Development of a User Interface for Processing Geometric Data from off-the-shelf CAD Packages, and Motion Planning for YAMAHA Zeta-1 Deburring Robot

YE SU

A Thesis

in

The Department

of

Mechanical Engineering

Presented in Partial Fulfilment of the Requirements

for the degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

March 5, 1999

0-612-39106-X

ABSTRACT

Development of a User Interface for Processing Geometric Data from off-the-shelf CAD Packages, and Motion Planning for YAMAHA Zeta-1 Deburring Robot

YE SU

This thesis presents the development of an interface to generate tool path for robot manipulators from CAD database. In engineering practice, CAD (Computer-Aided Design) has been utilized in different ways. Some utilize it to produce drawings and document designs. Some may employ it as a visual tool by generating shaded image and animated displays. Others may perform engineering analysis of some sort on geometric models, or perform process planning and generate NC part programs. This thesis is based on the idea of building a bridge between CAD systems and robot motion planning. The methodology of this thesis constitutes techniques for geometric data analysis, reorganizing and utilizing, and motion planning for an industrial robot–Yamaha Zeta-1 robot.

The thesis focuses on developing the tool path for a Yamaha Zeta-1 deburring robot that is designed specifically to debur complex engineering components, which is also used by the aerospace industries to edge debur and polish surfaces of impellers and bearing races in vane rings used in gas turbine engines. The developed interface will enable the user to design complex components using a standard off-the-shelf CAD package while machining can be carried out using robot manipulators designed for machining. The interface can also

be used to generate tool path for other machine tools by suitably modifying the interface sections that deal with the machine kinematics relating the tool location to joint coordinates of the machine tool. A similar method of translating the CAD data to tool path is commercially available for numerically controlled (NC) machines.

The CAD package chose for current study is the AutoCAD as it is commonly used in small to medium-size manufacturing industries. AutoCAD generates part database files in two formats, namely, the DXF and ACIS formats, that can be read by other CAD packages, such as CADKey, etc. These formats are chosen for analysis in this thesis.

A software interface has been developed to extract the geometric data from the database and re-organize the data so that connectivity between adjacent geometric entities can be established. Parametric representation of 2D and 3D curves and to a procedure to extract the geometric shape at the points of intersection of solids in different planes are carried out. Kinematic transformations and a kinematic model of the Yamaha Zeta-1 robot have been developed to relate the workpiece coordinates to the machining robot tool and world coordinates. Tool path is generated based on the geometric data of the component, machining data (tool velocity, cutting depth, tool geometries) and robot-workpiece transformations.

Experimental verification of the kinematic formulations and the generated tool path are carried out using the Yamaha Zeta-1 robot. The results indicate that the results from the kinematic formulations match the robot controller data and the correctness of the generated tool path.

ACKNOWLEDGEMENTS

Foremost, I would like to pay my sincerest gratitude and appreciation to Dr. R. M. H. Cheng and my supervisor Dr. Ramesh Rajagopalan, for their insightful guidance, encouragement, moral and financial support throughout the course of this research. Especially, my thanks are due to Drs. R. M. H. Cheng and Ramesh Rajagopalan, who initiated this topic as a part of the strategic grant and defined the methodology, and with whom I had a great deal of fruitful discussions. My thanks are also due to Mr. Karun Thanjavur and Mr. Bruce J. Roussel, who shared his invaluable experiences and opinions helping me to start with this work. I also appreciate the time spent by Mr. Gilles Huard for helping me from the course lab to the research work. I wish to thank my colleague, Mr. Jin Li, with whom I had the fruitful discussions and cooperation in the experiments of this work.

I am grateful to all my colleges and staff at Centre for Industrial Control for their help and friendship.

This work was supported by Strategic Grant STRO134360 and individual operating grant from NSERC awarded to Drs. R. M. H. Cheng and Ramesh Rajagopalan.

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

CHAPTER 2 - GEOMETRIC DATA REPRESENTATION IN CAD DATA FILES

CHAPTER 3 - GEOMETRIC MODELLING AND MATHEMATIC REPRESENTATION

CHAPTER 4 - TOOL PATH GENERATION

CHAPTER 5 - KINEMATICS OF YAMAHA ZETA-1 DEBURRING ROBOT

CHAPTER 6 - IMPLEMENTATION OF THE INTERFACE

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

θ	Turning angle of the main body of Yamaha Zeta-1 Deburring Robot
α	Turning angle of the wrist vertical joint of Yamaha Zeta-1 Deburring Robot
β	Turning angle of the wrist inclined joint of Yamaha Zeta-1 Deburring Robot
R	Radial displacement on Yamaha Zeta-1 Deburring Robot
A	Tool turning angle of the Yamaha Zeta-1 Deburring Robot
B	Tool collapse angle of the Yamaha Zeta-1 Deburring Robot
X_r	x-component of the machining point in the robotic Cartesian control system (mm)
Y_r	y-component of the machining point in the robotic Cartesian control system (mm)
Z_r	z-component of the machining point in the robotic Cartesian control system (mm)
X_w	x-component of the machining point in the workpiece coordinate system (mm)
Y_w	y-component of the machining point in the workpiece coordinate system (mm)
Z_w	z-component of the machining point in the workpiece coordinate system (mm)
nx_r	x normal component of the cutter axis in the robotic Cartesian control system
ny_r	y normal component of the cutter axis in the robotic Cartesian control system

nz_r z normal component of the cutter axis in the robotic Cartesian control system

nx_w x normal component of the cutter axis in the workpiece coordinate system

ny_w y normal component of the cutter axis in the workpiece coordinate system

nz_w z normal component of the cutter axis in the workpiece coordinate system

Chapter 1 INTRODUCTION

1.1 Industrial Robot and Robotic Deburring

Industrial robots do not look or behave like human beings, but they could do the work of humans. The industrial robot is a programmable mechanical manipulator, capable of moving along several directions, equipped at its end with a device called the end-effector, and performs factory work ordinarily carried out by human beings.

The term robot is used for a manipulator that has a built-in control system and is capable of stand-alone operation[1]. The first industrial robots were installed to replace people in dangerous operations, such as loading and unloading hot parts from processing furnaces, or in hazardous environments, for example, when workers were subjected to long exposures to toxic materials. Today, however, industrial robots are installed to improve productivity in manufacturing operations. Some machining operations that can be performed by robots are drilling, polishing and deburring. When machining is performed on metal parts, burrs are generated. The removal of these burrs has usually been done by hand operated tools, which is a monotonous and expensive task which will not result in consistent quality. By closely mimicking the manual method the industrial robots can carry out most deburring tasks, yielding consistent surface and edge finish while reducing health hazards to the operator at a reduced operating cost[1].

1.2 Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM)

Industrial robots have revolutionized industry. Use of industrial robots, along with computer-aided design (CAD), and computer-aided manufacturing (CAM) systems, characterizes the latest trends in automation of the manufacturing processes.

Computer-aided design (CAD) refers to the use of a computer to assist in the design of an individual part or a system, such as the impeller of an aircraft engine, or a milling machine. Computer-aided manufacturing (CAM) deals with the use of a computer to assist in the manufacture of a part. CAM can be divided into two main classes: on-line applications, namely, the use of a computer to control manufacturing systems in real-time, and off-line applications, namely, the use of the computer in production planning and non-real-time assistance in the manufacturing of parts. Example of off-line CAM is the tool path planning.

Any product begins with a need which is identified based on customers requirements and markets demands. A product goes through two main phases from the conceptualization of the idea to the finished product, which are classified as the design phase and the manufacturing phase. Traditionally, the design and manufacturing phases are separated and take place sequentially, with process planning as the activity that bridges the gap between the two phases (Figure 1.1). Process planning involves the translation of product and process design requirements into a set of manufacturing instructions which could be interpreted and carried out in the manufacturing facility.

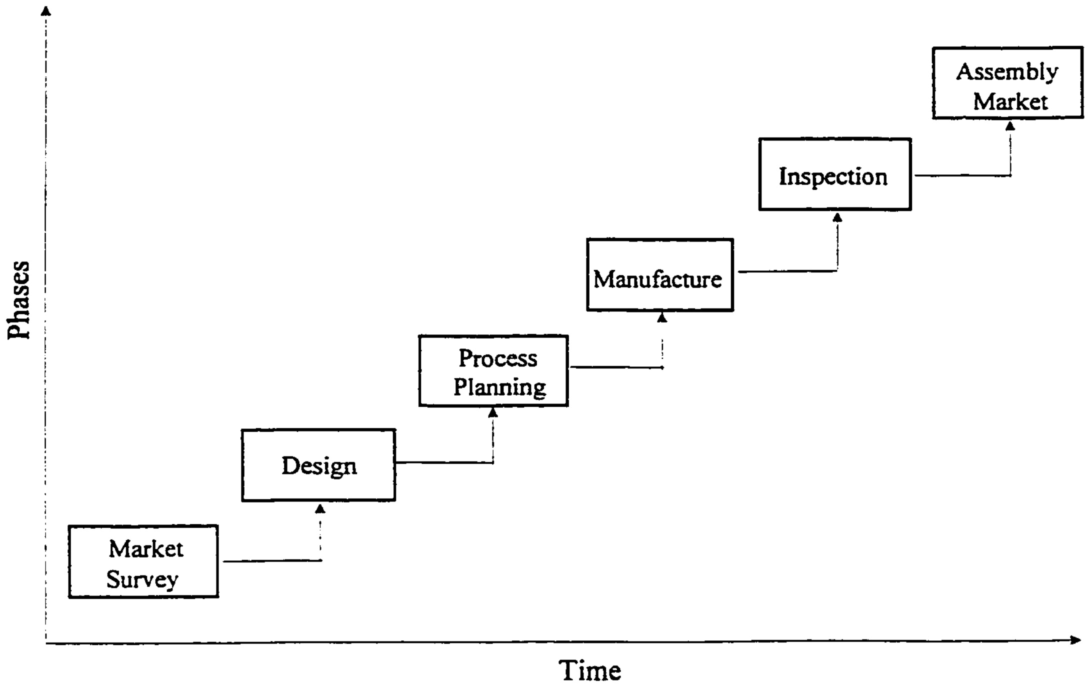


Figure 1.1 Product Development Processes[1]

A critical concern of CAD and CAM is the communication of design and manufacturing data within an engineering organization. CADCAM should be a system which involves the whole life of a product, in which the CAD portion is interfaced inside the computer with the CAM system. The end result of current CADCAM system is usually a part program in the form of a list or a punched tape. In advanced CADCAM systems part programs can be directly fed into the control computers of CNC machines and inspection stations.

The industrial robot systems can be regarded as a computer numerical control (CNC) machine. In order to easily prototype complex engineering workpieces, reduce the cost of inventory of workpieces and production costs, and improve flexibility of product lines, and consistency, a process planning between CAD and robots need to be developed. This is the objective of the research reported in this thesis. The robot system chosen for study is the Yamaha Zeta-1 deburring robot system as it is readily available at the Centre for Industrial Control and proven to be effective in deburring complex aerospace components in aerospace industries. The CAD package chosen for study is AutoCAD as it is commonly used in small- to medium-size manufacturing componies.

The key to the application of CAD/CAM is the extraction of data from the design model for use in analysis and in manufacturing. The main method to date of translating CAD data into manufacturing instructions is in the generation of data for numerically controlled machines. If the process planning function is to be completely automated, techniques must be developed to recognize part features from a CAD data file.

1.3. Deburring WorkCell at the Centre for Industrial Control (CIC)

1.3.1 Introduction

This section presents an overview of the deburring workcell at CIC consisting of a Yamaha Zeta-1 robot, developed to (i) modify the tool path of the robot in real-time; (ii) perform automated digitizing; (iii) carry out surface modelling; and (iv) a reject extraneous feature in new and refurbished components. Details concerning the workcell can be found in Ayyadevara[2] and Roussel[3].

The structure of the Yamaha Zeta-1 robot is composed of a main frame and a wrist at its end shown in Figure 1.2. The main frame is denoted as the arm, and the distal group

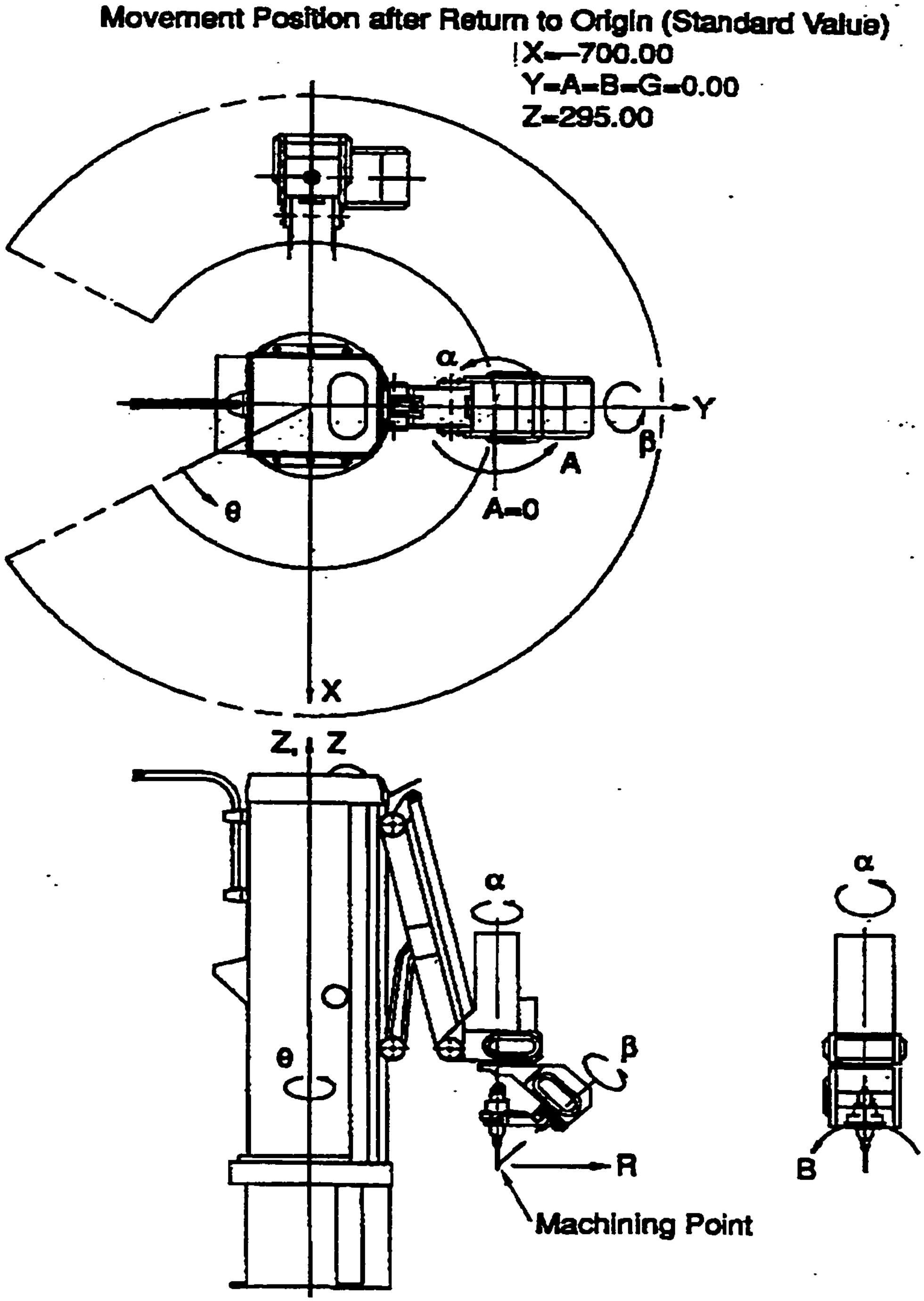


Figure 1.2 The Structure of the Yamaha Zeta-1 deburring Robot[4]

of joints affecting the motion of the end-effector is referred to as the wrist. The end-effector is a machining tool. The tool is mounted at the end of the robot and performs the work, and therefore is also denoted as the robot tool[4].

The Yamaha robot has five axes of motions (θ, R, Z, α, and β) to reach a point with a specific orientation in space. The NC table provides the additional degree of freedom needed to orient the component.

Each axis of motion is separately driven by a dc servomotor. Two drive elements are coupled directly to the mechanical links or joints, and the other drive elements are driven indirectly through gears, belts, and lead screws. Each axis of the robot includes a position feedback device, an incremental encoder, from which both velocity and position of the joints

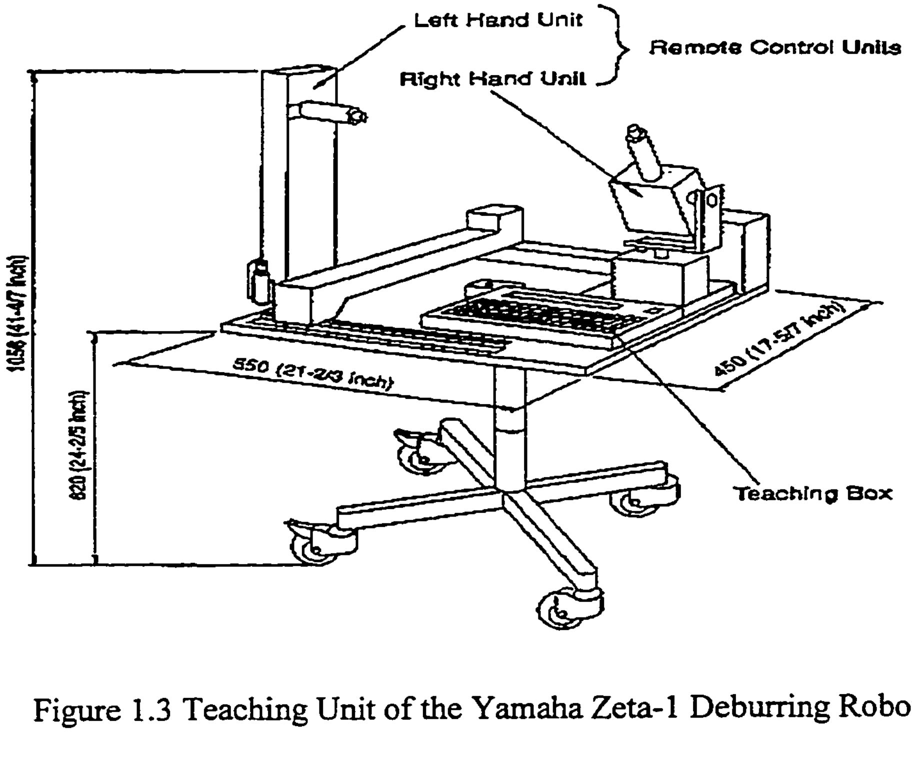


Figure 1.3 Teaching Unit of the Yamaha Zeta-1 Deburring Robot[4]

can be deduced. The lambda-mechanism extending radially from the main vertical column is designed to enhance the structural rigidity of the robot frame. The machining point of the robot is defined as the intersection point of two wrist joint axes[4].

The Yamaha Zeta-1 deburring robot is structurally a cylindrical coordinates type robot, but for control purposes, it moves with a Cartesian coordinate system. The Cartesian coordinate system is shown in Figure 1.2. The position of the machining point is defined by X, Y, and Z values, and the orientation of the end effector is defined by two angles A and B. A is the tool turning angle which is formed by the positive B direction of the tool's tumbled side and the positive direction of X axis. B is a tool collapse angle.

There are two ways to operate the robot: using a teaching unit, or using the workcell controller. The teaching unit consists of two joysticks and a teaching box. The left joystick is used to control the position of the machining point, and the right joystick is used to control the orientation of the tool. The teaching box with keypad can be used to manually control the position of the machining point and the orientation of the tool, or manually edit machining programs containing trajectory coordinates and speed, and to display the current position and orientation. The teaching unit is shown in Figure1.3.

Operation of the workcell consists of three levels: supervisory, process, and servo. Servo level control takes place within the Yamaha controller unit, where power signals are generated to drive the DC servomotor of each joint. Process level control include trajectory tracking command generation and communication with joint optical encoders. Supervisory level control consists of on-line tool path control and interaction with the user. The hardware connection of the workcell is shown in Figure 1.4.

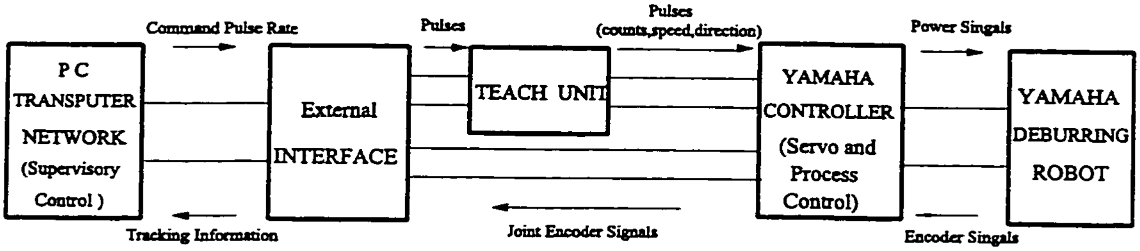


Figure 1.4 Data Link Between the Workcell and the Yamaha Zeta-1 Robot[2]

The architecture of the workcell controller, details of the design and performance data can be found in Ayyadevara[2]. The workcell consists of PC transputer network, interface with Yamaha controller, teach unit, Yamaha controller, and the Yamaha deburring robot.

The input data file is a script file which contains seven columns, X, Y, Z, A, B, V and the reset signal(optional). X, Y and Z describe the coordinate values of the machining point, A and B describe the tool orientation. V is the total velocity of the tool from previous machining location to the current machining location.

When the workcell is powered up, a homing sequence is first performed to initialize the optical shaft encoders of each joint. After the homing sequence, the robot is located at a predefined location called *the ready position.* The Yamaha Zeta-1 controller computes all subsequent displacement of the robot with respect to this initial position.

1.3.2 Automated Inspection and Reconditioning

Automated Inspection and Reconditioning of used and new components using the Yamaha Zeta-1 deburring robot was developed by Roussel[3].

Automated surface reconditioning is required during overhaul of gas turbine-powered aircraft engines, and other types of turbo machinery and fluid power control surfaces. In the course of normal operation, these components are subjected to thermal and inertial stresses, abrasion, impacts and corrosion. Typically, the rotor blades undergo localized defects such as cracks, dents, pitting and scratches. In addition, burrs resulting from primary machining may be present[3].

Surface reconditioning typically consists of light machining cuts for removal of small extraneous features from a surface, and restoration of the original surface finish. Cracks, pits, and scratches on used components will require grinding and subsequent buildup with weld prior to machining and surface polishing[3].

1.4 Previous Work in Robotic Deburring Process Control and Robotic Motion Planning

1.4.1 Robotic Deburring Process Control

In order to improve cost efficiency, there is an increasing demand for the development of an automated deburring system in manufacturing. The problem of automated deburring using a robot has received attention elsewhere, notable ones are in Stepien et al[5], and Kazerooni et al[6]. The work by Stepien et al was to model the deburring process as a normal contact force control problem. Kazerooni et al proposed an impedance control strategy to the automated deburring problem, by adding a special passive complaint end effector to a robot. Liu and Asada[7] have developed an adaptive control scheme for deburring robots whose design is based on a set of teaching data taken from human demonstrations. In their work, an associative mapping between burr characteristics and human responses is constructed using neural networks and this mapping is used as a basis for path planning on subsequent components. Smoothness of surface finish is found to be greatly improved by their method.

Her and Kazerooni [8] have focussed on deburring of two-dimensional workpieces where the workpieces geometry is not known. Contact forces measured by a roller bearing mounted on a force sensor at the robot endpoint are used to calculate the normal vector to the workpiece surface and generate compliancy in the robot. The cutting forces generated by the cutter is used by the deburring algorithm to produce a stable metal removal process. Haefer, Houpt, Baker and Dausch have addressed the development of a real-time robotic position/force control for edge contouring and deburring performed in two dimensions at low path velocities. The major factor limiting performance is reduced bandpass of the force control when the robot is used to apply force through tool and robot arm compliance. Bandpass improves by incorporating force control computation into the robot control.

R. Rajagopalan, et al.[9] developed an open architecture controller for a Yamaha Zeta-1 deburring robot. They identify the various means to modify the trajectory/toolpath in real-time from an external computer. They presented the system architecture and edge tracking performance of the workcell of the Yamaha Zeta-1 deburring robot which is capable of deburring used components[10]. Probing is accomplishing by means of a displacement sensor. The workcell controller has an information processing part consisting of a PC-Parallel Processor network, an interface to communicate with the controller of the deburring robot and a hardware interface for the probe. The surface of the workpiece is probed by the robot, directed along a path prescribed by the PC-Parallel Processor network, based on the original CAD database. The probed points collected are used to reconstruct the surface probed and extrapolate to obtain the edge profile. The tool path constructed from the edge profile is then used to direct the robot during machining.

1.4.2 Robotic Motion Planning

The motion planning problem is a well-known problem in the field of robots, and has received considerable attention in the past. Regardless of the robot types, the motion planning problem can be formulated in two fundamental different versions. In one version, a start configuration and a goal configuration are known in advance, and the objective is to compute a feasible path for robot from start position to goal position. In the second version, no start and goal configurations are specified, and the objective is to compute a data structure which can later be used for queries with arbitrary start and goal configurations. The first version is regard as one-shot problem, and the second version is called learning problem.

Generating collision-free motion of acceptable quality is one of the main concerns in robotics. A typical robot presents an arm manipulator with a fixed base operating in three-dimensional space. The research on the robotic motion planning focus mostly on the how to move purposely and safely in an often complex environment filled with known or unknown obstacles. John Canny[13] introduces a new algebraic method for robot motion planning and real geometry which solves the final path or generalized mover's problem in single exponential sequential time. This work is motivated by the abstract path planning problem in robotics. Given a way of specifying the configuration of a moving object, constraints on its motion define a subset of its configuration space within which the object is clear of obstacles. Ge and McCarthy [14] represent an algebraic formulation of the configuration obstacles for wrist-partitioned spatial robots. Positions of an object are represented as points in the image space. The set of positions that the end-effector of the robot can reach is characterized by an algebraic manifold in this space, called the Reachable manifold. This set of positions of one convex polyhedron that moves in contact with another is characterized by a piece-wise smooth manifold in the same space, called the Contact manifold. The intersection of these manifolds defines the constraint manifold of the end-effector as part of the closed chain formed by the robot in contact with an obstacle. This manifold is then parametrized using configuration variables of the robot to determine the boundaries of the configuration obstacle. Canny and Lin [15] describe a robot path planning algorithm that constructs a global skeleton of free-space by incremental local methods. By following the maxima of a well-designed potential field taking slice projections through critical points and at random values, an obstacle-avoiding path to guide a robot toward the desired goal is incrementally built.

Tsai-Yen li and Jean-Claude Latombe[16] describe an on-line planner for two cooperating arms, whose task is to grab parts of various types on a conveyor belt and transfer them to their respective goals while avoiding collisions with obstacles. Parts arrives at any time, in random order. The planner uses overall information provided by a vision system to time, in random order. The planner uses overall information provided by a vision system to break the overall planning into a stream of rather simple subproblems and orchestrate fast planning primitives solving these subproblems.

In order to increase the quality, consistency, and system productivity , the robotic motion planning also focus on the integration of two key field: CAD and Robotics. A. V. Trivedi [17] describes CAD/Robotics in the computer-integrated manufacture (CIM) environment. As CIM becomes a reality for most mid to large scale manufacturers, the importance of CAD increases dramatically. The need will no longer be the design process, but the ultimate culmination of the design into production, marketing, and meeting customer needs. CAD/Robotics is the stepping stone for technologies. CAD/Robotics is the breakthrough technology of today as numerical control was, for the past several decades. Bidanda and Narayanan [18] develop a CAD-based off-line programming system for robotic spray-glazing applications. The system algorithms access a geometric model of part in a CAD system. Using the parameters of the spray-glazing process it generates the robotic spray gun path. Path planning results are verified using graphic simulation software.

Vishnu and Cutkosky[19] have described an approach to automatic programming of robots for deburring precision workpieces. They have focussed on two major aspects of the problem: (1) automatically generating collision-free robot programs and (2) modelling the process to provide a basis for feedback control. The process of automatically generating robotic deburring programs began with interpreting a CAD model of the machined workpiece. A plan for deburring the workpiece is then composed, and checked at each step to ensure that the kinematic constraints are satisfied and that no collisions will occur. Due to the uncertainty in the real world, the robot cannot successfully accomplish its task without the help of sensors. Therefore, the data from sensors are used to negate the effects of uncertainty by modifying the nominal plan.

1.5 Scope and Outline of the Thesis

This thesis deals with the development of an interface between CAD output data files and the machining tools used to machine workpiece edges. The geometric entities of a workpiece are extracted from the DXF or ACIS, data files generated by a CAD package, such as an AutoCAD, and these extracted entities are reorganized and defined. The edges that are to be deburred and the machining requirements are specified by the user. Based on the requirements from users, toolpath is automatically generated. In order to utilize this interface for different kinds of robots or machining tools, the geometric definition and the toolpath for deburring the edges of components are in the workpiece coordinate system. A postprocessor is designed to transfer the toolpath from workpiece coordinate system to the robot control system, and the kinematic model is built to deduce the transformation equations used by the postprocessor. Toolpaths are planned for the approach phase during which the deburring tool is carried by the robot to approach the workpiece from the home position (ready position), the deburring phase during which the tool follows the contour of the workpiece, and the departing phase for the robot to return to its home position. Figure 1.5 represents the schematic layout of the interface. All processes are fulfilled by developing an interface which involves the following:

- Read and analyse the CAD data files (DXF and ACIS).

- Develop a filter to extract the geometric data from the data files, and reorganise the geometric data and develop three-dimensional wireframe models for display.
- Generate parametric representations of basic 2D and 3D curves.
- Generate toolpath in workpiece coordinate system.
- Deduce the kinematic model of the Yamaha Zeta-1 deburring robot
- Transform the toolpath from workpiece coordinate system into the robot control coordinate system. The final output of the interface is a script data file that can be directly used by the Yamaha Zeta-1 workcell controller.
- Carry out experimental verification of the kinematic formulations and the toolpath generated by the interface.

