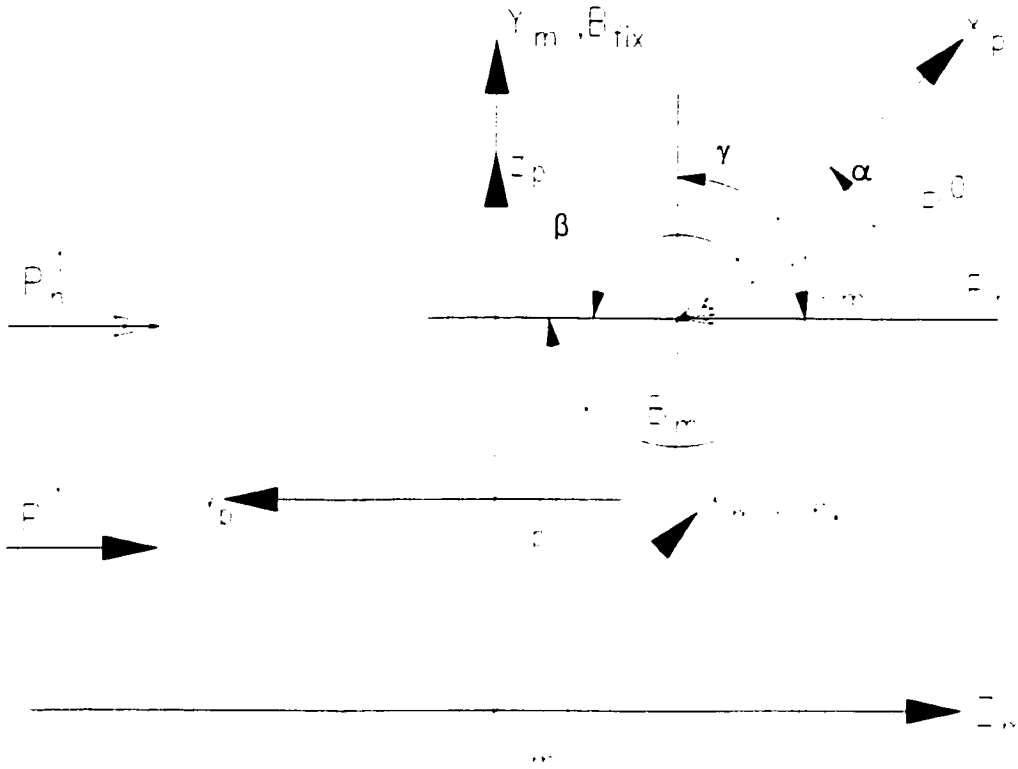
in Fig. 4.4. For the machine setup, the stack value is the distance of the workpiece coordinate system origin to the face of the fixture. The fixture thickness is set as F = 152.4 mm (6") and the stack value, in this case, is G = 0.0. Therefore, the workpiece coordinate system position in the machine coordinate system is known. The machine translational movements coordinates are measured relative to the machine zeros (0, 0, 0). The rotational angle about the B_m axis is measured from the positive z_m axis direction and the rotational angle about the C_m axis is measured from the positive y_m axis direction. The points that represent the machine variables coordinates usually are referred to as the Machine Control Point (MCP). In the case of the OM-1 milling centre, since the machine table and the spindle translate simultaneously during the machining process, the machine coordinate system origin and the spindle chuck are the MCP, as MCP-o and MCP-s respectively.

Modelling machine kinematic models, geometrical approaches use the geometrical configuration of individual machine tools and the geometrical intuition of the machining movements to determine the inverse and forward kinematic models. In this research work, based on the known OM-1 CNC milling centre configuration and the machining set-up data, a geometric approach is applied to obtain the inverse kinematic model. The forward kinematic model is then developed on the basis of the inverse kinematic model.

The procedure for deriving the inverse kinematic model is approached in two steps. In the first step, the machine rotational variables are determined by considering only the cutter orientation change that can be attained from the machine rotational movements. In the second

step, the machine translational variables are determined from the cutter position change that can be attained from both machine rotational and translational movements.

A frame is a coordinate system, where in addition to the orientation of the coordinate axes a position vector is given for locating the coordinate system origin to some other embedding coordinate system [68]. The machine coordinate system of the OM-1 milling centre forms a moving frame (hereafter referred to as the machine coordinate frame) since the machine coordinate system origin O_m moves relative to the machine zeros, and the orientations of the five axes vary during the machining process. The workpiece coordinate system with its coordinate axes and the origin in reference to the machine coordinate frame forms a frame (hereafter, referred to as the 'workpiece coordinate frame'). To formulate the machine rotational movements as functions of cutter orientations in a generalized machine setup, without loss of the generality, it is assumed that the workpiece coordinate frame is set at the position of the machine coordinate frame as shown in Fig.4.5, and the cutter vector, P^o, is a free vector in reference to the workpiece coordinate frame. A free vector refers to a vector which may be positioned anywhere in space without loss or change of meaning provided that magnitude and direction are preserved [68]. For generating tool path, the cutter vector is considered having unit magnitude and the direction cosines of the vector in the space of the workpiece coordinate frame. During the machining process, the workpiece coordinate frame translates and rotates simultaneously with the machine table. Hence, the cutter vector with the workpiece coordinate frame can be considered as a solid part.



(O_p, x_p, y_p, z_p) --- the part coordinate system
(O_m, x_m, y_m, z_m, B_m, C_m) --- the machine coordinate system
B_{fix} --- the initial position of the B_m axis
C_{fix} --- the initial position of the C_m axis
P^0 --- the initial location of the cutter vector with orientation angles (α,β,γ)
$P^0{}_h$, $P^1{}_h$ --- the cutter vector location after rotations
P^1 --- the spindle position

Figure 4.5 Schematics of Rotation Transformation

The machine rotational variables are the angles of the machine table together with the workpiece coordinate frame rotated about the machine moving axes B_{move} and C_{move}. To determine the machine rotational variables, one can pursue the solution by using the method of the spatial descriptions and transformations. Rotations can be defined as operators to transform a frame or a vector from one location to another [68]. Hence, a rotation matrix that represents the machine rotational movements can be used to operate on the workpiece coordinate frame or/and the cutter vector from one location to another. Since rotations can also be thought as descriptions of orientations, the rotational matrix can be determined by describing the orientation of the workpiece coordinate frame with the cutter vector. In representing the orientation of a frame, the angle sets convention includes fixed angle sets and Euler angle sets [68]. Fixed angle sets represent rotations specified about the fixed reference frame, while Euler angle sets describe the rotations relative to the axes of a moving frame. The duality of the fixed and Euler angle sets states that 'rotations taken about the fixed axes yield the same final orientation as the same rotations taken in opposite order about the axes of the moving frame' [68]. Based on this theory, the rotations about the moving machine coordinate frame can be obtained from the rotations about a fixed frame. Using the initial machining positions and orientations of the machine rotational axes as the fixed frame with the axes, B_{fix} and C_{fix} as shown in Fig.4.5, the rotational matrix can be determined as follows:

1) The machine C_m axis rotational movement rotates the workpiece coordinate frame and the cutter vector P^o about the C_{fix} axis to a new position which is on a horizontal plane as shown in Fig.4.5. The mathematical expression is:

$$P_h^{\,o} = Rot(C_{fix}\ ,\ C) * P^{\,o} \qquad (4.1)$$

where, P^o is the initial position of the cutter vector. $P_h^{\,o}$ is the position of the cutter vector after the rotation. Rot (C_{fix} , C) means a rotation about the fixed C axis and the rotation angle is C.

2) The machine table rotates the workpiece coordinate frame and the cutter vector P^o_h about the B_{fix} axis so that the cutter vector is transformed to a new position, P^1_h , which is parallel to the spindle vector P^1 (see Fig. 4.5), one may obtain:

$$P_h^{\,1} = Rot(B_{fix}\ ,\ -B) * P_h^{\,o} \qquad (4.2)$$

where, $P_h^{\,o}$ is the cutter vector before the rotation. $P_h^{\,1}$ is the cutter vector after the rotation. Rot (B_{fix} , -B) is the rotation about the fixed B axis, and the rotated angle is B in counterclockwise direction. Therefore, the rotations of the workpiece coordinate frame and the cutter vector from the initial position of P^o to the final position of $P_h^{\,1}$ is:

$$P_h^{\,1} = Rot(B_{fix}\ ,\ -B) * Rot(C_{fix}\ ,\ C) * P^{\,o}$$

$$P_h^{\,1} = R * P^{\,o} \qquad (4.3)$$

where, R is the rotation transformation matrix that is the product of the rotation matrices in the order of first about the C_{fix} axis then about the B_{fix} axis. Based on the

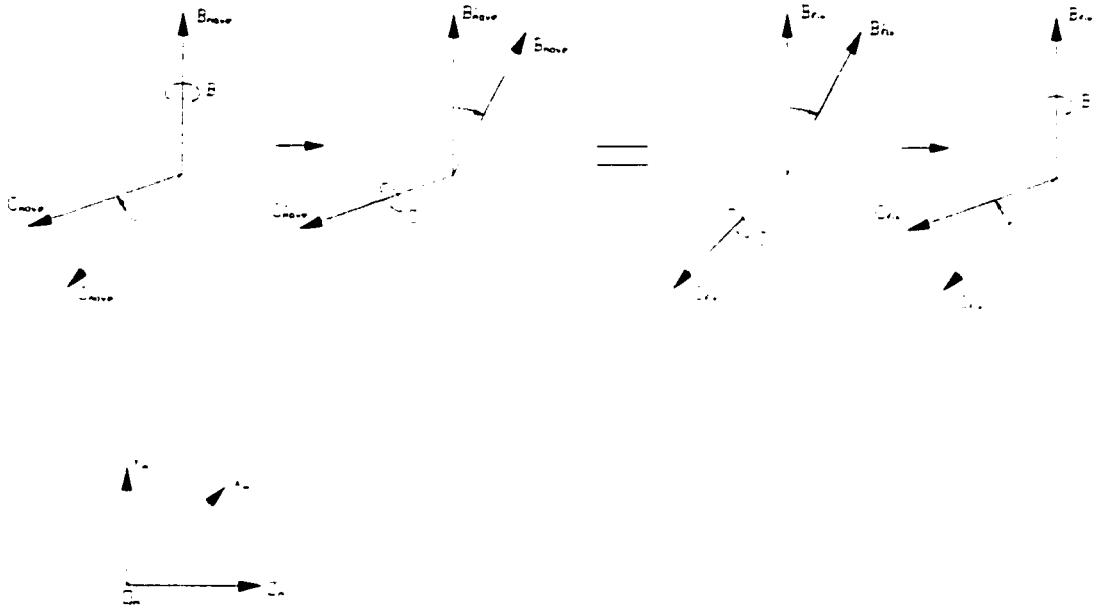
theory of the angle sets duality mentioned above, the rotation matrix about the actual machine moving axes B_{move} and C_{move} can be determined. As shown in Fig.4.6, the rotations taken about the moving axes of B_{move} and C_{move} are equal to the rotations taken about the fixed axes of C_{fix} and B_{fix} in opposite order as follows:

$$\begin{aligned} R &= Rot(C_{move}, C)\ Rot(B_{move}, -B) \\ &= Rot(B_{fix}, -B)\ Rot(C_{fix}, C) \end{aligned} \tag{4.4}$$

The rotation matrix Rot (C_{fix}, C) and Rot(B_{fix}, -B) can be determined as given in Appendix A. The mathematical representation of the rotation matrix R is:

$$R = \begin{bmatrix} cosB & -sinB\ .sinC & -sinB\ .cosC \\ -sinB & -cosB\ .sinC & -cosB\ .cosC \\ 0 & cosC & -sinC \end{bmatrix} \tag{4.5}$$

From Eq.(4.3), the cutter vector initially is at P^o with the orientation angle (α, β, γ) in reference to the workpiece coordinate frame and, after the rotations, the cutter vector is at P_h^I which is parallel to the machine spindle. Since the machine spindle has orientation of (cos90°, cos180° , cos90°) in reference to the workpiece coordinate frame, using the rotational matrix R as an operator, one may obtain:



C_{move} – machine moving C_m axis
B_{move} – machine moving B_m axis
C_{fix} – assumed fixed C_m axis
B_{fix} – assumed fixed B_m axis

Figure 4.6 The Rotations about the Moving and the Fixed Frames

$$P_h^{\,1}(\alpha,\beta,\gamma) = R * P^{o}(\alpha,\beta,\gamma)$$

$$\begin{bmatrix} \cos 90^{0} \\ \cos 180^{0} \\ \cos 90^{o} \end{bmatrix} = R * \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix} \qquad (4.6)$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} cosB & -sinB\,.sinC & -sinB\,.cosC \\ -sinB & -cosB\,.sinC & -cosB\,.cosC \\ 0 & cosC & -sinC \end{bmatrix} * \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix}$$

By solving the matrix equation (4.6), the machine rotational variables B_m and C_m are determined.

In the second step, considering the cutter vector initially at P^o (x, y, z) in reference to the initial workpiece coordinate frame, the cutter vector after the rotations is arrived at the position of $P_h^{\,1}$ (x_h, y_h, z_h) in reference to the initial workpiece coordinate frame, one may obtain:

$$P_h^{\,1}(x_h, y_h, z_h) = R * P^{o}(x, y, z) \qquad (4.7)$$

Since the workpiece coordinate frame is also rotated together with the machine table, its origin is moved to a new position referred to as:

$$\begin{bmatrix} O_x^1 \\ O_y^1 \\ O_z^1 \end{bmatrix} = R * \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} \qquad (4.8)$$

where, O represents the origin of the workpiece coordinate frame at the initial position. O^1 represents the origin at the new position. The subscripts x, y and z represents the coordinate of the origin in reference to the machine coordinate frame. Therefore, by the rotational movements, the new position of the cutter vector in reference to the fixed machine coordinate frame is:

$$\begin{aligned} x^1 &= x_h \pm O_x^1 \\ y^1 &= y_h \pm O_y^1 \\ z^1 &= z_h \pm O_z^1 \end{aligned} \qquad (4.9)$$

where, (x^1, y^1, z^1) is the position coordinate of the cutter vector P_h^1. The plus and the minus sign depends on the position of the origin of the workpiece coordinate frame in reference to the machine coordinate frame. This cutter vector's new position is equivalent to that caused by the machine rotational movements about the moving axes.

The cutter position change due to the machine translational movements are determined as follows: since the rotated cutter vector, P_h^1, is parallel to the machine spindle

and the position change of P_h^1 to the spindle can be attained only by the machine translational movements, the translational movements can be determined by moving P_h^1 to the spindle position. The spindle initial position relative to the fixed machine coordinate frame is at (x_m, y_m , z_m) = (PB, PC, 15.0") (where, PB represents the spindle position and is equal to the position x_m coordinate of the B_m axis pivot when the machine is at home position; PC represents the initial position of the spindle and is equal to the position of the C_m axis pivot in the y_m axis direction). The cutter vector, P_h^1, relative to the fixed machine coordinate frame has the coordinate of (x^1, y^1, z^1). Therefore, by translating the cutter vector P_h^1 to the spindle position, the machine translational variables are:

$$\begin{aligned} x_m &= PB \pm x^1 \\ y_m &= PC \pm z^1 \\ z_m &= PB \pm y^1 \end{aligned} \qquad (4.10)$$

where, the plus and the minus signs depend on the relative position of the P_h^1 and the spindle. In our case, the minus applies. Since the z axis of the workpiece coordinate frame is parallel to the y_m axis of the machine coordinate frame, the machine translational variable y_m is calculated from the z^1. The same condition applies to the machine variable z_m. The detailed calculation procedure is outlined in Appendix A, and the derived inverse kinematic model for the OM-1 five-axis CNC milling centre is as shown in the following equation:

$$x_m = PB - x\sqrt{1-\cos^2\alpha} + y\frac{\cos\alpha\cos\beta}{\sqrt{1-\cos^2\alpha}} + z\frac{\cos\alpha\cos\gamma}{\sqrt{1-\cos^2\alpha}} - (\frac{F}{2}+G)\sqrt{1-\cos^2\alpha}$$

$$y_m = PC - y\frac{\cos\gamma}{\sqrt{1-\cos^2\alpha}} - z\frac{\cos\beta}{\sqrt{1-\cos^2\alpha}}$$

$$z_m = PB + x\cos\alpha + y\cos\beta + z\cos\gamma + (\frac{F}{2}+G)\cos\alpha \qquad (4.11)$$

$$B_m = \tan^{-1}(\frac{\cos\alpha}{\sqrt{\cos\beta^2 + \cos\gamma^2}})$$

$$C_m = \tan^{-1}(\frac{\cos\beta}{\cos\gamma})$$

where, x, y, z are cutter position coordinates and $\cos\alpha$, $\cos\beta$, $\cos\gamma$ are the direction cosines of the cutter orientation coordinates in reference to the workpiece coordinate system. x_m, y_m, z_m are machine translational movement variables and B_m and C_m are machine rotational movement variables in reference to the machine coordinate system. F is the fixture thickness and G is the workpiece stacking position data.

The forward kinematic model is derived as follows. From Eq.(4. 6), one may obtain:

$$P^{o}(\cos(\alpha), \cos(\beta), \cos(\gamma)) = R^{-1} * P_h^{1}(\cos(90^{o}), \cos(180^{o}), \cos(90^{o})) \qquad (4.12)$$

Where, R^{-1} is the inverse of the rotation transformation matrix of Eq. (4. 5). From Eq.(4. 7), one may obtain:

$$P^{o}(x\ ,\ y\ ,\ z)\ =\ R^{-1}*\ P_{h}^{1}(x_{h}\ ,\ y_{h}\ ,\ z_{h}) \tag{4.13}$$

From Eq.(4.9) and Eq.(4.10), one may obtain:

$$\begin{aligned} x_{h} &= PB \mp O_{x}^{1} - x_{m} \\ y_{h} &= PB \mp O_{y}^{1} - z_{m} \\ z_{h} &= PC \mp O_{z}^{1} - y_{m} \end{aligned} \tag{4.14}$$

By substituting Eq.(4.14) into Eq.(4.13) and solving Eq.(4.12), the forward kinematic model is obtained:

$$\begin{aligned} x &= -(x_{m}-PB)cosB_{m}+(z_{m}-PB)sinB_{m}+(\frac{F}{2}+G) \\ y &= x_{m}\ sinB_{m}sinC_{m}+z_{m}\ cosB_{m}sinC_{m}+\ (PC-y_{m})\ cosC_{m} \\ &\quad -PB\ \ (sinB_{m}\ +\ cosB_{m})\ sinC_{m} \\ z &= x_{m}\ sinB_{m}cosC_{m}\ +\ z_{m}\ cosB_{m}cosC_{m}+\ (y_{m}-PC)\ sinC_{m} \\ &\quad -PB\ \ (sinB_{m}\ +\ cosB_{m})\ cosC_{m} \\ \cos\alpha &= sinB_{m} \\ \cos\beta &= cosB_{m}\ sinC_{m} \\ \cos\gamma &= cosB_{m}\ cosC_{m} \end{aligned} \tag{4.15}$$

The derivation procedure of the forward kinematic model is given in Appendix A.

The kinematic models of the OM-1 milling centre were verified as follows. The existing postprocessors, such as the AIGP mentioned above, have the function of performing

kinematic transformations between cutter variables and machine variables for a known machine configuration. Using the AIGP, a set of CLDATA for machining the airfoil surfaces of an impeller was transformed to the corresponding NC-codes. Then, the set of CLDATA was processed through the derived inverse kinematic model. The obtained machine variables are then compared with the NC-codes obtained from the AIGP. A comparison of results showed that the calculated machine variables (x_m , y_m , z_m , B_m , C_m) are exactly the same (up to five decimal digits) as the NC-codes computed by the AIGP. The result confirms the correctness of the derived inverse kinematic model. Similarly, the forward kinematic model is verified by transforming the set of NC-codes for machining the airfoil surfaces of the impeller into the corresponding CLDATA. The resultant CLDATA were the same as the set of data obtained from the AIGP.

4.4 Development of Machining Motion Trajectory Model

From the machine configuration shown in Fig.4.4 , the B_m axis coincides with the y_m axis which is perpendicular to the machine table, and the C_m axis is parallel to the z_m axis during machining process. Therefore, the B_m axis intersects the C_m axis at a space point, P. During the machining process, the intersection point P moves translationally only in the x_m axis and the z_m axis directions since the point is on both of the rotational axes. The point P also acts as a pivot for the machining spatial rotational movements. Thus, the CC point path is actually the result of the superimposition of the rotational movements of the machine table about the pivot P and the translational movements of the pivot P, which results in the combined 3D circular interpolated and the 3D linear interpolated path. From this geometrical perspective,

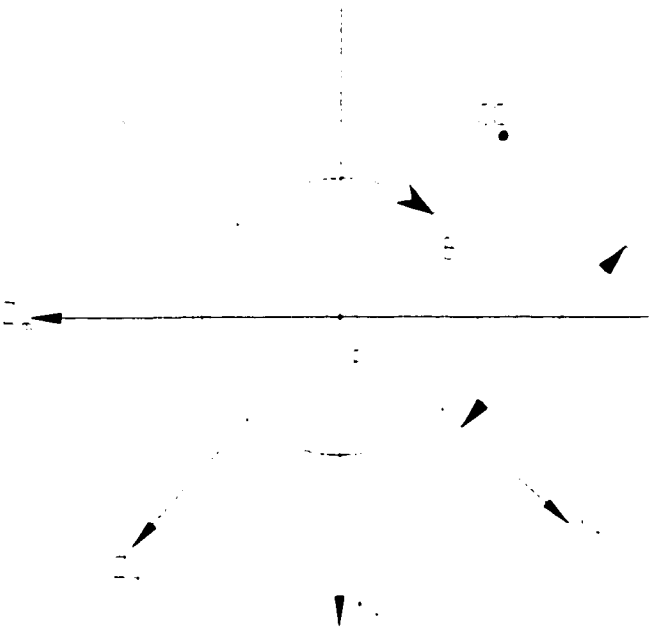
it is known that the CC point coordinate on the non-linear segment can be determined from the pivot P translational coordinates and the machine table rotational movements coordinates. Thus, the machining motion path can be formulated on the basis of the pivot translational movements and the machine rotational movements.