The detail about the CAD data file formats are discussed in Chapter 2. Chapter 3 deals with geometric modelling and mathematical representations based on the input data from DXF or ACIS files. Chapter 4 describes the toolpath generation methodology developed in this thesis. Detail on the kinematic model developed for the Yamaha Zeta-1 deburring robot and the steps to deduced transformation equations are shown in Chapter 5. A flowchart describing the interface software, the procedure used to manage geometric data in the software, and the method developed to interface and generate the toolpath are presented in Chapter 6. Chapter 7 presents the experimental verification of the toolpath generated by the interface. Conclusion and recommendations for future work are give in Chapter 8.

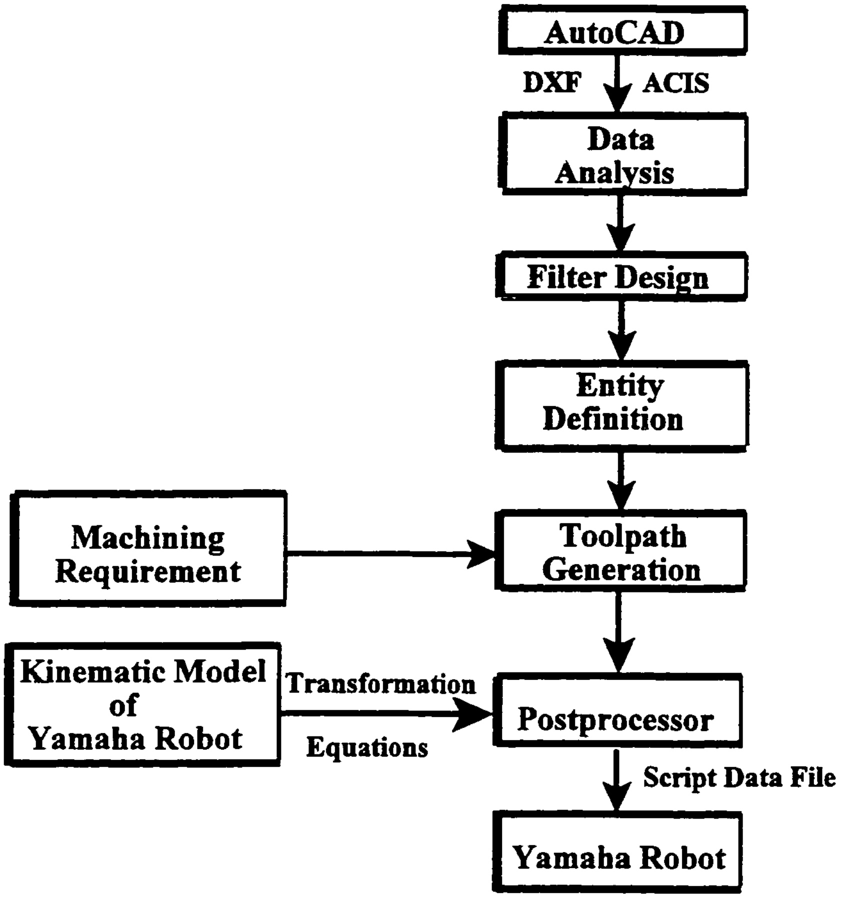


Figure 1.5 Schematic Layout of the Interface

CHAPTER 2 GEOMETRIC DATA REPRESENTATION IN CAD DATA FILES

2.1 Introduction

CAD databases are now replacing paper blueprints in defining product geometry and non-geometry for all phases of product design and manufacturing. Transferring data between similar CAD systems and dissimilar CAD systems must embrace the complete product description stored in its database. It becomes increasingly important to find effective procedures for exchanging these databases between CAD systems.

AutoCAD is a CAD software for PCs developed by Autodesk Inc., which has a large market share and has been very influential. Auto CAD provides a mechamism to interface external software to work with AutoCAD. This is accomplished by storing the data in compact binary forms and readable forms using ASCII. The format of these later forms are known as DXF (short for Data Exchange File), ans ACIS (specially store three-dimensional data) and is now widely used by many CAD systems as a means of storing data in a portable form.

In order to develop an interface that shares geometric data with CAD systems via the DXF or ACIS mechanism, the structures of DXF and ACIS files must be examined. The DXF and ACIS formats are structured in such a manner to easily extract only the required information. Basically, all entities in AutoCAD are divided into two categories: Three-dimensional entities and two-dimensional entities. Two-dimensional entities in DXF files and three-dimensional entities in ACIS files are considered in this thesis.

2.2 ASCII Drawing Exchange Files (DXF)

2.2.1 DXF File Format

A DXF file can be easily created by using the command DXFout in AutoCAD with a specified file name. A drawing Interchange File is an ASCII text file with a file type of ".dxf" and specially-formatted text. The complete DXF file for even the simplest drawing (one entity) is long and contains vast amount of information. The overall organization of a DXF file is as follows [20]:

1). HEADER section: General information about the drawing is found in this section of the DXF file. Each parameter has a variable name and an associated value.

2). TABLES section: This section contains definitions of named items.

- Line type table
- Layer table
- Text table
- View table
- User Coordinate System table
- Viewport configuration table
- Drawing manager table

3). BLOCK section: This section contains Block Definition entities describing the entities comprising each Block in the drawing.

4). ENTITIES section: This section contains the drawing entities, including any Block References.

5). END OF FILE

The HEADER, TABLE and BLOCK sections are useful when using the DXF file for file translations. However, when reading DXF with a program other than a CAD program, these parts are not useful, and only the ENTITY section need to be analysed.

A DXF file is composed of a multiplicity of groups, each of which occupies two lines in the DXF file. The first line of a group is a group code. The second line of a group is the group value, which depends on the type of the group as specified by the group code. The specific assignment of group codes depends upon the item being described in the file. Table 2.1 shows the group code ranges [20].

Table 2.1 Group Code Ranges [20]

Group Code Range	Following Value	Group Code Range	Following Value
0 - 9	String	999	Comment (String)
10 - 59	Floating-point	1000 - 1009	String
60 - 79	Integer	1010 - 1059	Floating-point
210 - 239	Floating-point	1060 - 1079	Integer

A program can be designed to read these values following a group code without knowing the particular use of this group in an item in the file. The appearance of values in the DXF file is not affected by the setting of the units command, i.e., coordinates are always represented as decimal numbers, and angles are always represented in decimal degrees.

Group codes are used both to indicate the type of the value of the group, and to indicate the general use of the group. The specific function of the group code depends on the actual variable, table item, or entity description. The general use of groups which appear in the section considered are tabulated in Table 2.2.

2.2.2 Geometric Data Representation in DXF File

In order to analyze and extract the geometric data of a workpiece, the representation of each entity in DXF file must be examined first. A workpiece is regarded as a wireframe model consisting of points, lines, circles, circular arcs, polylines. This section focus on the representations of these basic entities.

Since AutoCAD release 10, all coordinates in each record are in three-dimensional format, that is, there are X, Y and Z components for each coordinate point even if the Z value is 0. The coordinates of each record are all in the World Coordinate System, and it is the original Cartesian three-coordinate (X, Y and Z) system with coordinate {0,0,0} in the lower left-hand corner.

A two-dimensional workpiece consists of lines, circles, circular arcs, polylines:

1) Lines, Circles and Arcs

A line is represented by its two end points, a circle is represented by its center point and its radius, and an arc is represented by its center point, its radius, its starting angle, and its ending angle.

Table 2.2 Entity Group Codes [20]

Group Code	Value Type
0	Identifies the start of an entity
5	Entity handle expressed as a hexadecimal string
8	Layer name
10	Primary X coordinate (start point of a line, or center point of a circle, etc)
11-18	Other X coordinates
20	Primary Y coordinate (start point of a line, or center point of a circle, etc)
21-28	Other Y coordinates
30	Primary Z coordinate (start point of a line, center point of a circle, etc)
31-37	Other Z coordinates
39	This entity's thickness if nonzero
50-58	Angles
999	Comments

2) Polylines

A polyline looks like a series of lines and arcs on the screen, but their database description is completely different from that of lines and arcs. A polyline actually consists of a series of coordinate points and a bulge factor related to this vertex. Each coordinate point is called a vertex. AutoCAD takes the coordinate information for each vertex and draws straight line segments if the bulge factor is 0, or arcs between each vertex point if the polyline bulge factor is 1. All ellipses, polygons, and fitting curves, such as spline, cubic curves, are represented by polylines [21].

The representation of an ellipse is more complex than that of a line or a circle, or an arc. In a DXF file, an ellipse is represented by a polyline. An ellipse usually consists of 16 polyarcs, but each polyarc isn't constructed from a center point reference like ARCS and CIRCLES. A polyarc is described as having a start point (the current vertex), an end point (the next vertex), and a bulge factor (Figure 2.1). It is important to derive the center point and radius from a polyarc [21].

2.2.3 Example

Figure 2.2 shows a example of a drawing containing two lines and an arc. Because a DXF file is very long even for a simplest drawing, this example only presents some segments for the DXF file.

As mentioned before, only the ENTITIES SECTION of a DXF file has to be read and analysed. In Figure 2.2, the geometric data about a line following a key word "LINE"are two points: start point and end point. The x, y and z coordinates about the start point are the values follow the group codes 10, 20, and 30. The x, y and z coordinates of the end points are the values follow the group codes 11, 21, and 31. The geometric data about an arc follow a key word "ARC". For an arc, the x, y and z coordinates of the center point are the values follow the group codes 10, 20, and 30, the radius follows the group code 40, and the values follows group codes 50 and 51 are the start angle and the end angle of the arc.

```
 0
SECTION
 2
HEADER
....
 0
ENDSEC
 0
SECTION
 2
TABLES
 0
TABLE
......
ENDSEC
 0
SECTION
 2
BLOCKS
 0
ENDSEC
...

 0
SECTION
 2
ENTITIES
 0
LINE
 8
0
 10
10.08192
 20
29.39948
 30
0.0
 11
14.27223
 21
7.989556
 31
0.0
```

```
 0
ARC
 8
0
 10
25.26914
 20
27.21525
 30
0.0
 40
6.56144
 50
270.25955
 51
170.55586
 0
LINE
 8
0
 10
22.114124
 20
21.46214
 30
0.0
 11
33.82467
 21
8.78068
 31
0.0
 0
ENDSEC
 0
EOF
```

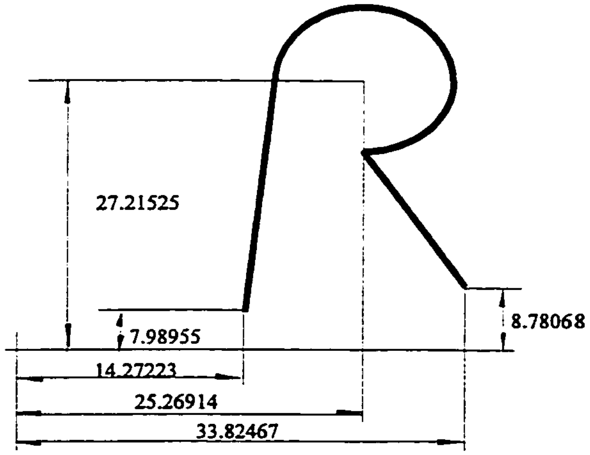


Figure 2.1 DXF File and the Related Drawing

2.3 ACIS Save Files (SAT)

ACIS is an object-oriented geometric modelling toolkit designed for use as a geometry engine within 3D modelling applications. ACIS stores geometry information to save files (.sat files). These files have an open format so that external applications can have access to the ACIS geometric model. These applications are then able to read the pure geometric data from or write information to a saved model.

Written in C++, ACIS provides an open architecture framework for wireframe, surface, and solid modelling from a common, unified data structure. Linear and quadratic geometry is represented analytically, and non-uniform rational B-splines represents free-form geometry. ACIS utilizes the C++ classes and structures to save the relative data [22].

2.3.1 Overview of ACIS

ACIS has several blocks, and these blocks are important for showing how certain data classes are derived. The data classes and their respective data elements appear in the save files. The data considered for future use in ACIS are contained in the kernel block. The ACIS kernel can be broken down into three main mechanisms: topology, geometry, and attributes. Geometry represents the physical objects, such as blocks, spheres, planes, and contains the mathematics that describe these objects. Topology is how ACIS implements boundary representation, and it holds the relationships between each type of boundary. It is hierarchically defined, where a "body" is the largest, encompassing type of boundary, to "vertex", which is the smallest, most specific type of boundary. Finally, the attributes attach other information to both the geometry and the topology to create flexible, application-specific models.

The topology block as seen in Figure 2.2 displays the data classes that are created for nearly all 3D models. It represents the basic classes. The topology explains how we traverse the model, so its data elements generally contains pointers to the other elements in the model.

The geometry block, as seen in Figure 2.3, displays the data classes that are created to represent specific aspects of the topology. The geometry is used to fix the topology in space. The topology points to elements in the geometry, which store the actual data for the model. Many of the geometry data classes can be broken down much further.

The attributes block display the data classes that are created to add special information of the topology and geometry, such as colour, system configuration parameters. This block has not been considered in this thesis as it is beyond the scope of current investigation.

2.3.2 Structure of ACIS File

An ACIS file can be easily created by using the command ACISOUT in AutoCAD with a specified file name. It consists of a single header record, an end marker for the file, and at least one data record between the header and end marker.

The ACIS part save file header includes four integer values: an encoded version number, the total number of saved data records, a count of the number of entities in the original entity list saved to the part file, and a reserved number for future use.

The header is followed by a sequence of entity records. Each entity record consists of a sequence number (optional), an entity type identifier, the entity data, and a terminator.

The top level entities (body entities) are always the first records in the sat file. Thereafter, the data records are in no particular order. Pointers between entities are saved as integer index values, with NULL pointers represented by the value -1. ACIS pointer indexes are preceded by $ in the sat file. The indices are generated as the entities are being saved [22].

2.3.3 ACIS Kernel and its Fundamental ENTITY

As mentioned in Section 2.3.1, the kernel is divided into topology, geometry, and attributes. All classes in the kernel are derived in some manner from ENTITY. The kernel classes explicitly appear in the save file format. All heading keywords that are enclosed within quotation marks ("") are used in the save file.

Tracing from body to vertex, or from vertex back to body, all individual information about one entity and all information about the relationship between two entities, all geometric data about a three-dimensional model are included in one ACIS file.

2.3.3.1 Kernel Topology

ENTITY represents common data and functionally that is mandatory in all classes that are permanent objects in the model. The classes derived from ENTITY that represent the topology of the model are: BODY, LUMP, SHELL, SUBSHELL, FACE, LOOP, COEDGE, EDGE, VERTEX, and WIRE [22].

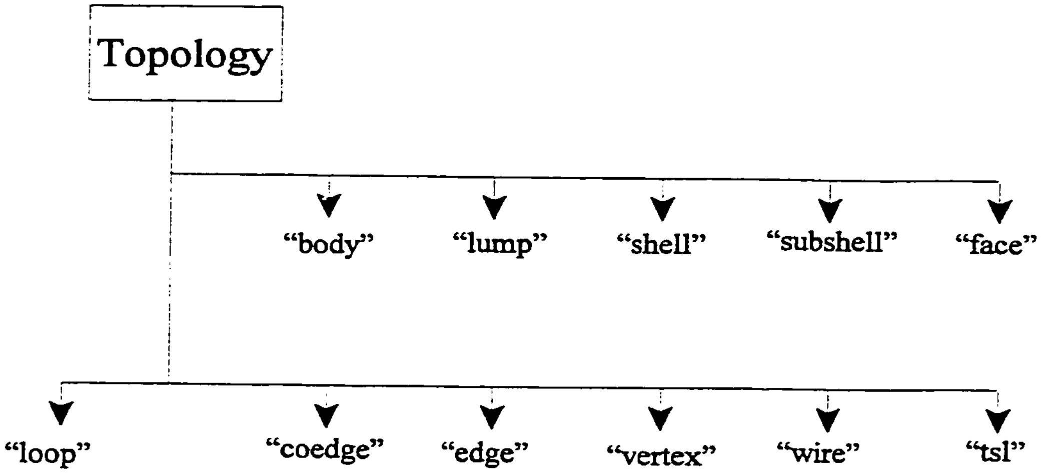


Figure 2.2 Kernel Topology Overview [22]

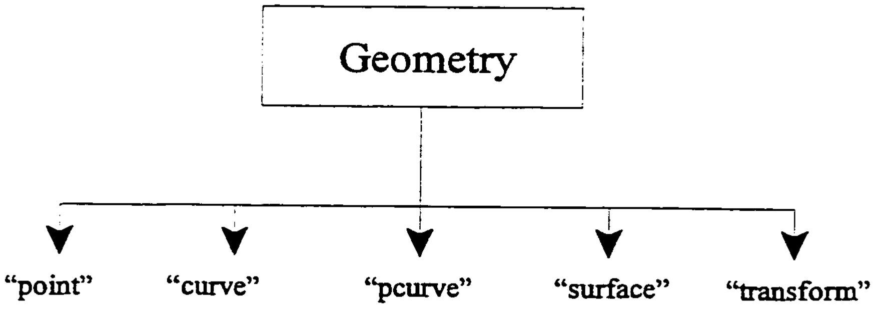


Figure 2.3 Kernel Geometry Overview [22]

"body" is the first record that follows the file header, and "body" represents a wire, sheet, or solid body. A pure wire body contains only wires, but no lumps, shells, or faces. A lump body represents partially-surfaced wireframes, sheets and solids. In principle, a body can contain both wire and lump components, but wire and lump components should not be connected to one another. A solid is represented by the boundaries of the regions of space that it encloses. Each solid is made up of one lump or a group of disjointed lumps. Lumps and wires are described in a local coordinate system. The local coordinate system is related to the universal coordinate system by a transformation stored with the body [22].

"lump" represents a bounded, connected portion of space, and "shell" bounds a LUMP peripherally. "Subshell" represents a subdivision of a SHELL or SUBSHELL. A "face" is a bounded portion of a single geometric surface in space, and "loop" bounds a FACE. A "coedge" relates EDGEs with adjacent EDGEs and owning ENTITY. When the EDGE is part of a well-formed solid body SHELL, it is adjacent to exactly two FACEs. A "vertex" represents a corner of a FACE and the end point of an EDGE[22].

2.3.3.2 Kernel Geometry

The fundamental kernel geometries considered are[22]: "cone", "ellipse", "intcurve", "pcurve", "plane", "point", "sphere", "straight" and "torus".

1) CONE

It defines the elliptical single cone by a base ellipse and the sine and cosine of the major half-angle of the cone. The ellipse has the data structure as center point, a unit_vector normal to the plane of the ellipse, a vector defining the major axis of the ellipse, and the radius ratio of the minor axis length to the major axis length. As special cases, the base may be circular, or the cone may be a cylinder.

The polarity (sign) of trigonometric functions define the slant of the surface of the cone and the sense of the surface.

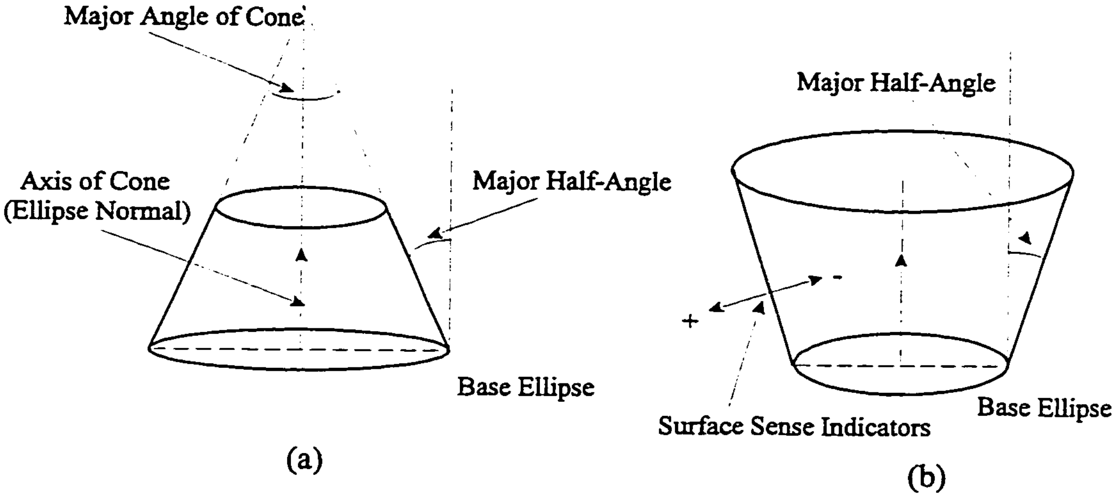


Figure 2.4 Cone Class Definition[22]

2) ELLIPSE

It represents circles and ellipses on any plane. An ellipse is defined by a center point, a unit_vector normal to the plane of the ellipse, a vector defining the major axis of the ellipse, and the radius ratio of the minor axis length to the major axis length. In a circle, the radius ration is exactly 1. The direction of the ellipse is defined by the right hand rule using the normal vector. Direction is important when defining an edge supported by this geometry.

3) INTCURVE

It represents parametric object-space curves that map an interval of the real line into a three-dimensional real vector space (object-space). This mapping is continuous, and one-to-one except possibly at the ends of the interval whose images may coincide. The direction is the positive sense of the curve. If the two ends of the curve are different, the curve is open. If they are the same, it is closed. If the curve joints itself smoothly, the curve is periodic, and its period is the length of the interval that it is primarily defined.

4) PCURVE

PCURVE is used to define parameter-space curves that map an interval of real line into a 2D real vector space (parameter space). This mapping is continuous, and one-to-one except possibly at the ends of the interval whose images may coincide. The direction is the positive sense of the curve. If the two ends of the curve are different, the curve is open. If they are the same, it is closed. If the curve joints itself smoothly, the curve is periodic, and its period is the length of the interval that it is primarily defined.

5) PLANE

A PLANE is defined by a point with a unit-vector normal to the plane. Usually, the point chosen to define the plane is near the center of interest. The normal represents the outside of the surface. This is important when a plane is chosen as machining surface.

6) POINT

A POINT represents the position of a VERTEX as an object in the model. It is defined by three coordinate values.

7) SPHERE

A SPHERE is defined by a center point and radius. Normally, the radius is positive, i.e. material inside, but it can be negative to indicate a hollow sphere, i.e., material outside.

8) STRAIGHT

A STRAIGHT define an infinite line by a point on the line and a unit vector specifying the direction. A straight line is an open curve that is not periodic.

9) TORUS

A TORUS is an object in the model. It is a circular thickening of a circular spline, defined by a center, normal, major radius, and minor radius. The minor radius is the radius of the thickening circle. The major radius is the radius of the torus from its center point. The normal determines the orientation in space of the torus.

2.3.4 Example

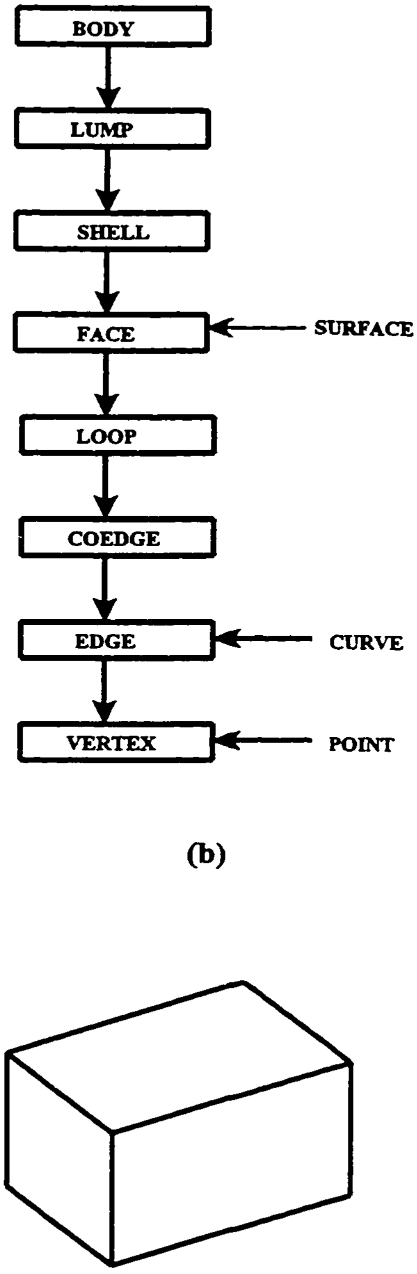
Figure 2.5 shows an ACIS data file of a cube. An ACIS file is very long even for this cube, so the example only shows a segment of the data file in Figure 2.5(a). Figure2.5(b) shows the relationship between the Topology and the Geometry. Based on this relationship, we can easily understand the ACIS format, catch the pure geometric data and extract these data.

2.4 Conclusion

Geometric representations in CAD data files have been read and analysed. The related data are extracted by the interface. The data contained in the ENTITY SECTION in DXF file are directly used by the interface to generate the off-line tool path. The data saved in ACIS file is in the form of solid models, these data are transferred into wireframe models.

```
106 191 1 0
body $1 $2 $-1 $-1 #
f_body-lwd-attrib $-1 $3 $-1 $0 #
lump $4 $-1 $5 $0 #
ref_vt-lwd-attrib $-1 $-1 $1 $0 $6 $7 #
ref_vt-lwd-attrib $-1 $-1 $-1 $2 $6 $7 #
shell $8 $-1 $-1 $9 $2 #
refinement $-1 0 0 0 0 0.086666665971279144 0 0 0 2 #
vertex_template $-1 3 0 1 8 #
ref_vt-lwd-attrib $-1 $-1 $-1 $5 $6 $7 #
face $10 $11 $12 $5 $-1 $13 reversed single #
color-adesk-attrib $-1 $14 $-1 $9 256 #
face $15 $16 $17 $5 $-1 $18 reversed single #
loop $-1 $19 $20 $9 #
cone-surface $-1 0 0 25 0 0 1 15 0 0 1 I I 0 1 0 I I I I #
fface-lwd-attrib $-1 $21 $10 $9 #
color-adesk-attrib $-1 $22 $-1 $11 256 #
face $23 $24 $25 $5 $-1 $26 forward single #
loop $-1 $-1 $27 $11 #
plane-surface $-1 0 0 0 0 0 -1 -1 0 0 0 I I I I #
loop $-1 $-1 $28 $9 #
coedge $29 $20 $20 $30 $31 1 $12 $-1 #
ref_vt-lwd-attrib $-1 $-1 $14 $9 $6 $-1 #
fface-lwd-attrib $-1 $32 $15 $11 #
color-adesk-attrib $-1 $33 $-1 $16 256 #
face $34 $35 $36 $5 $-1 $37 reversed single #
loop $-1 $38 $30 $16 #
plane-surface $-1 0 0 20 0 0 1 1 0 0 0 I I I I #
coedge $39 $27 $27 $28 $40 1 $17 $-1 #
coedge $41 $28 $28 $27 $40 0 $19 $-1 #
copar-lwd-attrib $-1 $-1 $-1 $20 #
coedge $42 $30 $30 $20 $31 0 $25 $-1 #
edge $43 $44 $44 $30 $45 0 #
ref_vt-lwd-attrib $-1 $-1 $22 $11 $6 $-1 #
fface-lwd-attrib $-1 $46 $23 $16 #
color-adesk-attrib $-1 $47 $-1 $24 256 #
face $48 $49 $50 $5 $-1 $51 reversed single #
loop $-1 $-1 $52 $24 #
plane-surface $-1 0 0 -20 0 0 1 1 0 0 0 I I I I #
loop $-1 $-1 $53 $16 #
copar-lwd-attrib $-1 $-1 $-1 $27 #
edge $54 $55 $55 $27 $56 0 #
copar-lwd-attrib $-1 $-1 $-1 $28 #
copar-lwd-attrib $-1 $-1 $-1 $30 #
color-adesk-attrib $-1 $57 $-1 $31 256 #
vertex $-1 $31 $58 #
ellipse-curve $-1 0 0 20 0 0 -1 15 0 0 1 I I #
```

(a)



(b)

(c)

Figure 2.5 ACIS Format of a Cube

CHAPTER 3 GEOMETRIC MODELLING

and

MATHEMATICAL REPRESENTATION

3.1 Introduction

A workpiece to be manufactured is defined first in terms of its geometry and second in terms of other parameters, such as material type and functionality. Workpiece consists of manufacturable entities, such as cubes, cones, torus, which can be defined as sets of workpiece surfaces and edges. This thesis focus on extracting the geometric information and tool path generation for edges of workpieces.

This part deals with geometric modelling of entities, their related mathematical representations, and the relationship between two entities. A geometric model of an object is constructed by providing the geometric data of the object. For automatic generation of these models, a user interface is needed. The interface converts such data into a mathematical representation of the object for further processing.

There are three types of geometric models, wireframe, surfaces, and solids, used to represent objects. This thesis deals with the wireframe model. A wireframe model of an object is the simplest geometric model. It is sometimes referred to as a stick figure or an edge representation of the object. The word "wireframe" is given to this form of representation as it is equivalent to bending wires to follow the object edges to generate the model. Wireframe modelling is the most commonly used technique.

Wireframe entities are divided into analytical and synthetical entities. Analytical entities are points, lines, arcs and circles, conics. Synthetical entities includes various types of spline and Bezier curve. Only the analytic entities are included in this thesis.

Definition of analytical entities are discussed in Section 3.2, which includes point, line, circle and circular arc, ellipse, cone (cylinder), and torus. Section 3.3 deals with the relationship between two planar entities. Intersection curves of two regular surfaces are represented in Section 3.4.

3.2 Definition of Analytical Entities

3.2.1 POINT

In a workpiece, a point is the location of the intersection of two lines, or the intersection of three planes.

A point M in space has three coordinates with respect to the frame {W}, and is represented by M{x, y, z}. For each such set of three real numbers there is a uniquely determined point in space.

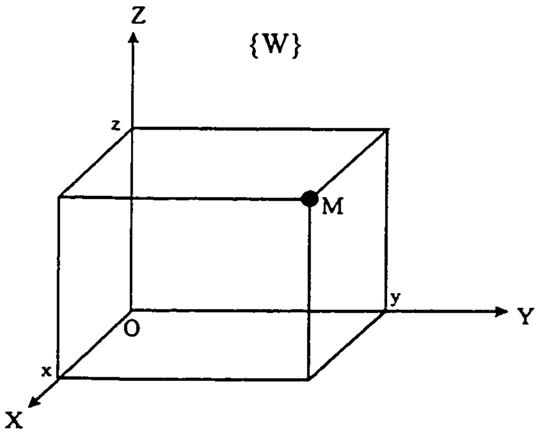


Figure 3.1 A Point in Space

3.2.2 LINE

A line is defined as the intersection of two planes. A vector **L** represents a line which passes through the origin of the frame. If point M (p, q, r) is any point on the line then the position vector **L** = **OP** can always be written in the form

$$\mathbf{OP} = p\,\mathbf{i} + q\,\mathbf{j} + r\,\mathbf{k} \qquad (3.1)$$

The length of vector **L** is given by

$$|\,\mathbf{L}\,| = \sqrt{p^2 + q^2 + r^2} \qquad (3.2)$$

The direction cosines of the vector **L** are given by cos α, cos β and cos γ.

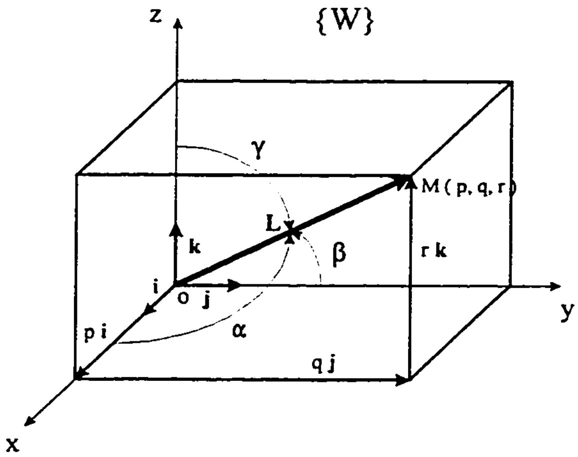


Figure 3.2 A Line in Space

1) Data Extracted from DXF File

From a DXF file, the coordinates of two points, $M_1(x_1, y_1, z_1)$ and $M_2(x_2, y_2, z_2)$, can be extracted. The length of the vector $\mathbf{M_1M_2}$ is:

$$|\mathbf{M_1M_2}| = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}$$

$$|\mathbf{M_1M_2}| = |\mathbf{OM_2}| - |\mathbf{OM_1}|$$

$$= (x_2 - x_1)\,\mathbf{i} + (y_2 - y_1)\,\mathbf{j} + (z_2 - z_1)\,\mathbf{k}$$

$$p = x_2 - x_1$$

$$q = y_2 - y_1$$

$$r = z_2 - z_1$$

Assume M is any point on this line. To obtain a vector equation for the line, we observe that we can reach point M on the line by proceeding first from origin O to M_1 and then travelling along the line a suitable multiple t of the vector $\mathbf{M_1M_2}$ to M as illustrated in Figure 3.3. Thus,

$$|\mathbf{L}| = |\mathbf{OM_1}| + t\,|\mathbf{M_1M_2}| \tag{3.3}$$

Substituting $\mathbf{L} = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k}$ in equation (3.3), we get

$$x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k} = x_1\,\mathbf{i} + y_1\,\mathbf{j} + z_1\,\mathbf{k} + t\,(x_2 - x_1)\,\mathbf{i} + t\,(y_2 - y_1)\,\mathbf{j} + t\,(z_2 - z_1)\,\mathbf{k} \tag{3.4}$$

Equating coefficients of the vector components on both sides of equation 3.4 leads to

$$\left.\begin{array}{l} x = x_1 + t(x_2 - x_1) = x_1 + pt \\ y = y_1 + t(y_2 - y_1) = y_1 + qt \\ z = z_1 + t(z_2 - z_1) = z_1 + rt \end{array}\right\} \tag{3.5}$$

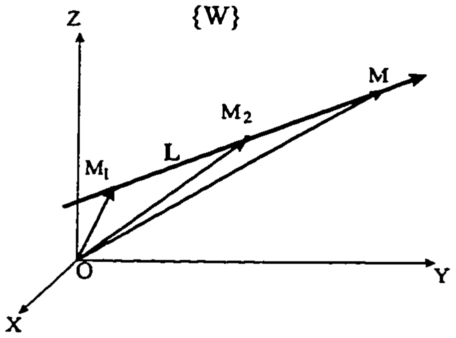


Figure 3.3 Vector Representation of a Line

2) Data Extracted from ACIS Files

From ACIS files, a point $M_1(x_1, y_1, z_1)$ on the line and a unit vector (p, q, r) specifying the direction of the line are extracted. Line **L** can be represented by

$$\left.\begin{aligned} x &= x_1 + p\,t \\ y &= y_1 + q\,t \\ z &= z_1 + r\,t \end{aligned}\right\} \qquad (3.6)$$

3.2.3 Ellipses, Circles and Circular Arcs

Mathematically the ellipse is a curve generated by a point moving in space such that at any position the sum of its distance from two fixed points (foci) is constant and equal to the major diameter. Each focus is located on the major axis of the ellipse at a distance from its center equal to $\sqrt{R_a^2 - R_b^2}$ (R_a and R_b are the major and minor radii) [23].

Circles and circular arcs are the special cases of the ellipses, which major and minor radii are equal.

Besides other information (accuracy and orientation of the ellipse), half the length of the major axis and minor axis, and the coordinates of the center point, are required to define the geometric shape of an ellipse. The parameter equations of the ellipse is:

$$\left.\begin{array}{l} x = x_c + R_a \cos(u) \sin(\phi) - R_b \sin(u) \sin(\phi) \\ y = y_a + R_a \cos(u) \sin(\phi) + R_b \sin(u) \cos(\phi) \\ z = z_c \end{array}\right\} \qquad (3.7)$$

where $0 \leq u \leq 2\pi$. The parameter u is the angle as in the case of a circle. However, for a point M shown in Figure 3.4 and 3.5, u is not the angle between the line MM_c and the major axis of the ellipse. Instead, to find coordinates of M on the ellipse that corresponds to an angle u, two concentric circles C_1 and C_2 are constructed with centers at M_c and radii of R_a and R_b respectively as shown in Figures 3.4 and 3.5. A radial line is constructed at the angle u to intersect both circles at points M_1 and M_2 respectively. If a line parallel to the minor axis is drawn from M_1 and a line parallel to the major axis is drawn from M_2, the intersection of these two lines defines the point M. ϕ isthe angle between the major axis of an ellipse and the X-axis as shown in Figure 3.5.

An ellipse is represented in DXF files as a polyline as mentioned before. The center point, the major and minor axes cannot be obtained directly. Several circular arc segments are used instead of a single elliptical curve.

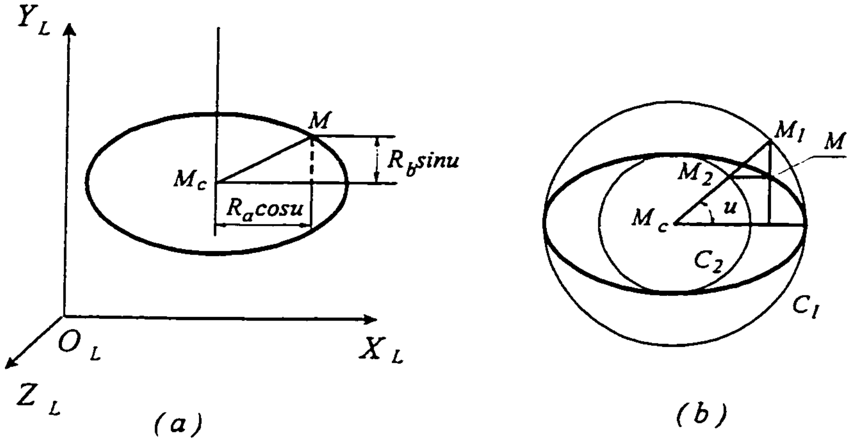


Figure 3.4 Representation of an Ellipse (I)

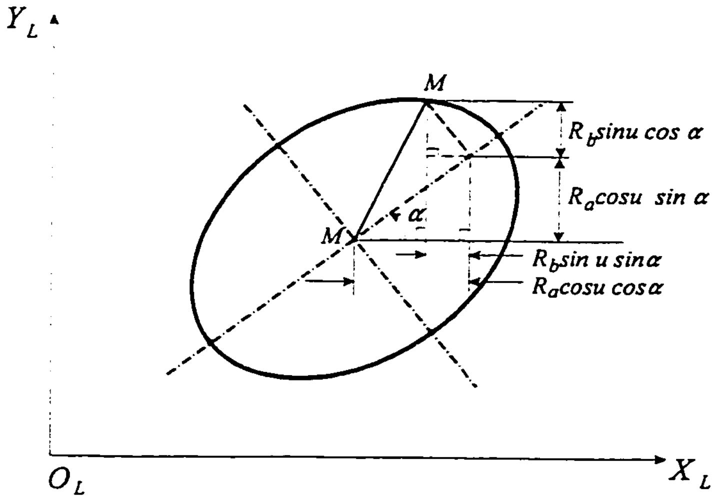


Figure 3.5 Representation of an Ellipse (II)

In ACIS file, the ellipse is represented by a center point Mc (x_c, y_c, z_c), a vector normal (**N**) to the plane of the ellipse, a vector (**U**) defining the major axis of the ellipse, and the radius ratio of the minor axis length to the major axis length. The major and the minor axes can be directly calculated based on these information. The major axis equals to the length of the vector defining the major axis of the ellipse, and the minor axis equals to the product of radius ratio and the major axis:

$$\left.\begin{aligned} \mathbf{R_a} &= |\mathbf{N}| \\ \mathbf{R_b} &= \mathbf{r} \times \mathbf{R_a} \end{aligned}\right\} \qquad (3.8)$$

where r is the radius ratio.

3.3 Intersection of Two Regular Surfaces

When two solids intersect each other, the geometric form of the resulting intersection curve depends upon the geometry of each of the surfaces and relative position of these surfaces. Thus if both surfaces consist of a series of plane faces, they will intersect each other in a series of straight lines; or if one or both of the surfaces be curved, the intersection will consist of one or more curves; these curves may be plane, but generally are tortuous curves; that is, such as cannot be contained in a plane.

One solid may completely penetrate the other, entering at a closed curve where their surfaces intersect, and emerging at a second closed curve . Or the inter-penetration may be only partial, in which case the line of intersection generally consists of only one closed curve.

The curve of intersection of a plane and a surface is a plane curve and is called a plane section of the surface. The determination of a plane section may be carried out if the equation of the surface is known relative to a coordinate system in which the given plane is one of the coordinate planes. If the plane is not on one of the coordinate planes, it is always possible to perform a transformation of coordinates so as to obtain a new coordinate system in which the given plane becomes one of the coordinate planes. The surface is expressed relative to the transformed coordinate system using a transformation matrix. The transformation matrix is discussed in Chapter 5.

The intersection problem involving surfaces is complex and nonlinear in nature. In most of the cases commonly considered in analytical geometry, the intersection of two surfaces is a space curve. Assume $\mathbf{P}(u_1, t_1) = 0$ and $\mathbf{P}(u_2, t_2) = 0$ are the equations of the two surfaces, $\mathbf{P}(u_1, t_1) - \mathbf{P}(u_2, t_2) = 0$ is the parametric equations of the curve of intersection since the intersection of two surfaces is the locus of all points on both surfaces. The shape of the intersection curve depends on the geometry and relative position of two surfaces. For regular surfaces such as revolution surfaces the parametric equations are easily deduced. The curve of intersection of these surfaces is deduced based on the principle that the intersection curve lies on both surfaces [23].

The rotation of a planar curve by an angle (u) about an axis of rotation creates a circle (if u is 360) for each point on the curve whose center lies on the axis of rotation and whose radius r(t) is variable. The planar curve and the circles are called the profile and parallels respectively while the various positions of the profile around the axis are called meridians [23].

The planar curve and the axis of rotation form the plane of zero angle, that is u = 0. To derive the parametric equation of a surface of revolution, a local coordinate system with the Y axis parallel to the axis of rotation is as shown in the Figure 3.6, with the subscript L. Consider a point **G** on the profile when u = 0, this point rotates an angle u about Y_L when the profile rotates the same angle. The parametric equation of the surface of rotation can be written as

$$\mathbf{P}_L(u, t) = (x_c + r(t) \cos(u))\, \mathbf{i} + y_L(t)\, \mathbf{j} + (z_c + r(t) \sin(u))\, \mathbf{k} \tag{3.9}$$

where x_c, y_c and z_c is any point on the axis of rotation.

that is
$$\left.\begin{aligned} x_L &= x_c + r(t) \cos(u) \\ y_L &= y_L(t) \\ z_L &= z_c + r(t) \sin(u) \end{aligned}\right\} \tag{3.10}$$

where $0 \le u \le 2\pi$ and $t_1 \le t \le t_2$. For a cylindrical surface, r(t) is constant r, (3.18) becomes

$$\left.\begin{aligned} x_L &= x_c + r \cos(u) \\ y_L &= y_L(t) \\ z_L &= z_c + r \sin(u) \end{aligned}\right\} \tag{3.11}$$

If Z_L is coincident with the axis of the cylinder,

$$\left.\begin{aligned} x_L &= x_c + r \cos(u) \\ y_L &= y_c + r \sin(u) \\ z_L &= z_L(t) \end{aligned}\right\} \tag{3.12}$$

If x_L is coincident with the axis of the cylinder,

$$\left.\begin{aligned} x_L &= x_L(t) \\ y_L &= y_c + r \cos(u) \\ z_L &= z_c + r \sin(u) \end{aligned}\right\} \tag{3.13}$$

There are four possible cases: (1) the axes of two cylinders are perpendicular to and intersecting each other, (2) perpendicular to but not intersecting each other, (3) not perpendicular to but intersecting each other, and they are neither perpendicular to nor intersecting each other.

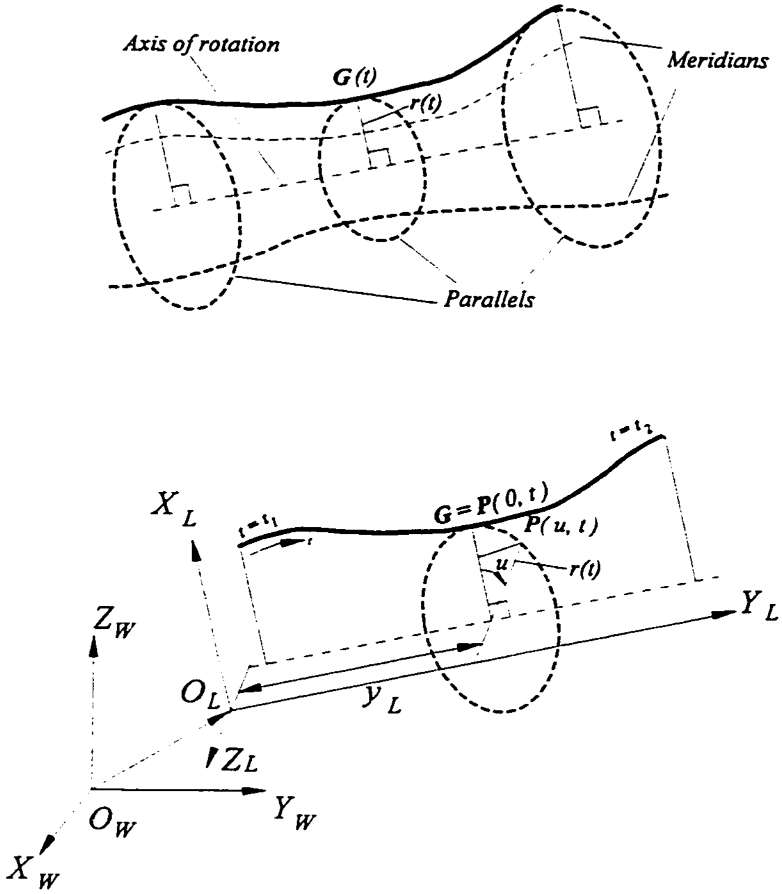


Figure 3.6 Representation of a Revolution Surface[23]

3.4.1 Intersecting Curve of Two Cylinders

A coordinate system {L} is assumed such that Y_L is coincident with the axis of the cylinder $\mathbf{C}_2$, a coordinate system {L'} is fixed on the cylinder $\mathbf{C}_1$ as shown in Figure 3.7. The coordinate system {L'} translates along X_L by a distance (d), rotates about X_L by ϕ_2 degrees, and/or rotates about Y_L by ϕ_1 degrees.

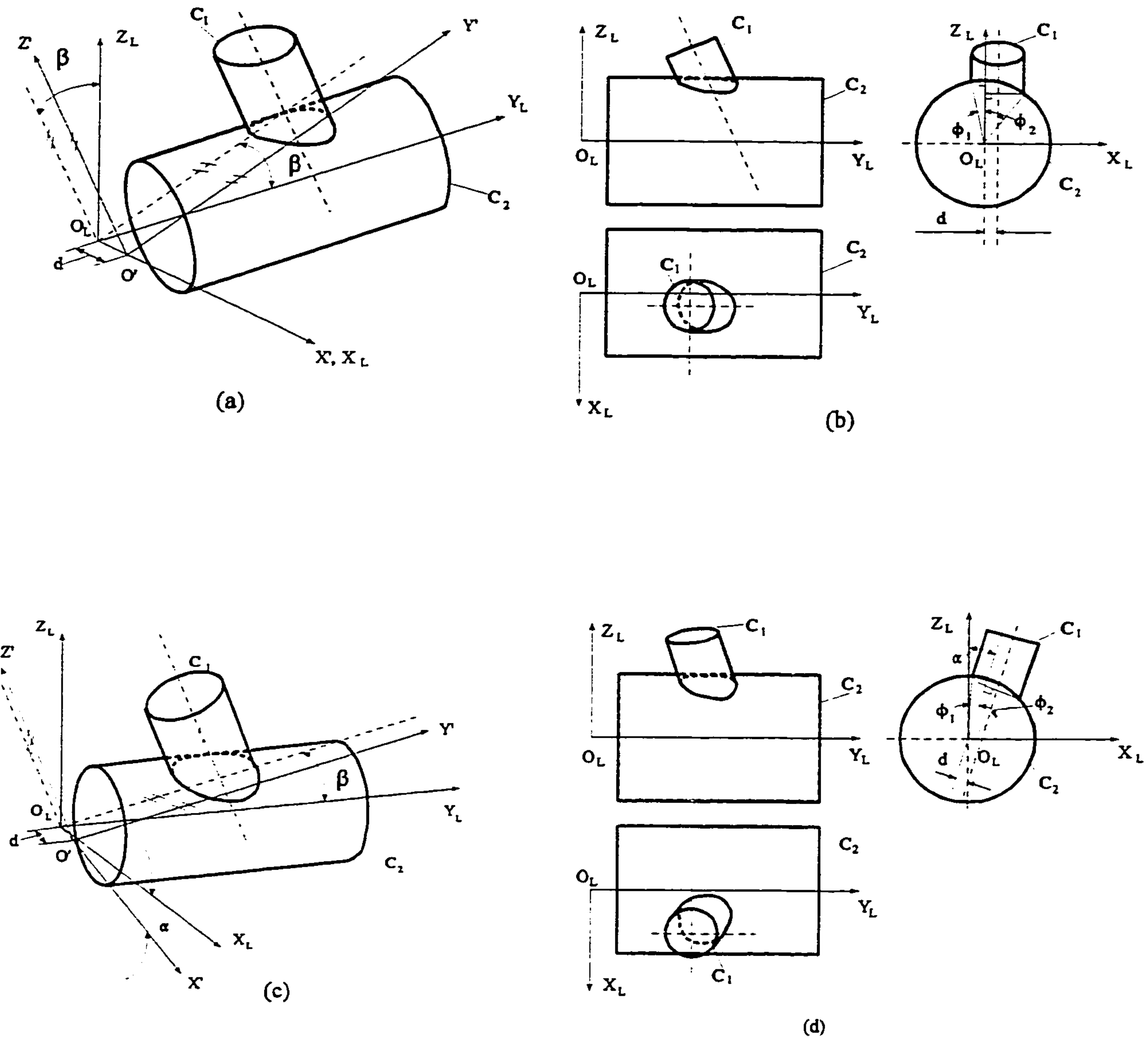


Figure 3.7 Representation of Two Intersecting Cylinders

The parametric equations of two cylinders in the coordinate {L} are:

$$\mathbf{C_1}: \quad \left.\begin{aligned} x_L &= x_{c1} + R_1 \cos(u_1)\cos(\phi_1) + t_1 \sin(\phi_1) \\ y_L &= y_{c1} + R_1 \cos(u_1)\sin(\phi_1)\sin(\phi_2) + R_1 \sin(u_1)\cos(\phi_2) \\ &\quad - t_1 \sin(\phi_2)\cos(\phi_1) \\ z_L &= z_{c1} - R_1 \cos(u_1)\cos(\phi_1)\sin(\phi_2) + R_1 \sin(u_1)\sin(\phi_2) \\ &\quad + t_1 \cos(\phi_2)\cos(\phi_1) \end{aligned}\right\} \qquad (3.14)$$

$$\mathbf{C_2}: \quad \left.\begin{aligned} x_L &= x_{c2} + R_2 \cos(u_2) \\ y_L &= t_2 \\ z_L &= z_{c2} + R_2 \sin(u_2) \end{aligned}\right\} \qquad (3.15)$$

The intersection curve can be represented by following equations in the frame {L}:

$$\left.\begin{aligned} x_L &= x_{c2} + R_2 \cos(u_2) \\ y_L &= y_{c1} + R_1 \sin(\phi_1)\sin(\phi_2)\cos(u_1) + R_1 \cos(\phi_2)\sin(u_1) \\ &\quad - \cos(\phi_1)\sin(\phi_2)\frac{x_{c2} + R_2 \cos(u_2) - x_{c1} - R_1 \cos(\phi_1)\cos(u_1)}{\sin(\phi_1)} \\ z_L &= z_{c2} + R_2 \sin(u_2) \end{aligned}\right\} \qquad (3.16)$$

where u_1 is from 0 to 2π, and the value of u_2 is determined by u_1. Figure 3.8 shows the relationship between u_1 and u_2.

$$u_2 = \sin^{-1}\left[\frac{R_1 \cos u_1 + d}{R_2}\right] + \gamma' \qquad (3.17)$$

where $\gamma' = \tan^{-1}\frac{ny}{nz}$ (3.18)

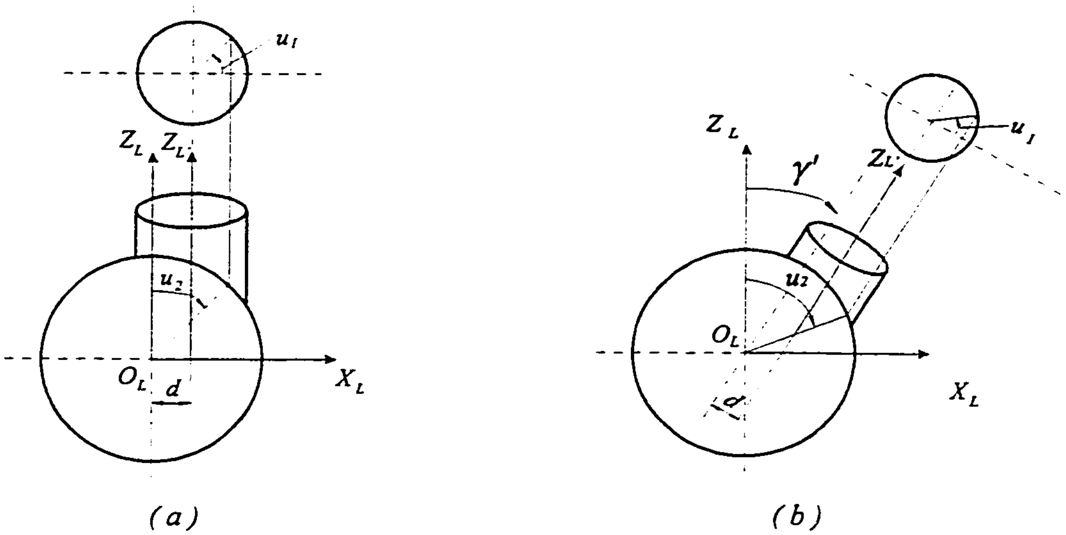


Figure 3.8 The Relationship Between Parameters u_1 and u_2 of Two Cylinders

3.4.2 Intersection Curve of a Torus and a Cylinder

This section considers only the case in which the axes of the torus and the axis of the cylinder are perpendicular and intersecting each other is discussed. Assume a frame {L} such that the Z_L is coincident with the axis of torus, and a frame {L'} such that the $Z_{L'}$ is coincident with the axis of the cylinder as shown in Figure 3.9.

The parametric equations of a torus in the frame {L} is:

$$\left.\begin{aligned} x_L &= (R_1 + R_2 \sin(u_2))\cos(u_1) \\ y_L &= (R_1 + R_2 \sin(u_2))\sin(u_1) \\ z_L &= R_2 \cos(u_2) \end{aligned}\right\} \qquad (3.19)$$

The parametric equations of the cylinder is:

$$\left.\begin{aligned} x_L &= x_{c3} + t\cos(u_1) - R_3\sin(u_1)\cos(u_3) \\ y_L &= y_{c3} + t\sin(u_1) + R_3\cos(u_1)\cos(u_3) \\ z_L &= R_3\sin(u_3) \end{aligned}\right\} \qquad (3.20)$$

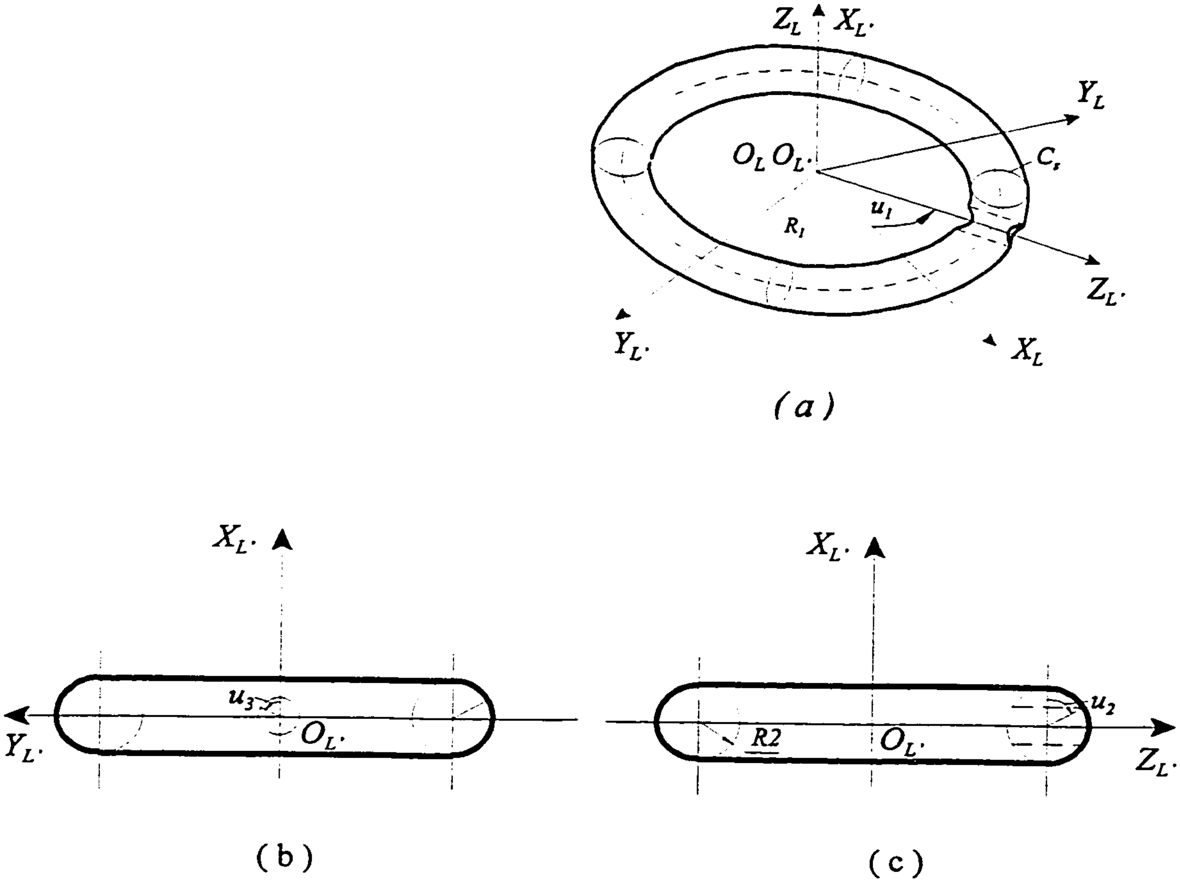


Figure 3.9 Representation of a Cylinder Intersecting with a Torus

The intersection curve can be represented by:

$$\left.\begin{aligned} x_L &= (R_1 + R_2 \sin(u_2))\cos(u_1) \\ y_L &= (R_1 + R_2 \sin(u_2))\sin(u_1) \\ z_L &= R_3 \sin(u_3) \end{aligned}\right\} \tag{3.21}$$

The relationship between u_1, u_2 and u_3 is shown in Figure 3.10.

$$u_1 = \sin^{-1} \frac{R_3 \cos u_3}{R_1} + \phi \tag{3.22}$$

$$u_2 = \sin^{-1} \frac{R_3 \sin u_3}{R_2} \tag{3.23}$$

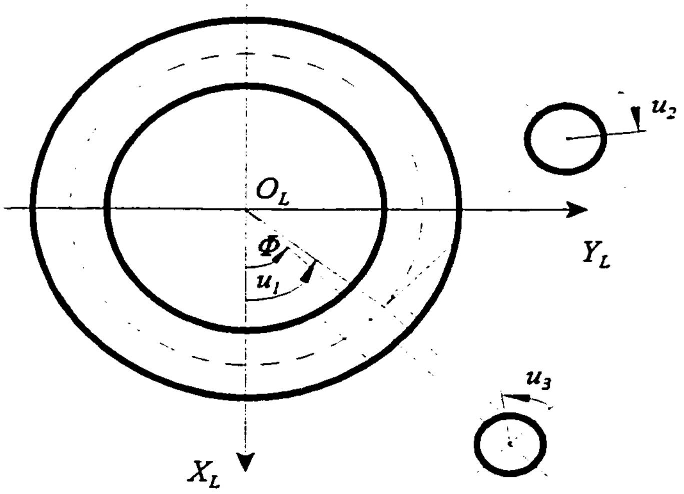


Figure 3.10 The Relationship between u_1, u_2 and u_3

3.5 Conclusion

The mathematical definition of fundamental entities are represented above. As mentioned before, any workpiece consists of these fundamental entities: points, lines, arcs, and circles, ellipses, intersecting curves. The database of a workpiece in CAD system are parameters related to all of these entities. Once the related mathematic representations are built, we can easily use the database from CAD systems, and automatically generate the required tool path for any robot designed to machine components. The derived equations will be used in the designed interface which is described in Section 4.3.2.3..