The machine home position is defined by the machine table coordinates (x_m, y_m, z_m, B_m, C_m) = (381.0 mm, 0.0 mm, 711.2 mm, 0°, 0°) = (15.0", 0", 28.0", 0°, 0°) and by the spindle coordinates (x_m, y_m, z_m) = (381.0 mm , 615.95 mm , 0.0 mm) = (15.0", 24.25", 0.0"). The machining start point is chosen as the machine home position. Since the B_m axis coincides with the y_m axis, the circular motion in the horizontal plane (the x_m-z_m plane) about the B_m axis changes the x_m and the z_m coordinates as shown in Fig.4.7, one may obtain:

$$\begin{aligned} x_{cc} &= l \sin(B_m) \\ z_{cc} &= l \cos(B_m) \end{aligned} \tag{4.16}$$

Similarly, because the C_m axis is parallel to the z_m axis, the circular motion in the vertical plane (the x_m-y_m plane) about the C_m axis changes the x_m and the y_m coordinates as shown in Fig. 4.8, one may obtain:

$$\begin{aligned} x_{cc} &= l \sin(C_m) \\ y_{cc} &= l \cos(C_m) \end{aligned} \tag{4.17}$$



$$x = l \sin B$$
$$z = l \cos B$$

Figure 4.7 Position Change due to Rotations in X-Z Plane

The rotation pivot coordinates, x_p and z_p, are equal to the x_m and z_m coordinates of the machine coordinate system origin since the B_m axis goes through the machine table top centre, and the rotation pivot y_p coordinate is the same as the pivot of the rotation C_m axis, C_{pivot}. Also, since the CC point is on both the cutter and the workpiece, the CC point coordinates that correspond to the NC-codes can be determined from the spindle chuck coordinates (x_{home}, y_m, 0) and the tool gage length (GL) as shown in Fig.4.9 (please refer to Fig.4.4 for definition of coordinate x_{home}, C_{pivot} and GL). Hence, the distance between the CC point and the rotation pivot P is:

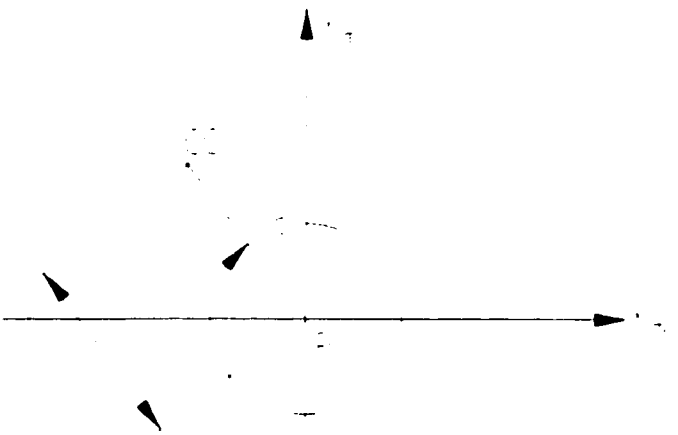
$$l = \sqrt{(x_m - x_{home})^2 + (y_m - C_{pivot})^2 + (z_m - GL)^2} \tag{4.18}$$

Thus, combining the linear motion of the pivot P and the rotary movements about the pivot P, the OM-1 machine motion trajectory model can be constructed as :

$$\begin{aligned} x_{cc} &= x_p + l * \sin(B_m) * \sin(C_m) \\ y_{cc} &= y_p + l * \cos(C_m) \\ z_{cc} &= z_p + l * \cos(B_m) \end{aligned} \tag{4.19}$$

with:

$$\begin{aligned} x_p &= x_m \\ y_p &= C_{pivot} \\ z_p &= z_m \end{aligned} \tag{4.20}$$



$$x = l \sin C$$
$$y = l \cos C$$

Figure 4.8 Position Change due to Rotations in X-Y Plane

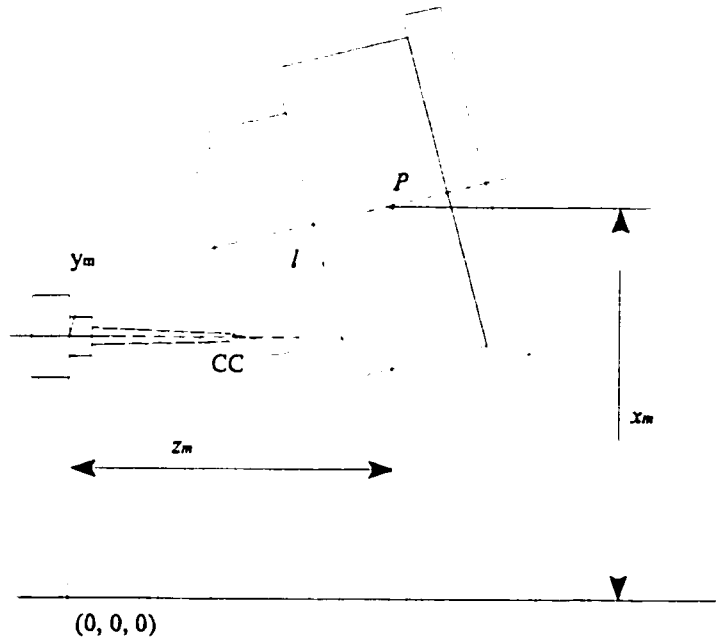


Figure 4.9 A Machining Position of the OM-1 Milling Centre

where, (x_{cc}, y_{cc}, z_{cc}) are the coordinates of the CC point, (x_p, y_P, z_p) are the coordinates of the rotation pivot P in reference to the machine coordinate system, l is the distance between the CC point and the pivot P of Eq.(4.18), B_m and C_m are the rotational movements of the machine table, and C_{pivot} is the C_m axis pivot coordinate which is a constant.

4.5 Minimum Error Tool Path Generation Methodology

A methodology to determine the optimum CLDATA for minimum machining error based on machine kinematics and machine motion trajectory analysis is explored. The machine kinematic models of Eq.(4.11) and Eq.(4.15) give the OM-1 machining geometrical relations between the cutter orientations and the machine rotational variables. The machine rotational movements are kinematically related to the spatially changed cutter orientations, and therefore the non-linearity errors depend upon the cutter orientation changes. The OM-1 machining motion trajectory model of Eq.(4.19) mathematically shows that the non-linear path curve segments depend upon machine rotational variables for the determined translational steps. Hence, non-linearity errors are a result of the machine 3D rotational movements. Also, by using the motion trajectory model, the coordinates on each non-linear tool path segment can be determined, which in turn, can be used with known linearity errors to determine the non-linearity errors. Particularly, the maximum non-linearity error on each segment can be determined. In order to calculate the linearity errors, the surface curvature at the machining data points must be known. To determine the geometrical properties, such as the local surface curvature, the cutting curve mathematical function is required. The cubic spline curve function is the minimum order function that can ensure the second order continuity [69]. Therefore,

the cutting curve can be closely approximated by the cubic spline of the known CLDATA. The development of the cubic spline functions to represent cutting curves is presented in Appendix B.

The proposed methodology reduces non-linearity errors by determining the acceptable machining rotational variables using the motion trajectory model, and also modifies the cutter orientations through the kinematic relations with the machine rotational variables. For a specified machining tolerance, the machine rotary movements which will result in an acceptable machining motion trajectory can therefore be determined. Then, the optimum cutter orientations which correspond to the obtained rotational variables can be determined. The procedure starts with the transformation of the CLDATA to the machining NC-codes by employing the inverse kinematic model of Eq.(4.11). The actual machining motion trajectory is then determined by using the OM-1 motion trajectory model of Eq.(4.19). The local surface curvature on the cutting curve and the step-forward distance, i.e., the linear spacing between a pair of the cutting points, determine the amount of linearity error. From the cubic spline cutting curve function, the local surface curvature can be determined (refer to Appendix B). Thus, the linearity error for each move can be obtained from the adjacent CC point data and the determined curvature. The allowable non-linearity error can be computed by subtracting the linearity error from the specified machining tolerance. The actual non-linearity error for each move is the deviation of the non-linear motion trajectory from the linear interpolated line segment. Representing the linear segment using linear interpolation function based on the CC point coordinates determined from the OM-1 motion trajectory model given in Eq.(4.19), the

maximum deviation between the non-linear motion trajectory and the linearly interpolated line segment, i.e., the maximum non-linearity error, for the step can be determined. In the steps where the maximum non-linearity error exceed the allowable non-linearity error, the machine rotational movements change will be modified. The new machine rotational movement variables are determined based on the modified rotational movement change. These new rotational movement variables are then used to calculate the resultant non-linearity error, which in turn is compared again with the allowable non-linearity error. Thus, by using the difference between the allowable non-linearity error and the modified non-linearity error as the criterion, the acceptable machine rotational movements can be determined iteratively. Finally, from the modified machine rotational movements, the corresponding cutter orientations can be determined by performing the forward kinematic transformation using Eq.(4.15).

It must be emphasized that the machining geometrical properties and the motion trajectory are machine dependent. Hence, the modification of CLDATA has to be carried out in terms of machining variables and subsequent use of the kinematic transformation to determine the modified CLDATA. For a set of CLDATA, the modification procedure can be performed by using the following algorithm.

Tool Path Modification Algorithm:

1) Transform the initial CLDATA into its corresponding machining NC-codes by using the inverse kinematic model for a particular CNC machine tool (for OM-1 see Eq. (4.11));

2) Determine the CC point coordinates by employing the machine dependent motion trajectory model (for OM-1 see Eq.(4.19));

3) Compute the desired tool path by cubic spline the CLDATA and calculate the curvatures k_f of the tool path at the CLDATA points;

4) Compute the linearity error by using the formula given by Faux and Pratt [70], based on the CC point coordinates from step (2) and the surface curvatures from step (3):

$$\delta_t = \frac{1}{8} k_f (\Delta s)^2 \tag{4.21}$$

where, k_f is the surface curvature at the CLDATA points, and Δs is the segment length between a pair of adjacent CC points.

5) Compute the allowable value of the non-linearity error: $\delta_{a,n}$ = tolerance -δ_t.

6) Determine the points on the straight line segment and on the machine motion trajectory segment that correspond to the maximum non-linearity error;

7) Compute the maximum non-linearity error, δ_{max}, using the points from step (6);

8) Modify the machine rotational angle change if $\delta_{max} > \delta_{a,n}$, i.e., increase or decrease ΔB_m and ΔC_m such that the non-linearity error, δ_{nl}, will satisfy $(\delta_{nl} - \delta_{a,n}) < 0$;

9) Compute the machining NC-codes of $B_{m,i+1}$ and $C_{m,i+1}$ based on the machine rotational angle variation from step (8):

$$\begin{aligned} B_{m,i+1} &= B_{m,i} \pm \Delta B_m \\ C_{m,i+1} &= C_{m,i} \pm \Delta C_m \end{aligned} \tag{4.22}$$

where, ($B_{m,i}$, $C_{m,i}$) are the i-th rotational variables, ($B_{m,i+1}$, $C_{m,i+1}$) are the (i+1)-th rotational variables, and (ΔB_m ,ΔC_m) are the modified rotation angle change. The ' + ' and ' - ' sign depends on if the angles in step (i+1) increases or decreases.

10) Determine the optimum cutter orientations by transforming the modified machining NC-codes using the forward kinematic model (for OM-1, use Eq.(4.15)).

4.6 Software for Implementing the Algorithm

A software program has been written to implement the proposed algorithm described above. The computation procedure is shown in the program flow chart shown in Fig.4.10. The program starts by inputting the known CLDATA and machining parameters, which include machining tolerance, fixture thickness and stack value. These machining parameters can be changed for different machining requirements and machine setup. The program includes a function for constructing the tool path curve based on the CLDATA by using the cubic spline technique, so that the curvatures at the cutter positions can be calculated. The program consists of an inverse kinematic transformation function of the CLDATA generating the machining NC-codes, and a function of OM-1 motion trajectory determining the CC point coordinates. For each step, based on the determined curvature and the CC point coordinate, the linearity error can be calculated. The allowable non-linearity error is determined from the linearity error and the specified machining tolerance. Considering that the maximum non-linearity error usually occurs at the middle point of each linear line segment, the program determines the points on the linear line segment and on the motion trajectory segment, which yields the maximum non-linearity error. By testing whether a non-linearity error exceeds the

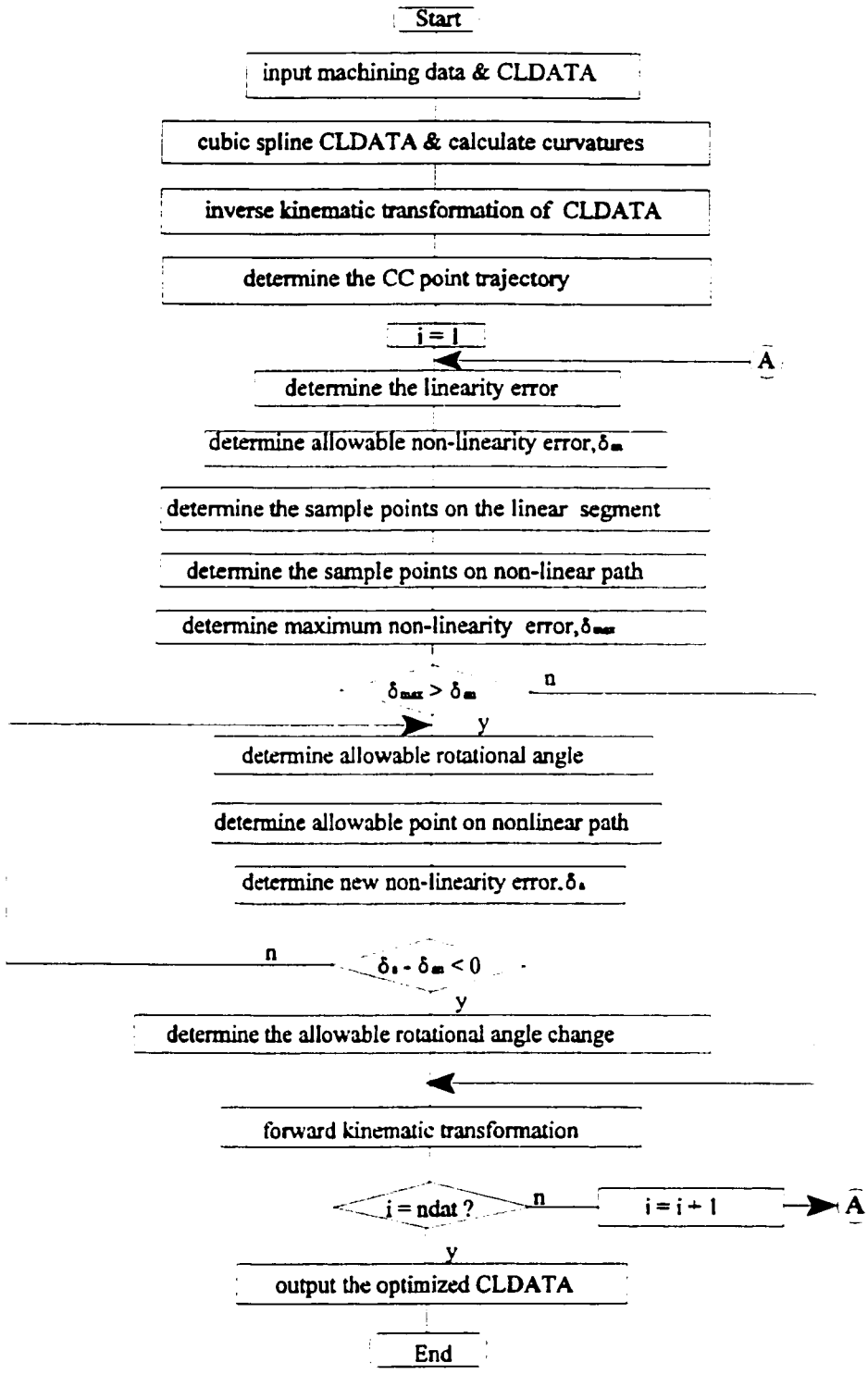


Figure 4.10 Flow Chart of the Minimum Error Tool Path Generation Method Software

allowable non-linearity error determined previously, a procedure for determining the acceptable rotational angle change is conducted iteratively as follows. Once the test of whether the actual non-linearity error exceeds the allowable value is true, the machine rotational angles' changes, ΔB_m and ΔC_m, are adjusted in each iteration and the corresponding machine rotational variables B_m and C_m are calculated. The adjusted B_m and C_m are then used to determine the resultant non-linearity error which in turn is compared with the allowable value. The iteration is conducted until the final non-linearity error is equal to or less than the allowable value, and the corresponding B_m and C_m which are the desired machine rotational variables can be determined. Repeating the computation for each step, the whole set of NC-codes can be processed. For the steps, the non-linearity error is smaller than the allowable value, the NC-codes remain unchanged. Finally, the NC-codes including the modified points and the remaining points are transformed into the CLDATA.

4.7 Summary

An off-line tool path generation methodology, which modifies the cutter orientations to reduce the machining errors, has been proposed. The methodology employs the machine type-specific kinematic models and the machining motion trajectory model. The new method reduces non-linearity errors to be within the specified tolerance range by calculating the allowable machine rotary angle changes. The optimum cutter orientations are then determined based on these allowable machine rotary angle changes. The new method improves the machining precision with smoother machining data distribution because the modified CLDATA contains the original cutter positions. Comparing with the existing methods, the

'linearization processes' insert cutter positions to reduce non-linearity errors which results in the feedrate fidelity problem and increases machining time. Alternately, the new method ensures the machining precision by tackling the cause of the non-linearity errors problem, the cutter orientation changes, without undesired consequences.

As the result, the software for implementing the proposed methodology can be used to process CLDATA specifically suitable to the OM-1 milling centre, and to generate NC-codes for the rotary table type five-axis CNC machine tools by changing the configuration data.

CHAPTER 5

AN APPLICATION OF THE

MINIMUM ERROR TOOL PATH GENERATION METHOD

5.1 Introduction

The highly twisted airfoil of a turbine impeller is a typical example of sculptured surfaces. For machining such complex surfaces, five-axis CNC machining with conical shaped ball-mills is usually used because of the highly twisted airfoil surfaces and the narrow passages between the airfoils. The CLDATA generation technique for machining of such impeller airfoil surfaces was described in section 4.2. It determines the cutter positions and its axis vector orientations based solely on the geometrical properties of the machined surfaces. As an application of the proposed 'minimum error tool path generation method', a set of CLDATA for the machining of the impeller airfoil surface is processed and the optimum CLDATA with minimum machining errors is obtained. In the following, a simulation study for processing the set of CLDATA by using the existing 'linearization process' is described in section 5.2. The procedure and results for processing the CLDATA by using the proposed methodology are outlined in section 5.3.

5.2 Case Study Using the 'Linearization Process '

In order to compare the optimized results with those obtained through the 'linearization process' mentioned in section 2.2, the set of CLDATA is processed by the extensively used postprocessor, the AIGP[31]. The AIGP generates machining NC-codes using the

'linearization process' as follows:

1) Transforms a pair of adjacent CLDATA points into corresponding machine NC-codes coordinates;

2) Constructs a straight line segment using the two NC-code position coordinates;

3) Divides the straight line segment and the corresponding rotational movements angle change into two portions;

4) Calculates the middle point NC-code coordinate based on the result of step (3);

5) Transforms the calculated new NC-code into the corresponding new CLDATA;

6) Tests the deviation of the new CLDATA point from the straight line constructed between the two considered CLDATA;

7) Goes to step 8) if the considered CLDATA are the original data, otherwise, goes to step 10).

8) Inserts a middle point between the two initial CLDATA if the deviation is greater than the specified machining tolerance;

9) Repeats the process from step 1) to step 6) using the start CLDATA point and the inserted median point of CLDATA;

10) Inserts another intermediate point between the considered two CLDATA points if the new deviation exceeds the specified tolerance;

11) Repeats the process from step 1) until either all points are within the machining tolerance range or a maximum of 63 points have been generated;

12) Repeats the steps 1) to 11) to the next adjacent original CLDATA pairs until all data points are processed.

The results for processing the CLDATA of machining the airfoil surfaces using AIGP's 'linearization process' are as follows.

With the specified machining tolerance of $12.7x10^{-3}$ mm ($0.5"x10^{-3}$), there was no additional data inserted between any adjacent machining data points. However, the 'linearization process' inserted data points when the machining tolerance was reduced. With the tolerance of $7.62x10^{-3}$ mm ($0.3"x10^{-3}$), one data point was inserted in the first motion step (between the first two original CLDATA), in the third motion step (between the third and the fourth original CLDATA), and in the ninth motion step (between the ninth and the tenth original CLDATA). Furthermore, by reducing the tolerance to $5.08x10^{-3}$ mm ($0.2"x10^{-3}$), more data points were inserted. Table 5.1 shows 25 sets of inserted data (shown as underlined points) among the complete set of 255 machining data points for the highest attainable machining precision of $5.08x10^{-3}$ mm ($0.2" \times 10^{-3}$). The resultant tool path data are more dense than the original data. Furthermore, the cutter position points are not equally spaced and the cutter orientation varies incorrectly.

5.3 Case Study Using the 'Minimum Error Tool Path Generation Method'

Using the proposed methodology, the set of CLDATA was also processed with the same machining parameters as used for the AIGP. With the specified machining tolerance of $12.7x10^{-3}$ mm ($0.5"x10^{-3}$), the maximum non-linearity error calculated from the NC-codes were all smaller than the allowable non-linearity error. Hence, there was no need to conduct the cutter orientation modifications. This result is the same as the one from the AIGP

Table 5.1 Sample NC-codes from the AIGP's Linearization Process

n	x_m	y_m	z_m	B_m	C_m
1	18.5829	12.4387	14.4904	353.571	359.271
	18.5720	12.4224	14.5123	353.752	359.112
	18.5610	12.4060	14.5340	353.933	358.960
	18.5499	12.3896	14.5556	353.113	359.812
2	18.5386	12.3731	14.5770	354.295	358.660
3	18.5099	12.3366	14.6190	354.610	358.668
	18.4882	12.3093	14.6505	354.849	358.671
4	18.4662	12.2820	14.6818	355.088	358.688
9	18.3979	12.1049	14.6199	352.548	357.906
	18.3822	12.0824	14.6421	352.761	357.672
10	18.3662	12.0597	14.6639	352.972	357.443
19	18.1932	11.8937	14.8453	355.036	357.101
	18.1833	11.8843	14.8542	355.143	357.071
20	18.1735	11.8748	14.8629	355.250	357.073
73	16.8410	11.5210	15.6696	8.346	19.523
	16.8258	11.5223	15.6769	8.495	19.841
74	16.8107	11.5238	15.6841	8.644	20.174
82	16.5579	11.5384	15.7981	11.204	24.770
	16.5420	11.5394	15.8048	11.359	25.051
83	16.5261	11.5405	15.8113	11.513	25.325
103	15.8425	11.5423	16.0427	18.365	33.451
	15.8246	11.5421	16.0475	18.543	33.622
104	15.8067	11.5421	16.0521	18.721	33.795
	15.7888	11.5420	16.0568	18.898	33.963
105	15.7709	11.5420	16.0613	19.074	34.128
122	15.5142	11.5397	16.1199	21.644	36.253
	15.4958	11.5398	16.1237	21.825	36.401
123	15.4774	11.5398	16.1273	22.005	36.551
130	15.2132	11.5373	16.1740	24.658	38.405
	15.1942	11.5372	16.1770	24.849	38.530
131	15.1751	11.5370	16.1798	25.040	38.657
133	15.0981	11.5360	16.1909	25.819	39.125
	15.0789	11.5358	16.1936	26.013	39.245
134	15.0597	11.5357	16.1961	26.207	39.361
139	14.8676	11.5355	16.2197	28.133	40.512
	14.8483	11.5355	16.2219	28.307	40.625
140	14.8289	11.5355	16.2239	28.500	40.726

Table 5.1 Sample NC-codes from the AIGP's Linearization Process (Con't)

n	x_m	y_m	z_m	B_m	C_m
145	14.6340	11.5368	16.2424	30.451	41.790
	14.6146	11.5372	16.2441	30.645	41.900
146	14.5951	11.5377	16.2458	30.838	42.000
147	14.5562	11.5387	16.2487	31.223	42.220
	14.5367	11.5393	16.2503	31.415	42.335
148	14.5173	11.5400	16.2517	31.606	42.440
152	14.3601	11.5462	16.2615	33.177	43.298
	14.3405	11.5470	16.2626	33.372	43.390
153	14.3209	11.5480	16.2636	33.566	43.500
154	14.2828	11.5498	16.2655	33.953	43.715
	14.2623	11.5506	16.2665	34.145	43.814
155	14.2429	11.5515	16.2673	34.337	43.911
166	13.8220	11.5546	16.2782	38.429	45.672
	13.8032	11.5545	16.2785	38.610	45.746
167	13.7843	11.5544	16.2786	38.791	45.813
178	13.3755	11.5577	16.2764	42.676	47.333
	13.3573	11.5578	16.2762	42.846	47.402
179	13.3392	11.5579	16.2758	43.016	47.460
187	13.0522	11.5525	16.2676	45.692	48.371
	13.0348	11.5519	16.2672	45.851	48.422
188	13.0173	11.5512	16.2666	46.009	48.479
195	12.7720	11.5409	16.2532	48.307	49.211
	12.7550	11.5412	16.2525	48.460	49.288
196	12.7379	11.5414	16.2516	48.613	49.351
198	12.6693	11.5427	16.2474	49.246	49.650
	12.6520	11.5429	16.2461	49.409	49.721
199	12.6347	11.5432	16.2446	49.572	49.801
	12.6173	11.5435	16.2431	49.739	49.875
200	12.6000	11.5437	16.2414	49.905	49.952
	12.5825	11.5437	16.2397	50.075	50.021
201	12.5651	11.5438	16.2378	50.244	50.098
254	11.5011	11.5970	16.0367	60.766	56.692
	11.4846	11.5987	16.0315	60.924	56.821
255	11.4716	11.6005	16.0264	61.082	56.951

'linearization process' mentioned above. By specifying the tolerance = 7.62×10^{-3} mm ($0.3" \times 10^{-3}$), the program processed the allowable machine rotational angle changes at some steps and modified the corresponding cutter orientations. For instance, for the first step and the third step, the non-linearity error exceeded the specified tolerance. Hence, the cutter orientations in the second block of CLDATA and the fourth block of CLDATA were modified. Furthermore, by tightening the machining tolerance to 5.08x10-3 mm ($0.2" \times 10^{-3}$), the allowable machine rotary angle changes were calculated iteratively at more motion steps. The cutter orientation modification process was conducted up to nine iterations for the first step until the non-linearity error was within the allowable range. Three iterations were performed for the third and the ninth steps. Similarly, the process was carried out for each motion step of the whole set of CLDATA. The acceptable machine rotational movements B_m and C_m were determined by adjusting the rotation angle changes: ΔB_m and ΔC_m, based on Eq.(4.22). The corresponding CLDATA were modified by transforming the processed NC-codes using the forward kinematic model. For the data points where the maximum non-linearity error calculated from the NC-codes were smaller than the allowable non-linearity error, the data remained unchanged.