CHAPTER 4 TOOLPATH GENERATION

4.1 Introduction

This Chapter deals with off-line tool path generation for the Yamaha Zeta-1 deburring robot. The Yamaha robot is a contouring-path robot, it allows the user to define the path to be taken between successive points. A linear interpolation is used by the robot controller to move between successive points along a straight-line in Cartesian space. When a workpiece is machined, it is fixed on the machining table, while the tool (end effector of the robot) is moved along X_r, Y_r, and Z_r of the robot to reach the machining point, and rotating about A and B axes to orientate itself to coincident with the normal of the machining surfaces or edges.

The formulation of tool paths in the world Cartesian space is considered, that is the coordinate values of X_w, Y_w, Z_w, and the normal nx_w, ny_w, nz_w are calculated. The transformation between world coordinate system $\{X_w, Y_w, Z_w, nx_w, ny_w, nz_w\}$ and robot coordinate system $\{X_r, Y_r, Z_r, nx_r, ny_r, nz_r\}$, and transformation between the normal components nx_r, ny_r, nz_r and the values of A and B are described in Chapter 5.

All entities of a workpiece are defined by their parametric equations. The entities that are to be machined are defined based on the specified machining edges. Once the machining edges are defined, all entities on these edges are searched, chosen, and arranged to form a machining toolpath. Depending on the relationship between different entities and the tool used, the machining points are determined, and depending on the parametric equation of different entities, the normals are calculated. In this thesis, tool path generation to machine edges are alone considered.

4.2 Approach and Depart Procedures

While machining using the Yamaha Zeta-1 deburring robot, the tool moves from the ready position, which is fixed as { -699.990, 0.000, 295.000}, to the first set point. The set point represents that location at which the longitudinal axis of the tool is to be aligned prior to beginning the machining operation, and is located above the first machining point (beginning location) by a certain distance as shown in the Figure 4.1. The detail on calculating the first machining point is presented in next section.

For a planar edge, a well-defined normal exists. However, for a three-dimensional curve the normal used to orientate the tool is called bi-normal of the curve. The detail about the calculation of the bi-normal of a three-dimensional curve is presented in 4.3.2.3. The set point of the three-dimensional curve is always located above the highest point (with maximum z value).

Once the tool is set in motion, continuous path motion sequences are executed until the entire sequence of operations on the edges is completed. At this time, the tool must be retracted from the workpiece and returned to its ready position. Such sequence ending operations are also needed for tool changing and clamp repositioning.

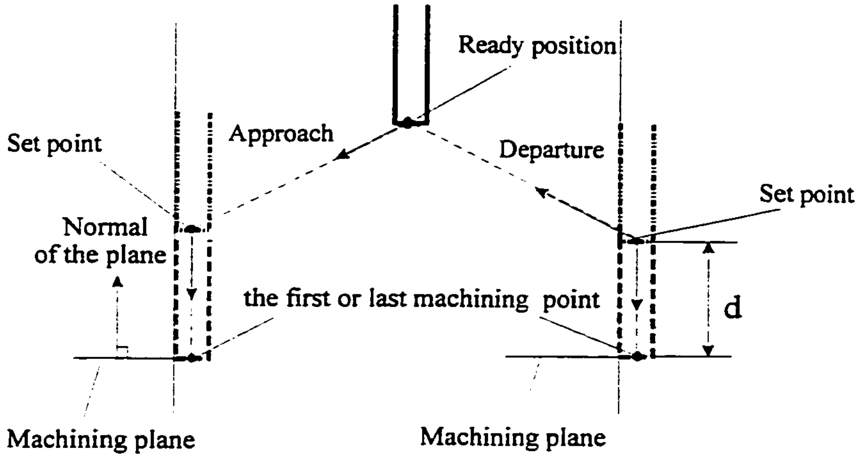


Figure 4.1 Approach and Departure Procedures[24]

The tool is retracted to a location that does not interfere with the component and above the last machining point. The robot return to the ready position from this location.

4.3 Machining Toolpath Generation

To generate the machining tool path, the main issue is how to guide the tool within machining tolerance of the edge along the edge to be machined. Section 4.3.1 discusses the tolerance specification and accuracy of the generated toolpath. The detail about toolpath computation is presented in Section 4.3.2.

4.3.1 Tolerance Specification and Tool Path Accuracy

The robot moves between successive points along a straight-line in Cartesian space, resulting in toolpath that contains successive straight-line segments. Allowing for the tool radius when following these straight-line segments, the profile of the machining edges should lie within specified maximum allowable deviations from the shape of the edges as programmed. The tolerance should be specified for the edges to be machined. Factors that control the selection of tolerances include the machine tool position error, tool tolerances, tool and workpiece deflection, and temperature. The tolerance specification suitable for two-dimensional machining is shown in Figure 4.2. Where INTOL, OUTTOL are the tolerance values for the edge. A zero-valued tolerance is permissible, but both INTOL and OUTTOL for a given edge may not be simultaneously zero. INTOL and OUTTOL may be used individually or in combination[24].

The value for INTOL specifies the maximum allowable deviation permitted between the straight-line segments and the programmed curve (desired curve) on the workpiece side of the curve. This is illustrated in Figure 4.2(a) with the OUTTOL value zero and where, relative to the tool position, the straight-line segments are tangent to convex portions, or chordal to concave portions of the curve, respectively. The straight-line segments always lie within the programmed curve (on the workpiece side), and the tool diameter results in fillets where adjacent chordal segments meet[24].

The value for OUTTOL specifies the maximum allowable deviation permitted between the straight-line segments and the programmed curve on the tool side of the curve. This is illustrated in Figure 4.2(b) with the INTOL value zero and where the straight-line

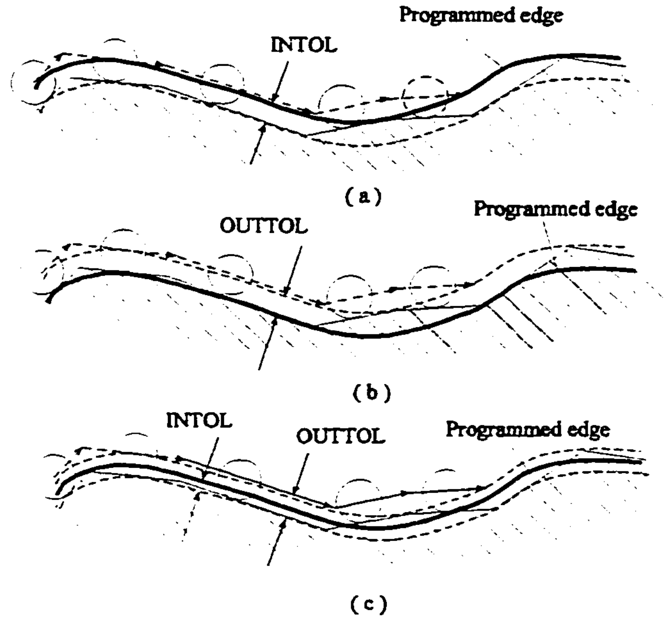


Figure 4.2 Tolerance Specification[24]

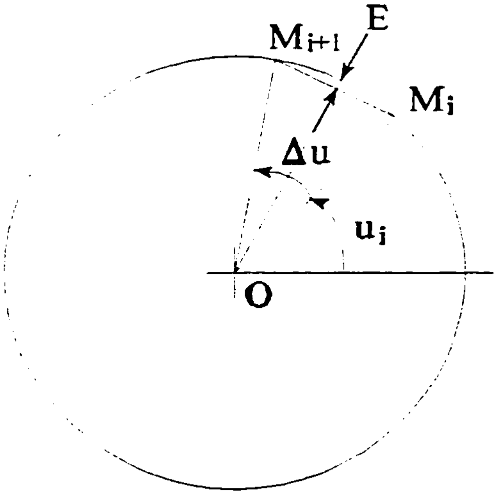


Figure 4.3 Machining Error and Tolerances

segments are also tangential or chordal to the programmed curve. The straight-line segments always lie external to the programmed curve[24].

When both INTOL and OUTTOL are non-zero, the straight-line segments will lie alternately on the workpiece and tool sides of the programmed curve, as shown in Figure 4.2(c). The values of INTOL and OUTTOL are determined by users. The desired toolpath will always be within tolerance of the programmed curve. In the case of a circular arc, the main factor affecting the accuracy of the machining is the selection of the step values of u of the relative parametric equation. The step value is calculated as:

$$E = R - R\cos(0.5\,\Delta u) \leq TOL \tag{4.1}$$

$$\Delta u \leq 2\cos^{-1}(R - TOL) \tag{4.2}$$

where TOL = INTOL if the curve has the positive bulge factor, and TOL = OUTTOL if the curve has the negative bulge factor. Because all points are calculated according to the relative parametric equation, every point lies on the curve.

4.3.2 Tool Path Computation and Decision

This section deals with the computation of the toolpath for machining, which includes the points on the machining edges and the normals at each point on the machining edges. Because the machining edges always consist of many curved segments, the connections between two adjacent curve segments need to be analysed, and the connecting point of two toolpath segments with respect to two curves need to be determined.

The interface developed for toolpath generation employs edge geometry techniques.

The interface has the capability to parametrically describe, and therefore to generate toolpath for all analytical curves. Each entity has a corresponding geometric structure that takes the required input parameters from a data file, and each structure relates to a parametric equation. It depends on the data structures and relative parametric equations to automatically generate the required tool positions and orientations. The tool will be directed counterclockwise around the edges starting from the beginning point.

Section 4.3.2.1 represents the method used to search the beginning point of the toolpath. Machining tool path for two-dimensional edges is discussed in Section 4.3.2.2, and the toolpath of three-dimensional edges is discussed in Section 4.3.2.3.

4.3.2.1 Beginning Point Search Scheme

The startup motion is from the ready position to set point with a change in orientation of the tool in order to align the tool to the required orientation, and from the set point to the beginning point with no change in the orientation of the tool. To calculate the toolpath of the startup procedure, the following steps have been developed:

- Calculate the nearest distances from the ready position to each entity on the machining edges, and the nearest point on each entity.

  The distances from ready position { x_s, y_s, z_s }to any straight curve segment is:

$$d = \left| \frac{\begin{vmatrix} i & j & k \\ p & q & r \\ x_s-x_1 & y_s-y_1 & z_s-z_1 \end{vmatrix}}{\sqrt{p^2+q^2+r^2}} \right| \tag{4.3}$$

where p, q, and r are the direction numbers of the straight curve, and x_1, y_1, and z_1 are coordinates of a point on the straight curve.

The nearest point is the intersection point M_n of the straight curve and a line passing through the ready position and perpendicular to the straight curve as shown in Figure 4.4(a) - (f). If the point M_n lies on the machining edge, the distance from the ready position to the point M_n is the nearest distance of this entity, and the point M_n is the nearest point. If the point M_n does not lie on the machining edge, then the end point M_n' which is near the ready position than another end point is the nearest point, and the distance from the ready position to the nearer end point M_n' is the nearest distance of the straight curve segment.

The distance from the ready position to any circular arc segment is:

$$d = \sqrt{(x_c-x_s)^2+(y_c-y_s)^2+(z_c-z_s)^2} - R \tag{4.4 }$$

where x_c, y_c, and z_c are coordinates of the center point of the circular arc, and R is the radius of the circular arc.

The nearest point is the intersection of the circular arc and a line passing through both ready position and the center point of the circular arc shown in Figure 4.4 (g) - (j). If the point M_n lies on range of the machining edge, the distance from the ready position to this point is the nearest distance of this entity, and the point M_n is the nearest point. If the point M_n does not lie on the range of the machining edge, then the end point M_n' which is near the ready position than another end point is the nearest point, and the distance from the ready position to the nearer end point M_n' is the nearest distance of the circular arc.

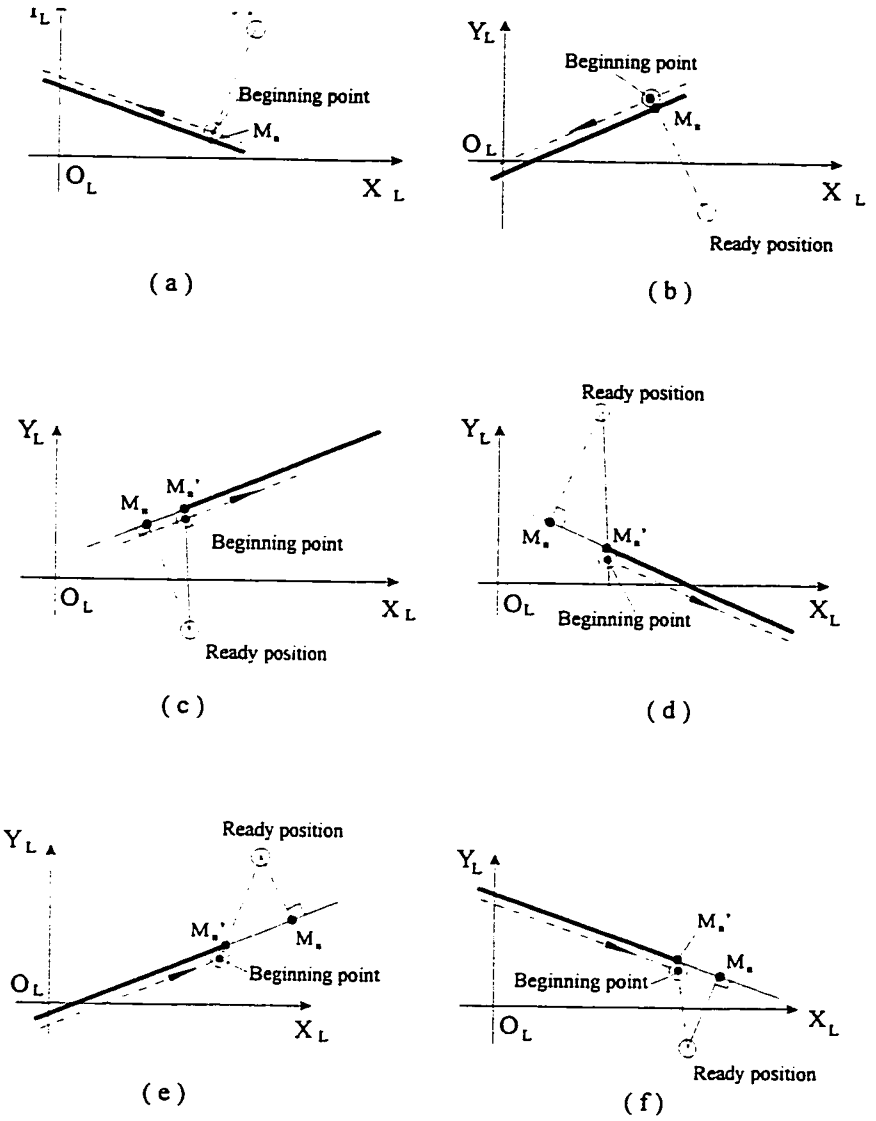


Figure 4.4(a) The Definition of the Nearest Point for Different Cases

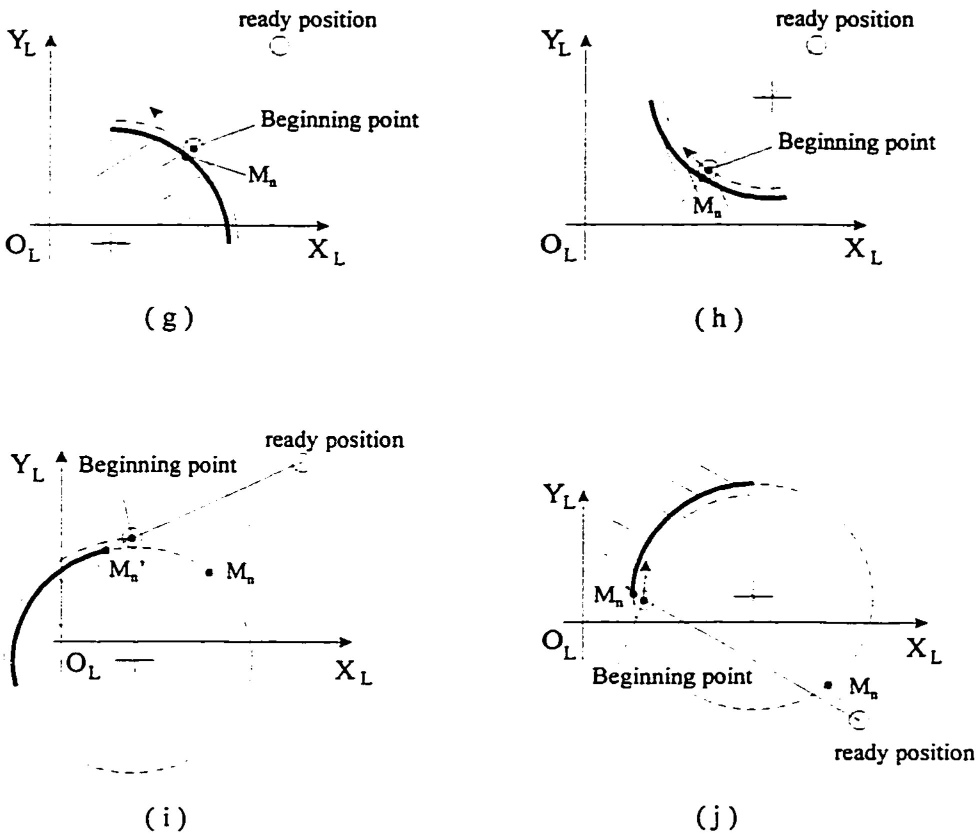


Figure 4.4(b) The Definition of the Nearest Point for Different Cases

- Search for the nearest point on the edges from the ready position by comparing the nearest distance of each entity on the edges, and take the entity with a smallest d value as the nearest entity. The nearest point on this entity is the nearest point on the edges.
- Based on the position of the nearest entity and the relationship between the ready position and the nearest entity, determine the beginning point on the toolpath segment (Figure 4.4).

  The beginning point is located on the line that passes through the ready position and the nearest point of the curve, and at a distance equal to the radius of the tool from the curve.
- The set point is selected to be above the first machining point with fixed distance d (Figure 4.1).

  It is necessary that the tool have a specific orientation at the set point as per the normals of the edges. The normal of the planar edge is always perpendicular to the plane, so the tool orientation is easily obtained, and it is fixed. For three-dimensional edges, the calculation of the normal is more complex, and described in Section 4.3.2.3.

4.3.2.2 Machining ToolPath for Two-Dimensional Edges

Machining toolpath for a two-dimensional curve consists of machining point on the toolpath and normals of the edges. This section deals with the calculations of these machining points and the normals.

From the beginning point, the tool starts machining. For each curve, a specific toolpath segment has to be assigned. A machining tool path consists of one or more straight or circular segments. The start point M_{1s} of the first segment and the position of the tool relative to the curve are determined in the previous section. The start point M_{is} of i^{th} segment and the position of the tool relative to the relating curve are determined based on the relationship between the i^{th} curve and the previous $(i-1)^{th}$ curve, and the end point M_{ie} of the i^{th} segment is calculated based the relationship between the i^{th} curve and the following $(i+1)^{th}$ curve. If the machining toolpath is closed, the end point M_{ie} of the last segment is the beginning point of the toolpath. If the machining tool path is not closed, the end point M_{ie} of the last is determined based on the last curve and the relation between the tool and the curve. In this section, the following cases are considered to generate the tool path.

Case 1: the first toolpath segment

The first segment of the toolpath is determined by the beginning point and the first entity. If the first entity is a line $\mathbf{L}_1$, let the parametric equations of the line $\mathbf{L}_i$ as

$$\mathbf{L}_1: \quad x_L = x_{11} + p_1 t$$

$$y_L = y_{11} + q_1 t \qquad (4.5)$$

The first segment of the toolpath, TP_1, is a line which is parallel to line $\mathbf{L}_1$ and with a distance $\frac{D_t}{2}$, where D_t is the diameter of the tool. The parametric equation of TP_1 is:

$$TP_1: \quad x_L = x_n + p_1 t$$

$$y_L = y_n + q_1 t \qquad (4.6)$$

where x_n and y_n are the coordinates of the beginning point.

If the first entity is a circular arc **C**$_1$, let the parametric equations of the line **L**$_i$ as

$$\mathbf{C}_1: \quad x_L = x_{c1} + R_1 \cos u_1$$

$$y_L = y_{c1} + R_1 \sin u_1 \tag{4.7}$$

The first segment of the toolpath, TP$_1$, is a circular arc which has the same center point as **C**$_1$ and with a distance $\frac{D_t}{2}$. The parametric equation of TP$_1$ is:

$$TP_1: \quad x_L = x_{c1} + (R_1 + F\frac{D_t}{2})\cos u_1$$

$$y_L = y_{c1} + (R_1 + F\frac{D_t}{2}) \sin u_1 \tag{4.8}$$

where F represents the bulge factor of the circular arc.

If the beginning point is inside **C**$_1$, the bulge factor of the first segment is set to be -1, if the beginning point is outside **C**$_1$, the bulge factor of the first segment is set to be +1. (See Figure 4.4)

Case 2: ith line **L**$_i$ followed by the (i+1)th line **L**$_{i+1}$

Let the parametric equations of the lines **L**$_i$ and **L**$_{i+1}$ as

$$\mathbf{L}_i: \quad x_L = x_{1i} + p_i\, t$$

$$y_L = y_{1i} + q_i\, t$$

$$\mathbf{L}_{i+1}: \quad x_L = x_{1(i+1)} + p_{i+1}\, t$$

$$y_L = y_{1(i+1)} + q_{i+1}\, t \tag{4.9}$$

The position of the tool relative to the line **L**$_i$ is known, that is the toolpath of the ith line, TP$_i$, is a line which is parallel to the line **L**$_i$, and at a distance $\frac{D_t}{2}$ from **L**$_i$. The parametric equation can be represented as

TP$_i$:
$$x_L = x_{1it} + p_i\, t$$
$$y_L = y_{1it} + q_i\, t \qquad (4.10)$$

Assume the tool moves to the point M$_{id}$, where the tool center lie on the line **L**$_{i+1}$ or on the extension line of **L**$_{i+1}$. The coordinates of this point in the local coordinate system is computed using the following:

$$x_{id} = \frac{b_{i+1} - b_{TPi}}{k_{TPi} - k_{i+1}} \qquad y_{id} = \frac{b_{i+1}k_{TPi} - b_{TPi}k_{i+1}}{k_{TPi} - k_{i+1}} \qquad (4.11)$$

where $k_i = \dfrac{q_i}{p_i}$ $\qquad b_i = y_{1i} - k_i\, x_{1i}$

$k_{TPi} = k_i \qquad b_{TPi} = b_i \pm \dfrac{D}{2\cos\alpha_i}$

$k_{i+1} = \dfrac{q_{i+1}}{p_{i+1}} \qquad b_{i+1} = y_{1(i+1)} - k_{i+1}\, x_{1(i+1)}$

p_i and q_i , are the direction numbers of the line **L**$_i$

p_{i+1} and q_{i+1} are the direction numbers of the line **L**$_{i+1}$

α_i is the angle between the line **L**$_i$ and the positive directive direction of X_L

α_{i+1} is the angle between the line **L**$_{i+1}$ and the positive directive direction of X_L

$\{x_{1i}, y_{1i}\}$ and $\{x_{1(i+1)}, y_{1(i+1)}\}$ are points on **L**$_i$ and **L**$_{i+1}$, respectively

The end point M$_{ie}$ of the toolpath segment TP$_i$ is the start point of the toolpath TP$_{i+1}$. The value of $k_{TP(i+1)}$ is the same as k_{i+1}, and the value of $b_{TP(i+1)}$ is determined from the coordinates of a point M$_{id}$. The point M$_{id}$ is the intersection point of the toolpath segment TP$_i$ and the (i+1)th curve. If y coordinate y_{is} of the starting point on TP$_i$, is smaller than y_{id} and M$_{id}$ is within the range of **L**$_{i+1}$ as shown in Figures 4.5(b) and (f), or y_{is} is larger than y_{id} and M$_{id}$ is outside the range of **L**$_{i+1}$ as shown in Figures 4.5(a) and (g), then

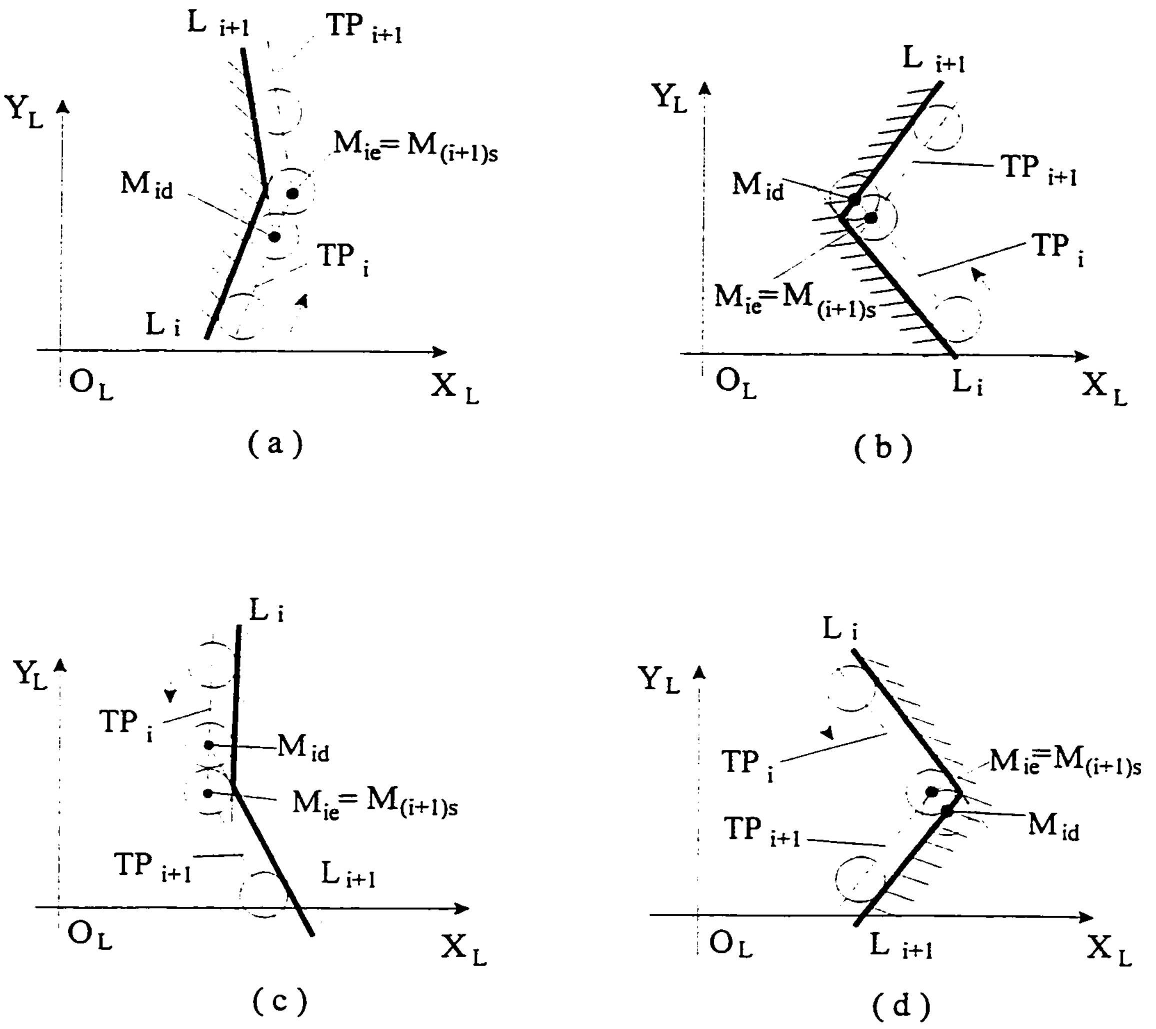


Figure 4.5 Locations of the Toolpath Segments for two Intersecting Straight Curves

$$k_{TP(i+1)} = k_{i+1} \qquad b_{TP(i+1)} = b_{i+1} - \frac{D_t}{2\cos\alpha}$$

or if y_{is} is smaller than y_{id} and M_{id} is out of the range of $\mathbf{L}_{i+1}$ as shown in Figures 4.5(c) and (e), or if y_{is} is larger than y_{id} and M_{id} is within the range of $\mathbf{L}_{i+1}$ as shown in Figures 4.5(d) and (h):

$$k_{TP(i+1)} = k_{i+1} \qquad b_{TP(i+1)} = b_{i+1} + \frac{D_t}{2\cos\alpha}$$

The end point of TP_i is:

$$x_{ie} = \frac{b_{TP(i+1)} - b_{TPi}}{k_{TPi} - k_{TP(i+1)}} \qquad y_{ie} = \frac{b_{TP(i+1)}k_{TPi} - b_{TPi}k_{TP(i+1)}}{k_{TPi} - k_{TP(i+1)}} \tag{4.12}$$

Case 3: i^{th} line $\mathbf{L}_i$ followed by the $(i+1)^{th}$ circular arc $\mathbf{C}_{i+1}$

Let the parametric equations of the line $\mathbf{L}_i$ and the circular arc $\mathbf{C}_{i+1}$ are:

$\mathbf{L}_i$: $x_L = x_{1i} + p_i\,t$

$$y_L = y_{1i} + q_i\,t \tag{4.13}$$

$\mathbf{C}_{i+1}$: $x_L = x_{c(i+1)} + R_{i+1}\cos u_{i+1}$

$$y_L = y_{c(i+1)} + R_{i+1}\sin u_{i+1} \tag{4.14}$$

If the end point of the line $\mathbf{L}_1$ is the start point of the circular arc $\mathbf{C}_{i+1}$, the bulge factor of the segment of the toolpath is set to be +1. If the end point of the line $\mathbf{L}_1$ is the end point of the circular arc $\mathbf{C}_{i+1}$, the bulge factor of the segment of the toolpath is set to be -1.

The tool path segment TP_i is a line which is parallel to the line $\mathbf{L}_i$, and with the distance $\frac{D_t}{2}$. The location of the toolpath segment TP_i depends on the calculation of previous entity (see the cases of the i^{th} line $\mathbf{L}_i$ followed by the $(i+1)^{th}$ line $\mathbf{L}_{i+1}$ and the i^{th} circular arc $\mathbf{C}_i$ followed by the $(i+1)^{th}$ line $\mathbf{L}_{i+1}$), that is the equation of TP_i is known.

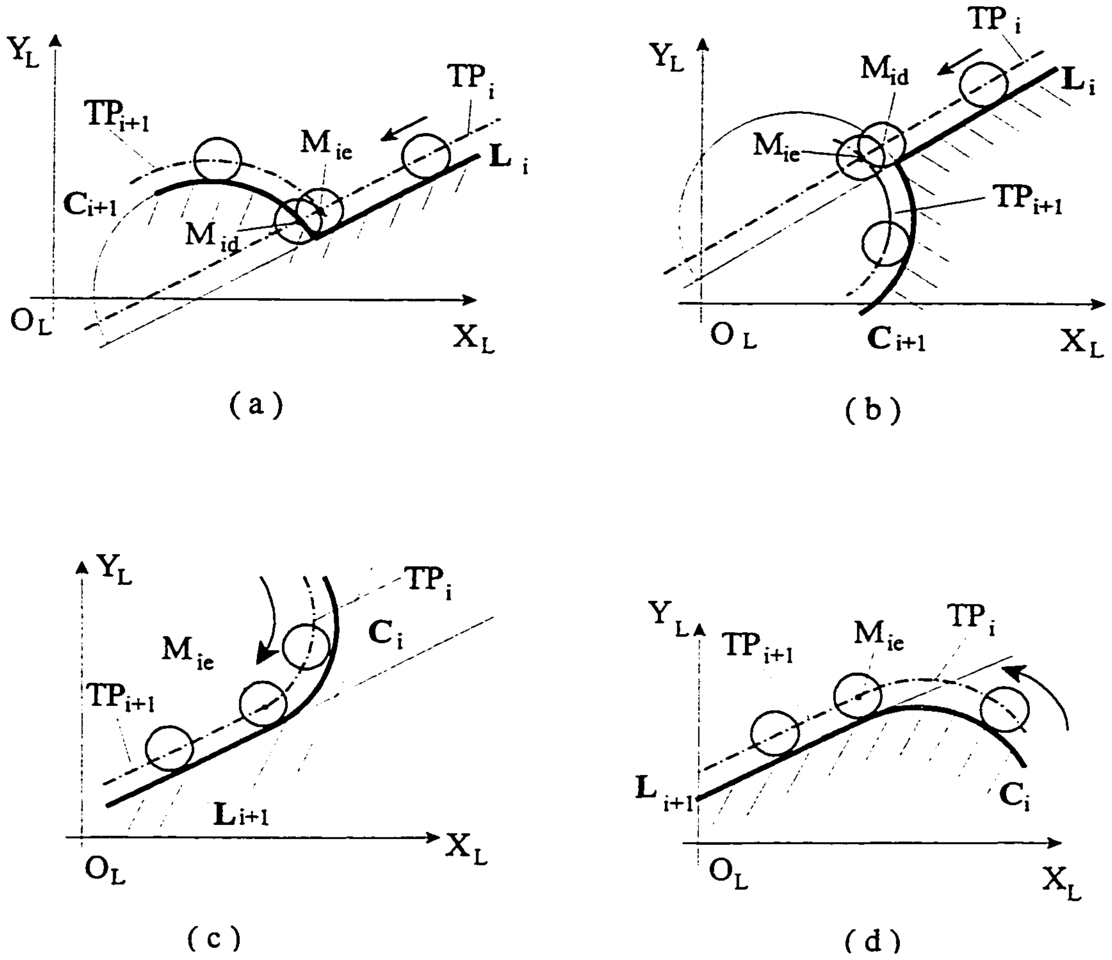


Figure 4.6 Connection between a Line and a Circular Arc (I)

Let:

$$\mathrm{TP}_i: \quad x_L = x_{1it} + p_i\, t$$

$$y_L = y_{1it} + q_i\, t \qquad (4.15)$$

According to the bulge factor, the parametric equation of the toolpath segment TP_{i+1} are obtained as

$$\mathrm{TP}_{i+1}: \quad x_L = x_{c(i+1)} + (\, R_{i+1} + F\ \frac{D_t}{2}\ )\cos u_{i+1}$$

$$y_L = y_{c(i+1)} + (\, R_{i+1} + F\ \frac{D_t}{2}\ )\sin u_{i+1} \qquad (4.16)$$

The end machining point of the toolpath TP_i is the intersection of TP_i ans TP_{i+1}. Solving the equations (4.11) and (4.12) simultaneously, obtain:

$$x_{ie1} = \frac{x_{c(i+1)} + y_{c(i+1)} k_{TP_i} - b_{TP_i} k_{TP_i}\ +\ \sqrt{H}}{1 + k_{TP_i}^2}$$

$$y_{ie1} = k_{TPi}\, x_{ie1} + b_{TPi}$$

$$x_{ie2} = \frac{x_{c(i+1)} + y_{c(i+1)} k_{TP_i} - b_{TP_i} k_{TP_i}\ +\ \sqrt{H}}{1 + k_{TP_i}^2}$$

$$y_{ie2} = k_{TPi}\, x_{ie2} + b_{TPi}$$

where

$$H = (x_{c(i+1)} + y_{c(i+1)}\, k_{TPi} - b_{TPi} k_{TPi}\,)^2 - (1 + k_{TPi}{}^2\,)(\, x_{c(i+1)}{}^2 + y_{c(i+1)}{}^2 + b_i^2 - \ (R_{i+1} \pm \frac{D_t}{2}\ )^2)$$

$$k_{TPi} = k_i = \ \frac{q_i}{p_i}$$

$$b_{TPi} = b_i \pm \frac{D_t}{2} / \cos\alpha = y_{1i} - k_i\, x_{1i} \pm \frac{D_t}{2} / \cos\alpha$$

If $H > 0$, then there are two intersection points, and the end machining point of TP_i is one of them which is nearer to the end point of the line $\mathbf{L}_i$ as shown in Figures 4.6.

If H = 0, the tool path segment TP_i is tangent to the toolpath segment TP_{i+1} as shown in Figures 4.6 (c) and (d). The coordinates of the tangent point is

$$x_{ie} = x_{ie1} = x_{ie2} = \frac{x_{c(i+1)} + y_{c(i+1)} k_{TPi} - b_{TPi} k_{TPi}}{1 + k_{TPi}^2}$$

H < 0 is not valid and should never take place because two adjacent curves are always intersecting or tangent to each other.

Case 4: ith circular arc $\mathbf{C}_i$ followed by the (i+1)th line $\mathbf{L}_{i+1}$

The parametric equations of the circular arc and the line are:

$$\mathbf{C}_i: \quad x_L = x_{ci} + R_i \cos u_i$$

$$y_L = y_{ci} + R_i \sin u_i \qquad (4.17)$$

$$\mathbf{L}_{i+1}: \quad x_L = x_{1(i+1)} + p_{i+1}\, t$$

$$y_L = y_{1(i+1)} + q_{i+1}\, t \qquad (4.18)$$

The toolpath segment TP_i is a circular arc which center point is $\{x_{ci}, y_{ci}\}$, and which radius is $R_t = R_i \pm \frac{D_t}{2}$. The parametric equation is

$$TP_i: \quad x_L = x_{ci} + R_t \cos u_t$$

$$y_L = y_{ci} + R_t \sin u_t \qquad (4.19)$$

Assume the tool moves along TP_i until the point M_{id}, at which the center of the tool lies on the line $\mathbf{L}_{i+1}$, or on the extension line of $\mathbf{L}_{i+1}$, as shown in Figure 4.7. The intersection points of the toolpath segment TP_i and the line $\mathbf{L}_{i+1}$ are obtained by solving the equations (4.14) and (4.15) simultaneously:

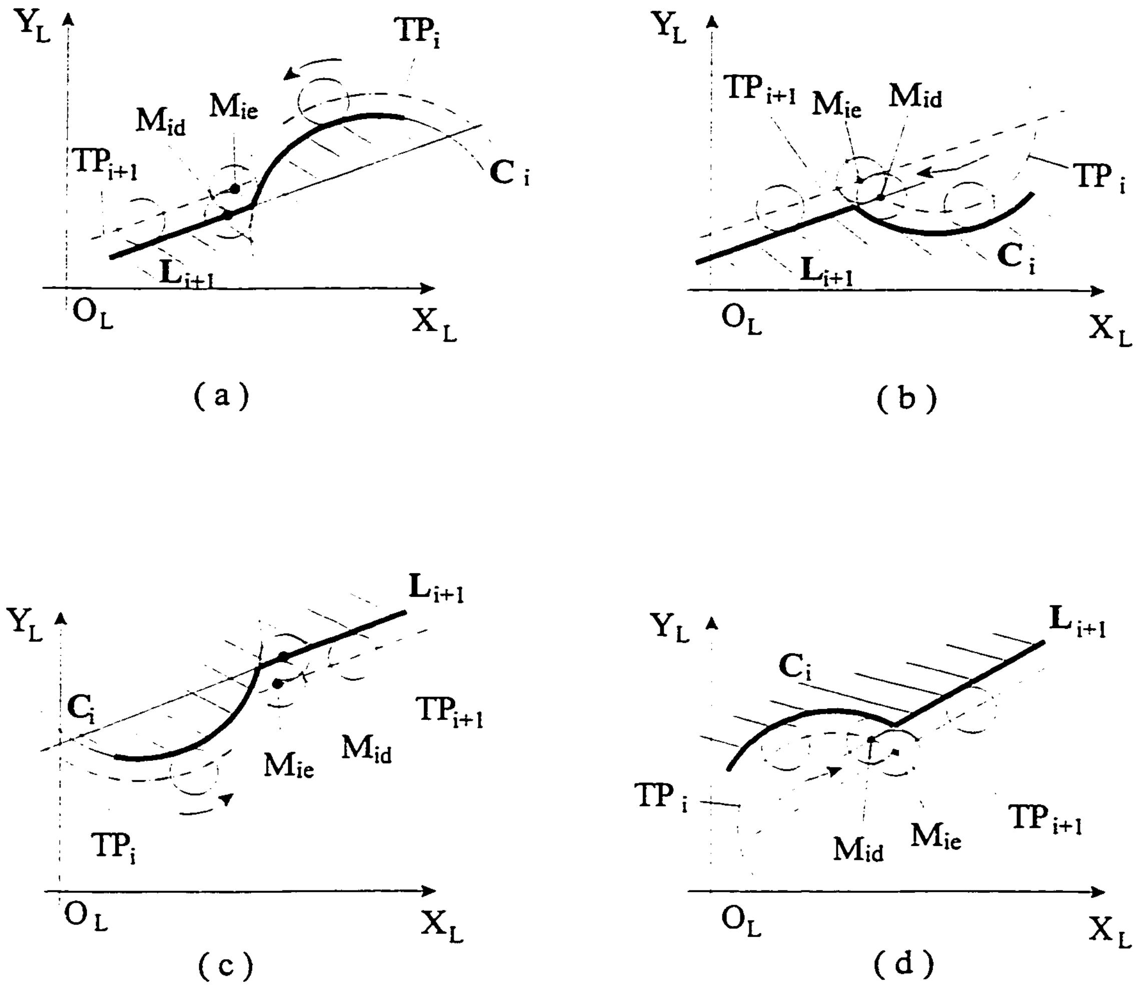


Figure 4.7 Connection between a Line and a Circular Arc (II)

$$x_{id1} = \frac{x_{ci} + y_{ci}k_{i+1} - b_{i+1}k_{i+1} + \sqrt{H_d}}{1 + k_{i+1}^2}$$

$$y_{id1} = k_{i+1}\, x_{id1} + b_{i+1}$$

$$x_{id2} = \frac{x_{ci} + y_{ci}k_{i+1} - b_{i+1}k_{i+1} - \sqrt{H_d}}{1 + k_{i+1}^2}$$

$$y_{id2} = k_{i+1}\, x_{id2} + b_{i+1}$$

where $k_{i+1} = \dfrac{q_{i+1}}{p_{i+1}}$ $\qquad b_{i+1} = y_{1i+1} - k_{i+1}\, x_{1i+1}$

$$H_d = (x_{ci} + y_{ci}\, k_{i+1} - b_{i+1}k_{i+1})^2 - (1 + k_{i+1}^2)(x_{ci}^2 + y_{ci}^2 + b_{i+1}^2 - R_t^2)$$

- If $H_d \geq 0$, the line $\mathbf{L}_{i+1}$ intersects with the TP_i, there are two intersection points. M_{id} is one of them which is nearer the end point of the circular arc $\mathbf{C}_i$ as shown in Figure 4.7. If the point M_{id} is in the range of the line $\mathbf{L}_{i+1}$ and the previous tool center M_{id}' $\{x_{id} - \Delta x, y_{id} - \Delta y\}$ is below the line $\mathbf{L}_{i+1}$, or if the point M_{id} is not in the range of the line $\mathbf{L}_{i+1}$ and the previous tool center M_{id}' $\{x_{id} - \Delta x, y_{id} - \Delta y\}$ is above the line $\mathbf{L}_{i+1}$, the parametric equation of the toolpath segment TP_{i+1} is

$$TP_{i+1}: \quad x_L = x_{1(i+1)} + p_{i+1}\, t$$

$$y_L = y_{1(i+1)} - \frac{D_t}{2\cos\alpha_{i+1}} + q_{i+1}\, t \qquad (4.20a)$$

- If the point M_{id} is in the range of the line $\mathbf{L}_{i+1}$ and the previous tool center M_{id}' $\{x_{id} - \Delta x, y_{id} - \Delta y\}$ is above the line $\mathbf{L}_{i+1}$, or if the point M_{id} is not in the range of the line $\mathbf{L}_{i+1}$ and the previous tool center M_{id}' $\{x_{id} - \Delta x, y_{id} - \Delta y\}$ is below the line $\mathbf{L}_{i+1}$, the parametric equation of the toolpath segment TP_{i+1} is

$$TP_{i+1}: \quad x_L = x_{1(i+1)} + p_{i+1}\, t$$

$$y_L = y_{1(i+1)} + \frac{D_t}{2\cos\alpha_{i+1}} + q_{i+1}\, t \qquad (4.20b)$$

- If $H_d < 0$, there exist two cases as shown in Figure 4.8: the line $\mathbf{L}_{i+1}$ is tangent to the circular arc $\mathbf{C}_i$; the line $\mathbf{L}_{i+1}$ intersects with the circular arc $\mathbf{C}_i$. the bulge factor of the circular arc $\mathbf{C}_i$ is -1, and the two intersection points are very close.

  Assume the tool moves along TP_i until the point M_{id}, at which point the machining point of the tool is the end point of the circular arc $\mathbf{C}_i$. If the tool center is below the line $\mathbf{L}_{i+1}$, the toolpath segment TP_{i+1} is below the line $\mathbf{L}_{i+1}$, and the parametric equation is

$$TP_{i+1}: \quad x_L = x_{1(i+1)} + p_{i+1}\, t$$

$$y_L = y_{1(i+1)} - \frac{D_t}{2\cos\alpha_{i+1}} + q_{i+1}\, t \qquad (4.20a)$$

If the tool center is above the line $\mathbf{L}_{i+1}$, the toolpath segment TP_{i+1} is above the line $\mathbf{L}_{i+1}$, and the parametric equation is

$$TP_{i+1}: \quad x_L = x_{1(i+1)} + p_{i+1}\, t$$

$$y_L = y_{1(i+1)} + \frac{D_t}{2\cos\alpha_{i+1}} + q_{i+1}\, t \qquad (4.20b)$$

The end point of the toolpath TP_i is the intersection point of the TP_i and TP_{i+1} and the coordinates are obtained by solving equations (4.15) and (4.16).

$$x_{ie1} = \frac{x_{ci} + y_{ci} k_{TP(i+1)} - b_{TP(i+1)} k_{TP(i+1)} + \sqrt{H}}{1 + k^2_{TP(i+1)}}$$

$$y_{ie1} = k_{TP(i+1)}\, x_{ie1} + b_{TP(i+1)}$$

$$x_{ie2} = \frac{x_{ci} + y_{ci} k_{TP(i+1)} - b_{TP(i+1)} k_{TP(i+1)} - \sqrt{H}}{1 + k^2_{TP(i+1)}}$$

$$y_{ie2} = k_{TP(i+1)}\, x_{ie2} + b_{TP(i+1)}$$

where $H = (x_{ci} + y_{ci}\, k_{TP(i+1)} - b_{TP(i+1)} k_{TP(i+1)})^2 - (1 + k_{TP(i+1)}{}^2)(x_{ci}{}^2 + y_{ci}{}^2 + b_{TP(i+1)}{}^2 - R_t{}^2)$

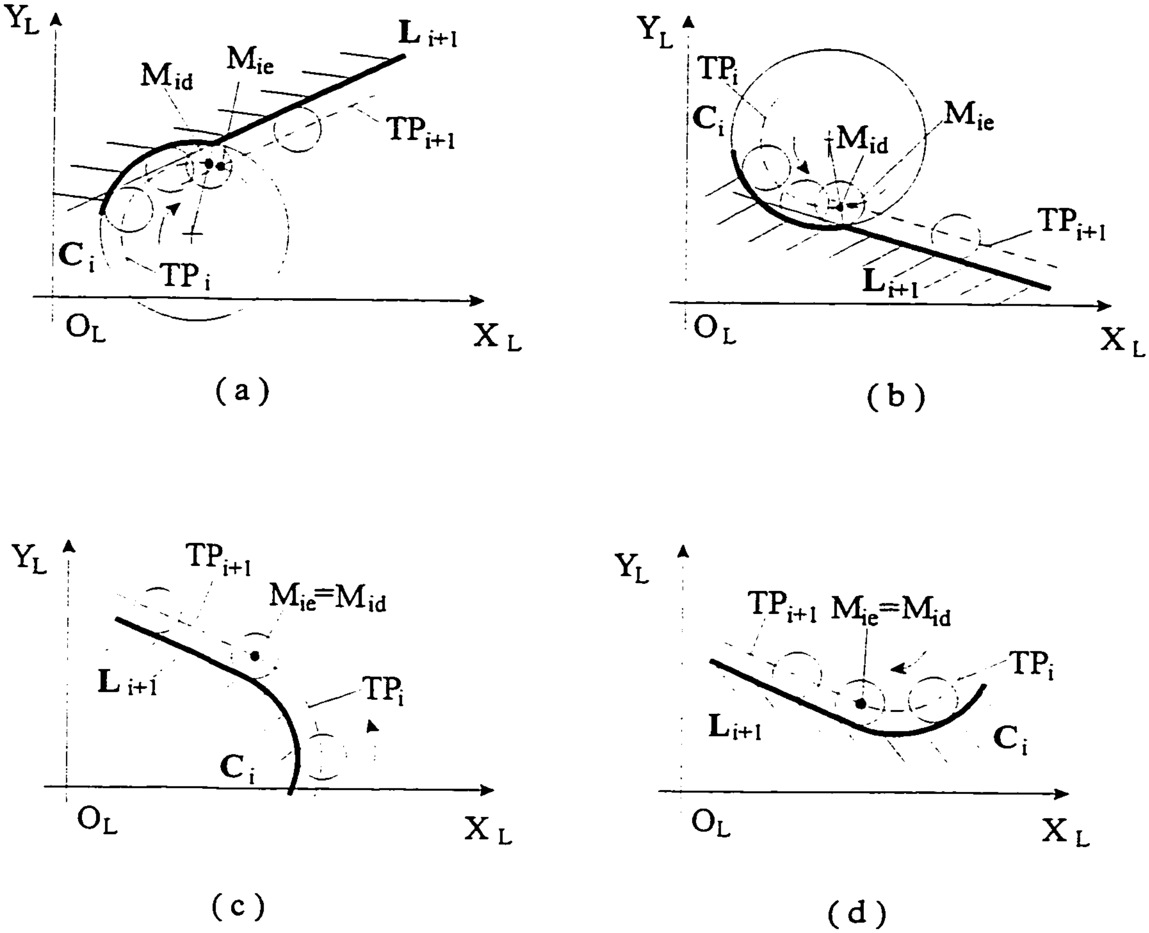


Figure 4.8 Connection between a Line and a Circular Arc (III)

M_{ie} is one of these two intersection points of TP_i and TP_{i+1}, which is near the end point of the circular arc $\mathbf{C}_i$.

$$x_{ie} = \frac{x_{ci} + y_{ci} k_{TP(i+1)} - b_{TP(i+1)} k_{TP(i+1)}}{1 + k^2_{TP(i+1)}}$$

$$y_{ie} = k_{TP(i+1)}\, x_{ie} + b_{TP(i+1)}$$

Case 5: ith circular arc $\mathbf{C}_i$ followed by the (i+1)th circular arc $\mathbf{C}_{i+1}$

Let the equations of the circular arcs of $\mathbf{C}_i$, $\mathbf{C}_{i+1}$, and TP_i as

$\mathbf{C}_i$: $x_L = x_{ci} + R_i \cos u_i$

$y_L = y_{ci} + R_i \sin u_i$ (4.21)

$\mathbf{C}_{i+1}$: $x_L = x_{c(i+1)} + R_{i+1} \cos u_{(i+1)}$

$y_L = y_{c(i+1)} + R_{i+1} \sin u_{(i+1)}$ (4.22)

TP_i: $x_L = x_{ci} + R_{TPi} \cos u_{TPi}$

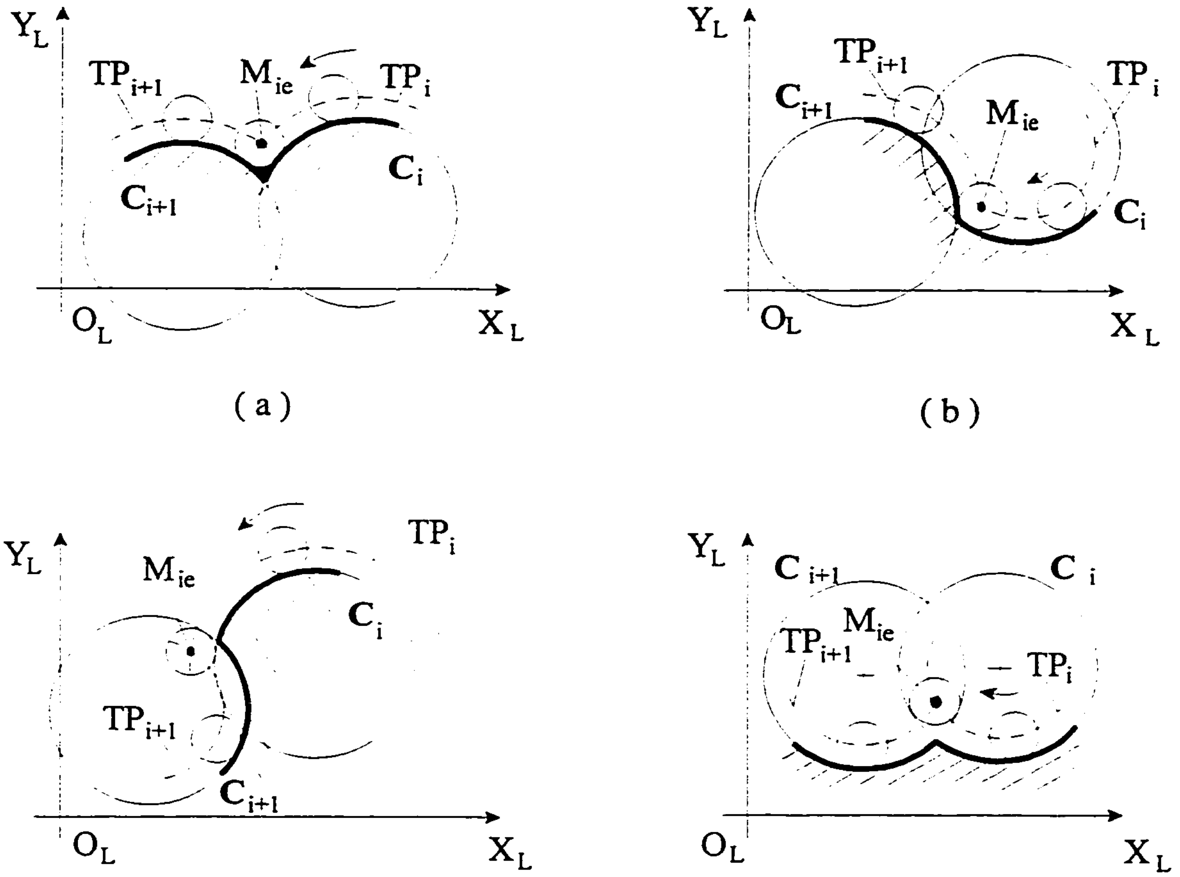
$y_L = y_{ci} + R_{TPi} \sin u_{TPi}$ (4.23)

The bulge factor of the circular arc $\mathbf{C}_{i+1}$ is determined as follows: if the end point of the circular arc $\mathbf{C}_i$, $\{x_{edi}, y_{edi}\}$, is the start point of the circular arc $\mathbf{C}_{i+1}$, the tool will be outside the circular arc $\mathbf{C}_{i+1}$, and the bulge factor equals to 1; if the point $\{x_{edi}, y_{edi}\}$ is the end point of the circular arc $\mathbf{C}_{i+1}$, the tool will be inside the circular arc $\mathbf{C}_{i+1}$, and the bulge factor equals to -1. The parametric equation of the toolpath segment TP_{i+1} is

TP_{i+1}: $x_L = x_{c(i+1)} + R_{TP(i+1)} \cos u_{TP(i+1)}$

$y_L = y_{c(i+1)} + R_{TP(i+1)} \sin u_{TP(i+1)}$ (4.24)

The intersection points of the toolpath segment TP_i and TP_{i+1} are obtained by solving equations (4.19) and (4.20).



(a)

(b)

(c)

(d)

Figure 4.9 Connection between Two Circular Arcs (I)

$$x_{ie1} = \frac{x_{ci} + D_1 y_{ci} - D_1 D_2 \ + \ \sqrt{H}}{1 + D_1^2}$$

$$y_{ie1} = D_1 \, x_{ie1} + D_2$$

$$x_{ie2} = \frac{x_{ci} + D_1 y_{ci} - D_1 D_2 \ - \ \sqrt{H}}{1 + D_1^2}$$

$$y_{ie2} = D_1 \, x_{ie2} + D_2$$

where $D_1 = \dfrac{x_{ci} - x_{c(i+1)}}{y_{c(i+1)} - y_{ci}}$

$$D_2 = \frac{R_{TPi}^2 - R_{TP(i+1)}^2 + x_{c(i+1)}^2 - x_{ci}^2 + y_{c(i+1)}^2 - y_{ci}^2}{2(y_{c(i+1)} - y_{ci})}$$

$$H = (x_{ci} + D_1 \, y_{ci} - D_1 D_2)^2 - (1 - D_1^{\ 2})(x_{xi}^{\ 2} + y_{ci}^{\ 2} + D_2^{\ 2} - R_{TPi}^{\ 2})$$

- If H > 0, two circular arcs $\mathbf{C}_i$ and $\mathbf{C}_{i+1}$ intersect each other, and there are two intersection points, as shown in Figure 4.9. The end point of TP_i is one of them (M_{ie1}, M_{ie2}) which is nearer to the end point of the circular arc $\mathbf{C}_i$. When the bulge factors for two intersecting circular arcs are 1, a fillet is present.
- If H = 0, two circular arcs $\mathbf{C}_i$ and $\mathbf{C}_{i+1}$ are tangent to each other, and there is a tangent point, as shown in Figures 4.10(a) and (b). The end point of the toolpath segment TP_i is the tangent point of TP_i and TP_{i+1} .
- If H < 0, there exist two cases: one is the two circular arcs are tangent to each other and one is outside the another; the other is the two circular arcs intersect and the two intersection points are very close. In both cases, the bulge factors for two circular arcs

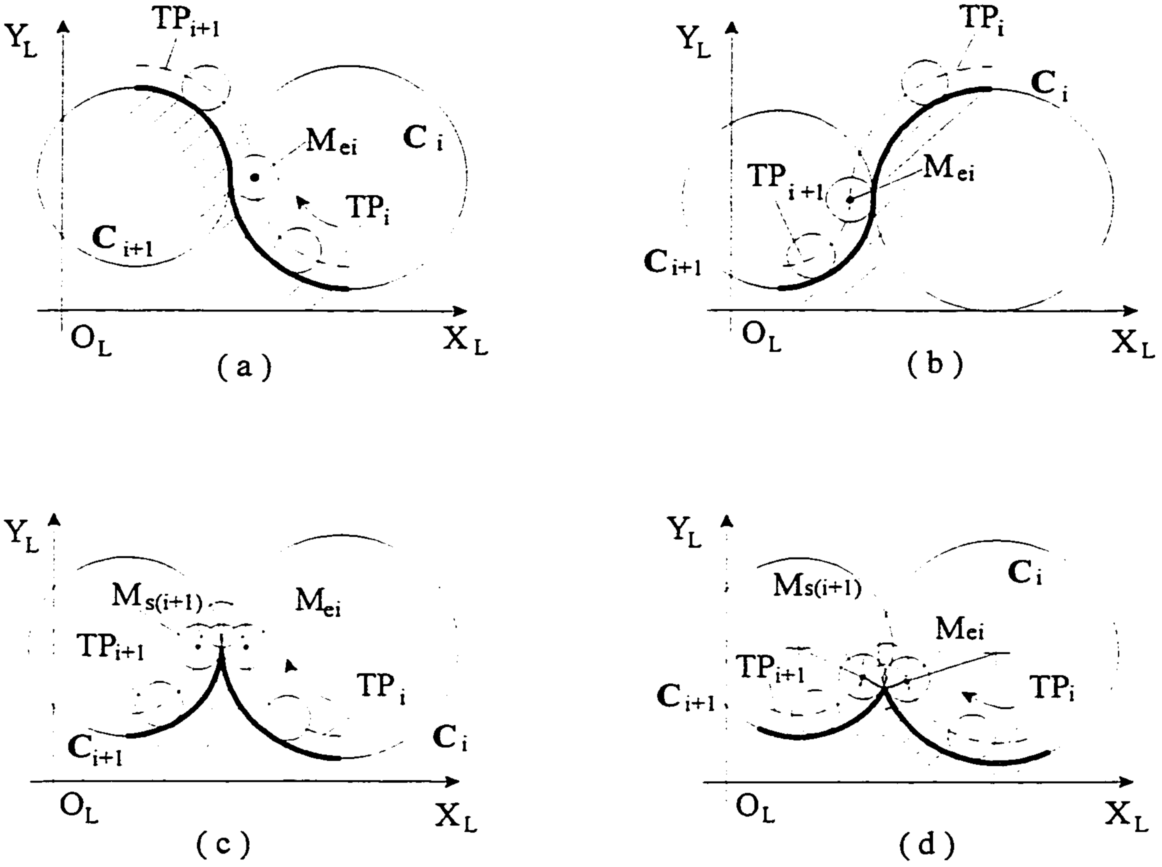


Figure 4.10 Connection between Two Circular Arcs (II)

are -1, and the tool is within the relative circular arc as shown in Figures 4.10 (c) and (d). The end point M_{ie} is the point at which the machining point of the tool is the end point of the circular arc $\mathbf{C}_i$, and the start point $M_{(i+1)s}$ of the toolpath segment TP_{i+1} is the point at which the machining point of the tool is the start point of the circular arc $\mathbf{C}_{i+1}$. The transition from M_{ie} to $M_{(i+1)s}$ is also a circular arc whose center is the intersection point of two circular arcs $\mathbf{C}_i$ and $\mathbf{C}_{i+1}$, and its radius is the same as the tool radius $\frac{D_t}{2}$.