Table 5.2 outlines some modified cutter orientation data and the corresponding NC-codes. The set (i , j , k) = ($\cos\alpha$, $\cos\beta$, $\cos\gamma$) are the direction cosines of the cutter axis vector relative to the workpiece coordinate system. For the first step (from point 1 to point 2), the adjusted rotary angle change ΔB_m was increased, and the adjusted ΔC_m was decreased. The modified direction cosines (i, j, k) result in the modified cutter orientation

Table 5.2 Sample Modified Rotary Angle Changes & Cutter Orientations

point	CLDATA		cutter angle		angle change		Bm, Cm		ΔBm, ΔCm	
N	original i, j, k	modified i, j, k	original α, β, γ	modified α, β, γ	original dα dβ dγ	modified dα dβ dγ	original Bm, Cm	modified Bm, Cm	original ΔBm, ΔCm	modified ΔBm, ΔCm
1	0.11196, -.01263, 0.99363	0.11196, -.01263, 0.99363	83.571°, 90.723°, 6.469°	83.571°, 90.723°, 6.469°	initial angles	initial angls	353.571° 359.271°	353.571° 359.271°	initial angles	initial angles
2	0.09940, -.02326, 0.00477	0.09787, -.02336, 0.99492	84.295°, 91.332°, 5.859°	84.383°, 91.338°, 5.774°	0.724°, 0.609°, -0.610°	0.812°, 0.615°, -0.695°	354.295° 358.660°	354.385° 358.656°	0.724°, -0.611°	0.814°, -0.615°
3	0.09399, -.02316, 0.99531	0.09399, -.02316, 0.99531	84.608°, 91.327°, 5.554°	84.608°, 91.327°, 5.554°	unchanged	unchanged	354.610° 358.668°	354.610° 358.668°	unchanged	unchanged
4	0.08562, -.02280, 0.99606	0.08513, -.02286, 0.99611	85.088°, 91.306°, 5.083°	85.116°, 91.309°, 5.057°	0.480°, -.0204°, -.470°	0.508°, -.017°, -.0496°	355.088° 358.688°	355.118° 358.687°	0.478°, 0.020°	0.508°, 0.018°
10	0.12234, -.04427, 0.99150	0.12273, -.04430, 0.99144	82.972°, 92.537°, 7.475°	82.950°, 92.539°, 7.502°	0.462°, 0.460°, -.269°	0.404°, 0.462°, -.242°	352.972°, 357.443°	352.853°, 357.543°	0.424°, -.234°	0.305°, -.363°
20	0.08281, -.05124, 0.99524	0.08268, -.05130, 0.99512	85.249°, 92.937°, 5.592°	85.257°, 92.941°, 5.663°	0.215°, 0.052°, -.157°	0.223°, 0.056°, -.086°	355.250°, 357.037°	355.163°, 357.068°	0.214°, -.028°	0.127°, -.033°
74	-.14515, 0.33060, 0.93254	-.14489, 0.33052, 0.93342	98.346°, 70.694°, 21.166°	98.330°, 70.699°, 21.025°	0.303°, -.628°, 0.711°	0.287°, -.623°, 0.570°	8.644°, 20.173°	8.525°, 19.976°	0.298°, 0.651°,	0.179°, 0.453°
83	-.19429, 0.41111, 0.89063	-.19593, 0.40763, 0.89138	101.203°, 65.725°, 27.047°	101.29° 65.943°, 26.952°	0.316°, -.510°, 0.622°	0.412°, -.292°, 0.527°	11.513°, 25.325°	11.495°, 25.152°	0.309°, 0.555°	0.291°, 0.382°
104	-.31507, 0.52325, 0.79178	-.31432, 0.52421, 0.79183	108.365°, 57.775°, 37.647°	108.31° 58.384°, 37.643°	0.359°, -.913°, 0.441°	0.313°, -.304°, 0.437°	18.721°, 33.795°	18.665°, 33.692°	0.356°, 0.344°	0.300°, 0.241°
105	-.32096, 0.52677, 0.78708	-.32178, 0.52125, 0.78691	108.721°, 58.213°, 38.086°	108.77° 58.583°, 38.102°	0.356°, 0.438°, 0.440°	0.452°, 0.199°, 0.459°	19.074°, 34.128°	18.898°, 33.968°	0.353°, 0.333°	0.177°, 0.173°
123	-.42941, 0.56704, 0.70290	-.43024, 0.56615, 0.70392	115.430°, 55.456°, 45.339°	115.48° 55.517°, 45.257°	0.390°, -.076°, 0.372°	0.442°, -.0150°, 0.290°	22.005°, 36.551°	21.925°, 36.491°	0.341°, 0.298°	0.281°, 0.238°

angle changes ($d\alpha$, $d\beta$, $d\gamma$) which were greater than the original cutter orientation angle changes. For the second step, since the maximum non-linearity error was smaller than the allowable value, there was no need to perform the cutter orientation modification and the direction cosines (i, j, k) at the third point of cutter orientation remained unchanged. For the third step (from point 3 to point 4), the adjusted rotary angle change ΔB_m was increased, and the adjusted ΔC_m was decreased. The cutter orientation (i, j, k) at the forth point were modified. The resultant cutter orientation angle changes of $d\alpha$ and $d\gamma$ were increased comparing with the original cutter orientation changes, and the cutter orientation angle change of $d\beta$ was decreased comparing with the original value. Similarly, the original and the modified cutter orientations (i, j, k) at points 10, 20, 74, 83, 104, 105 and 123 are shown in the table. The increases/decreases of the machine rotational movement angles of (ΔB_m, ΔC_m) and the corresponding cutter orientation changes of ($d\alpha$, $d\beta$, $d\gamma$) are due to the fact that the machining motion trajectory is constructed by combining the 3D circular and linear motions. Although the cutter orientation changes either increased or decreased, the non-linearity errors were reduced as shown in Table 5.3. From Table 5.3 the original non-linearity error for the step 1 was 8.788×10^{-3} mm ($0.346" \times 10^{-3}$). The non-linearity error after cutter orientation modification was reduced to 2.362×10^{-3} mm ($0.093" \times 10^{-3}$) which was smaller than the allowable non-linearity error of 2.514×10^{-3} mm ($0.099" \times 10^{-3}$). The reduced total machining error for this step was up to 56.5%. For step 3, with the modified cutter orientation data the non-linearity error was reduced from 5.486×10^{-3} mm ($0.216" \times 10^{-3}$) to 3.403×10^{-3} mm ($0.134" \times 10^{3}$), which was smaller than the allowable value of 3.784×10^{-3} mm ($0.149" \times 10^{-3}$). The reduced total machining for this step was up to 30.7%. Similarly, for steps at which

Table 5.3. Machining Errors at Sample Moves (tolerance = 0.005mm=0.0002")

Step\Error	linearity (inch x 10^{-3})	original non-linearity (inch x 10^{-3})	modified non-linearity (inch x 10^{-3})	reduced total (%)
1	0.101	0.346	0.093	56.5
3	0.051	0.216	0.134	30.7
9	0.005	0.210	0.185	11.6
19	0.007	0.197	0.182	7.4
73	0.002	0.200	0.189	5.5
82	0.005	0.210	0.187	10.7
103	0.001	0.230	0.190	17.3
104	0.025	0.195	0.172	10.5
122	0.001	0.205	0.193	5.8
130	0.004	0.210	0.195	7.0
133	0.003	0.208	0.192	7.6
139	0.002	0.203	0.196	3.4
145	0.027	0.181	0.170	5.3
147	0.029	0.198	0.179	8.4
152	0.018	0.190	0.183	5.2
154	0.005	0.200	0.194	2.9
166	0.004	0.198	0.193	2.5
178	0.016	0.192	0.182	4.8
187	0.001	0.205	0.198	3.4
195	0.025	0.183	0.172	5.3
196	0.027	0.179	0.169	4.9
198	0.022	0.184	0.175	4.4
199	0.019	0.183	0.170	6.4
200	0.017	0.191	0.182	4.3
254	0.003	0.210	0.195	7.0

AIGP's 'linearization process' was inserted data points (see Table 5.1), the original non-linearity errors were reduced to the modified values which were within the allowable ranges. For each step, the percentage of the reduced total machining error is also calculated as shown in Table 5.3. It should be noted that the allowable non-linearity error changes depend on the linearity error changes in each step. For example, in step 3, the linearity error was reduced, so the allowable non-linearity error was increased.

For the specified machining tolerance of 5.08×10^{-3} mm ($0.2" \times 10^{-3}$), the AIGP must insert the additional cutter positions to meet the machining precision requirements. Alternately, the new method modifying the cutter orientations of CLDATA also ensures the machining precision without data-point insertion. Specifically, the modified CLDATA has fewer data points than the one processed by the 'linearization process'. The proposed methodology improves the machining precision from 12.7×10^{-3} mm ($0.5" \times 10^{-3}$) to 5.08×10^{-3} mm ($0.2" \times 10^{-3}$) without reducing the cutter step-forward distance. This result overcomes the drawbacks from the 'linearization process'. It illustrates that the proposed methodology results in higher machining precision and reduced data storage space which in turn reduces machining time.

CHAPTER 6

A 3D COMBINED LINEAR AND CIRCULAR INTERPOLATOR DESIGN TECHNIQUE

6.1 Introduction

Continuous path contouring requires continuous control of spindle head movements with respect to the machine table (or of the table relative to the spindle), not only of the start and the end points of movements, but also all of the intermediate points as well. The process of filling in data about curved surfaces, from a set of point data describing some points on the curve, is termed interpolation. In conventional multi-axis CNC machining of sculptured surfaces, the 'position contouring' technique with linear interpolation method are commonly used to generate the required commands for driving the motion axes. In five-axis CNC machining, the three translational axes movements are linearly interpolated along the straight line segment and, simultaneously, the rotational movements are interpolated along space circles, which have coupling effects on the linearly interpolated translational movement positions. Consequently, the actual CC point moves along curve segments, which results in the non-linearity error problem.

Many factors contribute to the CNC machining errors. One way solving the non-linearity error problem in ultra-precision five-axis CNC machining process is to modify off-line the cutter orientation changes to reduce non-linearity errors within the machining tolerance. This method has been explored as the 'minimum error tool path generation method'

in ultra-precision five-axis CNC machining as outlined in Chapter 4. Another route is to pursue new interpolator designs, because one of the factors causing the non-linearity machining errors is due to the machining interpolation method. The five-axis CNC machining motion trajectories are non-linear curve segments, therefore, it is desired to design new interpolators which are capable of tracing non-linear curves and provide solutions to the non-linearity error problem. This chapter presents a procedure for designing a new interpolator for five-axis CNC machining systems. The proposed interpolator will coordinate the translational movements to move along a predesigned curve path, and conducts the CC point to move along the straight line segment connecting each pair of adjacent machining data points. Thus, non-linearity errors in five-axis CNC machining are expected to be eliminated. In the following, in section 6.2, the conventional interpolation methods in CNC systems are discussed. In section 6.3, the 2D and 3D Digital Differential Analyzer (DDA) linear interpolation principles in conventional linear interpolation methods are reviewed. Then, the deriving procedure of the 2D DDA circular interpolation principle is reviewed in section 6.4 since it forms a basis for the development of a new 3D circular interpolation principle presented in section 6.5. This is followed by the development of an innovative 3D combined linear and circular interpolation principle in section 6.6. An algorithm for eliminating non-linearity errors in five-axis CNC machining by using the 3D combined linear and circular interpolator is presented in section 6.6. Finally, a software interpolation routine for implementing the proposed algorithm is developed and is outlined in section 6.7.

6.2 The Conventional Interpolation Methods

The function of an interpolator is to generate intermediate points lying between two consecutive machining data points. Careful programming of cutter path motions allows the point-to-point type of milling machine to follow almost any curve within reasonable tolerances. In CNC machining, an interpolation method is the core of the CNC system since the accuracy of the calculated intermediate position directly affects the machining precision of the whole CNC system and the time for computing the intermediate positions affects the controlled axis velocity, which in turn, affects the quality of the machined surface and the machining time. Basically, the conventional interpolation methods can be divided as the reference-pulse methods and the reference-word method [44].

In the reference-pulse system, the interpolator routine in the MCU calculates the intermediate positions and send reference pulse signals to each axis control loop. The operation of the reference-pulse interpolators is based on an iterative technique controlled by an interrupt clock. At each interrupt, a single iteration of the routine is executed and produces an output pulse. Each pulse generates a motion of one basic length-unit (BLU) for the controlled axis. The accumulated number of pulses represents the axis position and the pulse (interrupt) frequency expresses the axis velocity. Since each pulse is equivalent to one BLU and the pulse frequency proportional to the axis velocity, the maximum velocity is proportional to the attainable interrupt frequency. While the pulse frequency depends upon the interpolation execution time, the axis velocity (the feedrate) is inversely proportional to the interpolation execution time of a single iteration. Thus, the maximum axis velocity is

proportional to the maximum attainable interrupt frequency, which in turn, depends on the execution time of the interpolator algorithm. For this reason, the reference-pulse interpolation technique is unsuitable for manufacturing systems requiring high axis velocities. The common reference-pulse interpolation techniques include the software DDA method, the stairs approximation method and the direct search method [52].

With the reference-word technique in the sampled-data system, the control loop of each axis is closed by the computer interpolation software itself, which generates reference binary words of line increments, rather than the pulses. In fact, the interpolator operates iteratively at an interpolation period of T. At each iteration, with the desired feedrate F, the interpolator computes the line increment of FT which is transmitted to the corresponding software loop comparators as the reference word. Then, the control program compares the reference word with the feedback signal to determine the position error. The error is fed at fixed time intervals to a digital-to-analog converter, which in turn supplies a voltage proportional to the required axis velocity. The reference-word interpolation techniques adopt the time-division idea to divide the curve segment into the line increments and function in an on-line mode corresponding to the data-sampling of the control system. Thus, the maximum velocity in the sampled-data system is not limited by the interpolation execution time. The practical reference-word linear interpolation techniques include the feedrate number method and the direction cosine method. The reference-word circular interpolation methods include the linear function method, the reference word DDA method, the extended DDA method and the improved Tustin method [49, 71].

Conventional CNC machines support the functions of 3D linear and 2D circular interpolations. With linear interpolation, the actual path formed consists of either the BLU increments in the reference pulse system or a set of straight line increments in the reference-word system. The BLU increments or line increments are used to approximate the line segments. Using the BLU increments to coordinate the intermediate data points, the intermediate positions either fall on the straight line segment or positioned within the band of one BLU. In reference-word linear interpolation, the linearly interpolated line increments are all on the line segment [49, 61]. The circular interpolation uses the circular arc increments to approximate the required contour path. Circular interpolated arcs actually are comprised of either BLU increments in the reference pulse system or straight line increments in the reference-word system. The basic idea of circular interpolations is to use straight line increments either in the chord direction or the tangent direction to approximate the arc increments. The number of the line increments determines the accuracy of the generated circular arcs. The optimal number of increments is the smallest one which maintains the path error within the band of BLU.

In selecting the interpolation method, one criterion is the uniformity of velocity along the path, because variations in milling velocity cause feed variations which affect the surface finish of the workpiece. It has been shown that the appropriate reference-pulse interpolation technique for machine tool systems is the software DDA method [52], which provides a uniform feedrate along the circular path. On the basis of the reference-pulse DDA interpolation algorithm, the reference-word DDA interpolation method has been designed

[49] with a selected interpolation period that replaces the differential time interval in the reference-pulse method. The reference-word DDA method coordinates the intermediate positions with the specified line increments and functions in an on-line mode, so that it overcomes the drawback on the limitation of the maximum feedrate in the reference-pulse DDA method. For this reason, a 3D circular software interpolator and a 3D combined linear and circular software interpolator for five-axis CNC machining have been designed based on the reference-word DDA technique. To demonstrate the developments of the new 3D interpolators, the basic 2D DDA interpolation principles, which are commonly used [49, 50, 51], are first discussed in the following sections.

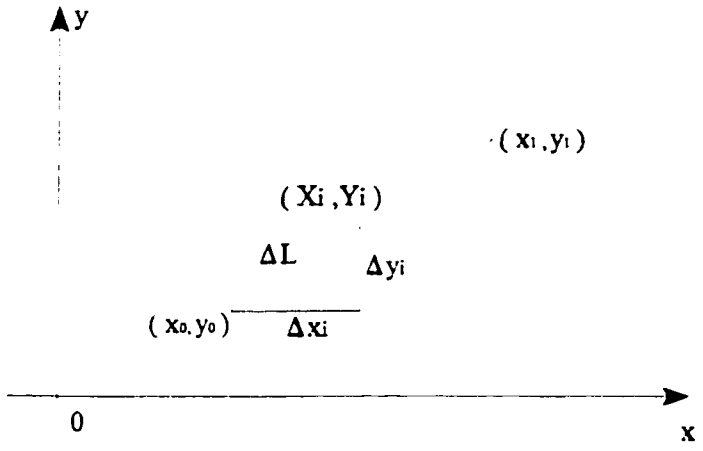
6.3 The 2D and 3D DDA Linear Interpolation Principles

6.3.1 The 2D DDA linear interpolation principle[48,49]

To drive the cutting tool moving along a straight line segment as shown in Fig. 6.1, with the start point of $p_0\,(x_0\,,\,y_0)$ and the end point of $p_1\,(x_1\,,\,y_1)$, the i-th increment from the i-th point to the (i+1)-th point for each axis can be determined from the start and the end points:

$$\frac{\Delta X_i}{x_1 - x_0} = \frac{\Delta L}{L}$$
$$\frac{\Delta Y_i}{y_1 - y_0} = \frac{\Delta L}{L} \qquad (6.1)$$

where, ΔL represents the line increment length in the direction of the velocity. L represents the length of the segment between the start point $p_0(x_0\,,\,y_0)$ and the end point $p_1(x_1\,,\,y_1)$. If the specified feedrate for the segment is F and the interpolation period is T, the line increment



(x_0 , y_0) --- start point coordinate;

(x_1 , y_1) --- end point coordinate;

(X_i , Y_i) --- the interpolated point;

Δx_i --- x-axis interpolation increment ;

Δy_i --- y-axis interpolation increment ;

ΔL --- interpolation increment ;

Figure 6.1 2D linear interpolation

length is $\Delta L = FT$ and the increments of each axis are:

$$\Delta X_i = \frac{FT}{L}(x_1 - x_0)$$
$$\Delta Y_i = \frac{FT}{L}(y_1 - y_0) \tag{6.2}$$

The 2D DDA linear interpolation can be operated as follows:

1) Preparatory calculation of the scale factor:

$$\lambda_l = \frac{FT}{L} \tag{6.3}$$

2) Computation of the segment length for each axis:

$$\Delta X_i = \lambda_l (x_1 - x_0)$$
$$\Delta Y_i = \lambda_l (y_1 - y_0) \tag{6.4}$$

3) Computation of the (i+1)-th 2D linear interpolated point:

$$X_{i+1} = X_i + \Delta X_i$$
$$Y_{i+1} = Y_i + \Delta Y_i \tag{6.5}$$

6.3.2 The 3D DDA linear interpolation principle

Based on the 2D DDA linear interpolation principle, the 3D DDA linear interpolation formula is derived as follows. Assume the start point and the end point of the segment are p_0 (x_0, y_0, z_0) and p_1 (x_1 , y_1 , z_1) as shown in Fig.6.2. The i-th increment for each axis are:

$$\begin{aligned} \Delta X_i &= \frac{\Delta L}{L}(x_1 - x_0) \\ \Delta Y_i &= \frac{\Delta L}{L}(y_1 - y_0) \\ \Delta Z_i &= \frac{\Delta L}{L}(z_1 - z_0) \end{aligned} \tag{6.6}$$

where, $\Delta L = FT$ is the line increment length in one interpolation with the feedrate F and the interpolation period T. L is the length of the segment.

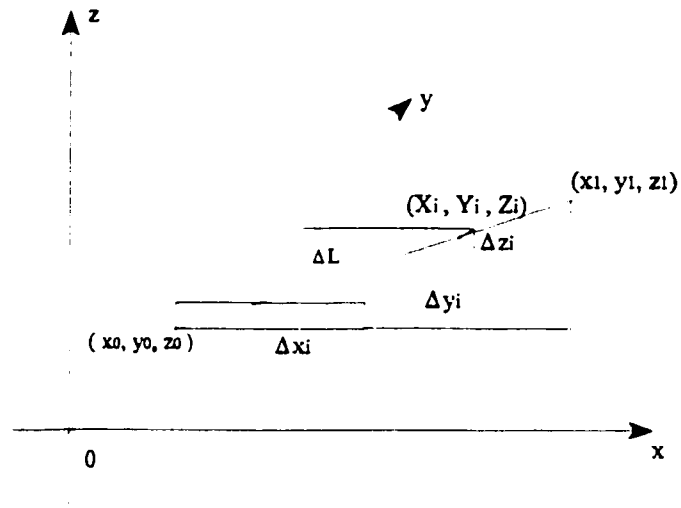
The 3D DDA linear interpolation can be iterated as follows:

1) Preparatory calculation of the linear interpolation scale factor:

$$\lambda_i = \frac{FT}{L} \tag{6.7}$$

2) Computation of the segment length for each axis:

$$\begin{aligned} \Delta X_i &= \lambda_i (x_1 - x_0) \\ \Delta Y_i &= \lambda_i (y_1 - y_0) \\ \Delta Z_i &= \lambda_i (z_1 - z_0) \end{aligned} \tag{6.8}$$



(x_0 , y_0 , z_0) --- start point coordinate;

(x_1 , y_1 , z_1) --- end point coordinate;

(X_i , Y_i , Z_i) --- the interpolated point;

Δx_i --- x-axis interpolation increment;

Δy_i --- y-axis interpolation increment;

Δz_i --- z-axis interpolation increment;

ΔL --- interpolation increment ;

Figure 6.2 3D Linear Interpolation

3) Computation of the (i+1)-th 3D linear interpolated point:

$$\begin{aligned} X_{i+1} &= X_i + \Delta X_i \\ Y_{i+1} &= Y_i + \Delta Y_i \\ Z_{i+1} &= Z_i + \Delta Z_i \end{aligned} \tag{6.9}$$

6.4 The 2D DDA Circular Interpolation Principle [49]

The 2D DDA circular interpolation principle is based on the solution of differential equations. As shown in Fig.6.3, a partial circular arc in the x-y plane is:

$$(X_i - x)^2 + (Y_i - y)^2 = R^2 \tag{6.10}$$

where, $X_i = X(t)$ and $Y_i = Y(t)$ represent the interpolation position variable with parameter t , (x, y) represents the circle centre coordinate and R represents the radius of the circle. By differentiating, it results in:

$$(X_i - x)\frac{dX_i}{dt} + (Y_i - y)\frac{dY_i}{dt} = 0 \tag{6.11}$$

or,

$$\frac{v_y}{v_x} = \frac{\dfrac{dY_i}{dt}}{\dfrac{dX_i}{dt}} = -\frac{(X_i - x)}{(Y_i - y)} = -\frac{k(X_i - x)}{k(Y_i - y)} \tag{6.12}$$

where, k represents a scale constant, and dt represents the differential time interval.