After the starting point and end point are decided, the middle points of a toolpath segment are calculated. The relative parametric equation of the toolpath segment derived above can be used to generate these middle points on the toolpath segment by increasing t from t_s to t_e, or u from u_s to u_e, where t_s to t_e are determined by the starting point and the end point of the straight line toolpath segment, u_s and u_e are determined by the starting point and the end point of the toolpath segment.

4.3.2.3 Machining Toolpath for Three-Dimensional Edges

The calculation of a toolpath of a three-dimensional entity includes two steps. The first is to calculate the coordinates of the machining points on the toolpath segment. The second is to calculate the normals of the curve. Assume a three-dimensional curve is approximated by a sequence of line segments, and the toolpath for these segments for ball-end cutters. According to the calculated normal, the coordinates of the ball center of the tool can be determined.

In this thesis, only the intersection curves of two regular surfaces are considered. All calculations are based on the relative parametric equations derived in Chapter 3. In an ACIS data file, a certain number of points on the intersection curve can be extracted, and these points are used as the desired machining points on the toolpath.

(1) Points on the Machining Toolpath

Consider a coordinate system {L} such that Y_L is coincident with the axis of the cylinder $\mathbf{C}_2$, a coordinate system {L'} is fixed on the cylinder $\mathbf{C}_1$ such that the Z_L axis is parallel to the axis of the smaller cylinder as shown in Figure 4.11. The coordinate system {L'} is obtained by translating along X_L by a distance "d", rotating about X_L by α_2 degrees, or/and rotates about $Y_{L'}$ by α_1 degrees.

The intersection curve of two cylindrical surfaces can be expressed as:

$$\mathbf{P}_L(u_1, u_2) = x_L(u_1, u_2)\,\mathbf{i} + y_L(u_1, u_2)\,\mathbf{j} + z_L(u_1, u_2)\,\mathbf{k} \tag{4.25}$$

where $x_L = x_{c2} + R_2 \cos(u_2)$

$$y_L = y_{c1} + R_1 \sin(\phi_1)\sin(\phi_2)\cos(u_1) + R_1\cos(\phi_2)\sin(u_1)$$

$$z_L = z_{c2} + R_2\sin(u_2)$$

$x_{c1}, y_{c1}, z_{c1}, x_{c2}, y_{c2}, z_{c2}$ are the coordinates of two points that lie on the axes of the two cylinders respectively. R_1 and R_2 are the radii of two cylinders respectively. γ' is the angle between Z_L and $Z_{L'}$;

$$0 \le u_1 \le 2\pi;$$

$$(\gamma' - \phi_1) \le u_2 \le (\gamma' + \phi_2);$$

$$\phi_1 = \sin^{-1}\left[\frac{R_1 + d}{R_2}\right], \text{ and } \phi_2 = \sin^{-1}\left[\frac{R_1 - d}{R_2}\right];$$

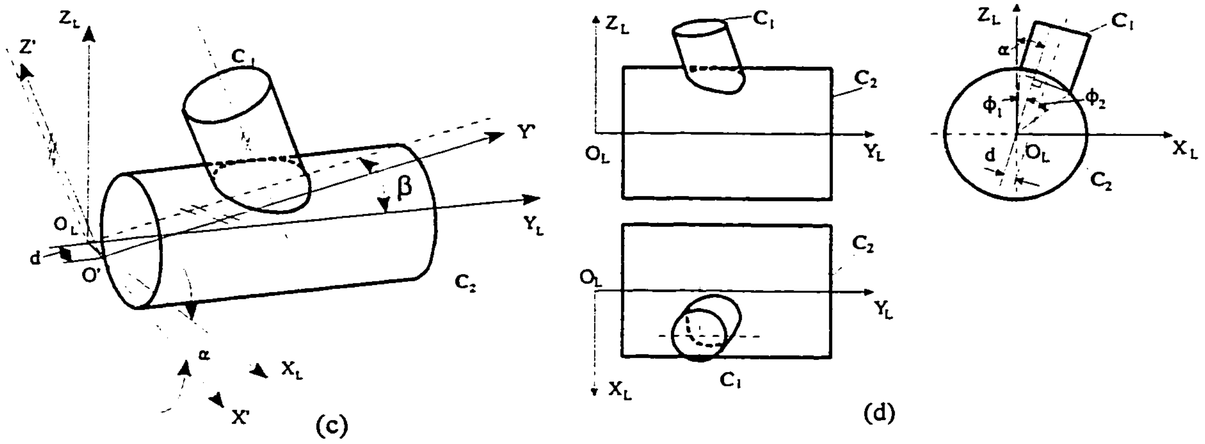


Figure 4.11 Representation of Two Intersecting Cylinders

$$\gamma' = \tan^{-1} [\,\frac{ny}{nz}\,];$$

The relationship between u_1 and u_2 is:

$$u_2 = \sin^{-1} [\,\frac{R_1 \cos u_1}{R_2}\,] + \gamma'$$

The parametric equation of a three-dimensional curve can be represented as[23]

$$\mathbf{P}_L(u_1) = x_L(u_1)\,\mathbf{i} + y_L(u_1)\,\mathbf{j} + z_L(u_1)\,\mathbf{k} \qquad (4.26)$$

For each value of u_1, there exists a unique point in the 3D space.

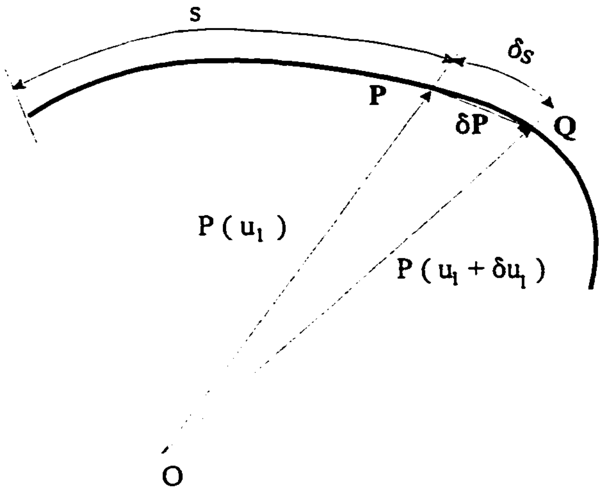


Figure 4.12 Representation of a Three-Dimensional Curve[25]

(2) The Normal of the Machining Toolpath

The intersection curve of two cylindrical surfaces is a twisted space curve, and represented by (4.22). In Figure 4.12, the vector $\delta \mathbf{P} = \mathbf{P}(u_1 + \delta u_1) - \mathbf{P}(u_1)$ represents the chord PQ that joints two machining points P and Q with parameters u_1 and $u_1 + \delta u_1$ on the curve represented by (4.22). As $\delta u_1 \rightarrow 0$, the direction of the vector $\delta \mathbf{P}/\delta u_1$ approaches the direction of the tangent at P, if the curve has a well-defined tangent[25].

$$\frac{d\boldsymbol{P}}{du_1} = \lim_{\delta u_1 \to 0} \frac{\delta \boldsymbol{P}}{\delta u_1}$$

$$\text{If } \frac{d\boldsymbol{P}}{du_1} \neq 0, \ \mathbf{T} = \frac{\delta \boldsymbol{P}/\delta u_1}{|\delta \boldsymbol{P}/\delta u_1|} \tag{4.27}$$

T is a vector of unit length in the direction of the tangent to the curve, and it is known as the unit tangent vector.

For a two-dimensional curve, a well-defined normal exists. However, for a three-dimensional curve any vector perpendicular to the tangent vector **T** is a normal vector. In particular, because **T** is a unit vector, the vector $\dot{T} = \frac{dT}{du_1}$ is normal to **T**. The unit vector **N** in the direction of $\dot{T}$ is known as the principal normal vector[25].

$$\mathrm{N} = \dot{T} = \frac{dT}{du_1} \tag{4.28}$$

The vector product **T** × **N** defines a third unit vector perpendicular to **T** and **N**, known as the binormal vector **B**.

The three vectors **T**, **N**, and **B** form a right-handed set of mutually orthogonal unit vectors such that **B** = **T** × **N**, **T** = **N** × **B** and **N** = **B** × **T**. The planes through a given point on the curve which contain the vectors **T** and **N** , **N** and **B**, and **B** and **T** respectively are known as the osculating plane, the normal plane and the rectifying plane (see Figure 4.13)[25].

The osculating circle of the curve at P is the limit as $\delta u_1 \to 0$ of the circle drawn through the points N, P and Q on the curve which have parameters $u_1 - \delta u_1$, u_1 and $u_1 + \delta u_1$. This circle lies on the osculating plane by noting the plane of the circle contains both

$\overrightarrow{PQ}$ and $\overrightarrow{PN}$. Thus the normal to the plane of the circle has direction of the vector product of $\overrightarrow{PQ}$ and $\overrightarrow{PN}$

$$\overrightarrow{PQ} \times \overrightarrow{PN} = [\mathbf{P}(u_1 + \delta u_1) - \mathbf{P}(u_1)] \times [\mathbf{P}(u_1 - \delta u_1) - \mathbf{P}(u_1)]$$

$$= [\, \delta u_1 \, \dot{\boldsymbol{P}}(u_1) + \frac{\delta u_1^2}{2} \, \ddot{\boldsymbol{P}} \, (u_1)] \times [\, - \delta u_1 \, \dot{\boldsymbol{P}}(u_1) + \frac{\delta u_1^2}{2} \ddot{\boldsymbol{P}}(u_1)] + O(\, \delta u_1^4)$$

$$= \; \delta u_1^3 \, [\, \delta u_1 \, \dot{\boldsymbol{P}}(u_1) \times \, \ddot{\boldsymbol{P}} \, (u_1)\,] + \; O(\, \delta u_1^4)$$

by the use of Taylor's Theorem. Thus the normal to the plane of the osculating circle has the direction of $\dot{\boldsymbol{P}} \times \ddot{\boldsymbol{P}}$, which is used to align the tool to machining.

$$\mathbf{B} = \; \dot{\boldsymbol{P}} \times \ddot{\boldsymbol{P}} = \mathbf{T} \times \mathbf{N} \qquad (4.29)$$

For a intersection curve,

$$\mathrm{P}_L(\, u_1\,) = x_L(\, u_1\,)\, \mathbf{i} + y_L(\, u_1\,)\, \mathbf{j} + \; z_L(\, u_1\,)\, \mathbf{k}$$

$$\frac{d\boldsymbol{P}_L(u_1)}{du_1} = \frac{dx_L(u_1)}{du_1}\,\mathbf{i} + \frac{dy_L(u_1)}{du_1}\,\mathbf{j} + \frac{dz_L(u_1)}{du_1}\,\mathbf{k} \qquad (4.30)$$

Assume

$$\mathrm{H} = | \; \frac{d\boldsymbol{P}_L(u_1)}{du_1} \; | = \sqrt{\; [\; dx_L(u_1)/du_1 \;]^2 + \; [\; dy_L(u_1)/du_1 \;]^2 + \; [\; dz_L(u_1)/du_1 \;]^2}$$

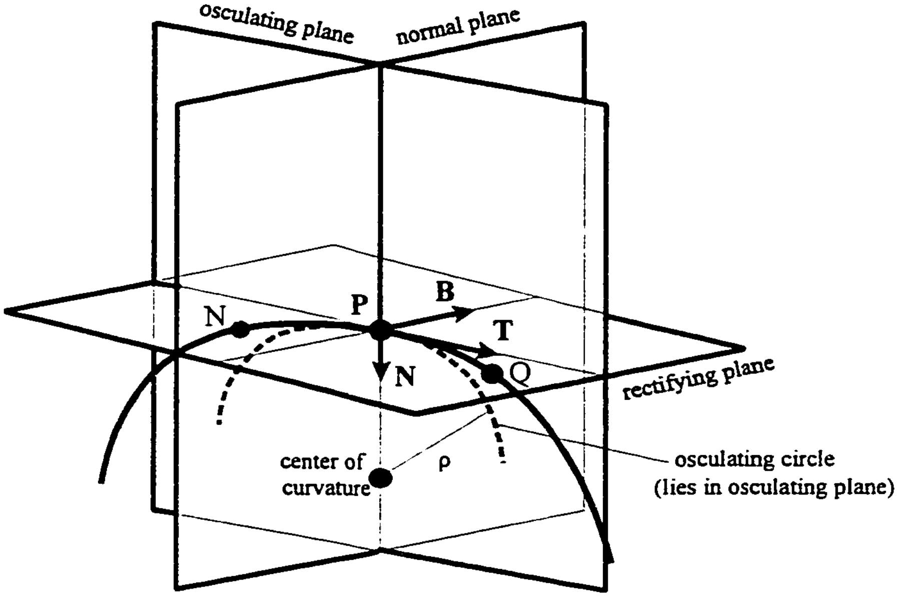


Figure 4.13 Osculating, Normal and Rectifying Planes of a Curve [25]

$$\mathbf{T} = \frac{d\boldsymbol{P}_L(u_1)/du_1}{|d\boldsymbol{P}_L(u_1)/du_1|} = \frac{d\boldsymbol{P}_L(u_1)/du_1}{H}$$

$$= [\ \frac{dx_L(u_1)/du_1}{H}]\ \mathbf{i} + [\frac{dy_L(u_1)/du_1}{H}\ ]\ \mathbf{j} + [\frac{dz_L(u_1)/du_1}{H}\ ]\ \mathbf{k}$$

$$= [\ \frac{\dot{x}_L(u_1)}{H}]\ \mathbf{i} + [\frac{\dot{y}_L(u_1)}{H}\ ]\ \mathbf{j} + [\frac{\dot{z}_L(u_1)}{H}\ ]\ \mathbf{k}$$

$$= \mathrm{f}_x'\ \mathbf{i} + \ \mathrm{f}_y'\ \mathbf{j}\ + \mathrm{f}_z'\ \mathbf{k}$$

$$\mathrm{f}_x' = \frac{\dot{x}_L(u_1)}{H}$$

$$\mathrm{f}_y' = \frac{\dot{y}_L(u_1)}{H}$$

$$\mathrm{f}_z' = \frac{\dot{z}_L(u_1)}{H}$$

$$\mathbf{N} = \dot{\boldsymbol{T}}$$

$$= \mathrm{f}_x''\ \mathbf{i} + \mathrm{f}_y''\,\mathbf{j} + \mathrm{f}_z''\,\mathbf{k}$$

$$\mathrm{f}_x'' = \frac{\ddot{x}_L(u_1)\ H - x_L(u_1)\ \dot{H}}{H^2}$$

$$\mathrm{f}_y'' = \frac{\ddot{y}_L(u_1)\ H - y_L(u_1)\ \dot{H}}{H^2}$$

$$\mathrm{f}_z'' = \frac{\ddot{z}_L(u_1)\ H - z_L(u_1)\ \dot{H}}{H^2}$$

$$\mathbf{B} = \begin{vmatrix} i & j & k \\ f_x' & f_y' & f_z' \\ f_x'' & f_y'' & f_z'' \end{vmatrix}$$

where

$$\dot{H} = \frac{\dot{x}_L(u_1)\ \ddot{x}_L(u_1)\ +\ \dot{y}_L(u_1)\ \ddot{y}_L(u_1)\ +\ \dot{z}_L(u_1)\ \ddot{z}_L(u_1\ )}{\sqrt{H}}$$

For a intersecting curve presented in Figure 4.11:

$$\mathbf{T} = \mathrm{f}_x'\ \mathbf{i} + \mathrm{f}_y'\,\mathbf{j} + \mathrm{f}_z'\,\mathbf{k}$$

where

$$\mathrm{f}_x' = \frac{dx_L(u_1)}{du_1} / \mathrm{H}$$

$$= [(- R_1 \sin u_1 \cos\alpha) + \frac{R_2 \cos u_2 \dot{u}_2 - R_1 \sin u_1 \sin\alpha \sin\beta + R_1 \cos u_1 \cos\beta}{\cos\alpha \sin\beta} \sin\alpha] / H$$

$$f_y' = \frac{dy_L(u_1)}{du_1} / H$$

$$= [- R_1 \sin u_1 \cos\beta \sin\alpha + R_1 \cos u_1 \sin\beta - R_2 \cos u_2 \dot{u}_2 + R_1 \sin u_1 \sin\alpha \sin\beta$$

$$- R_1 \cos u_1 \cos\beta] / H$$

$$f_z' = \frac{dz_L(u_1)}{du_1} / H$$

$$= [R_2 \cos u_2 \dot{u}_2] / H$$

$$\dot{u}_2 = \frac{R_1 \sin u_1}{R_2 \sqrt{1 - (\frac{R_1 \cos u_1}{R_2})^2}}$$

$$\mathbf{N} = \frac{\ddot{x}_L(u_1)\ H - x_L(u_1)\ \dot{H}}{H^2}\ \mathbf{i} + \frac{\ddot{y}_L(u_1)\ H - y_L(u_1)\ \dot{H}}{H^2}\ \mathbf{j} + \frac{\ddot{z}_L(u_1)\ H - z_L(u_1)\ \dot{H}}{H^2}\ \mathbf{k}$$

where

$$\ddot{x}_L(u_1) = -R_1 \cos u_1 \cos\alpha +$$

$$\frac{-R_2 \cos u_2 \dot{u}_2^2 + R_2 \cos u_2 \ddot{u}_2 - R_1 \cos u_1 \sin\alpha \sin\beta - R_1 \sin u_1 \cos\beta}{\cos\alpha \sin\beta} \sin\alpha$$

$$\ddot{y}_L(u_1) = - R_1 \cos u_1 \sin\alpha \cos\beta - R_1 \sin u_1 \sin\beta + R_2 \sin u_2 \dot{u}_2^2 - R_2 \cos u_2 \ddot{u}_2$$

$$+ R_1 \cos u_1 \sin\alpha \sin\beta + R_1 \sin u_1 \cos\beta$$

$$\ddot{z}_L(u_1) = - R_2 \sin u_2 \dot{u}_2^2 + R_2 \cos u_2 \ddot{u}_2$$

$$\ddot{u}_2 = \frac{R_1 R_2 (1 - (\frac{R_1 \cos u_1}{R_2})^2) \cos u_1 - R_1^2 \sin u_1^2}{R_2 \sqrt[3]{(1 - (\frac{R_1 \cos u_1}{R_2})^2}}$$

The step value of the parameter u_1 affect the normal. Whether the method of the equal angular increments or the method of equal distances between two adjacent points is used to calculate the machining points and relative normal, the results will be unacceptable, because the normals change significantly for some ranges. In the interface, the method of the equal angle increments combined with smaller step values in the special ranges.

4.4. Conclusion

Generation of toolpath is discussed in this Chapter. The points on the toolpath are all calculated based on the parametric equations derived in last Chapter. For two-dimensional curve, the cutter offset, accuracy of toolpath are considered, and for three-dimensional curve, zero cutter offset is assumed. The toolpath calculations in the interface software is coded based on the steps described here. The results of calculations are shown in Chapter 7.

CHAPTER 5 KINEMATICS OF YAMAHA ZETA-1 DEBURRING ROBOT

5.1 Introduction

The Yamaha Zeta-1 robot is a programmable mechanical manipulator, capable of moving along X, Y, and Z directions and rotating about α and β axes as shown in described in Chapter 1, equipped at its end with a tool to machine components. The Yamaha Zeta-1 robot is developed by Yamaha Corporation, Japan, especially for the task of edge deburring and surface polishing operations. The end-effector of the Yamaha Zeta-1 robot is the deburring/polishing tool.

Kinematics is the science of motion which treats motion without regard to the forces. There are two types of formulations in robot kinematics analysis: the forward kinematics and the inverse kinematics. For a given set of joint angles, the forward kinematic formulation computes the position and orientation of the end-effector. For a given position and orientation of the end-effector, the inverse kinematic formulation calculates all possible sets of joint angles.

A robotic manipulator may be thought of as a set of rigid bodies connected in a chain through joints. These bodies are called the links. Joints form a connection between a neighbouring pair of links. Due to mechanical design considerations, manipulators are generally constructed from joints, and each joint exhibits one degree of freedom. In order to develop a kinematic model of the robot, a coordinate frame is embedded in each link of the manipulator. The mechanism and structure of the Yamaha Zeta-1 robot is presented in Section 5.2. Section 5.3 presents the kinematic formulations developed for the Yamaha Zeta-1 robot, and the relationship between two sets of systems; the workpiece coordinate system X_w, Y_w, Z_w, nx, ny, nz and the Yamaha Zeta-1 robot control coordinate system X_r, Y_r, Z_r, A, B is described in Section 5.4

5.2 Mechanism and Structure of the Yamaha Zeta-1 Robot

The Yamaha Zeta-1 deburring robot is structurally a cylindrical coordinates type robot. It has five degrees of freedom: R, θ, Z, α, and β. R is arm's radial motion; θ is main body's rotation about the Z-axis (vertical to the floor); Z is arm's up-down motion; α is the turning of the wrist about a vertical axis; β is the turning of the wrist about an inclined axis. For control purposes, the robot movement is computed with respect to a Cartesian coordinate system $\{X_r, Y_r, Z_r, A, B\}$, where A is the tool turning angle, and B is the tool collapse angle. A transformation between these two system is needed in real-time control of the robot. Usually a workpiece is defined in the workpiece coordinate system $\{X_w, Y_w, Z_w, nx_w, ny_w, nz_w\}$, where X_w, Y_w and Z_w are the coordinates of a point on the machining edge, nx_w, ny_w, and nz_w are the direction cosines of the normal of machining edge. As mentioned before, in order to utilize this interface to different kinds robots or machining tools, a processor is built based on the transformation equations between workpiece coordinate system and the robot control system, which are deduced from the kinematic formulation, refer to Figure 1.2. The kinematic models of the Yamaha Zeta-1 robot are developed using algebraic approach. In Appendix B, the geometric approach is presented. The results from two different approaches are used to verify the correctness of the formulation developed.

5.3 Kinematics of the Yamaha Zeta-1 Deburring Robot

The Yamaha Zeta-1 robot has four revolute active joints, a sliding active joint, and one passive revolute joint. Two active revolute joints and the sliding joint provide the θ, R, and Z, while the remaining two active revolute joints provide α and β. An algebraic approach is used to obtain explicit expressions to compute the joint parameters (θ, R, Z, α and β) and Cartesian parameters (X, Y, Z, A, and B). The transformation approach to deduce the kinematics of the Yamah robot is developed in this section.

5.3.1 Link Parameters

The links are numbered starting from the immobile base of the robot, which is called link 0. The first moving body is link 1, and so on, out to the free end of the arm, which is link 5. A link is considered only as a rigid body which defines the relationship between two neighbouring joint axes of the robot. Joint axes are defined by lines in space. Joint axis i is defined by a line in a vector direction, about which link i rotates relative to link i-1 [26].

For any two axes in 3D-space there exists a well-defined measure of distance between them. This distance is measured along a line which is mutually perpendicular to both axes. This mutual perpendicularity always exists and is unique except when both axes are parallel.

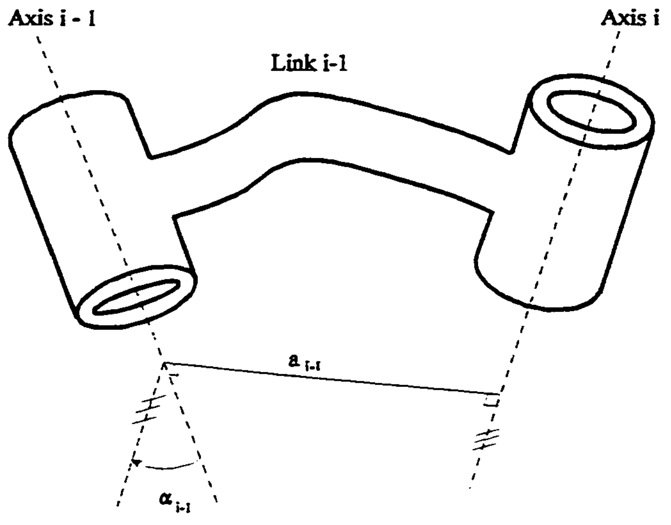


Figure 5.1 Link Length a_{i-1} and Link Twist α_{i-1}[26]

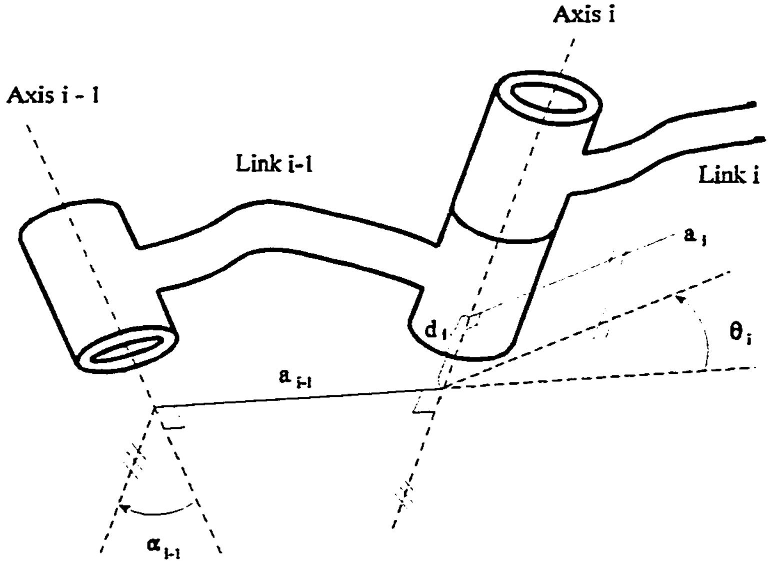


Figure 5.2 Link Offset d_i and Joint Angle θ_i[26]

Figure 5.1 shows link i-1 and the mutually perpendicular line along which the link length, a_{i-1}, is measured. The second parameter needed to define the relative location of the two axes is called the link twist, α_{i-1}. In Figure 5.1, α_{i-1} is indicated as the angle between axis i-1 and axis i (the lines with the triple hash marks are parallel). This angle is measured from axis i-1 to axis i in the right-hand sense about a_{i-1}[26].

Neighbouring links have a common joint axis between them. One parameter of interconnection has to do with the distance along this common axis from one link to the next. This parameter is called the link offset. The offset at joint axis i is donated as d_i. The second parameter describes the amount of rotation about this common axis between one link and its neighbour. This is called the joint angle, θ_i [26]. Figure 5.2 shows the interconnection between links i-1 and i.

The Yamaha Zeta-1 robot can be described kinematically by specifying the values of four quantities for each link. Two of them describe the link itself, and the other two describe the link's connection to a neighbouring link. For a revolute joint, θ_i is called the joint variable, and other three quantities would be fixed link parameters. For a sliding joint, d_i is the joint variable and the other three quantities are fixed link parameters.

5.3.2 Coordinate Frame Assignment

In order to describe the location of each link relative to its neighbours, a frame is attached to each link. The frames of active links are named by number according to the link to which they are attached. That is, frame i is attached rigidly to link i. The frames on the links are located as follows: The frame of a passive link is numbered as the previous frame with a single quotation mark as shown in Figure 5.4. The Z-axis of frame {i}, called Z_i, is coincident with the joint axis i. The origin of the frame {i} is located where a_i intersects the joint i axis. X_i points along a_i in the direction from joint i to joint i+1. Y_i is formed by the right-hand rule to complete the ith frame. Figure 5.4 shows the initial locations (when A=0 and B=0) of frames assigned to each link. The values of L_i are constants, L_0 (= 682.4 mm) is the initial distance between joint 1 and joint 2. L_1 (= 511 mm) is the initial distance from joint 4 to the rotation axis of the main body. L_2 is the length of the main arm, and equals to 700 mm.