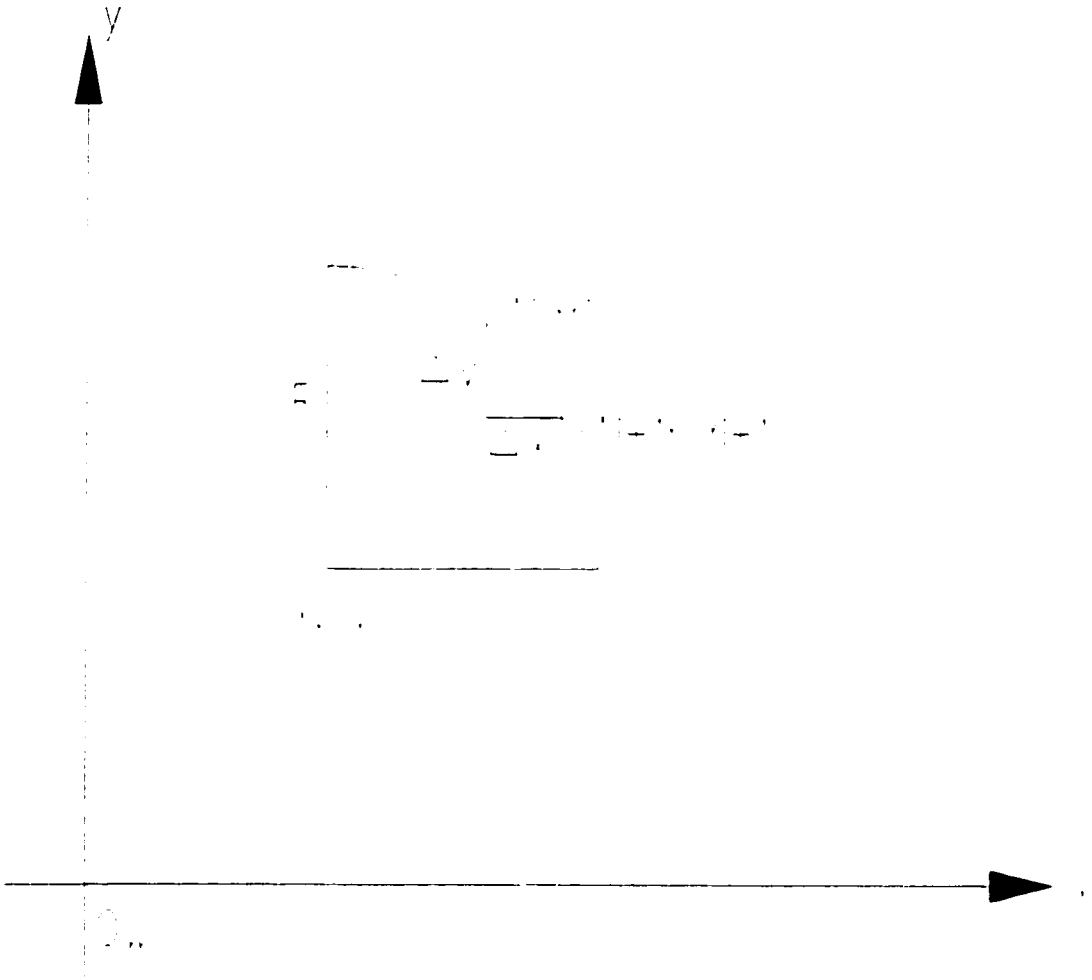


Figure 6.3 2D DDA Circular Interpolation

From Eq.(6.12), it is known that each axis velocity depends on the cross-coupled i-th interpolated point, i.e., the x-axis velocity v_x depends on the value of the interpolated coordinate Y_i and v_y depends on the interpolated coordinate X_i. Thus, the i-th step increment for each axis is:

$$\begin{aligned} \Delta X_i &= v_x \, \Delta t = k \, (Y_i - y)\Delta t \\ \Delta Y_i &= v_y \, \Delta t = -k \, (X_i - x)\Delta t \end{aligned} \tag{6.13}$$

where, the plus sign for ΔX_i and the minus sign for ΔY_i is for the case of the clockwise circular interpolation. From Eq.(6.13), it can be seen that each axis increment is in the direction of the axis velocity vector. Representing the feedrate by F and replacing the differential time interval Δt by the interpolation period T, and letting k = F/R, the i-th increment for each axis is:

$$\begin{aligned} \Delta X_i &= \frac{FT}{R}(Y_i - y) \\ \Delta Y_i &= -\frac{FT}{R}(X_i - x) \end{aligned} \tag{6.14}$$

Similarly, for the arc on the z-x plane, the i-th step increment for each axis is:

$$\begin{aligned} \Delta X_i &= v_x \, \Delta t = -\frac{FT}{R}(Z_i - z) \\ \Delta Z_i &= v_z \, \Delta t = \frac{FT}{R}(X_i - x) \end{aligned} \tag{6.15}$$

And, for the arc on the z-y plane, the i-th step increment for each axis is:

$$\Delta Z_i = v_z \Delta t = -\frac{FT}{R}(Y_i - y)$$
$$\Delta Y_i = v_y \Delta t = \frac{FT}{R}(Z_i - z) \tag{6.16}$$

The 2D DDA circular interpolation can be operated as follows:

1) Preparatory calculation of the circular interpolation scale factor:

$$\lambda_c = \frac{FT}{R} \tag{6.17}$$

2) Computation of the (i+1)-th point of interpolation on the x-y plane if the interpolated circle is on the plane:

$$X_{i+1} = X_i + \lambda_c (Y_i - y)$$
$$Y_{i+1} = Y_i - \lambda_c (X_i - x) \tag{6.18}$$

3) Computation of the (i+1)-th point of interpolation on the x-z plane if the interpolated circle is on the plane:

$$X_{i+1} = X_i - \lambda_c (Z_i - z)$$
$$Z_{i+1} = Z_i + \lambda_c (X_i - x) \tag{6.19}$$

4) Computation of the (i+1)-th point of interpolation on the z-y plane if the interpolated circle is on the plane:

$$\begin{aligned} Z_{i+1} &= Z_i - \lambda_c (Y_i - y) \\ Y_{i+1} &= Y_i + \lambda_c (Z_i - z) \end{aligned} \tag{6.20}$$

6.5 Development of a 3D DDA Circular Interpolation Principle

From the 2D DDA circular interpolation formula, the i-th increment in the x-axis direction includes the increments which resulted from the circular interpolations on both the x-y plane:

$$\Delta X_i = \lambda_c (Y_i - y) \tag{6.21}$$

and the z-x plane:

$$\Delta X_i = - \lambda_c (Z_i - z) \tag{6.22}$$

The total increment in the x-axis direction is the summation of Eq.(6.21) and Eq.(6.22):

$$\Delta X_i = \lambda_c [n_z (Y_i - y) - n_y (Z_i - z)] \tag{6.23}$$

where, Y_i represents the i-th interpolated y coordinate and Z_i represents the i-th interpolated z coordinate, y and z represent the interpolation circle centre's y and z coordinate respectively, n_z is the unit normal vector of the x-y plane, n_y is the unit normal vector of z-x plane, and λ_c is the circular interpolation scale factor as given by Eq. (6.17).

The i-th increment in the y-axis direction consists of the increments which resulted from the circular interpolations on the x-y plane:

$$\Delta Y_i = - \lambda_c (X_i - x) \tag{6.24}$$

and on the y-z plane:

$$\Delta Y_i = \lambda_c (Z_i - z) \tag{6.25}$$

The total interpolated increment in the y-axis direction is the summation of Eq.(6.24) and Eq.(6.25):

$$\Delta Y_i = \lambda_c [n_x (Z_i - z) - n_z (X_i - x)] \tag{6.26}$$

where, X_i and Z_i represent the i-th interpolated x and z coordinates respectively, x and z represent the interpolated circle centre coordinates respectively, λ_c is a scale factor as given by Eq.(6.17). n_x is the unit normal vector of the y-z plane, and n_z is the unit normal vector of x-y plane.

Similarly, the z increment due to the interpolation on the z-x plane is:

$$\Delta Z_i = \lambda_c (X_i - x) \tag{6.27}$$

and the z increment due to the interpolation on the y-z plane is:

$$\Delta Z_i = - \lambda_c (Y_i - y) \tag{6.28}$$

The total interpolated increment in the z-axis direction is the summation of Eq.(6.27) and Eq.(6.28):

$$\Delta Z_i = \lambda_c [n_y (X_i - x) - n_x (Y_i - y)] \qquad (6.29)$$

where, X_i and Y_i represent the i-th interpolated x and y coordinates respectively, x and y represent the interpolated circle centre coordinates respectively, λ_c is a scale factor as given by Eq.(6.17), n_y and n_x are the unit normal vectors of the x-z plane and the y-z plane.

Considering that the computer carries out the interpolations in the sequence of: (1) x-axis position; (2) y-axis position; (3) z-axis position, the x-axis i-th interpolated increment is determined on the basis of the i-th interpolated coordinate of (X_i , Y_i , Z_i) as follows:

$$\Delta X_i = \lambda_c [n_z (Y_i - y) - n_y (Z_i - z)] \qquad (6.30)$$

The interpolation procedure goes to the new position (X_{i-1} , Y_i , Z). The interpolated y-axis i-th increment , then, is determined based on the coordinate of (X_{i-1} , Y_i , Z_i) as:

$$\Delta Y_i = \lambda_c [n_x (Z_i - z) - n_z (X_{i-1} - x)] \qquad (6.31)$$

Next the interpolation goes to another new point of (X_{i-1} , Y_{i-1} , Z_i). The interpolated i-th z-axis increment is then determined based on the new point coordinate of (X_{i-1} , Y_{i-1} , Z_i):

$$\Delta Z_i = \lambda_c [n_y (X_{i-1} - x) - n_x (Y_{i-1} - y)] \qquad (6.32)$$

The (i+1)-th interpolated coordinates can be determined as:

$$\begin{aligned} X_{i+1} &= X_i + \Delta X_i \\ Y_{i+1} &= Y_i + \Delta Y_i \\ Z_{i+1} &= Z_i + \Delta Z_i \end{aligned} \tag{6.33}$$

Therefore, by substituting Eq.(6.30), Eq.(6.31) and Eq.(6.32) into Eq.(6.33), one may obtain the 3D DDA circular interpolation principle as follows:

$$\begin{aligned} X_{i+1} &= X_i + \lambda_c \left[n_z (Y_i - y) - n_y (Z_i - z) \right] \\ Y_{i+1} &= Y_i + \lambda_c \left[n_x (Z_i - z) - n_z (X_{i+1} - x) \right] \\ Z_{i+1} &= Z_i + \lambda_c \left[n_y (X_{i+1} - x) - n_x (Y_{i+1} - y) \right] \end{aligned} \tag{6.34}$$

The iterative operation of the 3D DDA circular interpolation can be described in the following steps:

1) Calculate off-line the circular interpolation scale factor based on Eq.(6.17);

2) Compute on-line the i-th increments in each axis direction and the (i+1)-th interpolated point in the order of x-y-z coordinates based on equations (6.30), (6.31), (6.32) and (6.33).

To verify the 3D DDA circular interpolation principle of Eq.(6.34), a simulation study was carried out. The objective of 3D circular interpolation is to interpolate data points along a space circle or along a path on a sphere. Hence, by assuming a set of data points on a sphere surface and using the 3D DDA circular interpolation formula of Eq.(6.34), the simulated 3D

circular interpolated path was generated. A sphere centred at the origin of the coordinate system and having the radius of 76.2 mm (3") was chosen for the simulation as shown in Fig.6.4 . The assumed data points are on the space circle with the longitude angle $\phi = 60°$ and the latitude angle θ varying from 0° to 90° on the surface of the sphere. In order to investigate the interpolation accuracy, the segment arc length between two consecutive assumed data points was set to be varying. The segment arc lengths were assumed 10° latitude angle apart for the region that the latitude angle is between 60° and 90° of the sphere circle. The segment arc lengths were assumed 15° latitude angle apart for the region that the latitude angle is from 30° to 60° of the sphere circle, and for the region that the latitude angle is between 0° to 30° of the sphere circle, the segment length was assumed corresponding to 30° latitude angle apart. The interpolation feedrate was chosen as F=1524 mm/min. (60 ipm), and the interpolation period was chosen as T = 0.2 sec. Hence, the interpolation increment FT = 5.08 mm (0.2"). A detailed discussion on how to select the interpolation parameters F and T is given in Chapter 7.

The simulation was conducted as follows:

1) Based on the chosen interpolation feedrate, the interpolation period and the radius of the sphere, the circular interpolation scale factor was calculated by using Eq.(6. 17). This is an off-line calculation and the scale factor is a constant.

2) Based on the assumed data points on the sphere surface, the calculations of each segment length on each axis was performed:

Figure 6.4 3D DDA Circular Interpolation

$$L_x(k) = x(k+1) - x(k)$$
$$L_y(k) = y(k+1) - y(k) \qquad (6.35)$$
$$L_z(k) = z(k+1) - z(k)$$

where, (x(k), y(k), z(k)) and (x(k+1), y(k+1), z(k+1)) represent the start and the end points of the k-th segment, and $L_x(k)$, $L_y(k)$ and $L_z(k)$ are the corresponding k-th segment lengths along each axis direction.

3) The interpolation start point (X_0, Y_0, Z_0) for each segment was initialized:

$$X_0 = x(k)$$
$$Y_0 = y(k) \qquad (6.36)$$
$$Z_0 = z(k)$$

4) The interpolation procedure was continued from step 5) when the interpolated coordinates, X_i, Y_i and Z_i, had not arrived at the end point of the segment, in which case they would have satisfied the following condition:

$$X_i - X_0 < L_x$$
$$Y_i - Y_0 < L_y \qquad (6.37)$$
$$Z_i - Z_0 < L_z$$

when one or two of the conditions of Eq.(6.37) were not met, the interpolation procedure in the axes directions would not be performed and the other axis direction of interpolation continued from step 5). If all of the conditions were not met, the

interpolation procedures went on to the next segment.

5) The differences between the interpolated point (X_i , Y_i , Z_i) and the end point of the segment (X_1 , Y_1 , Z_1) were calculated:

$$\begin{aligned} D_x &= L_x(k) - (X_i - X_0(k)) \\ D_y &= L_y(k) - (Y_i - Y_0(k)) \\ D_z &= L_z(k) - (Z_i - z_0(k)) \end{aligned} \tag{6.38}$$

6) The interpolation procedure was continued from step 7) when the maximum difference among the D_x , D_y and D_z calculated in step 5) satisfied the following condition:

$$\max (D_x , D_y , D_z) \geq FT \tag{6.39}$$

If the condition of Eq.(6.39) was not met, the interpolation procedure repeated from step 3). In fact, this happened when the interpolation procedure approached the end point of the segment. Since the maximum difference was smaller than the interpolation increment and usually was a very small value, it was then skipped and interpolation went on to the next segment.

7) The interpolation increment in the x-axis direction was calculated based on Eq.(6.30);

8) The next interpolation point in the x-axis direction was calculated as follows:

$$X_{i+1} = X_i \pm \Delta X_i \tag{6.40}$$

where, the plus and the minus sign depended upon whether the interpolation increased or decreased.

9) The interpolation increment in the y-axis direction was calculated based on Eq.(6.31);

10) The next interpolated point in the y-axis direction was calculated as follows:

$$Y_{i+1} = Y_i \pm \Delta Y_i \tag{6.41}$$

where, the plus and the minus sign depended upon whether the interpolation increased or decreased.

11) The interpolation increment in the z-axis direction was calculated based on Eq.(6.32);

12) The next interpolated point in the z-axis direction was calculated as follows:

$$Z_{i+1} = Z_i \pm \Delta Z_i \tag{6.42}$$

where, the plus and the minus sign depended upon whether the interpolation increased or decreased.

The total number of interpolated points were 33 points among the assumed 7 data points on the sphere surface. Since the segment arc lengths were purposely set varying along the sphere circle path, the amount of interpolated points for each segment were different. The shorter the segment arc length, the less points were interpolated. In fact, in the region when the latitude angle was close to the equator, the amount of interpolated points were less than that close to the pole. Fig. 6.4 shows the interpolated path and the space curve on the sphere surface. It is obvious that the interpolated path is along the space circle and the accuracy is

better at the region near to the equator, which is due to the fact that the shorter segment arc length was assigned. The accuracy study results of the interpolation are outlined in Table 6.1. The maximum interpolation error is 5.0521mm (0.1989") which is smaller than the interpolation increment: FT= 5.08 mm (0.2"). When the segment arc length is 10° apart, the minimum interpolation error of 0.0203 mm (0.0008") resulted. The interpolation accuracy can also be improved by choosing a smaller interpolation period. This simulated result verifies the usefulness of the new 3D DDA circular interpolation principle.

Table 6.1 3D DDA Circular Interpolation Errors

Segment arc angle	max. error	min. error	interpolation increment
10°	-0.6147 mm (-0.0242")	-0.0203 mm (-0.0008")	5.08 mm (0.2")
15°	2.8804 mm (0.1134")	0.3912 mm (0.0154")	5.08 mm (0.2")
30°	5.0521mm (0.1989")	1.1252 mm (0.0443")	5.08 mm (0.2")

6.6. A 3D Combined Linear and Circular Interpolation Principle

6.6.1 The Motion Trajectory of the OM-1 Milling Centre

The motion trajectory in five-axis CNC machining depends on different machine configurations and rotational movements. The common configuration of five-axis CNC machine tools include both the swivel-head and the rotary-table types. In the swivel-head type, the spindle chuck acts as the swivel (the rotational) movements pivot. The 3D circular swivel movements about the pivot superimposed on the 3D linear motion of the pivot construct the combined 3D linear and circular motion trajectory. In the rotary-table type, the intersection point of two axes acts as the pivot for the rotational movements. The machine table's 3D rotational movements, superimposed on the translational movements of the pivot, form the 3D combined linear and circular motion trajectory. To analyze the 3D combined motion trajectory, without loss of generality, the OM-1 milling centre, which is the rotary-table type, is considered in this work. From the OM-1 configuration (see Fig.4.4), it is known that the rotational B_m axis and C_m axis are perpendicular and intersect at a space point, P, at all time during the machining processes while the machine table moves. Since the intersection point P is on both the B_m axis and the C_m axis, it moves translationally only. Thus, the intersection point P acts as a pivot for the machining rotational movements, such that the part together with the machine table rotates about the moving pivot P tracing out a 3D non-linear path. In other words, the actual CC point trajectory is constructed by the 3D circular movements (about the pivot P) which are superimposed on the linearly interpolated movements of the pivot P. From this geometrical perspective, the CC point trajectory model for the OM-1 was determined as given by Eq.(4.19) and Eq.(4.20).

6.6.2 The Rotational Movements Interpolation Principle

In multi-axis CNC machining, the rotational movement increments are angles measured in degrees. The common interpolation method for rotational movements is using the direct function algorithm [71] as follows :

$$\begin{aligned} B_i &= B_0 + \tau\ (\ B_1 - B_0\ ) \\ C_i &= C_0 + \tau\ (\ C_1 - C_0\ ) \end{aligned} \qquad (6.43)$$

where, (B_i, C_i) is the interpolated coordinate of the rotational movements in degree units, (B_0, C_0) and (B_1, C_1) represent the start and the end coordinates of the rotational movements for the move respectively, τ represents a parameter which is proportional to time and varies in the range of (0 , 1), i.e., $\tau = 0$ at (B_0, C_0) and $\tau = 1$ at (B_1, C_1).

The direct function algorithm is designed for parametric space curves and is particularly suitable to rotational movement angles. The 3D machine movements are parametric functions of time, because each axis coordinate (linear position in length or circular position in degree units) is a parametric function of time. Therefore, the direct function algorithm is used to interpolate the rotational movements in five-axis machining.

6.6.3 A 3D Combined Linear and Circular Interpolation Principle

In conventional five-axis CNC machining, 3D linear interpolation principle is used to coordinate the translational movements along each 3D straight line segment and the rotational

movements are interpolated based on the direct function algorithm of Eq. (6. 43). As a result, the CC point trajectories are the 3D linear motions of the rotation pivot combined with the 3D circular motions about the rotation pivot as shown by the machining motion trajectory model of Eq.(4.19). The curved trajectories cause non-linearity errors in multi-axis CNC machining. To eliminate non-linearity errors, it is desired to conduct the CC point to move along the 3D straight line segment connecting the consecutive movement points. This can be achieved if the rotational pivot (or the translational movements) is interpolated along a predesigned curve, say a 3D combined linear and circular curve. Based on this geometrical analysis, a combined 3D linear and circular interpolation principle is developed. The corresponding 3D DDA linear and circular interpolation formula is:

$$
\begin{aligned}
X_{i+1} &= X_i + \lambda_c \left[(x_1 - x_0) + n_z(Y_i - y_i) - n_y(Z_i - z_i)\right] \\
Y_{i+1} &= Y_i + \lambda_c \left[(y_1 - y_0) + n_x(Z_i - z_i) - n_z(X_{i+1} - x_{i+1})\right] \\
Z_{i+1} &= Z_i + \lambda_c \left[(z_1 - z_0) + n_y(X_{i+1} - x_{i+1}) - n_x(Y_{i+1} - y_{i+1})\right]
\end{aligned}
\tag{6.44}
$$

The detailed deriving procedure is provided in Appendix C.

Interpolating the rotational movements based on the direct function of Eq.(6.43) and interpolating the pivot P along the curve of Eq.(6.44), i.e., coordinate the translational axes based on Eq.(6. 44), the CC point is expected to move along the straight line segment with constant feedrate at each move:

$$x_{cc}(i+1) = x_{cc}(i) + \lambda_l (x_{cc}(1) - x_{cc}(0))$$

$$y_{cc}(i+1) = y_{cc}(i) + \lambda_l (y_{cc}(1) - y_{cc}(0)) \qquad (6.45)$$

$$z_{cc}(i+1) = z_{cc}(i) + \lambda_l (z_{cc}(1) - z_{cc}(0))$$

where, $(x_{cc}(i), y_{cc}(i), z_{cc}(i))$and $(x_{cc}(i+1), y_{cc}(i+1), z_{cc}(i+1))$ are the present and the next CC point coordinates respectively, $(x_{cc}(0), y_{cc}(0), z_{cc}(0))$ and $(x_{cc}(1), y_{cc}(1), z_{cc}(1))$ are the start and the end CC point coordinates of the segment, and λ_l is the linear interpolation scale factor.

6.6.4. Proposed Methodology for Solving the Non-Linearity Error Problem

What follows is the proposal of a methodology which can be utilized to solve the multi-axis CNC machining precision problem. The method, applying the 3D combined linear and circular interpolation principle, coordinates the translational axes, or the rotation pivot, along the predesigned curve segment and conducts the CC point to move along the 3D straight line segment connecting each two consecutive machining data points. The method interpolates the CC point coordinates which are determined from the machine motion trajectory model. By comparing the coordinates of the interpolated point and the end point of the segment, the iterative interpolation is performed. The procedure for performing the iterative interpolation can be presented in the following algorithm.

3D Combined Linear & Circular Interpolation Algorithm

1) Compute off-line the start and the ending CC point coordinates for each segment by using the machine motion trajectory model (for OM-1, using Eqs. (4.19) and (4. 20));

2) Calculate off-line the interpolation scale factors for each segment:

2.1) Calculate each segment length:

$$L=\sqrt{(x_{cc}(e)-x_{cc}(s))^2+(y_{cc}(e)-y_{cc}(s))^2+(z_{cc}(e)-z_{cc}(s))^2} \tag{6.46}$$

where, the index, e, represents the end point and the index, s, represents the start point.

2.2) Calculate the linear interpolation scale factor: $\lambda_l = FT/L$;

2.3) Calculate the rotational (3D circular) movements radius, R, which equals to the distance between the CC point and the rotation pivot. Hence, one may have:

$$R = \sqrt{(x_m - x_{home})^2 + (y_m - C_{pivot})^2 + (z_m - GL)^2} \tag{6.47}$$

where, (x_m, y_m , z_m) are the NC-codes coordinates, x_{home} is the spindle position x-coordinate, C_{pivot} is the C_m axis pivot constant and GL is the cutter gage length as shown in Fig.4.4.