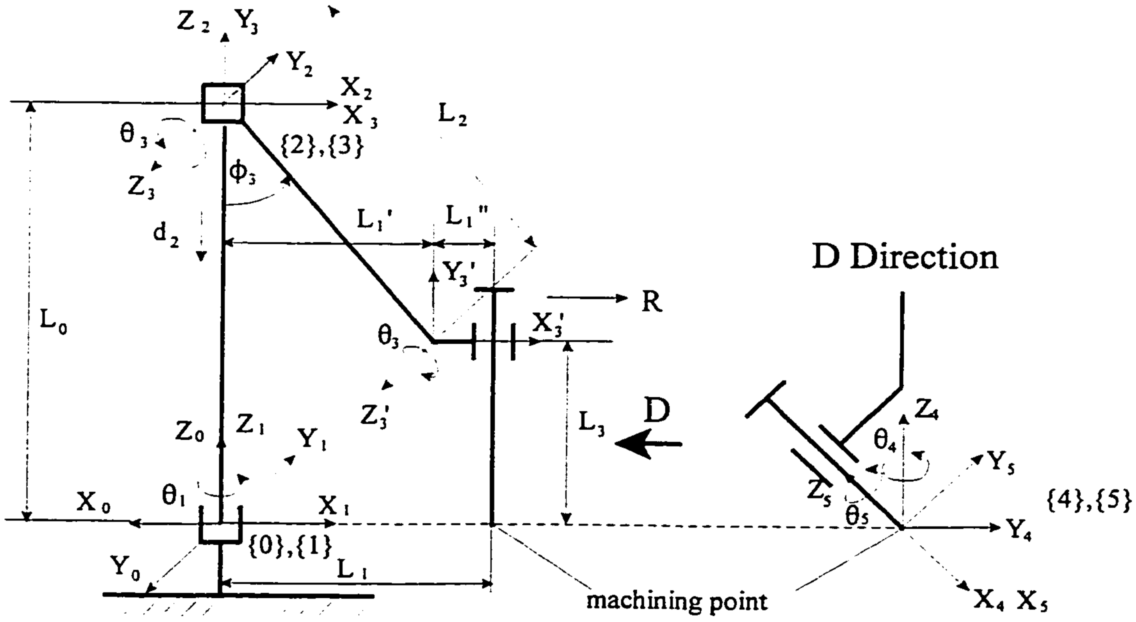


Figure 5.3 Frames Assigned to Each Link of the Robot

5.3.3 Derivation of Link Transformations

The transformation methodology developed by Paul[27] is used in this thesis. By defining a frame for each link, the kinematic problem has been divided into 5 subproblems. In order to solve each of these subproblems, namely ${}^{i-1}T_i$ transformation, the problem will further be divided into four sub-subproblems. Each of these four transformations will be a function of one parameter of a link only. Hence intermediate frame {P}, {Q}, and {R} are defined for each link. Figure 5.5 shows the same pair of joints as before with frames {P}, {Q}, and {R} defined. Frame {R} differs from {i-1} only by a rotation of α_{i-1}. Frame {Q} differs from {R} by a translation a_{i-1}. Frame {P} differs from {Q} by a rotation θ_i, and frame {i} differs from {P} by a translation d_i. The transformation which transforms vectors defined in {i} to their description in {i-1} can be written as

$$ {}^{i-1}\mathbf{P} = {}^{i-1}T_R \, {}^{R}T_Q \, {}^{Q}T_P \, {}^{P}T_i \, {}^{i}\mathbf{P} \qquad (5.1)$$

or $$ {}^{i-1}\mathbf{P} = {}^{i-1}T_i \, {}^{i}\mathbf{P} \qquad (5.2)$$

where $$ {}^{i-1}T_i = {}^{i-1}T_R \, {}^{R}T_Q \, {}^{Q}T_P \, {}^{P}T_i \qquad (5.3)$$

$$ = R_X(\alpha_{i-1}) \, D_X(a_{i-1}) \, R_Z(\theta_i) \, D_Z(d_i) \qquad (5.4)$$

where $R_X(\alpha_{i-1})$ stands for a rotation about X-axis of frame {i-1} by an angle α_{i-1}.

$D_X(a_{i-1})$ stands for a translation along X-axis of frame {R} by a distance a_{i-1}.

$R_Z(\theta_i)$ stands for a rotation about Z-axis of frame {Q} by an angle θ_i.

$D_X(d_i)$ stands for a translation along X-axis of frame {P} by a distance d_i.

Multiplying the rotational and translation transformation of (5.4), the general form of ${}^{i-1}T_i$ is obtained as

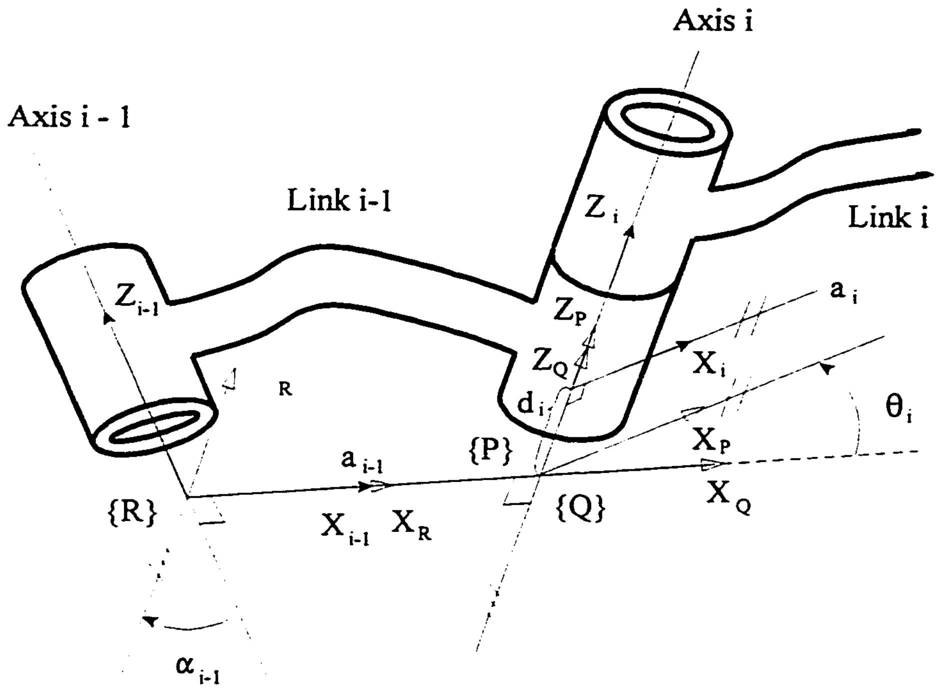


Figure 5.4 Location of Intermediate Frames {P}, {Q} and {R}[27]

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1}d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.5\ ) $$

5.3.4 Link Parameters of the Yamaha Zeta-1 Robot

Figure 5.3 is a schematic representation of the Yamaha Zeta-1 robot. Frames are assigned to each link. Frame {0}, the reference frame, is fixed to the base. It is positioned as shown in Figure 5.3 with Z_0 aligned with the joint 1 axis. For Yamaha robot, the ready position is regard as the initial position, when the X axis of frame {1} is aligned with negative X axis of frame {0}. Joint 2 is a prismatic joint, and its Z axis is assigned to be along the joint axis, which is parallel to Z_1. The joint variable is d_2, but the distance between X_1 and X_2 is $(L_0 - d_2)$. The frame {2} is always parallel to the frame {1}. Joint 3 is an active rotational joint, and its Z axis is along the joint axis, which is perpendicular to Z_2. X_3 is assigned to point along the common perpendicular of the axes X_3 and X_4, and from the axis of the joint 3 to the axis of the joint 4. Joint 3' is a passive rotational joint, and it only effects the orientation of the end-effector. The frame {3'} is aligned with the frame {3}. The frames {4}is located at the machining point of the robot, which is the intersecting point of the axes of joints 4 and 5. X_4 is assigned to be normal to the plane containing the axes of joints 4 and 5 as shown in Figure 5.3. For the last frame {5}, the direction of X_5 is chosen such that it aligns with X_4 when $\theta_5 = 0.0$, and the origin of frame {5} is chosen so that the distance between X_4 and X_5 is zero. The corresponding link parameters of the Yamaha robot are shown in Table 5.1.

Table 5.1 Link Parameters

Link i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	$\theta_1+180°$
2	0	0	$L_0 - d_2$	0
3	90°	0	0	θ_3
3'	0	0	0	$-\theta_3$
4	-90°	$L_1 + R$	L_0	θ_4
5	45°	0	0	θ_5

Substituting the link parameters shown in Table 5.1 for the Yamaha robot in equation (5.5), the individual transformations for each link are obtained as given below:

$$^{0}T_{1} = \begin{bmatrix} \cos(\theta_1+180°) & -\sin(\theta_1+180°) & 0 & 0 \\ \sin(\theta_1+180°) & \cos(\theta_1+180°) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & -\cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}T_{2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_0-d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}^{2}T_{3} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{2}T_{3} = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3'}T_{4} = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & L_1+\Delta R \\ 0 & 0 & 1 & 0 \\ -\sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{2}T_{3} = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ \frac{\sqrt{2}}{2}\sin\theta_5 & \frac{\sqrt{2}}{2}\cos\theta_5 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2}\sin\theta_5 & \frac{\sqrt{2}}{2}\cos\theta_5 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

where $\Delta R = L_2 \sin(\phi_3 + \theta_3) - L_2 \sin\phi_3 = L_2 \sin(\phi_3 + \theta_3) - L_1$

These link transformations can be multiplied together to obtain a single transformation that relates the frame {5} to the base frame {0}:

$$ {}^{0}T_{5} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\,{}^{3}T_{3'}\,{}^{3'}T_{4}\,{}^{4}T_{5} \tag{5.6} $$

$$ {}^{0}T_{5} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.7} $$

where

$$ r_{11} = -\cos(\theta_1 + \theta_4)\cos\theta_5 + \frac{\sqrt{2}}{2}\sin(\theta_1 + \theta_4)\sin\theta_5 $$

$$ r_{12} = \cos(\theta_1 + \theta_4)\sin\theta_5 + \frac{\sqrt{2}}{2}\sin(\theta_1 + \theta_4)\cos\theta_5 $$

$r_{13} = - \frac{\sqrt{2}}{2} \sin(\theta_1 + \theta_4)$

$r_{14} = - (L_1 + \Delta R) \cos \theta_1$

$r_{21} = - \sin(\theta_1 + \theta_4) \cos \theta_5 - \frac{\sqrt{2}}{2} \cos(\theta_1 + \theta_4) \sin \theta_5$

$r_{22} = \sin(\theta_1 + \theta_4) \sin \theta_5 - \frac{\sqrt{2}}{2} \cos(\theta_1 + \theta_4) \cos \theta_5$

$r_{23} = \frac{\sqrt{2}}{2} \cos(\theta_1 + \theta_4)$

$r_{24} = - (L_1 + \Delta R) \sin \theta_1$

$r_{31} = \frac{\sqrt{2}}{2} \sin \theta_5$

$r_{32} = \frac{\sqrt{2}}{2} \cos \theta_5$

$r_{33} = \frac{\sqrt{2}}{2}$

$r_{34} = d_2$

5.3.5 Transformation between Component Space and Robot Joint Space

A frame {t} is used to represent a frame fixed on the tool, which is located such that the origin of the frame is coincident with the machining point, and the Z axis is along the length of the tool as shown in Figure 5.5. The transformation from the frame {t} to the frame {5} is:

$$^5T_t = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -\cos 45^\circ & \sin 45^\circ & 0 \\ 0 & \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

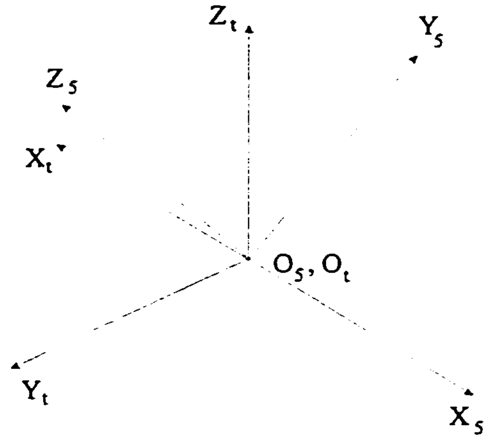
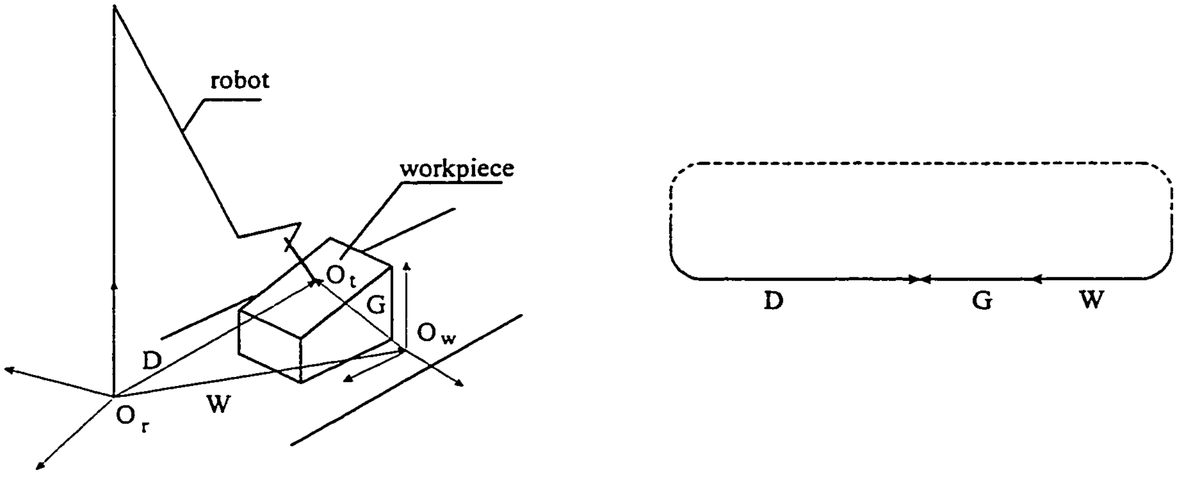


Figure 5.5 The Relationship Between Two Frames {5} and {t}



(a)

(b)

Figure 5.6 Relative Transformations in the Yamaha Zeta-1 Robot Work Space

The relative transformations in Yamaha robot work space is shown in Figure 5.6. The transformation between the base frame {0 } of the robot and the tool frame { t } is:

$$D = {}^{0}T_{t} = {}^{0}T_{5}\,{}^{5}T_{t} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & X_r \\ e_{21} & e_{22} & e_{23} & Y_r \\ e_{31} & e_{32} & e_{33} & Z_r \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.7)$$

where $e_{11} = \cos(\theta_1 + \theta_4)\cos\theta_5 - \frac{\sqrt{2}}{2}\sin(\theta_1 + \theta_4)\sin\theta_5$

$e_{12} = -\frac{\sqrt{2}}{2}\cos(\theta_1 + \theta_4)\sin\theta_5 - \frac{1}{2}\sin(\theta_1 + \theta_4)\cos\theta_5 - \frac{1}{2}\sin(\theta_1 + \theta_4)$

$e_{13} = \frac{\sqrt{2}}{2}\cos(\theta_1 + \theta_4)\sin\theta_5 + \frac{1}{2}\sin(\theta_1 + \theta_4)\cos\theta_5 - \frac{1}{2}\sin(\theta_1 + \theta_4)$

$X_r = -(L_1 + \Delta R)\cos\theta_1$

$e_{21} = \sin(\theta_1 + \theta_4)\cos\theta_5 + \frac{\sqrt{2}}{2}\cos(\theta_1 + \theta_4)\sin\theta_5$

$e_{22} = -\frac{\sqrt{2}}{2}\sin(\theta_1 + \theta_4)\sin\theta_5 + \frac{1}{2}\cos(\theta_1 + \theta_4)\cos\theta_5 + \frac{1}{2}\cos(\theta_1 + \theta_4)$

$e_{23} = \frac{\sqrt{2}}{2}\sin(\theta_1 + \theta_4)\sin\theta_5 - \frac{1}{2}\cos(\theta_1 + \theta_4)\cos\theta_5 + \frac{1}{2}\cos(\theta_1 + \theta_4)$

$Y_r = -(L_1 + \Delta R)\sin\theta_1$

$e_{31} = -\frac{\sqrt{2}}{2}\sin\theta_5$

$e_{32} = \frac{1}{2}\cos\theta_5 + \frac{1}{2}$

$e_{33} = \frac{1}{2}\cos\theta_5 + \frac{1}{2}$

$Z_z = d_2$

Figure 5.6 shows relative transformations in the Yamaha Zeta-1 robot work space, where T represents the transformation of the tool coordinate system with respect to the base coordinate system of the robot. W represents the transformation of workpiece coordinate system with respect to the base coordinate system of the robot, and is known. The frame {W} is initially coincident with the base frame of the robot, then it was translated to the point {F_x, F_y, F_z }, and rotated about Z axis by ϕ degrees.

$$W = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & F_x \\ \sin\phi & \cos\phi & 0 & F_y \\ 0 & 0 & 1 & F_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.8)$$

G represents the transformation of the coordinate system fixed on the machining point with respect to the workpiece coordinate system. It is known based on the geometric data of the machining edge in the workpiece coordinate system and is obtained from CAD data file.

$$G = \begin{bmatrix} t_x & m_x & b_x & X_w \\ t_y & m_y & b_y & Y_w \\ t_z & m_z & b_z & Z_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.9)$$

where t_x, t_y, and t_z are the projections of unit tangent vector of a curve m_x, m_y, and m_z are the projections of the main normal vector of a curve b_x, b_y, and b_z are the projections of binormal vector of a curve, which is the same as the components (nx_w, ny_w, nz_w) of the normal of the edge used to guide a tool, i.e.,

$$\begin{aligned} nx_w &= b_x \\ ny_w &= b_y \\ nz_w &= b_z \end{aligned} \tag{5.10}$$

During machining, the axis of the cutter should be parallel to the normal of the machining surface. This results in the following transformation:

$$D = W\,G$$

From (5.7)

$$D = {}^0T_t = {}^0T_5\,{}^5T_t = \begin{bmatrix} e_{11} & e_{12} & e_{13} & X_r \\ e_{21} & e_{22} & e_{23} & Y_r \\ e_{31} & e_{32} & e_{33} & Z_r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$WG = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & F_x \\ \sin\phi & \cos\phi & 0 & F_y \\ 0 & 0 & 1 & F_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x & m_x & b_x & X_w \\ t_y & m_y & b_y & Y_w \\ t_z & m_z & b_z & Z_w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} e_{11} & e_{12} & e_{13} & X_r \\ e_{21} & e_{22} & e_{23} & Y_r \\ e_{31} & e_{32} & e_{33} & Z_r \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.11}$$

where $e_{11} = t_x \cos \phi - t_y \sin \phi$ (5.12)

$e_{21} = t_x \sin \phi + t_y \cos \phi$ (5.13)

$e_{31} = t_z$ (5.14)

$e_{12} = m_x \cos \phi - m_y \sin \phi$ (5.15)

$e_{22} = m_x \sin \phi + m_y \cos \phi$ (5.16)

$e_{32} = m_z$ (5.17)

$e_{13} = nx_r = b_x \cos \phi - b_y \sin \phi = nx_w \cos \phi - ny_w \sin \phi$ (5.18)

$e_{23} = ny_r = b_x \sin \phi + b_y \cos \phi = nx_w \sin \phi + ny_w \cos \phi$ (5.19)

$e_{33} = nz_r = b_z = nz_w$ (5.20)

$X_r = X_w \cos \phi - Y_w \sin \phi + F_x$ (5.21)

$Y_r = X_w \sin \phi + Y_w \sin \phi + F_y$ (5.22)

$Z_r = Z_w + F_z$ (5.23)

By equating the corresponding elements of equations (5.7) and (5.11), we obtain

$$-(L_1 + \Delta R) \cos\theta_1 = X_r$$

$$-(L_1 + \Delta R) \sin\theta_1 = Y_r$$

$$d_2 = Z_r$$

$$\frac{1}{2} \cos\theta_5 + \frac{1}{2} = e_{33}$$

$$\frac{\sqrt{2}}{2}\cos(\theta_1 + \theta_4) \sin\theta_5 - \frac{1}{2} \sin(\theta_1 + \theta_4)(1 - \cos\theta_5) = e_{13}$$

$$\frac{\sqrt{2}}{2}\sin(\theta_1 + \theta_4) \sin\theta_5 + \frac{1}{2} \cos(\theta_1 + \theta_4)(1 - \cos\theta_5) = e_{23}$$

After rearranging, the following expressions result:

$$180° + \theta_1 = \tan^{-1} \frac{Y_r}{X_r} \tag{5.24}$$

$$d_2 = Z_r \tag{5.25}$$

$$\theta_5 = \cos^{-1} (2\, e_{33} - 1) = \cos^{-1} (2\, nz_r - 1) \tag{5.26}$$

$$\theta_4 = \tan^{-1} [\frac{\sqrt{2}e_{23}\sin\theta_5 + e_{13}\cos\theta_5 - e_{13}}{\sqrt{2}e_{13}\sin\theta_5 + e_{23} - e_{23}\cos\theta_5}] - \theta_1$$

$$= \tan^{-1} [\frac{ny_r}{nx_r}] - \tan^{-1} [\frac{1 - \cos\theta_5}{\sqrt{2}\sin\theta_5}] - \theta_1 \tag{5.27}$$

$$R = \sqrt{X_r^2 + Y_r^2} \tag{5.28}$$

The position coordinates of the machining point of the robot is controlled by manipulating the joint parameters θ_1 and d_2. The orientation of the cutter is modified whenever θ_4 and θ_5 are altered. It should be noted here that the variable of passive joint 3', θ_3', is always equal and opposite to joint 3 variable, θ_3. Hence, they cancel each other in the calculation.

For the Yamaha Zeta-1 robot, $\theta_1 = \theta$; $d_2 = Z$; $\theta_4 = \alpha$; $\theta_5 = \beta$. Substituting the values of θ_1, d_2, θ_4, θ_5 by θ, Z, α, β, respectively, we get

$$\theta = 180° + \theta_1 = \tan^{-1} \frac{Y_r}{X_r} \tag{5.29}$$

$$Z = Z_r \tag{5.30}$$

$$\alpha = \tan^{-1} (\frac{e_{23}}{e_{13}}) - \tan^{-1} (\frac{1 - \cos\beta}{\sqrt{2}\sin\beta}) - \theta_1$$

$$= \tan^{-1}\left(\frac{ny_r}{nx_r}\right) - \tan^{-1}\left(\frac{1-\cos\beta}{\sqrt{2}\sin\beta}\right) - \theta_1 \qquad (5.31)$$

$$\beta = \cos^{-1}(2\,e_{33} - 1) = \cos^{-1}(2\,nz_r - 1) \qquad (5.32)$$

5.4 Relationship Between Joint Space and Cartesian Space

For control purpose, X_r, Y_r, Z_r, A, B are used instead of θ, R, Z, α, β, i.e., the position of the tool tip is represented by coordinates of machining point in Cartesian space, and the orientation of the tool is represented by two angles A and B. According the definitions, A is the tool turning angle, and B is the tool collapsing angle. If B is zero, the tumbling surface passed through the tool and tangent to the cone surface which is the trace of the tool when it rotates about β axis, i.e. the angle between X and $X_{L'}$, as shown in Figure 5.7. If B has a certain value, the tumbling surface is a plane which passes through the tool and α axis. When the tool is in the state where the tumbling surface is machined with the X axis, A is equal to zero, and the initial value of A is 0°. When the tool turns counterclockwise when viewed from above, the value of A is positive. B is a tool collapse angle, which is measured in the tumbling surface. When the tool is in the state where it is vertical, the value of B is zero, and the tool collapses along the plus direction of β axis, the value of B is positive. Hence, the value A is affected by the joint θ, α and β, and the value B is only effected by the joint β.

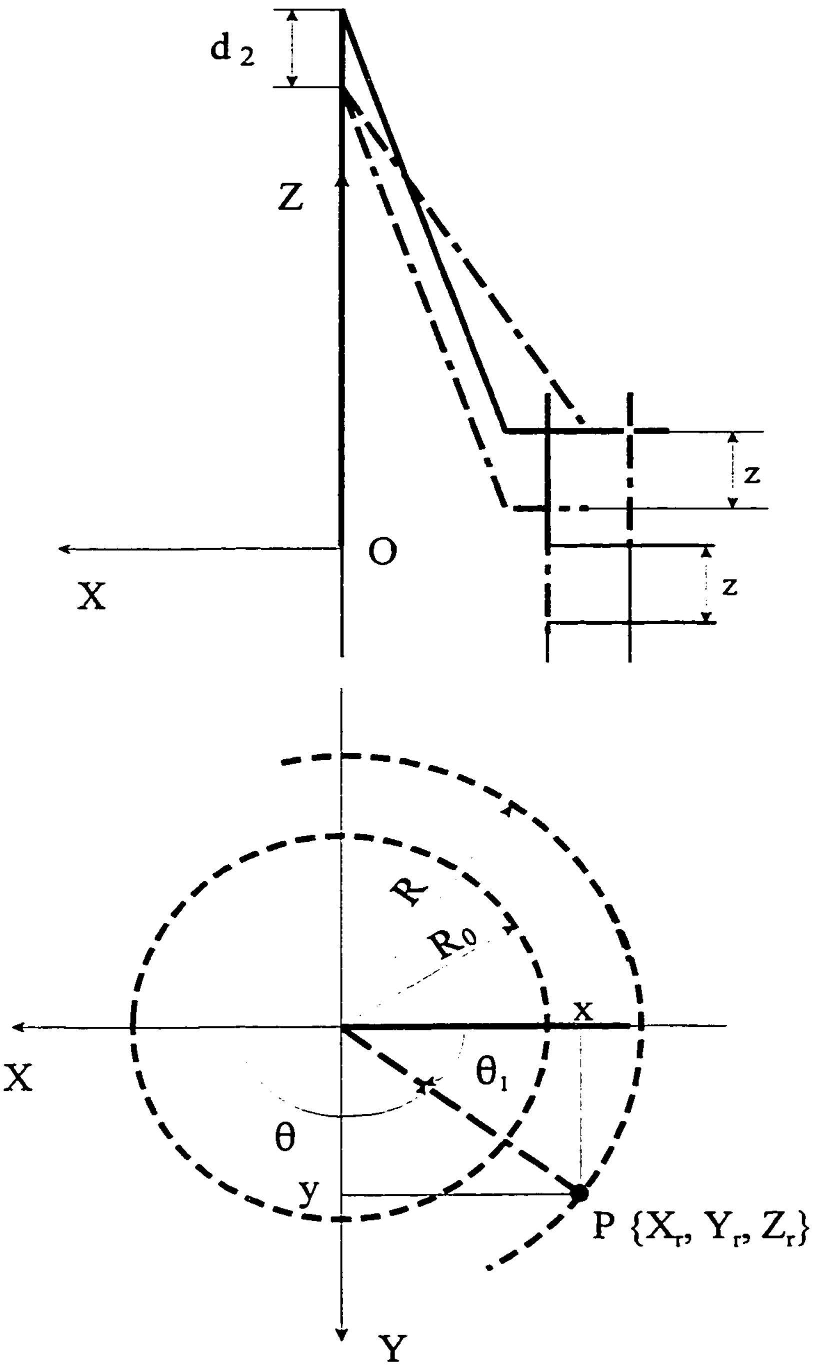


Figure 5.7 The Relationship between X_r, Y_r, Z_r and θ, R, Z

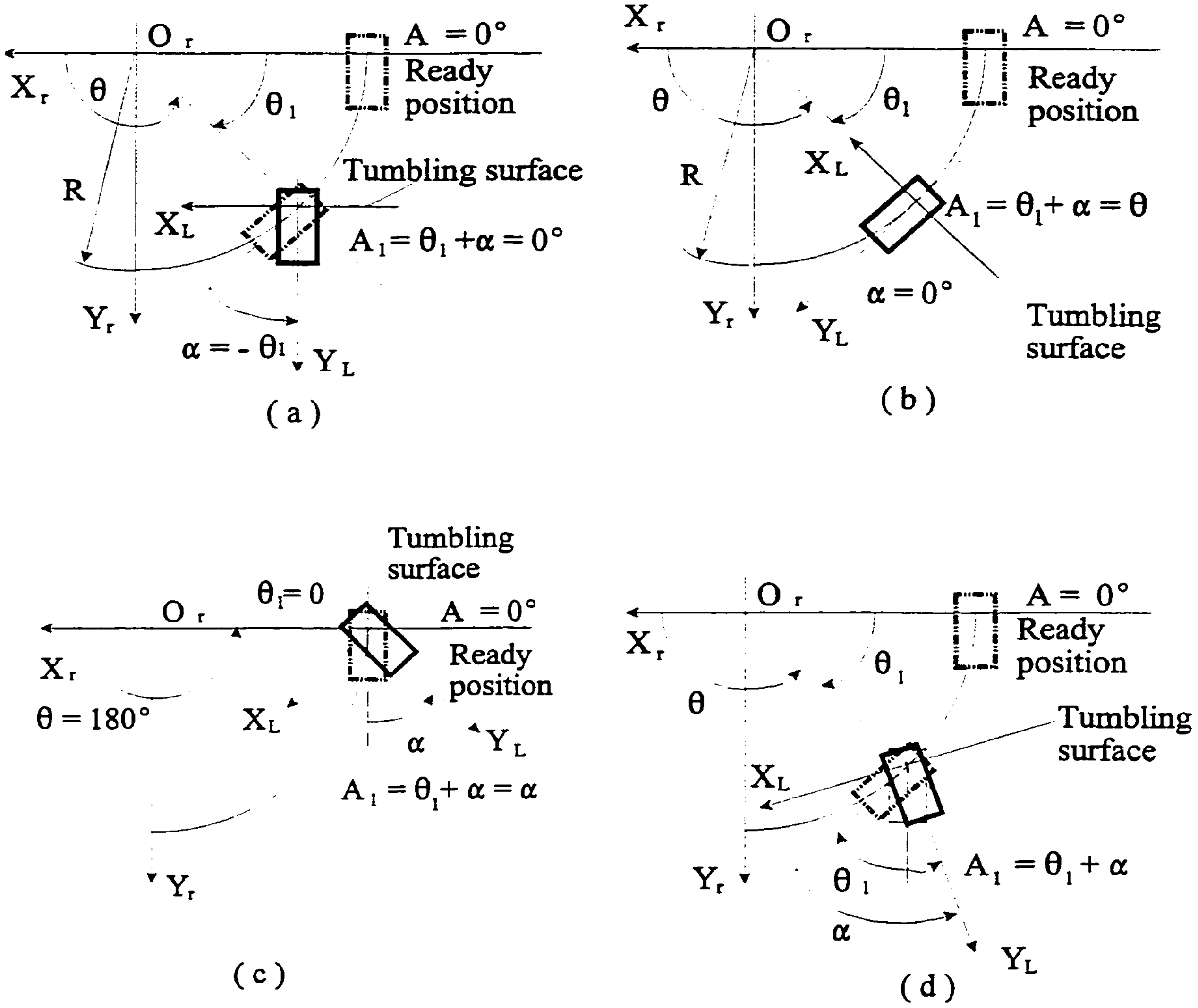


Figure 5.8 Relationship Between A and $\{\theta_1, \alpha\}$

The relationship between X_r, Y_r, Z_r and θ, R, Z is clearly shown in Figure 5.7, where $\theta_1 = \theta - 180°$, R_0 is the initial value of R.

$$X_r = R\cos(180° + \theta_1) = R\cos\theta \tag{5.33}$$

$$Y_r = R\cos(180° + \theta_1) = R\sin\theta \tag{5.34}$$

$$Z_r = Z \tag{5.35}$$

Figure 5.8 shows that when $\beta = 0$, $A_1 = \theta_1 + \alpha = \theta + \alpha - 180°$, where A_1 represents the value of A caused by the joints θ and α..