2.4) Calculate the circular interpolation scale factor: $\lambda_c = FT/R$;

3) On-line interpolation routine:

3.1) Interpolate the CC point coordinates based on the start point

(x_{cc} (s), y_{cc} (s), z_{cc}(s)) and the end point (x_{cc} (e), y_{cc} (e), z_{cc}(e)) of the segment:

$$
\begin{aligned}
x_{cc}(i+1) &= x_{cc}(i) + \lambda_l\,(x_{cc}(e) - x_{cc}(s)) \\
y_{cc}(i+1) &= y_{cc}(i) + \lambda_l\,(y_{cc}(e) - y_{cc}(s)) \\
z_{cc}(i+1) &= z_{cc}(i) + \lambda_l\,(z_{cc}(e) - z_{cc}(s))
\end{aligned}
\tag{6.48}
$$

3.2) Interpolate the translational axes coordinates in the following order: x-axis⇒ y-axis⇒ z-axis. Thus, the 3D linear and circular interpolation of the pivot centre is:

$$
\begin{aligned}
X(i+1) &= X(i)+\lambda_c[(x_{cc}(e)-x_{cc}(s))+(Y(i)-y_{cc}(i))-(Z(i)-z_{cc}(i))] \\
Y(i+1) &= Y(i)+\lambda_c[(y_{cc}(e)-y_{cc}(s))+(Z(i)-z_{cc}(i))-(X(i+1)-x_{cc}(i+1))] \\
Z(i+1) &= Z(i)+\lambda_c[(z_{cc}(e)-z_{cc}(s))+(X(i+1)-x_{cc}(i+1))-(Y(i+1)-y_{cc}(i+1))]
\end{aligned}
\tag{6.49}
$$

3.3) Interpolate the rotational axes circular movements according to the direct function algorithm as follows:

$$
\begin{aligned}
B(i) &= B(s) + \tau(\,B(e) - B(s)\,) \\
C(i) &= C(s) + \tau(\,C(e) - C(s)\,)
\end{aligned}
\tag{6.50}
$$

4) Calculate the length between the start point and the present interpolated CC point:

$$
l(i) = \sqrt{(x_{cc}(i)-x_{cc}(s))^2+(y_{cc}(i)-y_{cc}(s))^2+(z_{cc}(i)-z_{cc}(s))^2} \tag{6.51}
$$

5) Compare the coordinates of the interpolated point with the end point of the segment by performing the test:

$$(L - l(i)) \geq FT \tag{6.52}$$

If the test is true, perform step 6) , otherwise go to step 7).

6) Repeat the interpolation routine on the present segment by performing the steps 3) to 5).

7) Determine the modified feedrate if the test is false:

$$F' = \frac{(L - l(i))}{T} \tag{6.53}$$

and modify the interpolation scale factors: λ_l = F' T/L and λ_c = F' T/R;

and repeat the step 3) to reach the end point of the segment.

8) Repeat the steps 3) to 7) on next segment until the machining arrives at the end of the NC-codes.

6.7. The Software Interpolation Routine

The first step in the overall design of a CNC system involves the selection of an appropriate control technique and the optimal setting of the control loop parameters. Subsequently, an appropriate interpolator routine that generates the reference signals to the control system must be designed. The reference-word interpolation technique in the sampled-data system has been commonly employed in current CNC systems. Therefore, the new 3D combined linear and circular interpolation technique is designed as a reference-word type interpolator. In the sampled-data system, the interpolation calculation is performed by the software interpolation

routine in the MCU. The tool path data input into the MCU must be pre-processed, which includes decoding the tool path data into a set of computer system data, performing cutter compensations if needed, and distributing feedrate to each axis of motion. After preparatory processing, the data are inputted into the interpolation routine and the interpolator computes the intermediate coordinates of positions based on the appropriate interpolation algorithm. The proposed software interpolation routine computes the coordinates based on the algorithm developed in section 6.6. The software interpolation routine includes the following functions:

1) A function accepting the machining NC-codes and the machining parameters such as the specified feedrate and the chosen interpolation period;

2) A function determining the CC point coordinates that correspond to the end points of each movement (the machine motion trajectory model is employed);

3) Three functions computing the interpolated translational axes coordinates based on the 3D combined linear and circular interpolation principle of Eq. (6. 49);

4) Two functions calculating the interpolated rotational movements coordinates based on the direct function algorithm of Eq. (6. 50).

A flow chart of the new interpolation software is shown in Fig. 6.5. The program starts by inputting the machining parameters: the specified feedrate F and the chosen interpolation period T, and the machining NC-codes. Then, the CC point coordinates of the end points of each segment is computed. Using the machining parameters and the NC-codes, the linear and circular interpolation scale factors can be determined. These are constants for each machining step move and are calculated off-line prior to the on-line execution of the

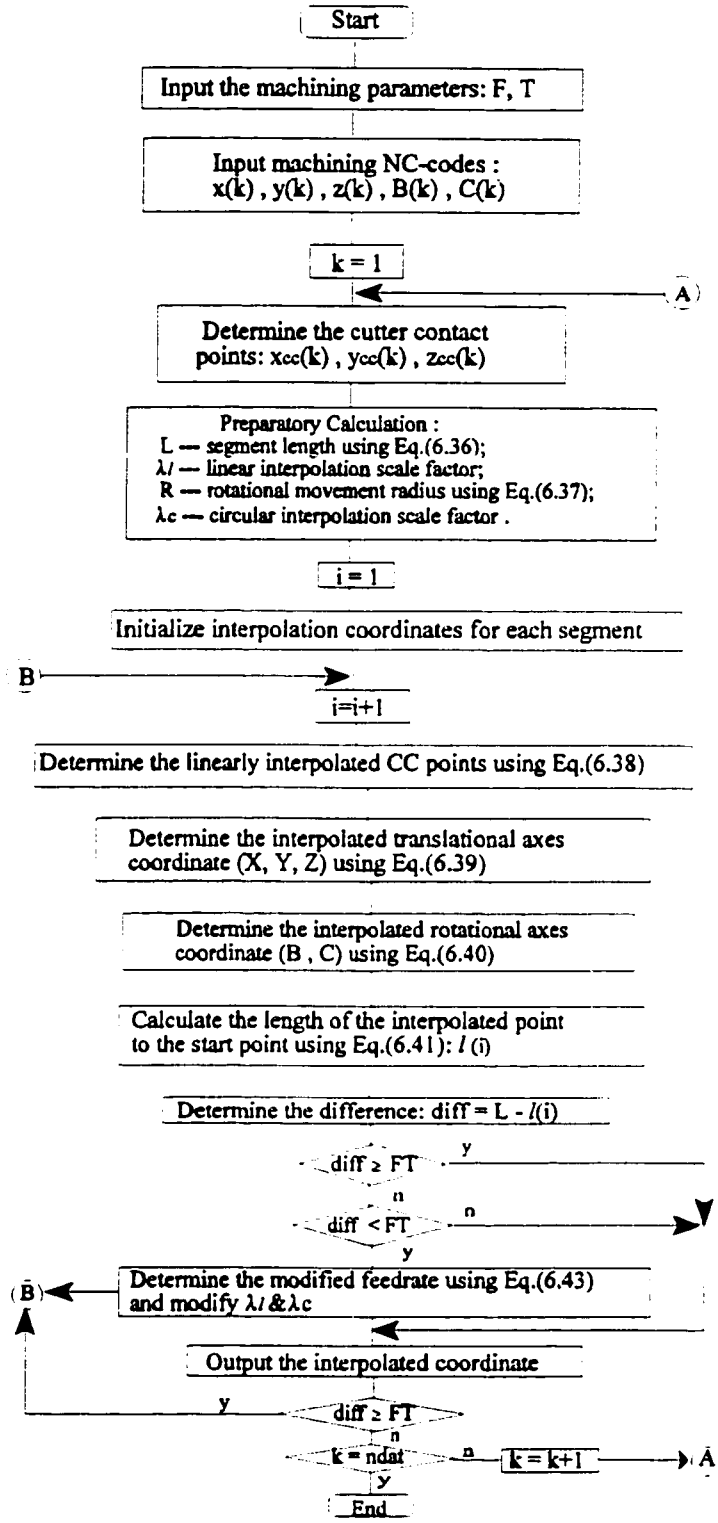


Figure 6.5 Flow Chart of the 3D Linear & Circular Interpolation Routine

interpolation. Then, real-time interpolation begins by initializing each axis coordinate for each segment. The interpolated positions of the translational and rotational axes are then computed successively. The iterative computation of the interpolated positions for each axis is performed under the conditions that the total interpolation incremented length of $l(i)$ is less than the segment length of L, and the remaining length of (L - $l(i)$) is greater than the interpolation increment as given in Eq. (6.52). In the case that the remaining length of (L - $l(i)$) is smaller than the interpolation increment, the remaining length is used to modify the feedrate which, in turn, is used to command the axes to reach the end point of the segment. This interpolation routine is repeated until all the NC-codes are completed. For the final interpolation point, it should be noted that each axis may not arrive at the segment end at the same time since each axis segment length is different. The interpolation process on next segment, however, is started at the same time for each axis since the iterative interpolation is performed under the condition of Eq.(6.52) and the feedrate adaptation procedure is used. The new software interpolation routine is coded in the programming language C, in order to adapt to general CNC machine tool control systems.

6.8 Summary

In this chapter, a methodology for solving the non-linearity error problem in ultra-precision five-axis CNC machining was proposed. Based on the 2D DDA circular interpolation principle, the 3D DDA circular interpolation formula was developed and a simulation using the proposed 3D circular interpolation principle was performed. The simulated result demonstrated the validity and the precision achieved by the new 3D circular interpolation

principle. Then, a 3D combined linear and circular interpolator was designed based on the 3D DDA linear and circular interpolation principles. A software interpolation routine based on the 3D combined linear and circular interpolation algorithm was developed. The software is capable of generating the reference-word signals to the machine axes control loops of the sampled-data systems, such that the machine translational axes can be driven to move along the predesigned curve and the machine rotational movements can be driven tc move circularly in space. The CC point thus can be conducted to move along the straight line segment, so that the non-linearity errors can be eliminated.

CHAPTER 7

AN APPLICATION OF

THE INTERPOLATOR DESIGN TECHNIQUE

7.1 Introduction

Applying the '3D combined linear and circular interpolation technique' proposed in Chapter 6, the process of machining the airfoil surfaces of an impeller is simulated and is presented in this chapter. The simulated results demonstrate that the proposed 3D linear and circular interpolator drives the translational axes along the predesigned 3D curve and conducts the CC point along the straight line segment that connects each two consecutive machining data points, so that elimination of non-linearity errors is achieved. In the following, the interpolation preparatory data processing procedure is outlined in section 7.2. Based on the data from the interpolation preparatory data processing, the conventional five-axis CNC machining process for machining the airfoil surfaces of an impeller is simulated and the result is outlined in section 7.3. This simulation result is then compared with that from the simulation of machining the same airfoil surface using the proposed 3D linear and circular interpolator (section 7.4).

7.2 Interpolation Preparatory Data Processing

An NC part program contains the data of the geometric information of the machined surfaces (the coordinates of the tool path) and the motion information of the machining (the feedrate and the auxiliary functions data). These data cannot be directly input into the interpolation

routine in the MCU of the control computer, and must be processed by the data processing unit in the MCU before the interpolation is executed. This interpolation preparatory data processing includes: (i) the decoding of the NC-codes into a set of computer system data based on standard format, (ii) the cutter compensations to obtain the cutter centre tool path if the tool path is generated based on the contour of the machined surface, and (iii) the feedrate calculations for each axis of motion to generate commands to the control loops. The procedure for interpolation preparatory data processing usually involves the following steps:

1) The NC-codes are input and stored in the NC-codes buffer;

2) The data decoding routine in the MCU processes a block of NC-codes data and store it in the decoded data buffer;

3) The cutter compensation routine in the MCU determines the cutter centre coordinate based on the surface geometry and the cutter radius if the tool path is generated based on the contour of the surface to be machined, and then stores the compensated data in the cutter compensator data buffer;

4) The feedrate routine in the MCU calculates the feedrate for each axis of motion for the machining movement and stores the data in the system working buffer;

5) The MCU transfers the data into the interpolation register and the interpolation process can be started.

The data decoding process is needed for different computer systems of CNC machine tools in order to perform the other MCU functions. In this study, however, for the purpose of simulating the function of the new interpolator, the NC-codes for machining the airfoil

surfaces of an impeller are directly used in the interpolation calculation.

In machining simple or regular geometry surfaces, such as a plane or circular shaped surface, the NC-codes are usually generated from the contour of the surface. Cutter compensation is required to obtain the cutter centre coordinates of motion. This can be accomplished by using the cutter compensation techniques as given by Liu and Lei [72] and Li, et. al. [73]. For machining complex 3D surfaces (i.e. the sculptured surfaces), the tool path is usually generated in terms of the cutter centre positions. Hence, there is no need to perform the cutter radius compensation process, and cutter length compensation is usually performed by compensating the spindle length. In this study, the available CLDATA for machining the airfoil surfaces of an impeller were generated in terms of the ball-end cutter centre position coordinates, therefore, the cutter compensation process is not required in the simulation study.

The feedrate calculation is closely related to the interpolation process of the machine axes movements. The feedrate calculation determines the step increment in an interpolation period for each axes of motion based on the specified machining feedrate. Many factors affect the machining feedrate, the specification of which must be based on the materials of the cutter and the workpiece, cutter geometry data (cutter radius and length), depth of cut and cutting velocity. In this study, assuming that the material of the workpiece is titanium and the material of the cutter is carbide, and considering the depth of cut is for the finish machining and the cutting velocity is that equivalents to the spindle speed of 1200 rpm, the feedrate is specified

as 25 mm/sec (60 ipm).

In conventional five-axis machining, the feedrate specified in terms of the inverse-time feedrate word (inverse-time FRN) is recommended, since the rotational movements are involved. The feedrate specified in terms of the direct feedrate word (length/time units) is defined for linear movements. Changes in rotational movements are measured in degrees. Therefore, the direct feedrate word is not suitable for angle/time units. The inverse-time FRN is defined as the time required for the cutter moving in the angular/linear position change of a machining step. By using the inverse-time FRN and the linear or angular position change on each axis, the feedrate along each axis can be determined. The specified feedrate represented as the inverse-time FRN in the NC-codes is by setting the G function as G93. For point milling, each segment length can be calculated from the cutter position coordinates. Using the segment length for the k-th machining step move, L_k, and the specified feedrate, F, the inverse-time FRN can be determined as:

$$FRN_k = \frac{F}{L_k} \quad [1/\text{min}] \tag{7.1}$$

Using the inverse-time FRN_k, the feedrate calculation routine in the MCU then calculates each axis speed for the k-th segment movement as follows:

$$
\begin{aligned}
F_x &= FRN_k * \Delta x_k \quad [ipm] \\
F_y &= FRN_k * \Delta y_k \quad [ipm] \\
F_z &= FRN_k * \Delta z_k \quad [ipm]
\end{aligned}
\tag{7.2}
$$

Based on these axis feedrates F_x , F_y , F_z and the interpolation period T, the linear interpolator generates the increments for each axis of motion:

$$
\begin{aligned}
dl_x &= F_x * T \\
dl_y &= F_y * T \\
dl_z &= F_z * T
\end{aligned}
\tag{7.3}
$$

The determination of the interpolation period must consider interpolation accuracy, which directly affects the machining precision of the whole CNC system. It should be greater than the time required for performing the actual interpolation calculations since the MCU performs not only the interpolations but also the other on-line tasks. The Pythagorean theorem used by Huang [74] gives the relational function of the machining precision measure, i.e., the machining tolerance t_r, the machining feedrate F and the interpolation period, T , as follows:

$$
T \leq \frac{\sqrt{8t_r\rho - 4t_r^2}}{F} \tag{7.4}
$$

where, ρ represents the radius of the local curvature of the curve; t_r is equal to or less than the machine BLU to ensure interpolation accuracy.

In this study, for the machining tolerance of t_r = 0.005 mm (0.0002"), the minimum radius of curvature of the curve ρ = 4 mm and the feedrate F = 25 mm/sec (60 ipm), the interpolation period T is determined based on Eq.(7.4), and the determined interpolation period range should be less than 15ms. By considering the required time for executing the interpolation calculation for the algorithm of the designed interpolator, the interpolation period is chosen as T = 10 ms.

The feedrate calculation procedure described above is employed when the linear interpolation method is used in conventional five-axis CNC machining. In the calculation of the DDA circular interpolation, the feedrate is in the direction of the tangent at the interpolation point on the circle, hence, the specified feedrate in the tool path data is directly used. In this study, the new interpolator performs the combined 3D DDA linear and circular interpolation function. The feedrate is in the direction of the tangent at the interpolation point on the predesigned 3D curve segment. Therefore, the specified machining feedrate F = 25 mm/sec (60 ipm) is directly used in the simulation.

7.3 Simulation of Machining Blade Surfaces Using Linear Interpolation

In order to verify the multi-axis CNC machining error analysis and the functions of the proposed 3D interpolator, the machining process for machining the airfoil surfaces of an

impeller with the linear interpolation technique was simulated. The procedure, as the conventional five-axis CNC machining, is described as follows.

A set of NC-codes for machining the airfoil surface of an impeller was determined by transforming the available CLDATA taken from a sample machining process in industry. The 3D cutting curve (or the tool path) on the airfoil surface of the impeller in reference to the workpiece coordinate system is shown in Fig.7.1. The cutter locations for machining the airfoil surfaces of the impeller in reference to the workpiece coordinate system is shown in Fig.7.2. Using the specified feedrate, F = 25 mm/sec (60 ipm), and the G93 function, the inverse-time FRN was determined for each segment using Eq.(7.1). The interpolation period determined in section 7.2 was used to calculate the interpolation increments. Then, from the preparatory data processing, the feedrates F_x, F_y and F_z were calculated for each axis based on Eq.(7.2) within the interpolation period. The linear interpolation scale factor λ_l was then determined based on the axis feedrates, the interpolation period, and each axis segment length. The translational axes, i.e. the coordinates of the machine rotation pivot P, were then interpolated based on the 3D linear interpolation principle of Eq.(6.8) and Eq.(6.9). The rotational axes were interpolated circularly based on the direct function algorithm of Eq. (6.43).

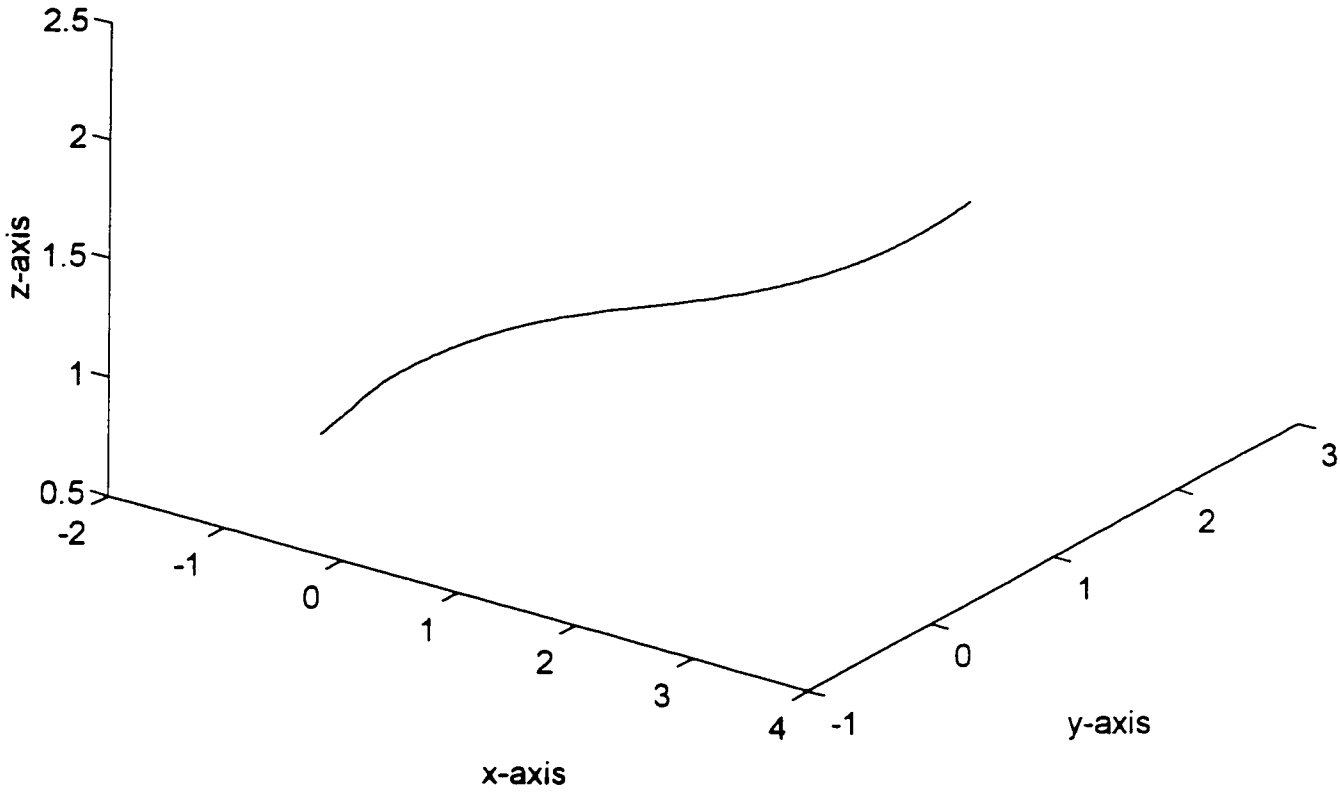


Figure 7.1 The Cutting Curve on the Blade Surface in Part Coordinate System

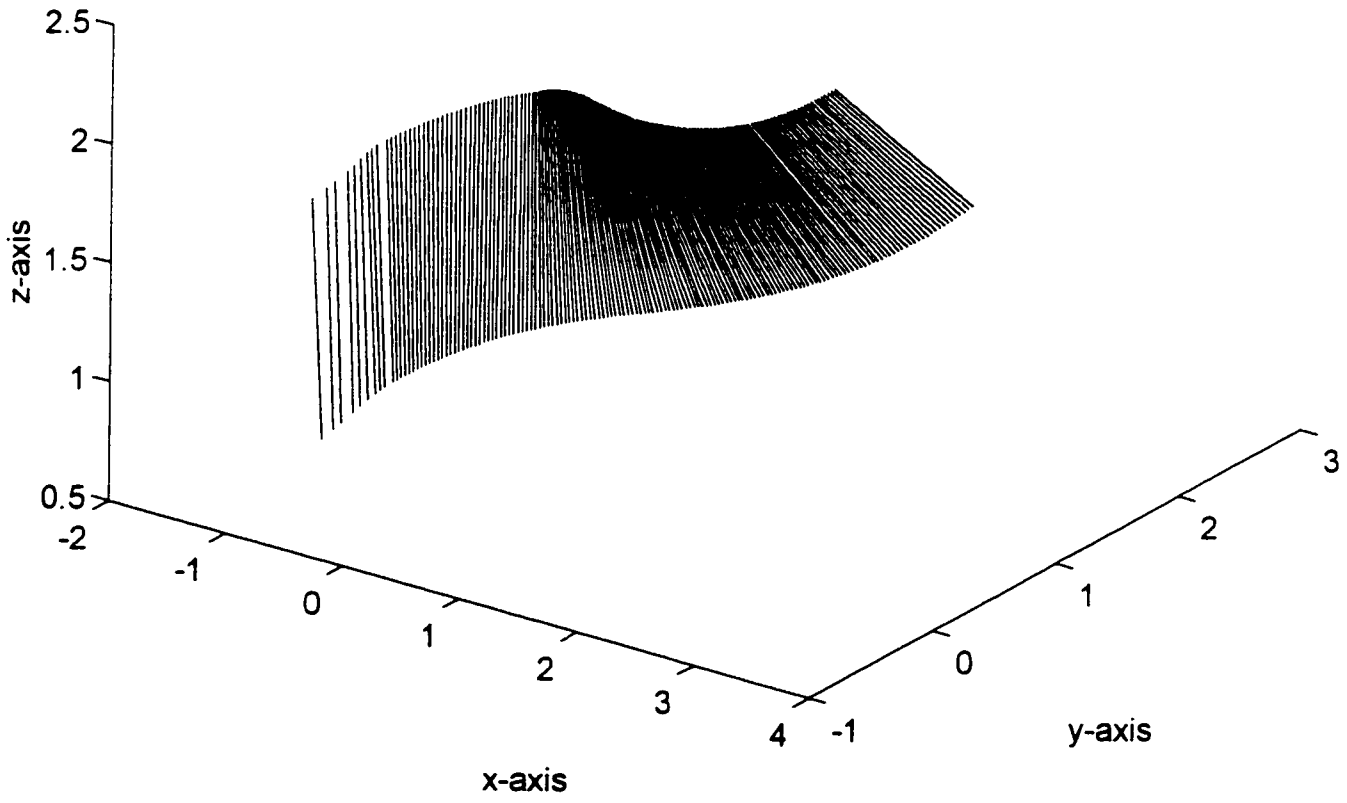


Figure 7.2 The Cutter Locations for Machining the Blade Surface

The linear interpolation and the direct function algorithms interpolated the translational and the rotational movement coordinates in each interpolation period as in the real machining process. The corresponding cutting point (CC point) moved along the machining motion trajectory. The CC point coordinates, for the OM-1 milling centre, was then calculated based on the OM-1 machining motion trajectory model of Eq.(4.19) and Eq.(4.20).

The resulting CC point path in reference to the machine coordinate system is plotted as shown in Fig.7.3. It can be seen that the CC point path is formed by a series of space curve segments, which is a fact that is obvious from the OM-1 non-linear machining motion trajectory model. The enlarged closer view of the CC point path between adjacent machining data points shows that the CC point moves along the non-linear curve segments which deviate from the straight line segments connecting each pair of the machining data points when the rotation pivot (or the translational axes) was linearly interpolated. The machining errors, resulting from the simulation using the linear interpolation method, were determined and a sample of the machining errors are shown in Table 7.1. It shows that the total machining error for each move consists of both the linearity error and the non-linearity error.

The simulation results confirm the machining error analysis indicating that the non-linear curve segments deviate from the linearly interpolated straight line segments, which causes non-linearity errors. The result also shows the inadequacy of the linear interpolation method when applied to current multi-axis CNC machining of convex surfaces.

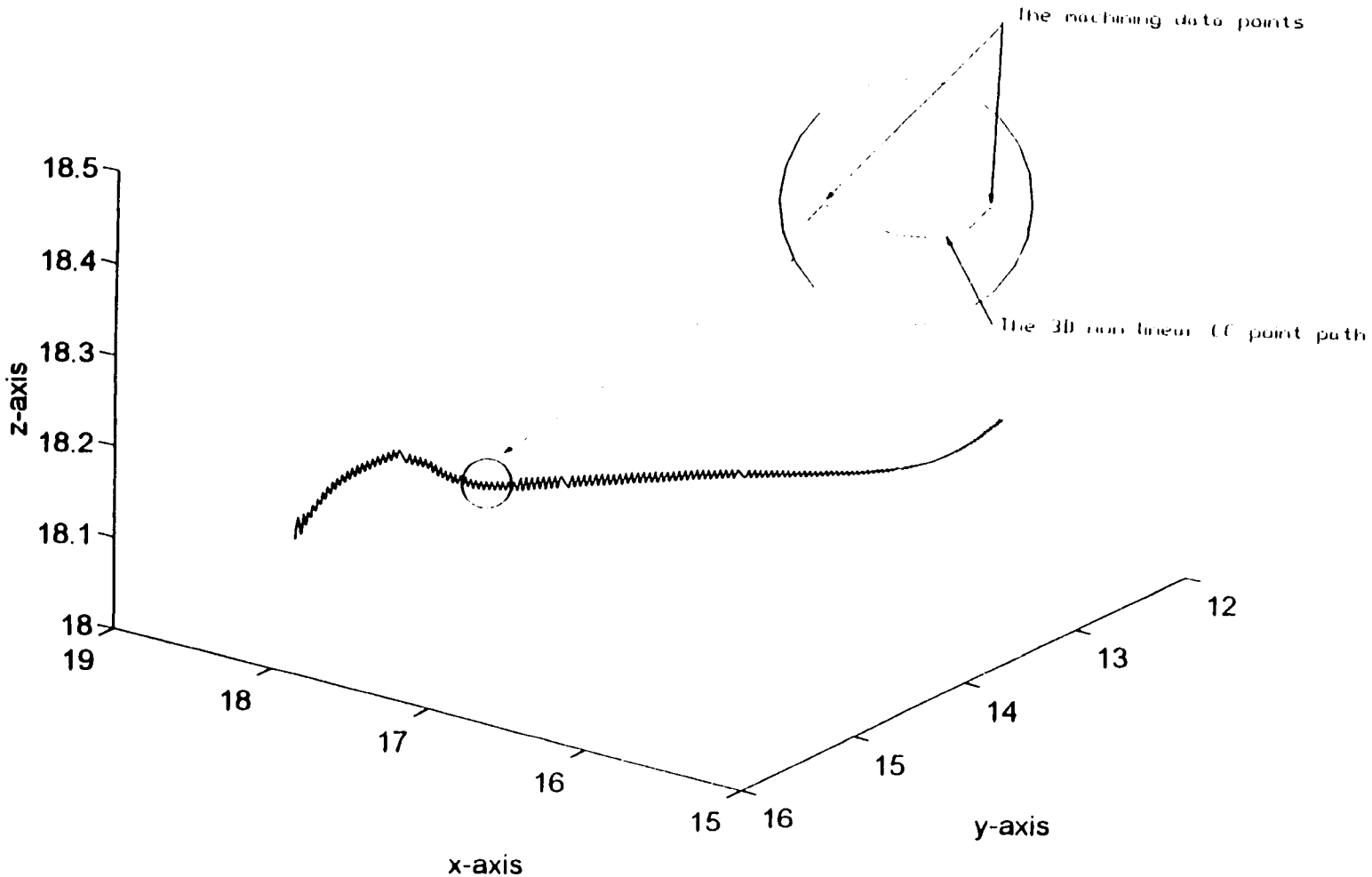


Figure 7.3 The Linearly Interpolated CC Point Path

7.4 Simulation of Machining Blade Surfaces Using the Proposed Interpolator

Applying the proposed interpolator, the '3D combined linear and circular interpolation technique,' to process the same set of NC-codes for machining the airfoil surface, a simulation was performed. Using the OM-1 inverse kinematic model, a set of NC-codes was determined by inverse transforming the available CLDATA as shown in Fig.7.2. The procedure followed the proposed algorithm in section 6.5 and is described next.

The 3D combined linear and circular interpolator is designed to interpolate the machining translational movements, i.e., the 3D coordinates of the rotational movements pivot, such that the rotation pivot moves along the predesigned 3D linear and circular curve segments and the CC point of the cutter moves a small straight line increment from one command to the next along the 3D straight line segment connecting the end CC points. In the interpolation calculation, as a precalculation, the CC point coordinates which correspond to the tool path data points were determined based on the already-known NC-codes and the OM-1 motion trajectory model. Then, these CC point coordinates were linearly interpolated and used in the calculation of interpolated positions of the rotation pivot. The specified feedrate of F = 25 mm/sec (60 ipm) and the interpolation period T=10 ms were used to calculate the interpolation scale factor for each interpolation increment. The initial interpolation coordinates of the rotation pivot P were assigned as the start point coordinates of each segment. In the first interpolation period, based on the linearly interpolated CC point coordinate, the first point of translational movements was interpolated by using the 3D combined linear and circular interpolation formula of Eq.(6.49). The rotational movements

were interpolated in the same way as in the linear interpolation simulation of section 7.3. The interpolated position and rotation coordinates were sent to each machine axis control loop as the reference-words. It should be clarified that the control loops were not simulated in this interpolator function simulation study. After sending the interpolated position coordinates, the interpolated CC point increment was compared with the present segment length, L, which was pre-determined using the CC point coordinates for the segment. Upon the comparison of the interpolation incremented length with the segment length, L, the interpolation was continued since the first increment was shorter than the segment length. In the second interpolation period, the second point of the translational axes and the rotational axes incremented coordinates were interpolated on the basis of the first interpolated coordinates. Next, the total length of the interpolated increments was compared with the present segment length, L. Upon the comparison, the interpolation was performed on the present segment or continued on to the next segment. This iterative interpolation procedure repeated until the interpolated point reached the end point of the segment. When the test showed the difference between the total length of the interpolated increments and the segment length was shorter than one interpolation increment F*T, the feedrate was modified as in the algorithm presented in section 6.5 and the interpolation scale factors were also re-calculated. The 3D linear and circular interpolation was continued to calculate the new increment. Thus, the end point of the segment was reached. The same procedure was performed on each consecutive segment of the complete set of NC-codes.

The interpolated pivot point path in reference to the machine coordinate system is

plotted as shown in Fig. 7.4. It can be seen that the pivot point path is constructed by a series of 3D curved segments, and the enlarged closer view of the pivot point path between adjacent machining data points shows that the pivot point moves along a 3D non-linear curve segment which deviates from the straight line segment connecting the machining data points. The corresponding interpolated CC point path in reference to the machine coordinate system is plotted as shown in Fig. 7.5. It clearly shows that the resulted CC point path is formed by a series of smooth segments. In fact, the CC point path is connected by a series of line segments when the rotation pivot was interpolated along the pre-designed 3D curve segments. Obviously, the non-linearity error for each segment is eliminated. The machining errors resulting from this simulation were also computed as shown in Table 7.1. The resulting machining errors using the proposed interpolator consist solely of linearity error for each move.

Compared with the result obtained from the linear interpolation simulation, the linear interpolation method is simpler, but it results in both linearity errors and non-linearity errors which cause difficulties for ensuring machining precision. Alternatively, the '3D combined linear and circular interpolator ' conducts the CC point to move along straight line segments, so that the only machining error for each segment is the linearity error. Although the proposed interpolator needs more calculations to interpolate the position points as compared to the linear interpolator, the non-linearity error for each segment is eliminated. The simulated results show that the proposed interpolation technique is superior to the linear interpolation method since the resulting motion trajectory causes only the linearity machining error.

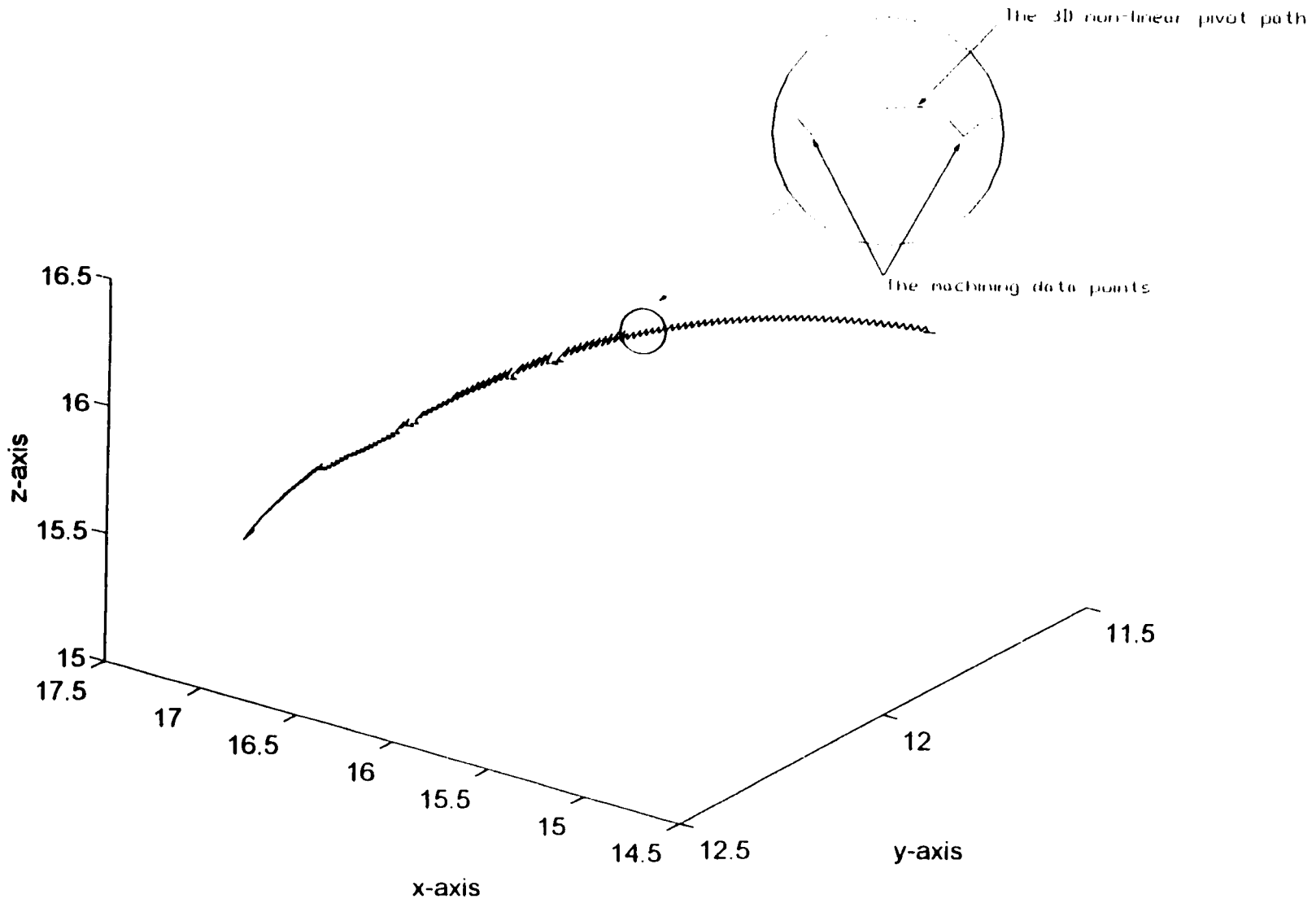


Figure 7.4 The Interpolated Pivot Point Path by the 3D L&C Interpolator

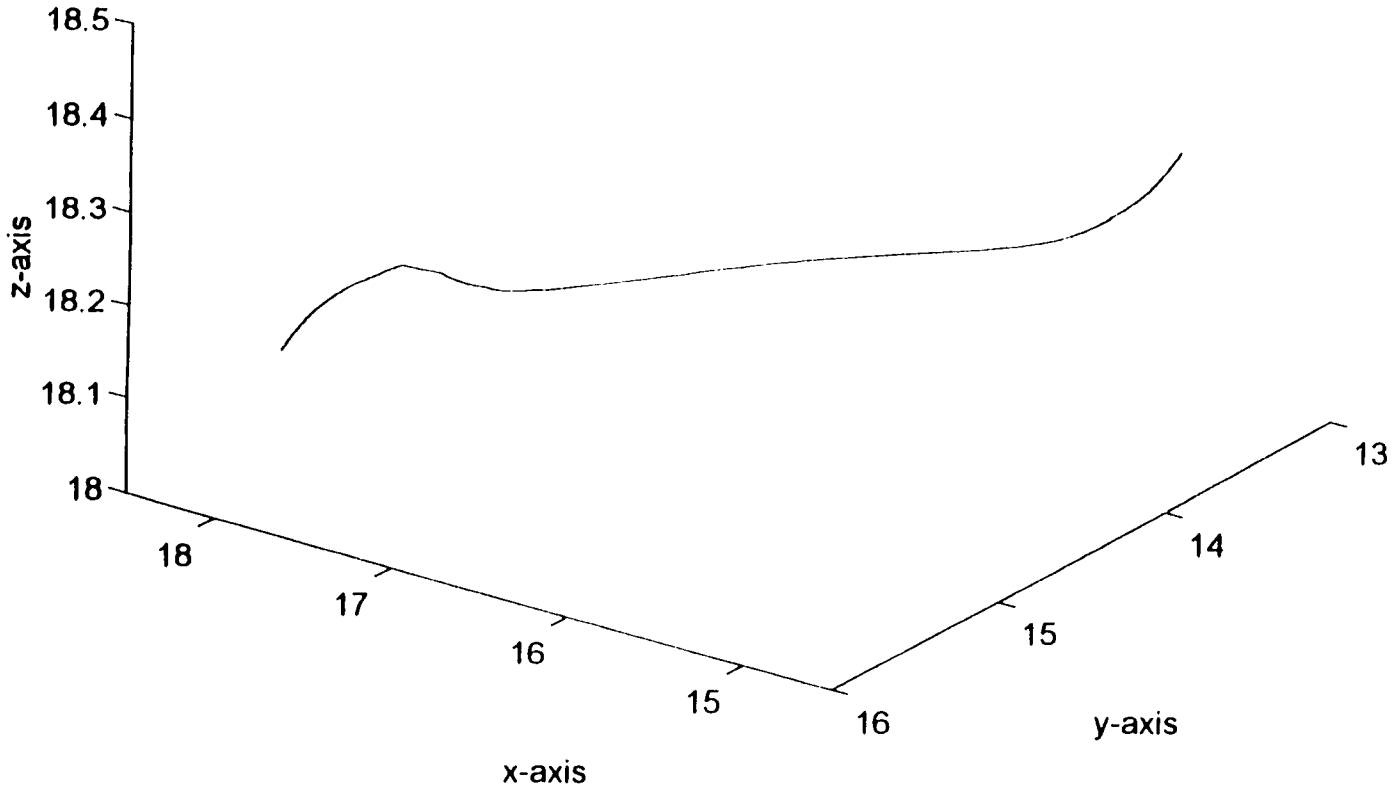


Figure 7.5 The Interpolated CC point Path by the 3D L&C Interpolator

Table 7.1 Comparison of the machining errors from the linear interpolation method and the '3D combined linear & circular interpolation technique'

	The linear interpolator			The 3D combined linear & circular interpolator	
	Linearity error (inch x 10^{-3})	Non-linearity error (inch x 10^{-3})	Total error (inch x 10^{-3})	Linearity error (inch x 10^{-3})	Total error (inch x 10^{-3})
1	0.101	0.346	0.447	0.101	0.101
3	0.051	0.216	0.267	0.051	0.051
9	0.005	0.210	0.215	0.005	0.005
19	0.007	0.197	0.204	0.007	0.007
73	0.002	0.200	0.202	0.002	0.002
82	0.005	0.210	0.215	0.005	0.005
103	0.001	0.230	0.211	0.001	0.001
104	0.025	0.195	0.220	0.025	0.025
122	0.001	0.205	0.206	0.001	0.001
130	0.004	0.210	0.214	0.004	0.004
133	0.003	0.208	0.211	0.003	0.003
139	0.002	0.203	0.205	0.002	0.002
145	0.027	0.181	0.208	0.027	0.027
147	0.029	0.198	0.227	0.029	0.029
152	0.018	0.190	0.208	0.018	0.018
154	0.005	0.200	0.205	0.005	0.005
166	0.004	0.198	0.202	0.004	0.004
178	0.016	0.192	0.208	0.016	0.016
187	0.001	0.205	0.206	0.001	0.001
195	0.025	0.183	0.208	0.025	0.025
245	0.003	0.210	0.213	0.003	0.003

CHAPTER 8

CONCLUSION
AND
RECOMMENDATION FOR FUTURE RESEARCH

8.1 Conclusion

The non-linearity error problem is the main obstacle to ensuring machining precision in multi-axis CNC machining of sculptured surfaces. The nature of the problem is that superimposing rotational movements on translational movements yields non-linear machining motion trajectories, which deviate from the linearly interpolated straight line segments resulting in non-linearity errors. Five-axis rotational movements are kinematically related to cutter orientations, therefore, one of the factors causing non-linearity errors is cutter orientation changes. Another factor contributing the non-linearity error problem is due to the linear interpolation method which is not able to trace the non-linear motion trajectory in five-axis machining. The objectives of this thesis work were to develop novel procedures to eliminate the problem. The first objective was to develop an off-line tool path generation method which tackles the problem from off-line cutter orientation generation. The second objective deals with the development of an on-line 3D interpolator design technique.

As the achievement of the first objective, an off-line tool path generation methodology, the 'minimum error tool path generation method', has been developed. This new method is capable of reducing non-linearity errors to meet machining precision requirements

and has the following characteristics:

1) Ability to obtain high machining precision is increased without undesired consequences, compared to the existing off-line methods.
2) Generation of CLDATA is based on the geometric properties of the machined surfaces, the machining kinematics and motion trajectory;
3) Optimum machining NC-codes can be generated by modifying off-line cutter orientations without inserting extra machining position data;
4) Capacity for generating smoother tool path is improved, thus ensuring feedrate fidelity and resulting in better machined surface finish;
5) Requirement for data storage memory size is reduced;
6) Machining time can be reduced.

A software package for implementing the proposed off-line 'minimum error tool path generation method' has been developed. The software can be used (i) to directly generate NC-codes specifically suitable for OM-1 milling centre; (ii) to generate NC-codes for other OM-series five-axis CNC machine tools by inputting different machine configuration data and setup data. Case studies were performed by applying the proposed 'minimum error tool path generation method'. It was verified through simulation that the new method increased machining precision from 0.0127mm (0.0005") to 0.005 mm (0.0002"). In other words, the proposed method increased the machining precision by an amount of 30% without undesired consequences such as the incorrect cutter orientation and the feedrate fidelity problems, compared with the result of 'linearization process'.

An on-line 3D interpolator design technique for solving the multi-axis CNC machining non-linearity error problem has been proposed which fulfills the second objective of this thesis. Conventional CNC machining systems support functions of 2D circular and 3D linear interpolations, which are unfortunately inadequate for the machining of highly complex 3D surfaces. A 3D software circular interpolation principle has been developed in this thesis, which is capable of tracking spherical curves with low position errors and uniform feedrate. To verify this 3D software circular interpolation, a computer simulation of interpolating a 3D curve has been performed. The interpolated space path shown in Fig. 6.4 illustrates the new interpolator's 3D circular interpolation capability, and the interpolation errors analysis results given in Table 6.1 showed small curve tracking errors. On the basis of this software-based 3D circular interpolator, a '3D combined linear and circular (L&C) interpolation technique' has been proposed, which provides a solution to the multi-axis CNC machining non-linearity error problem and has the following characteristics:

1) Applicable to generating command signals for accurate position-tracking of 3D non-linear trajectories;
2) Applicable to eliminating non-linearity errors in five-axis CNC machining;
3) Applicable to machining 3D curves on five-axis machine tool systems of all types;
4) Uniform machining feedrate for each move can be obtained;
5) Small data storage memory are required.

A software interpolation routine which implements the '3D combined linear and circular interpolation technique' has been developed. The software is applicable to the

machining on the rotary-table type five-axis CNC machine tools, and can be extended to other types of five-axis CNC machine tools. Computer simulations have been performed to verify the proposed '3D combined linear and circular interpolation technique' and to compare it with the linear interpolation method. In the simulation of the machining process in section 7.3, the elimination of non-linearity errors were conducted successfully. Such capability is unattainable using the existing linear interpolation method or using the 'linearization processes'.

The results of the research work in this thesis are innovative and contribute solutions to present day concerns in the manufacturing industry. The 'minimum error tool path generation method' is readily applicable in off-line tool path generation for OM-series five-axis CNC machine tools, and can be used to increase machining precision considerably. The off-line method can be further extended to all type five-axis machine tool configurations. The proposed '3D combined linear and circular interpolation technique' can be used in five-axis CNC machine tool systems once implemented and tested in the real-time machining environment. The proposed technique possesses the ability of eliminating non-linearity errors, and if the proposed technique is verified by successful experimental study and proves to be effective in on-line application, the benefits to industry would not only include reduced machining time and greater precision, but also reduced post-machining processing time.

8.2 Recommendation of Future Research

The following topics are suggested for further research to expand the present work:

1. Generalization of the 'minimum error tool path generation method' software:

   The 'minimum error tool path generation method' is machine type-specific since it uses the individual machine's kinematic models and motion trajectory model. The software of implementing the off-line method developed in this thesis contains the functions and machine configuration data which are specific for use with OM-1 milling centre. In order to adapt the off-line method to different machine tool configurations and machining set-up, the software requires extension to different machine kinematic models and machine motion trajectory models. Therefore, for future work, it is recommended that the software be extended into a general application which can apply the proposed off-line method to all types of five-axis CNC machine tool configurations.

2. Experimental study of the 3D combined linear and circular interpolator:

   The simulation of using the '3D combined linear and circular interpolation technique' has been performed, which demonstrated the ability of the interpolator to eliminate non-linearity errors. To verify and evaluate the whole performance of the proposed interpolator not only from the machining errors issue but also from actual machining time concerns, experimental studies are required. Through experiments, the adaptation of the proposed interpolation technique to CNC control systems can be implemented, and the required machining time and position tracking accuracy can be verified using the real-time feedrate and cutting force control. Experimental study can be performed by modifying the controllers of five-axis CNC machine tool systems.

3. Feedrate study of the 3D combined linear and circular interpolator:

The '3D combined linear and circular interpolator' is capable of driving the CC point along the space straight line segments with constant feedrate, but without considering the feedrate continuity between segments. In addition, the feedrate adaptation procedure in the proposed interpolator is not a best solution to ensure uniform feedrate in each machining move. Therefore, it is recommended that research work is continued on the subject of feedrate continuity and control in machine CNC systems.

BIBLIOGRAPHY

1. Choi, B. K., Lee, C. S., Hwang, J. S. and Jun, C. S., 1988, " Compound Surface Modelling and Machining", *Computer Aided Design*, Vol. 20, No.3, pp. 127-136.

2. Wu , C. Y. , 1994, "" Arbitrary Surface Flank Milling of Fan, Compressor, and Impeller Blades", *ASME , the International Gas Turbine and Aeroengin congress and Exposition, The Hague, netherlands, June 13-16, pp.* 1-7.

3. Bedi, S., Gravelle, S. and Chen, Y. H., 1997, " Principle Curvature Alignment Technique for Machining Complex Surfaces", *ASME, Journal of Manufacturing Science and Engineering* ,Vol. 119, No. 4(B), pp. 756-765.

4. Vickers, G. W. and Quan, K. W., 1989, " Ball-mills Versus End-mills for Curved Surface Machining", *Transaction of the ASME, Journal of Engineering for Industry*, Vol. 111, pp. 22-26.

5. Renker, H. J., 1993, "Collision-Free Five-Axis Milling of Twisted Ruled Surfaces", *Annals of the CIRP*, Vol. 42, No.1.

6. Liu, X. W., 1994, "*Numerical Control Machining Theory and NC programming Technology", (Chinese)*, Manufacturing Industrial Press.

7. Koren, Y., 1995, " Five-axis Surface Interpolators ", *Annals of the CIRP*, Vol. 44/1, pp. 379-382

8. Marciniak, K., 1991, "*Geometric Modelling for Numerical Controlled Machining*", Oxford University Press.

9. Chou, J. J and Yang, D. C. H., 1991, " Command Generation for Three-Axis CNC Machining", *Journal of Engineering for Industry*, Vol. 113, pp. 305-310

10. "*CATIA II User Manual*" , 1985, Version 2, Release 1, Dassault Systems.

11. "*CATIA IV User Manual*" , 1996, Version 4.1.8 , Dassault Systems.

12. "*SmartCAM Programming Manual*", 1988, Point Control, Oregan.

13. Elber, G. and Cohen, E. , 1994, " Toolpath Generation for Freeform Surface Models", *Computer Aided Design*, Vol. 26, No. 6, pp. 490-496.

14. Wysocki, D. A., 1987, " Generation, Verification, and Correction of Numerical Control Tool Paths", *Master Thesis, Michigan State University.*

15. Loney, G. C. and Ozsoy, T. M., 1987, "NC Machining of Free Form Surfaces", *Computer Aided Design*, Vol. 19, No. 2, pp. 85-90.

16. "*Unigraphics II User Manual* ", 1990, McDonnel Douglas Manufacturing System.

17. Oliver, J. H., Wysocki, D. A. and Goodman, E. D., 1993, " Gouge Detection Algorithms for Sculptured Surface NC Generation", *Journal of Engineering for Industry*, Vol. 115, pp.139-143.

18. Huang, Y. and Oliver, J. H., 1992, "Non-constant Parameter NC Tool Path Generation on Sculptured Surfaces ", *ASME, Computers in Engineering*, Vol. 1, pp. 411-419.

19. Marciniak, K., 1987, "Influence of Surface Shape on Admissible Tool Positions in 5-Axis Face Milling", *Computer Aided Design*, Vol. 19, No. 5, pp. 233-236.

20. Li, S. X. and Jerard, R. B., 1994, " 5-Axis Machining of Sculptured Surfaces with a Flat-End Cutter", *Computer Aided Design*, Vol. 26, No. 3, pp. 165-178.

21. Choi, B. K., Park, J. W. and Jun, C. S., 1993, " Cutter Location Data Optimization in 5-Axis Surface Machining", *Computer Aided Design*, Vol. 25, No. 6, pp. 377-386.

22. Cho, H. D., Jun, Y. T. and Yang, M. Y., 1993, "Five-Axis Milling for Effective Machining of Sculptured Surfaces ", *International Journal of Production and Research*, Vol. 31, No. 11, pp. 2559-2573.

23. Jensen, C. G. and Anderson, D. C., 1993, " Accurate Tool Placement and Orientation for Finish Surface Machining", *Journal of design and manufacturing*, Vol. 3, pp. 251-261.

24. Lee, Y. S. and Chang, T. C., 1996, "Machined Surface Error Analysis for 5-Axis Machining", *International Journal of Production & Research*, Vol.34, no. 1, pp.111- 135.

25. Kruth, J. P. and Klewais, P.,1994, "Optimization &Dynamic Adaptation of the Cutter Inclination During Five-Axis Milling of Sculptured Surfaces ", *Annals of the CIRP*, Vol. 43/1, pp. 443-448.

26. Lee, Y. S. and Chang, T. C., 1995, "2-Phase Approach to Global tool Interference Avoidance in 5-axis Machining", *Computer Aided Design*, Vol. 27, No. 10, pp. 715-729.

27. Rao, N., Ismail, F., and Bedi, S., " 5-Axis Sculptured Surface Machining Using the Principal Axis Method". *Submitted to International Journal of Advanced Manufacturing Technology.*

28. Liu, X. W. , 1995, "Five-axis NC Cylindrical Milling of Sculptured Surfaces", *Computer Aided Design*, Vol. 27, No.12, pp. 887-894.

29. Liu, X. W. , Yuan, Z. M. and Liu, H. M. , 1994, " Interpolating Surfaces in Practical Engineering", (in Chinese), *Mechanical and Eletronic Engineering*, Vol. 1. pp. 21-23.

30. Morishige, K., Takeuchi, Y. and Kase, K., 1999, "Tool Path Generation Using Cspace fro 5-Axis Control Machining", *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, Vol. 121, pp.144 - 149.

31. "*Automation Intelligence Generalized Postprocessor Reference Manual* ", 1996 , IntelliPost.

32. " *Vangard Custom PostprocessorReference Manual* ", 1986.

33. " *Vangard Custom PostprocessorReference Manual* ", 1996.

34. " *Ominimill Custom Postprocessor Reference Manual*", 1992.

35. " *Bosto Custom Postprocessor Reference Manual*", 1990.

36. "*AIX Numerical Control Post Generator* ", 1996.

37. "*ICAM Post: NC Pstprocessor Generator* ", 1996, ICAM Technologies Corporation.

38. Takeuchi, Y., Shimizu, H., Idemura, T., Watnabe, T., and Ito,T., 1990, "5-Axis Controlled Machining Based on Solid Models ", *Journal of the Japan Society for Precision Engineering*, 56, 2063, pp. 111-116

39. Paul, R. P., 1981," *Robot Manipulators: Mathematics, Programming, and Control*", The MIT Press.

40. Lee, C.S.G., and Ziegler, M., 1984, "Geometric Approach in Solving Inverse Kinematics of PUMA Robots ", *IEEE Transactions on Aerospace & Electronic Systems*, Vol. aes-20.

41. Chou, J. J and Yang, D. C. H. , 1992, "On the Generation of Coordinated Motion of Five-Axis CNC/CMM Machines", *Journal of Engineering for Industry*, Vol.114, pp.15-.

42. Lin, R. S., and Koren, Y. , 1994, "Real-Time Five Axis Interpolator for Machining Ruled Surfaces", *ASME Winter Annual Metting*, DSC-Vol. 55-2, pp. 951-960.

43. Masory, O. and Koren, Y. , 1982, " Reference-Word Circular Interpolatiors for CNC Systems", *Transaction of the ASME, Journal of Engineering for Industry*, Vol. 104, pp. 401-405.

44. Koren, Y., 1983, "*Computer Control of Manufacturing Systems*", McGraw-Hill, Inc.N.Y.

45. Mayorov, F. V., 1964, "*Digital Differential Analyzer*", Iliffe Books, London, England.

46. Sizer, T. R. H., 1968, "*The Digital Differential Analyzer*", Chapman & Hall, London, England.

47. Danielsson, P. E., 1970, "Incremental Curve Generation ", *IEEE Transaction on Computers*, Vol. c-19, pp. 783-793.

48. Milner, D. A., 1976, "Some Aspects of Computer Numerical Control with Reference to Interpolation" , *ASME Journal of Engineering for Industry*, pp. 883-889, Aug.

49. Bi, C. N. and Ding, N. J. , 1993, " *Modern Numerical Controlled Machine Tools", (Chinese)*, Mechanical Engineering Press, Beijing.

50. Li, C. R. and Wu, W. Q., 1996, "*Computer Numerical Control of Machine Tools" (Chinese)*, North-West Poletechnic University Press, Xian.

51. Koren, Y., 1976, " Interpolator for A Computer Numerical Control System", *IEEE Trans. on Computer*, Vol. C-25, No. 1, pp. 32-37.

52. Koren, Y. and O. Masory, 1981, "Reference-Pulse Circular Interpolator for CNC Systems", *ASME, Journal of Engineering for Industry*, Vol. 103, pp. 131-136.

53. Gan, J. G. and Woo , T. C. , 1992, " Error-Free Interpolation of Parametric Surfaces", *ASME Journal of Engineering for Industry*, Vol. 114, pp. 271-276.

54. Sata, T., F. Kimura, N. Okada and M. Hosaka, 1981, " A new Method of NC Interpolation for Machining the Sculptured Surface", *Annals of the CIRP*, Vol.30/1.

55. Stadelmann, R. , 1989, " Computation of Nominal Path Values to Generate Various Special Curves for Machine ", *Annals of the CIRP*, Vol. 38/1, pp. 373-376.

56. Makino, H. , 1988, " Clothoidal Interpolation - A New Tool for High-Speed Continuous Path Control", *Annals of the CIRP*, Vol. 37/1, pp. 25-28

57. Papaioannou, S. G. and D. Kiritsis, 1988, " Computer-Aided Manufacture of High Precision Cams", *Transaction of the ASME, Journal of Engineering for Industry*, Vol. 110, pp. 352-358.

58. Huang, J. T. and Yang, D. C. H., 1992, "A generalized interpolator for command generation of parametric curves in computer-controlled machines", *Jap./USA Symp. Flexible Automation* Vol. 1, pp. 393-399.

59. Renner, G. and Pochop ,V. , 1981, " A New Method for Local Smooth Interpolation", *Proc. Eurographics.*

60. Renner, G. , 1982, " A Method of Shape Description for Mechanical Engineering Practice", *Computer Industry,* Vol. 3, pp. 137-142.

61. Wang, F. C. and Yang, D. C. H. , 1993, " Nearly Arc-Length Parameterized Quintic-Spline Interpolation for Precision Machining", *Computer Aided Design*, Vol. 25, no.5. pp. 281-288.

62. Kiritsis, D., 1994, " High Precision Interpolation Algorithm for 3D Parametric Curve Generation", *Computer Aided Design*, Vol. 26, no.11, pp. 850-856.

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

64. Lo, C. C. and Hsiao, C. Y.,1998, "CNC machine tool interpolator with path compensation for repeated contour machining", *Computer-Aided Design*, Vol. 30, No. 1, pp. 55 - 62.

65. Shpitalni, M., Koren, Y. and Lo, C. C. , 1994, " Realtime Curve Interpolators", *Computer-Aided Design*, Vol. 26, No. 11. pp.832 - 838.

66. Lo, C. C., 1998, "A new approach to CNC tool path generation", *Computer-Aided Design*, Vol. 30, No. 8, pp. 649 - 655.

67. Lynch, M. " *Computer Numerical Control for Machining* ", McGraw-Hill, Inc. 1992.

68. Craig, J. J., 1989, " *Introduction to Robotics: Mechanics and Control* ", 2nd ed. Addison Wesley, 09528.

69. Zeid, L., 1991, "*CAD/CAM Theory and Practice*", McGraw-Hill, Inc. N.Y. 10020.

70. Faux, I. D., and Pratt, M. J., 1979, "*Computational Geometry for Design and Manufacture*", Halsted Press, A Division of John Wiley & Sons.

71. Wang, Y. Z. , 1995, "*NC Technology of Machine Tools*" (in Chinese), Harebin Politenique University Press.

72. Liu, Y. U. and Lei, X. D., 1998, "Machine Tool Computer Numerical Control and Applications" (in Chinese), Manufacturing Industrial Press.

73. Li, H. S., Chou, X. R. and Guo, W. C., 1997, "NC Principles and Systems" (in Chinese), China Numerical Control Trainning Network, Manufacturing Industrial Press.

74. Huang, J. T. , 1992, " Design and application of a new CNC command generator for CAD/CAM integration", *Ph. D thesis, University of California at Los Angeles, USA.*

75. Rogers, D. F. and Adams, J. A., 1976, "*Mathematical Elements for Computer Graphics*", McGraw-Hill, Inc. New York, 75-29930.

Appendix A

Development of Kinematic Models of OM-1 Milling Centre

1. Development of the rotation transformation matrix

In representing the orientation of a body with an attached coordinate system {B} in reference to a coordinate system {A}, a set of three vectors may be used to construct a 3 x 3 matrix with the three vectors as its columns [68]:

$$ {}^{A}R_{B} = \left[{}^{A}x_{B} \quad {}^{A}y_{B} \quad {}^{A}z_{B} \right] \tag{A.1} $$

where, ${}^{A}R_{B}$ is called a rotation matrix which describes the orientation of {B} and the body relative to {A}. ${}^{A}x_{B}$, ${}^{A}y_{B}$, ${}^{A}z_{B}$ denote the unit vectors giving the principle directions of coordinate system {B} in terms of coordinate system {A}.

Since the rotation matrix has orthonormal columns, the inverse of the rotation matrix is equal to its transpose:

$$ {}^{A}R_{B}^{-1} = {}^{B}R_{A} = {}^{A}R_{B}^{T} \tag{A.2} $$

where, ${}^{B}R_{A}$ is the rotation matrix that describes the orientation of {A} relative to {B}, hence, it is the inverse of the rotation matrix ${}^{A}R_{B}$.

A rotation matrix can be interpreted as a rotation operator which transforms a vector and changes that vector to a new vector by means of a rotation:

$$p_2 = R * p_1 \tag{A.3a}$$

A rotation matrix as an operator, with the axis k is being rotated about, can be represented as:

$$p_2 = Rot(k, \theta) * p_1 \tag{A.3b}$$

where, Rot(k , θ) represents a rotation about the axis k by an amount θ degrees.

When the axis of rotation is one of the principle axes (x, y, z) of a coordinate system with the x axis and y axis are in horizontal and with the z axis is in vertical, the rotation matrix is as the following:

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$Rot(y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{A.4}$$

$$Rot(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To determine the rotation transformation matrix which as an operator transforms the cutter vector about the machine moving axes B_{move} and C_{move} of the OM-1 milling centre, the fixed axes B_{fix} and C_{fix} are assumed to coincide with the initial position of the machine moving axes, B_{move} and C_{move},

and the cutter vector , P^0 , is assumed as a free vector in the space of the part coordinate frame. The rotation transformation matrix that operates the cutter vector P^0 to the vector $P_h{}^0$ about the axis, C_{fix} can be determined as follows.

From the OM-1 milling centre configuration, the C_{fix} originally is horizontal in the direction of the x axis, therefore, the rotation matrix is as:

$$Rot(C_{fix}, C) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cosC & -sinC \\ 0 & sinC & cosC \end{bmatrix} \qquad (A.5)$$

Since the machine C_m axis originally is parallel to the machine x_m axis, and it moves to the machining initial position which is parallel to the spindle, the rotation angle about the C_{fix} should be measured from the original position which is parallel to the machine x_m axis. i.e., the measured C angle should plus the angle, 90° , from the machine x_m axis direction to the machine spindle. Thus, one may obtain:

$$Rot(C_{fix}, C) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(90^\circ+C) & -\sin(90^\circ+C) \\ 0 & \sin(90^\circ+C) & \cos(90^\circ+C) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -sinC & -cosC \\ 0 & cosC & -sinC \end{bmatrix} \qquad (A.6)$$

Similarly, from the OM-1 milling centre configuration, the B_{fix} is vertical in the opposite direction of the z axis, the rotations are in counterclockwise direction. Hence, the rotation matrix for the rotation about the axis, B_{fix} , is as:

$$Rot(B_{fix}, -B) = \begin{bmatrix} \cos(-B) & -\sin(-B) & 0 \\ \sin(-B) & \cos(-B) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cosB & sinB & 0 \\ -sinB & cosB & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (A.7)$$

Therefore, the rotation transformation matrix that rotates the cutter vector in an position to the spindle position about the machine moving axes, B_{move} and C_{move} , is as :

$$\begin{aligned} R &= Rot(C_{move}\ ,\ C)\ Rot(B_{move}\ ,\ -B) \\ &= Rot(B_{fix}\ ,\ -B)\ Rot(C_{fix}\ ,\ C) \\ &= \begin{bmatrix} cosB & -sinB\ .sinC & -sinB\ .cosC \\ -sinB & -cosB\ .sinC & -cosB\ .cosC \\ 0 & cosC & -sinC \end{bmatrix} \end{aligned} \qquad (A.8)$$

2. Development of the inverse kinematic model

The cutter vector p^0 is operated on by the rotation matrix R to the position p_h^1 , one may have:

$$P_h^1\ (\alpha_h\ ,\ \beta_h\ ,\ \gamma_h) = R * P^0\ (\alpha\ ,\ \beta\ ,\ \gamma) \qquad (A.9)$$

where, P_h^1 is parallel to the machine spindle, hence:

$$P_h^1\ (\alpha_h\ ,\ \beta_h\ ,\ \gamma_h) = P_h^1\ (\ 90^\circ\ ,\ 180^\circ\ ,\ 90^\circ\ ) \qquad (A.10)$$

Thus, one may have:

$$\begin{bmatrix} \cos 90^0 \\ \cos 180^0 \\ \cos 90^\circ \end{bmatrix} = \begin{bmatrix} cosB & -sinBsinC & -sinBcosC \\ -sinB & -cosBsinC & -cosBcosC \\ 0 & cosC & -sinC \end{bmatrix} \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix} \qquad (A.11)$$

that is,

$$cosB\ \cos\alpha\ -\ sinB\ \ sinC\ \cos\beta\ -\ sinB\ \ cosC\ \cos\gamma\ =\ 0 \qquad (A.12a)$$

and,

$$cosC \cos\beta - sinC \cos\gamma = 0 \qquad \text{(A.12.b)}$$

From (A.12.b), one may have:

$$\tan C_m = \frac{\cos\beta}{\cos\gamma}$$

$$\sin C_m = \frac{\cos\beta}{\cos\gamma} cosC \qquad \text{(A.12.c)}$$

$$\cos C_m = \frac{\cos\gamma}{\sqrt{\cos^2\gamma + \cos^2\beta}}$$

Substitute (A.12.c) into (A. 12.a), one may have:

$$\begin{aligned} cosB \cos\alpha &= sinB\,(sinC \cos\beta + cosC \cos\gamma) \\ &= sinB \sqrt{\cos^2\gamma + \cos^2\beta} \end{aligned} \qquad \text{(A.13)}$$

or,

$$\tan B = \frac{\cos\alpha}{\sqrt{\cos^2\gamma + \cos^2\beta}} \qquad \text{(A.14)}$$

The cutter vector position initially is p^o (x, y, z), after rotated by the transformation matrix R, it is arrived at p_h^1 (x_h^1 , y_h^1, z_h^1) in reference to the initial part coordinate frame. In matrix form, it is:

$$\begin{bmatrix} x_h^1 \\ y_h^1 \\ z_h^1 \end{bmatrix} = \begin{bmatrix} cosB & -sinBsinC & -sinBcosC \\ -sinB & -cosBsinC & -cosBcosC \\ 0 & cosC & -sinC \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{A.15}$$

Thus, one may have:

$$\begin{aligned} x_h^1 &= x\ cosB - y\ sinB\ \ sinC - z\ sinB\ \ cosC \\ y_h^1 &= -x\ sinB - y\ cosB\ \ sinC - z\ cosB\ \ cosC \\ z_h^1 &= y\ cosC - z\ sinC \end{aligned} \tag{A.16}$$

Since the part coordinate frame is also rotated, the frame origin is moved to a new position, one have:

$$\begin{bmatrix} O_x^1 \\ O_y^1 \\ O_z^1 \end{bmatrix} = \begin{bmatrix} cosB & -sinBsinC & -sinBcosC \\ -sinB & -cosBsinC & -cosBcosC \\ 0 & cosC & -sinC \end{bmatrix} \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} \tag{A.17}$$

From the machine set up data, the fixture thickness, F , and the part stacking position data, G , the initial position of the part coordinate frame origin is as:

$$\begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} = \begin{bmatrix} \frac{F}{2}+G \\ 0 \\ 0 \end{bmatrix} \tag{A.18}$$

Hence, one may have:

$$O_x^1 = (\frac{F}{2} + G)\cos B$$

$$O_y^1 = -(\frac{F}{2} + G)\sin B \qquad \text{(A.19)}$$

$$O_z^1 = 0$$

Therefore, after rotations, the cutter position in reference to the 'fixed' machine coordinate frame is as:

$$x^1 = x_h^1 \pm O_x^1$$

$$y^1 = y_h^1 \pm O_y^1 \qquad \text{(A.20)}$$

$$z^1 = z_h^1 \pm O_z^1$$

where, the plus sign and the minus sign depends on the position of the origin of the part frame in reference to the machine coordinate frame. In our case, the minus applies.

The spindle MCP_s initial position in the 'fixed' machine coordinate frame is at:

$$(x_m , y_m , z_m) = (PB , PC , PB) \qquad \text{(A.21)}$$

where, PB represents the position of the B_m axis pivot in the x_m axis direction which is equal to the spindle initial position in x_m (see Fig. 4.4) and in z_m. PC represents the position of the C_m axis pivot in the y_m axis direction which is equal to the spindle initial position in y_m as shown in Fig.4.4.

To move the cutter vector P_h^1 from the position (x^1 , y^1 , z^1) to the spindle position (PB, PC, PB), the machine should translate the P_h^1 to the spindle by the amount of:

$$\begin{aligned} x_m &= PB \pm x^1 = PB \pm (x_h^1 \pm O_x^1) \\ y_m &= PC \pm z^1 = PC \pm (z_h^1 \pm O_z^1) \\ z_m &= PB \pm y^1 = PB \pm (y_h^1 \pm O_y^1) \end{aligned} \tag{A.22}$$

where, the plus or the minus sign of (x^1 , y^1 , z^1) depends on the relative position of the vector P_h^1 to the spindle, in our case, the minus sign applies.

Substitute Eq.(A.20) and using the rotational axes pivot constant, PB = 15.0" and PC = 11.75", the machine translational variables are obtained as:

$$\begin{aligned} x_m &= PB - [x\cos B - y\sin B\sin C - z\sin B\cos C + (\frac{F}{2}+G)\cos B] \\ y_m &= PC - [y\cos C - z\sin C] \\ z_m &= PB - [-x\sin B - y\cos B\sin C - z\cos B\cos C - (\frac{F}{2}+G)\sin B] \end{aligned} \tag{A.23}$$

Since,

$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1 \tag{A.24}$$

one may have:

$$cosC = \frac{1}{\sqrt{1 + \tan^2 C}} = \frac{\cos\gamma}{\sqrt{1 - \cos^2\alpha}}$$

$$cosB = \frac{1}{\sqrt{1 + \tan^2 B}} = \sin\alpha \qquad (A.25)$$

$$sinB = tanB \; cosB = \cos\alpha$$

$$sinC = tanC \; cosC = \frac{\cos\beta}{\sqrt{1 - \cos^2\alpha}}$$

Substitute Eq.(A.25) into Eq. (A.23), one may obtain:

$$x_m = PB - [x-(y\ tanC + z)\ cosC\ \ tanB\ ]\ cosB - (\frac{F}{2}+G)\ cosB$$

$$= PB - [x - y\frac{\cos\beta}{\sqrt{1-\cos^2\alpha}}\ \frac{\cos\alpha}{\sqrt{1-\cos^2\alpha}} - z\frac{\cos\gamma}{\sqrt{1-\cos^2\alpha}}\ \frac{\cos\alpha}{\sqrt{1-\cos^2\alpha}}]\sqrt{1-\cos^2\alpha} - (\frac{F}{2}+G)\ \sin\alpha$$

$$y_m = PC - [\ y\ cosC\ - z\ sinC\ ]$$

$$= PC - [\ y - z\frac{\cos\beta}{\cos\gamma}\ ]\ \frac{\cos\gamma}{\sqrt{1-\cos^2\alpha}}$$

$$z_m = PB + [x\ tanB + y\ (tanC + z)\ cosC\ ]\ cosB + (\frac{F}{2}+G)\ sinB$$

$$= PB + x\ \frac{\cos\alpha}{\sin\alpha} + y\ (\frac{\cos\beta}{\cos\gamma} + z)\ \frac{\cos\gamma}{\sin\alpha}\ ]\ \sin\alpha + (\frac{F}{2}+G)\ \cos\alpha$$

Therefore, the inverse kinematic model for the OM-1 5-axis CNC milling centre is obtained as the following:

$$x_m = PB - x\sqrt{1-\cos^2\alpha} + y\frac{\cos\alpha\cos\beta}{\sqrt{1-\cos^2\alpha}} + z\frac{\cos\alpha\cos\gamma}{\sqrt{1-\cos^2\alpha}} - (\frac{F}{2}+G)\sqrt{1-\cos^2\alpha}$$

$$y_m = PC - y\frac{\cos\gamma}{\sqrt{1-\cos^2\alpha}} - z\frac{\cos\beta}{\sqrt{1-\cos^2\alpha}}$$

$$z_m = PB + x\cos\alpha + y\cos\beta + z\cos\gamma + (\frac{F}{2}+G)\cos\alpha \qquad \text{(A.27)}$$

$$B_m = \tan^{-1}(\frac{\cos\alpha}{\sqrt{\cos\beta^2 + \cos\gamma^2}})$$

$$C_m = \tan^{-1}(\frac{\cos\beta}{\cos\gamma})$$

where, x, y, z are cutter position coordinates and $\cos\alpha$, $\cos\beta$, $\cos\gamma$ are the direction cosine of the cutter orientation coordinates in reference to the part coordinate system. x_m , y_m , z_m are machine translational movement variables and B_m and C_m are machine rotational movement variables in reference to the machine coordinate system.

3. Development of the forward kinematic model

The forward kinematics deal with the problem of determining cutter variables from the machine variables. From Eq.(A. 9):

$$P_h^1(\alpha_h, \beta_h, \gamma_h) = R * P^o(\alpha, \beta, \gamma) \qquad \text{(A.9)}$$

one may have:

$$P^o(\alpha, \beta, \gamma) = R^{-1} P_h^1(\alpha_h, \beta_h, \gamma_h) \qquad \text{(A.28)}$$

where, R^{-1} is the inverse of the rotation matrix R. Since a rotation matrix has the property that its inverse equal to its transpose as mentioned in Eq.(A. 2) above, one may have:

$$R^{-1} = R^{T} = \begin{bmatrix} cosB & -sinB & 0 \\ sinB\ sinC & -cosB\ sinC & cosC \\ -sinB\ cosC & -cosB\ cosC & -sinC \end{bmatrix} \tag{A.29}$$

Thus, one have:

$$\begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix} = \begin{bmatrix} cosB & -sinB & 0 \\ sinB\ sinC & -cosB\ sinC & cosC \\ -sinB\ cosC & -cosB\ cosC & -sinC \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \tag{A.30}$$

By solving the matrix equation of (A.30), one may obtain:

$$\begin{aligned} \cos\alpha &= sinB \\ \cos\beta &= cosB\ sinC \\ \cos\gamma &= cosB\ cosC \end{aligned} \tag{A.31}$$

From Eq.(A.15), one may have:

$$P_h^{1}(x_h^{1}, y_h^{1}, z_h^{1}) = R * P^{o}(x, y, z) \tag{A.32}$$

and,

$$P^{o}(x, y, z) = R^{-1} P_h^{1}(x_h^{1}, y_h^{1}, z_h^{1}) \tag{A.33}$$

Hence,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} cosB & -sinB & 0 \\ sinB\ sinC & -cosB\ sinC & cosC \\ -sinB\ cosC & -cosB\ cosC & -sinC \end{bmatrix} \begin{bmatrix} x_h^1 \\ y_h^1 \\ z_h^1 \end{bmatrix} \tag{A.34}$$

From Eq.(A. 22), one may have:

$$\begin{aligned} x_h^1 &= PB \mp O_x^1 - x_m \\ y_h^1 &= PB \mp O_y^1 - z_m \\ z_h^1 &= PC \mp O_z^1 - y_m \end{aligned} \tag{A.35}$$

By substituting Eq.(A.35) into Eq. (A.34), one may have:

$$\begin{aligned} x &= x_h^1\ cosB - y_h^1\ sinB \\ &= (PB + (\frac{F}{2}+G)\ cosB - x_m)\ cosB - (PB - (\frac{F}{2}+G)\ sinB - z_m)\ sinB \\ &= -(x_m - PB)\ cosB + (z_m - PB)\ sinB + (\frac{F}{2}+G) \end{aligned} \tag{A.36a}$$

From Eq.(A.19), one may have:

$$\begin{aligned} & O_x^1\ sinB\ sinC + O_y^1\ cosB\ sinC \\ &= (\frac{F}{2}+G)\ cosB\ sinB\ sinC - (\frac{F}{2}+G)\ sinB\ cosB\ sinC \\ &= 0 \end{aligned} \tag{A.36b}$$

Hence,

$$
\begin{aligned}
y &= -x_h^1 sinBsinC - y_h^1 cosBsinC + z_h^1 cosC \\
&= -(PB + O_x^1 - x_m)sinBsinC - (PB + O_y^1 - z_m)cosBsinC + (PC - y_m)cosC \\
&= x_m sinBsinC + z_m cosBsinC + (PC - y_m)cosC - PB(sinB + cosB)sinC
\end{aligned} \tag{A.36c}
$$

From Eq.(A.19), one may have:

$$
\begin{aligned}
& O_x^1 sinBcosC + O_y^1 cosBcosC \\
&= (\frac{F}{2} + G)cosBsinBcosC - (\frac{F}{2} + G)sinBcosBcosC \\
&= 0
\end{aligned} \tag{A.36d}
$$

Hence,

$$
\begin{aligned}
z &= -x_h^1 sinBcosC - y_h^1 cosBcosC + z_h^1 sinC \\
&= -(PB + O_x^1 - x_m)sinBcosC - (PB + O_y^1 - z_m)cosBcosC - (PC - y_m)sinC \\
&= x_m sinBcosC + z_m cosBcosC - PB(sinB + cosB)cosC - (PC - y_m)sinC
\end{aligned} \tag{A.36e}
$$

Therefore, the forward kinematic model of the OM-1 milling centre is obtained as the following equation:

$$
\begin{aligned}
x &= -(x_m - PB)cosB_m + (z_m - PB)sinB_m + (\frac{F}{2} + G) \\
y &= x_m sinB_m sinC_m + z_m cosB_m sinC_m + (PC - y_m)cosC_m \\
&\quad - PB(sinB_m + cosB_m)sinC_m \\
z &= x_m sinB_m cosC_m + z_m cosB_m cosC_m + (y_m - PC)sinC_m \\
&\quad - PB(sinB_m + cosB_m)cosC_m \\
\cos\alpha &= sinB_m \\
\cos\beta &= cosB_m sinC_m \\
\cos\gamma &= cosB_m cosC_m
\end{aligned} \tag{A.37}
$$

where, x, y, z are cutter position coordinates and $\cos\alpha$, $\cos\beta$, $\cos\gamma$ are the direction cosine of the cutter orientation coordinates in reference to the part coordinate system. x_m , y_m , z_m are machine translational movement variables and B_m and C_m are machine rotational movement variables in reference to the machine coordinate system.

Appendix B

Development of Cubic Spline Representation of Tool Path

A common technique is to use a series of cubic spline segments with each segment spanning only two points. The cubic spline is advantageous since it is the lowest degree curve which allows a point of inflection and which has the ability to twist through space. To represent the desired cutting curve on the machined surface and calculate the local surface geometrical properties, cubic spline technique can be used to approximate the desired tool path. Thus, the cubic polynomial is developed. The resultant formula are the same as that by Ragers and Adams[75].

The equation for a single parametric cubic spline segment is given by:

$$P(t) = B_1 + B_2\, t + B_3\, t^2 + B_4\, t^3 \qquad \text{(B.1)}$$

where t is the parameter which varies between the two end-point values t_1 and t_2 corresponding to the two end-points P_1 and P_2 of the cubic segment. By knowing the tangents at the two end-points of the cubic segment, P'_1 and P'_2, and considering the normalized parameters, i.e.,

$$t_1 = 0 \le t \le 1 = t_2 \qquad \text{(B.2)}$$

The cubic polynomial for each cubic segment can be determined as follows.

For t = 0,

$$P(0) = B_1 = P_1$$

$$\left.\frac{dP}{dt}\right|_{t=0} = B_2 + 2B_3\, t + 3B_4\, t^2 \Big|_{t=0} = B_2 = P_1' \tag{B.3}$$

For t = 1,

$$P(1) = B_1 + B_2 + B_3 + B_4 = P_2$$

$$\left.\frac{dP}{dt}\right|_{t=1} = B_2 + 2B_3\, t + 3B_4\, t^2 \Big|_{t=1} = B_2 + 2B_3 + 3B_4 = P_2' \tag{B.4}$$

By solving B_3 and B_4, one may have:

$$B_3 + B_4 = P_2 - P_1 - P_1'$$

$$2B_3 + 3B_4 = P_2' - P_1' \tag{B.5}$$

Hence,

$$B_3 = -3P_1 + 3P_2 - 2P_1' - P_2'$$

$$B_4 = 2P_1 - 2P_2 + P_1' + P_2' \tag{B.6}$$

Therefore, by substituting Eq. (B.3), Eq.(B.4), Eq.(B.5) and Eq.(B.6) into Eq.(B.1), the cubic polynomial is as:

$$P(t) = P_1 + P_1' t + (-3P_1 + 3P_2 - 2P_1' - P_2') t^2 + (2P_1 - 2P_2 + P_1' + P_2') t^3 \tag{B.7}$$

To ensure second-order continuity for a cubic spline, we impose the condition for constant curvature at the internal joint between the two spans, namely:

$$y'' = k = constant$$

where, k represents the curvature at a joint.

From Eq.(B.1), one may have:

$$P''(t) = 2B_3 + 6B_4 t \tag{B.8}$$

Since at the end of the first segment (s_1), $t = t_2 = 1$, one may have:

$$P''(t) = 2B_3 + 6B_4 \tag{B.8a}$$

and at the beginning of the second segment (s_2), $t = t_1 = 0$, on may have:

$$P''(t) = 2B_3 \tag{B.8b}$$

By equating Eq.(B8a) and Eq.(B.8b), one may have:

$$(2B_3 + 6B_4)\big|_{s1} = 2B_3\big|_{s2} \tag{B.9}$$

Thus, From Eq.(B.6) and Eq.(B.9), for joint between s_1 and s_2 , one may have:

$$2(-3P_1+3P_2-2P_1'-P_2')+6(2P_1-2P_2+P_1'+P_2')$$

$$=2(-3P_2+3P_3-2P_2'-P_3')$$

or,

$$6P_1-6P_2+2P_1'+4P_2' = 6P_2+6P_3-4P_2'-2P_3' \tag{B.10}$$

Therefore,

$$3P_1 - 3P_3 = -P_1' - 4P_2' - P_3' \tag{B.11}$$

For a given n = 5 data points, there are n-1 = 4 segments. At the joint between s_2 and s_3, one may have:

$$(2B_3 + 6B_4)\Big|_{s_2} = 2B_3\Big|_{s_3} \tag{B.12}$$

Hence,

$$2(-3P_2+3P_3-2P_2'-P_3')+6(2P_2-2P_3+P_2'+P_3')$$
$$=2(-3P_3+3P_4-2P_3'-P_4')$$

or,

$$6P_2-6P_3+2P_2'+4P_3' = 6P_3+6P_4-4P_3'-2P_4' \tag{B.13}$$

Therefore,

$$3P_2 - 3P_4 = -P_2' - 4P_3' - P_4' \tag{B.14}$$

Similarly, for the joint between s_3 and s_4, one may have:

$$(2B_3 + 6B_4)\Big|_{s_3} = 2B_3\Big|_{s_4} \tag{B.15}$$

Hence,

$$3P_3 - 3P_5 = -P_3' - 4P_4' - P_5' \tag{B.16}$$

Therefore, one may obtain:

$$\begin{aligned} P_1' + 4P_2' + P_3' &= -3P_1 + 3P_3 \\ P_2' + 4P_3' + P_4' &= -3P_2 + 3P_4 \\ P_3' + 4P_4' + P_5' &= -3P_3 + 3P_5 \end{aligned} \tag{B.17}$$

or,

$$\begin{aligned} 4P_2' + P_3' &= -3P_1 + 3P_3 - P_1' \\ P_2' + 4P_3' + P_4' &= -3P_2 + 3P_4 \\ P_3' + 4P_4' &= -3P_3 + 3P_5 - P_5' \end{aligned}$$

In matrix form,one may have:

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} P_2' \\ P_3' \\ P_4' \end{bmatrix} = \begin{bmatrix} -3P_1 + P_3 - P_1' \\ -3P_2 + 3P_4 \\ -3P_3 + 3P_5 - P_5' \end{bmatrix} \tag{B.18}$$

Using M to represents the left coefficient matrix, P' to represents the left tangents column, and N represents the right side matrix, one may have:

$$[\,M\,]\,[\,P'\,] = [\,N\,] \tag{B.19}$$

or,

$$[\,P'\,] = [\,M^{-1}\,]\,[\,N\,]$$

By solving Eq. (B. 19), the tangents at the inner joints of the cubic spline curve can be determined. The coefficients for cubic spline segment, then, can be determined. For example, for segment s_1, the four coefficients, from Eq. (B.3) and Eq.(B.6), are:

$$\begin{aligned} B_1 &= P_1 \\ B_2 &= P_1' \\ B_3 &= -3P_1 + 3P_2 - 2P_1' - P_2' \\ B_4 &= 2P_1 - 2P_2 + P_1' + P_2' \end{aligned} \tag{B.20}$$

In matrix form, one may have:

$$\begin{bmatrix} B_4 \\ B_3 \\ B_2 \\ B_1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_1' \\ P_2' \end{bmatrix} \tag{B.21}$$

In general, for all spans, the tangents at the intermediate joint points can be determined from the following matrix equation:

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 0. \\ . & & & & & & \\ . & & & & & & \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 \end{bmatrix} \begin{bmatrix} P_2' \\ P_3' \\ P_4' \\ . \\ . \\ P_{n-1}' \end{bmatrix} = \begin{bmatrix} 3(P_3-P_1)-P_1' \\ 3(P_4-P_2) \\ 3(P_5-P_3) \\ . \\ . \\ 3(P_n-P_{n-2})-P_n' \end{bmatrix} \qquad \text{(B.22)}$$

The four coefficients for each parametric cubic spline curve can be expressed in the following matrix form:

$$\begin{bmatrix} B_4 \\ B_3 \\ B_2 \\ B_1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P_k' \\ P_{k+1}' \end{bmatrix} \qquad \text{(B.23)}$$

$$(1 \le k \le n-1)$$

Appendix C

Development of A 3D Combined

Linear and Circular Interpolation Principle

The 3D DDA linear interpolation principle is as follows:

$$\begin{aligned} X_{i+1} &= X_i + \lambda_l (x_1 - x_0) \\ Y_{i+1} &= Y_i + \lambda_l (y_1 - y_0) \\ Z_{i+1} &= Z_i + \lambda_l (z_1 - z_0) \end{aligned} \qquad \text{(C.1)}$$

where, (x_0 , y_0 , z_0) and (x_1 , y_1 , z_1) represent the start point and the end point coordinates of the interpolation line segment respectively, (X_i , Y_i , Z_i) represents the present interpolated point. $(X_{i+1}, Y_{i+1} , Z_{i+1})$ represents the next interpolating point coordinate, λ_l represents the linear interpolation scale factor:

$$\lambda_l = \frac{FT}{L} \qquad \text{(C.2)}$$

where, F represents the feedrate, T represents the interpolation period, and L represents the length of the interpolation line segment.

The 3D DDA circular interpolation principle developed in section 6.5 is as follows:

$$
\begin{aligned}
X_{i+1} &= X_i + \lambda_c \left[n_z (Y_i - y) - n_y (Z_i - z) \right] \\
Y_{i+1} &= Y_i + \lambda_c \left[n_x (Z_i - z) - n_z (X_{i+1} - x) \right] \\
Z_{i+1} &= Z_i + \lambda_c \left[n_y (X_{i+1} - x) - n_x (Y_{i+1} - y) \right]
\end{aligned}
\qquad \text{(C.3)}
$$

where, (X_i , Y_i , Z_i) is the i-th interpolated point. (X_{i+1} , Y_{i+1} , Z_{i+1}) is the (i+1)-th interpolated point. (x, y, z) is the interpolated circle centre's coordinate. λ_c is the circular interpolation scale factor which can be determined from the feedrate, the interpolation period and the interpolation circle radius. (n_x , n_y , n_z) are the unit normal vectors of the coordinate planes in the Cartesian coordinate system.

Five-axis CNC machining consists of the simultaneous and coupled three translational movements and two rotational movements. In conventional five-axis CNC machining, the rotational movements are about a moving pivot which moves translationally in space. In other words, the rotation pivot's motion is interpolated based on the 3D linear interpolation principle as given by Eq.(C.1). The rotational movements are the 3D circular movements about the rotation pivot, which form a spherical path. This spherical motion can be interpolated by using the 3D DDA circular interpolation principle as given in Eq.(C.2). The five-axis simultaneous and coupled 3D translational and rotational movements result in the 3D non-linear motion trajectory. To conduct the cutting point moving along the space line segments connecting each consecutive machining data points (i.e., to eliminate the non-linearity errors), one may coordinate the rotation pivot to move along a predesigned 3D curve, say, a 3D combined linear and circular path. From the 3D linear and 3D circular interpolation principle, the spatial non-linear path can be constructed by coordinating linearly the 3D

circular interpolation centre. The development of the 3D combined linear and circular interpolation principle is as follows.

By substituting Eq. (C.1) into Eq. (C.3) as the interpolation circle centre, one may obtain the interpolated x coordinate:

$$\begin{aligned} X_{i+1} &= X_i+\lambda_c[\ n_z(Y_i-(y_i+\lambda_l(y_1-y_0)))-n_y(Z_i-(z_i+\lambda_l(z_1-z_0)))\ ] \\ &= X_i+\lambda_c[\ n_z(Y_i-y_i)-n_y(Z_i-z_i)+\lambda_l(n_y(z_1-z_0)-n_z(y_1-y_0))\ ] \end{aligned} \tag{C.4}$$

From the 3D DDA circular interpolation, replacing (X_i, Y_i, Z_i) by the initial point (x_0, y_0, z_0), and replacing (X_{i+1}, Y_{i+1}, Z_{i+1}) by the interpolated point (x_i, y_i, z_i), one may obtain the following equation:

$$x_i - x_0 = \lambda_c\ [\ n_y(z_i-z_0) - n_z(y_i-y_0)] \tag{C.5}$$

Thus, by replacing the interpolated point with the final end point (x_1, y_1, z_1) of the segment, and using the linear interpolation scale factor λ_l, one may obtain:

$$x_1 - x_0 = \lambda_l[\ n_y(z_1-z_0) - n_z(y_1-y_0)] \tag{C.6}$$

By substituting Eq.(C.6) into Eq.(C.4), the (i+1)-th interpolated x coordinate is obtained as:

$$X_{i+1} = X_i + \lambda_c\ [(x_1 - x_0) + n_z(Y_i - y_i) - n_y(Z_i - z_i)] \tag{C.7}$$

From Eq.(C.1) and Eq.(C.3), the interpolated y coordinate may be obtained as:

$$\begin{aligned} Y_{i+1} &= Y_i + \lambda_c[\ n_z(Z_i-(z_i+\lambda_f(z_1-z_0)))-n_z(X_{i+1}-(x_{i+1}+\lambda_f(x_1-x_0)))\ ] \\ &= Y_i + \lambda_c[\ n_z(Z_i-z_i)-n_z(X_{i+1}-x_{i+1})+\lambda_f(n_z(x_1-x_0)-n_x(z_1-z_0))\ ] \end{aligned} \tag{C.8}$$

From the 3D DDA circular interpolation, replacing (X_i, Y_i, Z_i) by the initial point (x_o, y_o, z_o) and $(X_{i+1}, Y_{i+1}, Z_{i+1})$ by the interpolated point (x_i, y_i, z_i), one may obtain the following equation:

$$y_i - y_0 = \lambda_c[n_z(x_i-x_0) - n_x(z_i-z_0)] \tag{C.9}$$

By replacing the interpolated point with the final end point (x_1, y_1, z_1) of the segment, and using the linear interpolation scale factor λ_f, one may obtain:

$$y_1 - y_0 = \lambda_f[n_z(x_1-x_0) - n_x(z_1-z_0)] \tag{C.10}$$

Therefore, by substituting Eq.(C.10) into Eq. (C.8), the (i+1)-th interpolated y coordinate is:

$$Y_{i+1} = Y_i + \lambda_c\ [(y_1 - y_0) + n_x(Z_i - z_i) - n_z(X_{i+1} - x_{i+1})] \tag{C.11}$$

Similarly, from Eq.(C.1) and Eq.(C.3), one may obtain the interpolated z coordinate as:

$$\begin{aligned} Z_{i+1} &= Z_i + \lambda_c[\ n_y(X_{i+1}-(x_{i+1}+\lambda_f(x_1-x_0)))-n_x(Y_{i+1}-(y_{i+1}+\lambda_f(y_1-y_0)))\ ] \\ &= Z_i + \lambda_c[\ n_y(X_{i+1}-x_{i+1})-n_x(Y_{i+1}-y_{i+1})+\lambda_f(n_x(y_1-y_0)-n_y(x_1-x_0))\ ] \end{aligned} \tag{C.12}$$

From the 3D DDA circular interpolation, replacing (X_i, Y_i, Z_i) by the initial point (x_o, y_o, z_o) and $(X_{i+1}, Y_{i+1}, Z_{i+1})$ by the interpolated point (x_i, y_i, z_i), one may obtain the following equation:

$$z_i - z_0 = \lambda_c[n_x(y_i - y_0) - n_z(x_i - x_0)] \tag{C.13}$$

By replacing the interpolated point with the final end point (x_1, y_1, z_1) of the segment, and using the linear interpolation scale factor λ_l, one may obtain:

$$z_1 - z_0 = \lambda_l[n_x(y_1 - y_0) - n_y(x_1 - x_0)] \tag{C.14}$$

Thus, by substituting Eq.(C.14) into Eq.(C.12), the (i+1)-th interpolated z coordinate is as:

$$Z_{i+1} = Z_i + \lambda_c\,[(z_1 - z_0) + n_y(X_{i+1} - x_{i+1}) - n_x(Y_{i+1} - y_{i+1})] \tag{C.15}$$

Therefore, by combining the Eq.(C.7), Eq. (C. 11) and Eq.(C.15), a combined 3D DDA linear and circular interpolation principle is obtained as follows:

$$\begin{aligned} X_{i+1} &= X_i + \lambda_c\,[(x_1 - x_0) + n_z(Y_i - y_i) - n_y(Z_i - z_i)] \\ Y_{i+1} &= Y_i + \lambda_c\,[(y_1 - y_0) + n_x(Z_i - z_i) - n_z(X_{i+1} - x_{i+1})] \\ Z_{i+1} &= Z_i + \lambda_c\,[(z_1 - z_0) + n_y(X_{i+1} - x_{i+1}) - n_x(Y_{i+1} - y_{i+1})] \end{aligned} \tag{C.16}$$