The contribution of β on A is not straightforward, and is derived below. Assume $\theta_1 = 0$, and $\alpha = 0$, and the tool collapse along β axis by $+\beta$ degrees, as shown in Figure 5.9.

In Figure 5.9, let A_2 represents the change of A caused by β, and A_2 is the angle between the tumbling surface and positive X_r axis.

$$A_2 = \angle M_1''M_2''M_3'' = |\tan^{-1}[\frac{M_1''M_3''}{M_2''M_3''}]| = |\tan^{-1}[\frac{1-\cos\beta}{\sqrt{2}\sin\beta}]|$$

if $\beta > 0$, $A_2 = -|\tan^{-1}\frac{1-\cos\beta}{\sqrt{2}\sin\beta}|$, if $\beta < 0$, $A_2 = |\tan^{-1}\frac{1-\cos\beta}{\sqrt{2}\sin\beta}|$

$$A_2 = -\tan^{-1}\frac{1-\cos\beta}{\sqrt{2}\sin\beta}$$

The total effect of θ, α and β on A is:

$$\begin{aligned} A &= A_1 + A_2 = \theta_1 + \alpha - \tan^{-1}[\frac{1-\cos\beta}{\sqrt{2}\sin\beta}] \\ &= \tan^{-1}(\frac{ny_r}{nx_r}) \end{aligned} \tag{5.36}$$

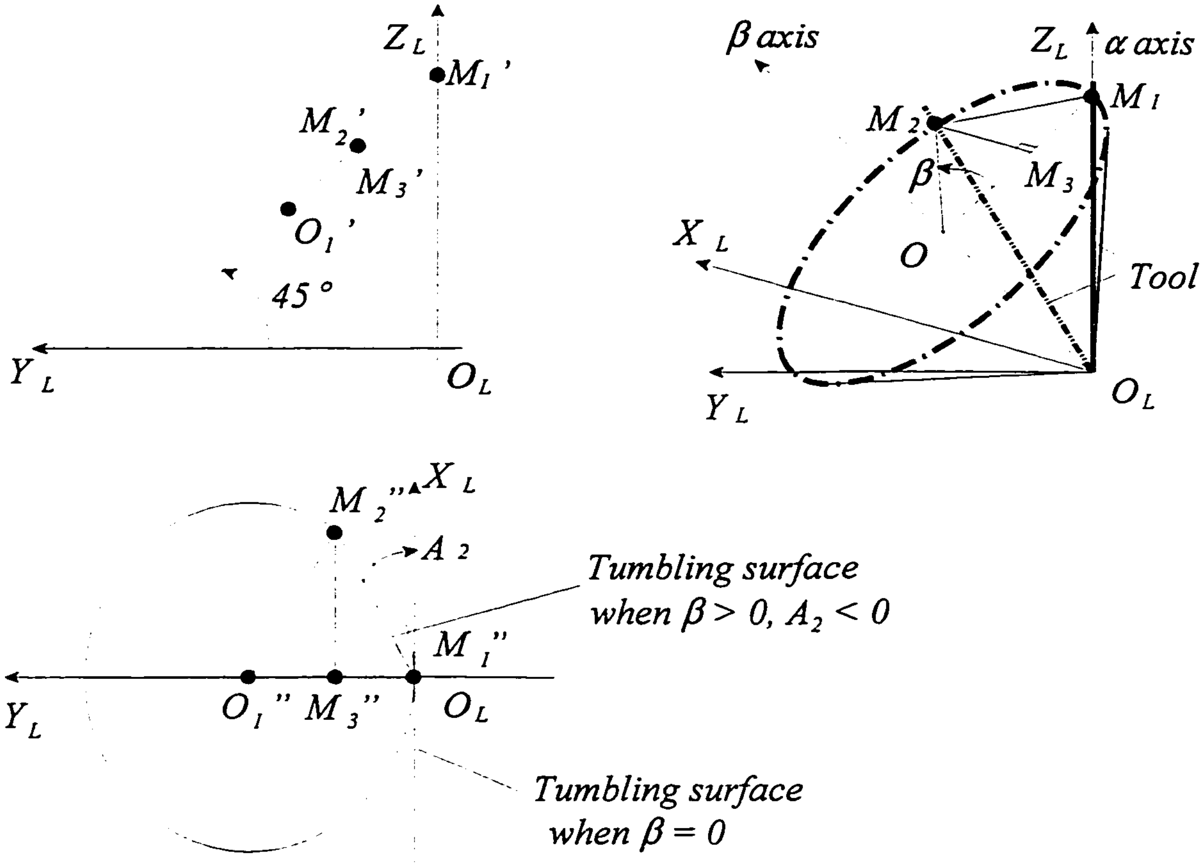


Figure 5.9(a) Relationship Between A_2 and β (β > 0)

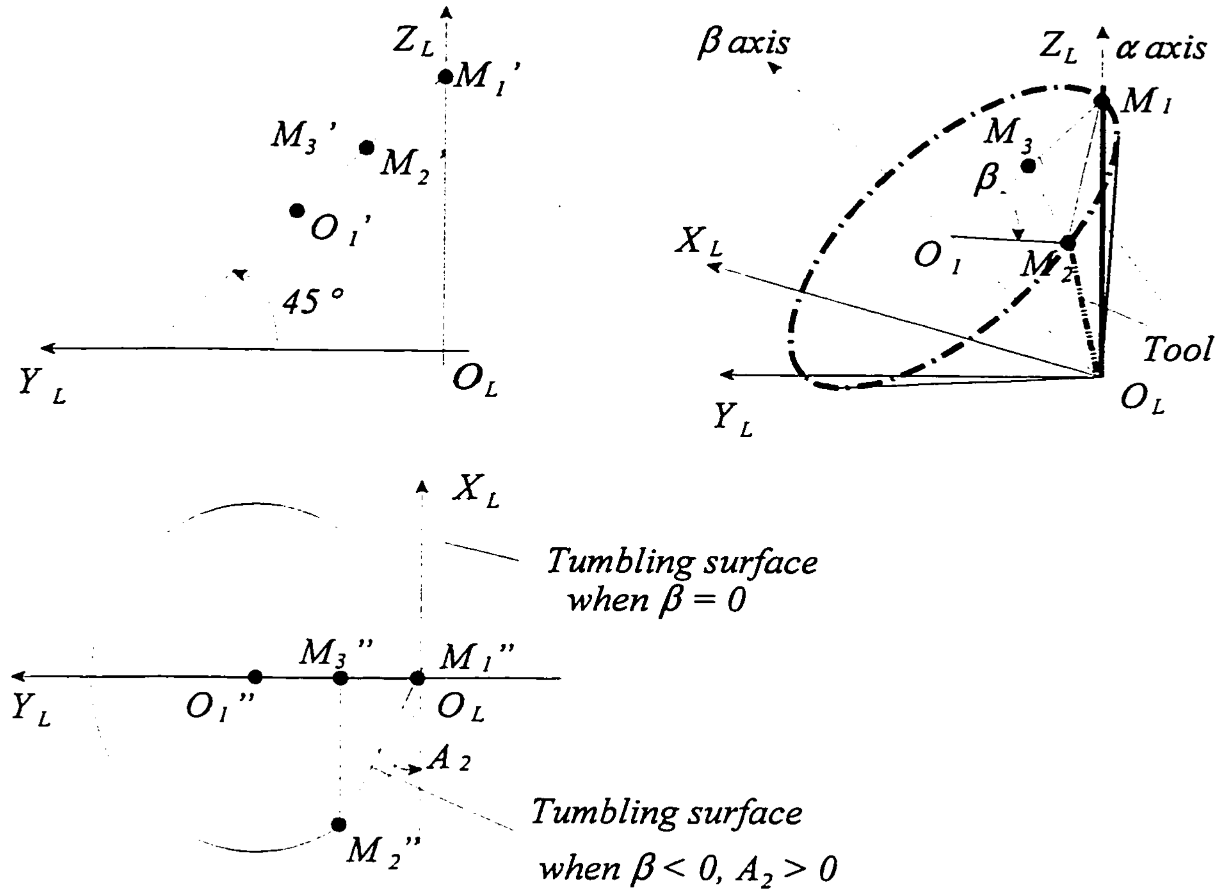


Figure 5.9 (b) Relationship Between A_2 and β ($\beta < 0$)

Only β affects B. B is the angle between the tool and the vertical axis, as shown in Figure 5.10. According to the design values, the length of O_1O_L and the length of O_1M_1 are same, so $O_LM_1 = O_LM_2 = \sqrt{2}$ R. In triangle $O_1M_1M_2$,

$$M_1M_2^2 = R^2 + R^2 - 2R^2 \cos \beta.$$

In triangle $O_LM_1M_2$,

$$M_1M_2^2 = (\sqrt{2}\,R)^2 + (\sqrt{2}\,R)^2 - 2(\sqrt{2}\,R)^2 \cos B$$

so, $$\cos B = \frac{1 + \cos\beta}{2}$$

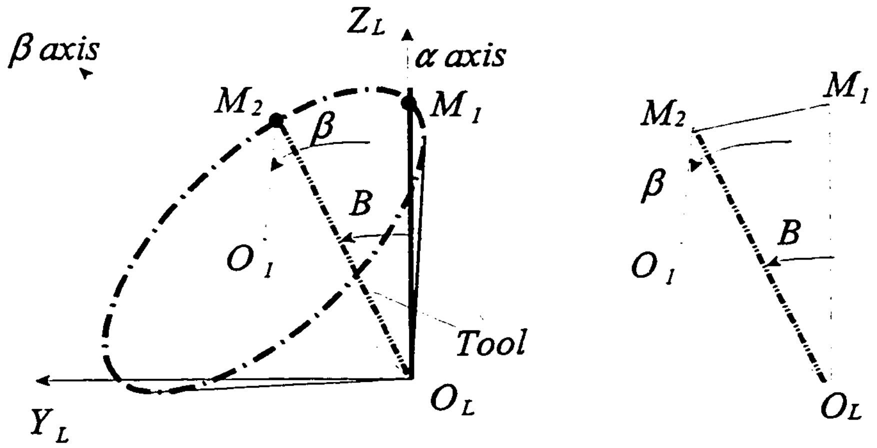


Figure 5.10 The Relationship between Angles B and β

B always has the same sign as β.

$$B = \cos \frac{1+\cos\beta}{2} \qquad \text{If } \beta > 0 \qquad (5.37a)$$

$$B = -\cos \frac{1+\cos\beta}{2} \qquad \text{If } \beta < 0 \qquad (5.37b)$$

5.5 Discussion

In this Chapter, the algebraic approach is used to derive the transformation equations between the workpiece coordinate system and the robot joint space. Appendix B introduce a geometric approach, and it is can be seen that the results from two different methods are the same as

$$R = \sqrt{X_r^2 + Y_r^2}$$

$$\theta = \tan^{-1} \frac{Y_r}{X_r}$$

$$Z = Z_r$$

$$\alpha = \tan^{-1}\left(\frac{ny_r}{nx_r}\right) - \tan^{-1}\left(\frac{1-\cos\beta}{\sqrt{2}\sin\beta}\right) - \theta$$

$$\beta = \cos^{-1}(2\, nz_r - 1)$$

In the interface, the tool path is calculated first in the workpiece coordinate system, and then it is transferred to the robot coordinate system. The transformation equations are:

$$X_r = X_w \cos\phi - Yw \sin\phi + T_x$$

$$Y_r = X_w \sin\phi + Yw \cos\phi + T_y$$

$$Z_r = Z_w + T_z$$

$$A = \tan^{-1}\left(\frac{ny_r}{nx_r}\right)$$

$$B = \cos^{-1} \frac{1 + \cos\beta}{2} \qquad \text{if } \beta > 0$$

$$B = -\cos^{-1} \frac{1 + \cos\beta}{2} \qquad \text{if } \beta < 0$$

The transformation equations derived above are verified using an experimental setup, which is represent in Chapter 7. The transformation part of the interface software makes use of these transformation equations.

CHAPTER 6 IMPLEMENTATION OF THE INTERFACE

6.1 Introduction

An interface is designed to link the CAD output packages and the robot control workcell, to process geometric data from CAD ASCII output file (DXF and ACIS data files), and generate a script data file which can be directly used by the robot control workcell. The interface is implemented in C.

The layout of the interface is presented in Section 6.2. The organization of geometric data extracted from AutoCAD file is discussed in Section 6.3, and the usage of these geometric data is detailed in Section 6.4. Section 6.5 discusses the generation of tool paths in workpiece coordinate system and transformation of tool paths between the control axes X, Y, Z, A, B and the general representation X, Y, Z, nx, ny, nz. It also concerns the problems existing in the interface software.

6.2 The Layout of the Interface

A workpiece consists of different entities, such as, points, straight lines, ellipse curves, planes, cone surfaces, and intersection curves. In order to rebuild the geometric models, all necessary parameters about each entity need to be known. These necessary data are all contained in the CAD data file, so the data file should be read first.

The geometric data of each entity is represented in a data file by a key word and its relative parameters. In a DXF data file, the key word “ARC” represented a circular arc segment; “LINE” represents a straight line segment; In 12 or lower versions, the ellipse curves are represent as polylines, in version 13 the key word “ELLIPSE” is used. In ACIS data file (.sat), the key word “ellipse-curve" represents a circle (circular arc) or ellipse (elliptical arc) depending on the ratio of the minor radius to major radius. “straight-curve” represents a infinite straight line in space. “point” represents a point in space. “plane” represents a infinite plane in space. “intcurve-curve” represents a 3D intersecting curve in space. “torus” represents a torus in space.

The interface reads the whole data file, and when it meets a key word representing an entity which is the same as what we look for, the data of that entity is extracted and saved into a relative structure. Each kind of entities has a relative structure, for example, a structure used to save the coordinates of points is named point, a structure array variable. When the key word "point" is met, the coordinates of the point is extract, and saved into the point structure variable.

Any workpiece contains many surfaces, which are parallel or perpendicular to the principal plane of the coordinate system, or which are in general positions, i.e., not parallel and not perpendicular to any principal plane of the coordinate system. According to the orientations of surfaces, different layers are assigned for a workpiece such that at least one surface lies on a layer, and a curve lies on two layers with different orientations. The end points of each curve are defined, and curves on the same layer are searched and ordered for future use.

The user can input the information about which surface or edges need to be machined, and the interface will find the specified edges. According to the geometric parameters, the relative parametric equations, and the required tolerances, the interface calculates the coordinates and normals of points on the desired tool path in workpiece coordinate system. Based on the transformations derived in Chapter 5, the tool path is transferred from the workpiece coordinate system into the robot control coordinate system. Finally, a script file contains the values of X_r, Y_r, Z_r, A and B is produced, which can be directly used by the robot workcell.

The interface follows the following steps: read files, search end points of each entity and the connection between two adjacent entities, tool path generation, plot and simulation, and tool path transformations. Figure 6.1 is the flow chart of the interface software.

6.3 Geometric Data Processing

6.3.1 Definitions of Structure Variables

For two-dimension workpiece (data extracted from DXF files), there are five basic types of entities: line, arc, circle, polyline and polyarc. Relatively, there are five structure variables are used in the 2D part of the interface. They are defined as: (1) line, (2) circle, (3) arc, (4) polyarc, and (5) polyline.

There are seven structure variables used in the three-dimensional part of the interface (data extracted from ACIS file): point, line, plane, ellipse_curve, cone_surface, and intersection. Each structure contains certain information represented by various variables. The definitions are shown as following:

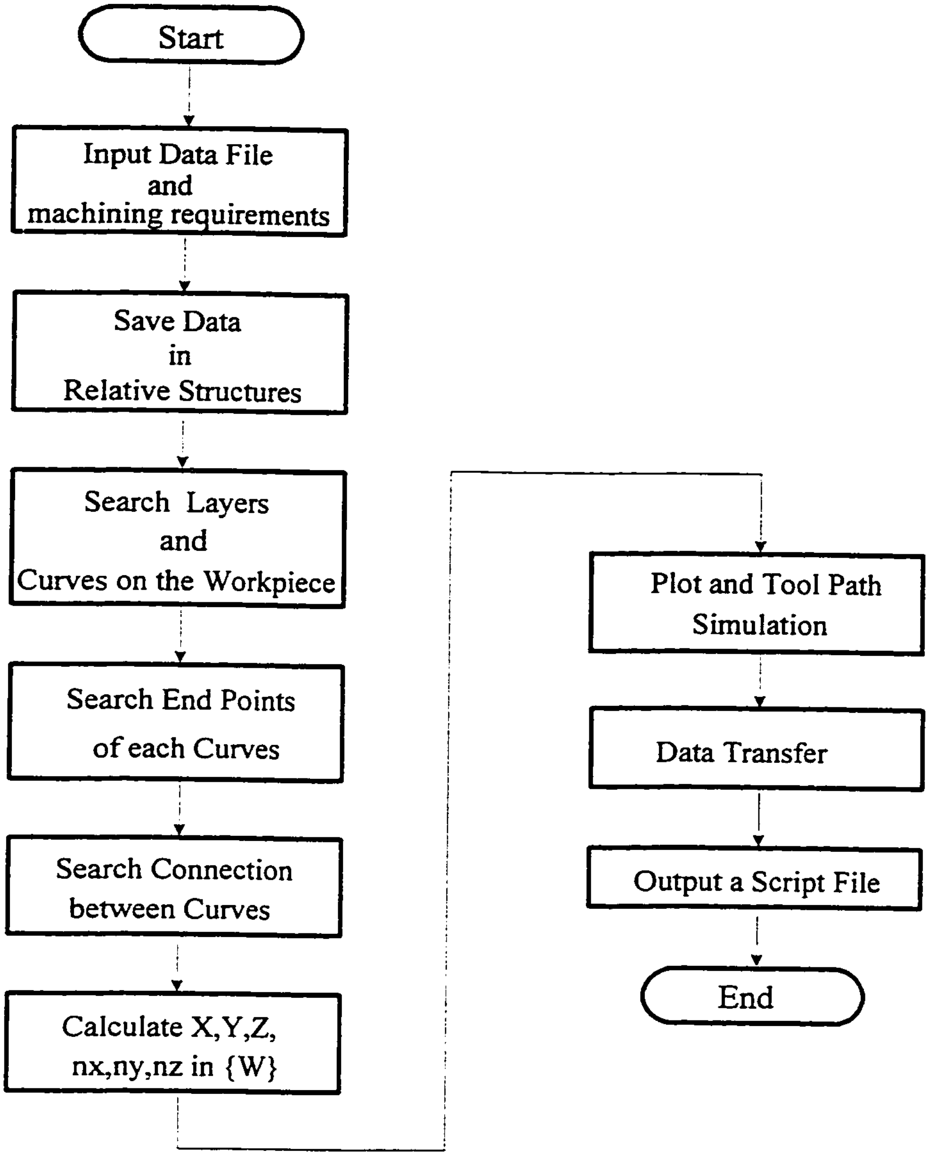

Start
Input Data File and machining requirements
Save Data in Relative Structures
Search Layers and Curves on the Workpiece
Search End Points of each Curves
Search Connection between Curves
Calculate X,Y,Z, nx,ny,nz in {W}
Plot and Tool Path Simulation
Data Transfer
Output a Script File
End

Figure 6.1 Flow Chart of the Interface Software

(6) structure point, (7) structure line, (8) structure plane, (9) structure ellipse_curve, (10) structure cone_surface, and (11) structure intersection

6.3.2 Determinations of the Location and the End Points of Each Curve

After the geometric data are extracted and saved in the relative structure, the locations of each edge on the workpiece need to be defined. As mentioned before, each surface of a workpiece lies on a layer. A layer may be parallel to XY plane, XZ plane, YZ plane or in a general position. A curve lies on two layers. If two intersecting surfaces are not planes, the intersection of them is a three-dimensional curve, so it cannot be simply defined in a layer, instead, the axes of two intersecting cylinders are defined to determine the location of the intersection curve. There may be several layers parallel to a principal plane of the workpiece coordinate system, so different values are assigned to each layer. For example, for a straight line, three flags are used to identify to which layer it belongs, if flagx = 1, flagy = 2, flagz = 0, the straight line lies on the intersection of two layers, one is parallel to YZ plane, the other is parallel to the XZ plane. Flagx = 1 means the layer parallel to the YZ plane has the maximum x values, and flagy = 2 means the layer parallel to XZ plane has the second maximum y value.

In DXF files, the data following a single key word represent a well defined curve segment, including the coordinates of the end points. By contrast, data saved in ACIS files are more complex. The data following a single key word only represent an infinite entity. In order to obtain well defined curve segments, the end points need to be defined according to other information containing in ACIS files. For example, an edge lies on a curve, and each edge has two vertices. Each vertex locates on a point. Based on these relationships, the end points of each curve can be determined.

6.6.3 Connection between Two Plan Curves

When the locations and end points of each curve are known, it is easy to search the connections between two curves on a layer. The intersection point of two curves is one of the end points of these two curves.

The connection search begins from the curve which is nearest to the ready position $\{x_{rp}, y_{rp}, z_{rp}\}$. In order to find such nearest curve, the nearest points on each curve from the ready position are searched. Assume the z value of the ready point is the same as the z value of the layer on which the curve lies, i.e., use the projection of the ready position (defined in Section 4.2) on the layer on which the curve lies. For a straight line, such nearest point is the intersection point of the known line and a line which is perpendicular to the known line and passes through the ready position. If $y = k_i x + b_i$ represents the known straight line, where $k_i = \frac{q_i}{p_i}$, $b_i = \frac{q_i}{p_i} x_{mi} + y_{mi}$, and i represents the ith straight line, $y = k_{rp} x + b_{rp}$ represents a line vertical to the ith line, where $k_{rp} = -\frac{1}{k_i}$, $b_{rp} = y_{rp} - k_{rp} x_{rp}$. The intersection point is:

$$x = \frac{b_{rp} - b_i}{k_i - k_{rp}}; \qquad y = \frac{k_i b_{rp} - k_{rp} b_i}{k_i - k_{rp}}$$

For a circle, $(x - x_c)^2 + (y - y_c)^2 = R^2$, and a line passes the center point of the circle and the ready position is $y = k_{rp} x + b_{rp}$, where $k_{rp} = \frac{y_{rp} - y_c}{x_{rp} - x_c}$, and $b_{rp} = \frac{x_{rp} y_c - x_c y_{rp}}{x_{rp} - x_c}$. The intersection point $\{x_{int}, y_{int}, z_{int}\}$ of them is obtained by solving these two equations.

The calculated intersection point should be checked if it lies within the range of the straight line segment. If the calculated intersection point does not in the range of the line, it cannot be used as the nearest point, one of two end points which is closer to the ready point will be chosen as the nearest point.

The nearest distance of a curve from the ready position is the distance between the ready position and the relative nearest point:

$$D = \sqrt{(x_n - x_{rp})^2 + (y_n - y_{rp})^2} \quad (6.1)$$

Comparing the nearest distances of all curve on the same layer, and chosen a curve with minimum D value as the first curve. During searching the connection, an order number is assigned to each curve starting from 1 to the first curve.

6.4 Step Size Determination

The machining tolerance specified by the user is used to compute the step size to define a series of machining points (locations) and interpolate between them. For a straight line, the parametric form is:

$$x = x_m + p\,t$$

$$y = y_m + q\,t$$

$$z = z_m + r\,t \quad (6.2)$$

The step size of t does not affect the machining precision because the machining points are all located on the line, and the paths through these points is coincident with the line.

For a circular arc, the parametric form is:

$x = x_c + R \cos u$

$y = y_c + R \sin u$

$z = z_c$ if the circular arc plane is parallel to the XY plane

$z = z_c + R \cos u$

$x = x_c + R \sin u$

$y = y_c$ if the circular arc plane is parallel to the XZ plane

$y = y_c + R \cos u$

$z = z_c + R \sin u$

$x = x_c$ if the circular arc plane is parallel to the YZ plane

(6.3)

These equations can be used to generate points on the circular arc by incrementing u from u_s to u_e. These points are in turn connected with line segments to represent the circular arc. The step size of u, Δu, affect the machining precision, the error is the distance between points M_m and M_m' as shown in the Figure 6.2.

$$\begin{aligned} \text{Error} &= |M_m M_m'| \\ &= R - |OM_m'| \\ &= R - R \cos(0.5\, \Delta u) \end{aligned}$$

For a circular arc, only unilateral tolerance exists, i.e., for a circular arc with bulge = 1, only INTOL exists, and for a circular arc with bulge = 0, only OUTTOL exists. The error should be smaller than the unilateral tolerance which has smaller value.

$$\text{Error} \leq \text{TOL} \tag{6.4}$$

$$R - R\cos(0.5\,\Delta u) < \text{TOL}$$

$$\Delta u \leq 2\arccos\left(1 - \frac{TOL}{R}\right) \tag{6.5}$$

where TOL is the smaller of INTOL and OUTTOL.

For a 3D curve, it is difficult to calculate the step size, and is assumed arbitrarily in the interface software, but it need to be modified in future.

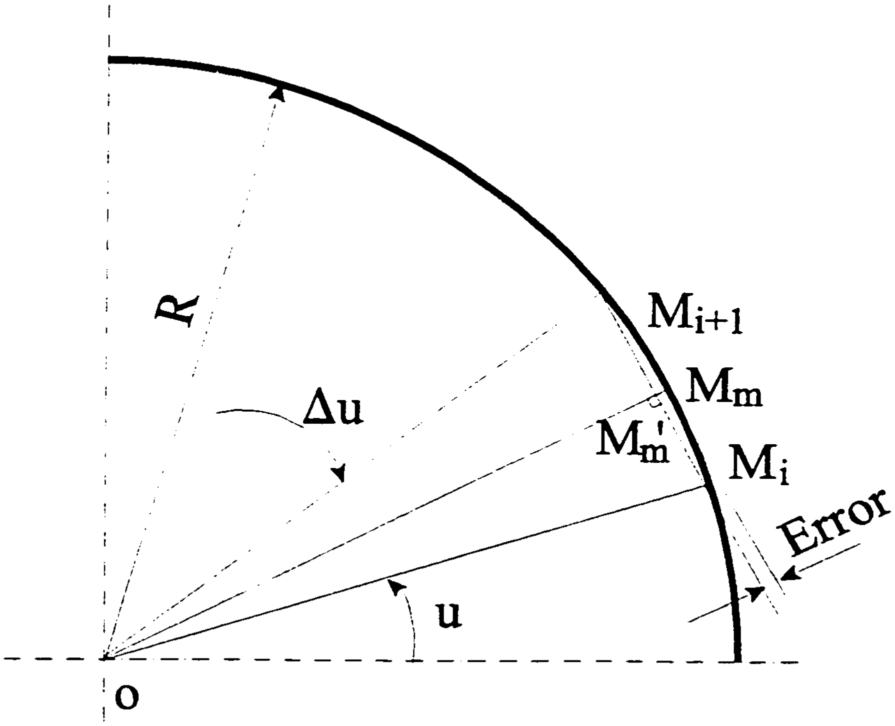


Figure 6.2 Representation of the Error of a Circular Arc

6.5 Tool Path Generation

Once the step size is determined, the coordinates of points and normals at each point on a tool path are calculated according to the relative parametric equations. The normal for a 2D curve is a constant vector which is always vertical to the plane on which the 2D curves lie. For a 3D curve, the calculation of the coordinates of machining points is the same as that of the 2D curves. The normal of a 3D curve is not a constant vector, and the details about the calculation is presented in Chapter 4.

6.6 Display of a Workpiece

In order to display a 3D workpiece, the method of isometric projection is used. The axes of the coordinate system is assumed as shown in the Figure 6.3. The Z axis is a vertical

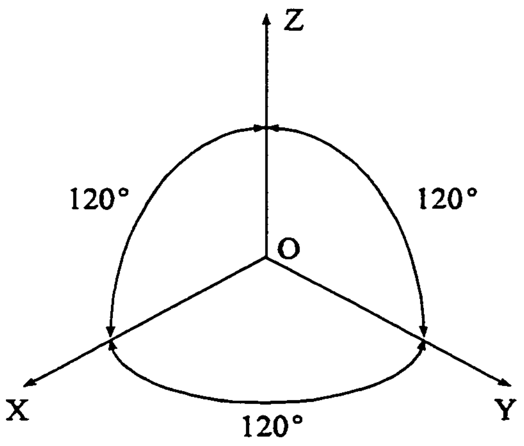


Figure 6.3   Display Coordinate System

line, X axis is a line with tangent angle 210°, and Y is a line with tangent angle -30°. In the software, k_xaxis represents the tangent of X axis, k_yaxis represents the tangent of Y axis.

For a point {x, y, z} on the plan parallel to XY plane, the isometric projection is the intersection point of two lines which are parallel to X and Y axes respectively. The projection of this point is:

$$x_p = \frac{(x-y)}{k_{xaxis}-k_{yaxis}}$$

$$y_p = \frac{xk_{xaxis}-yk_{yaxis}}{k_{axis}-k_{yaxis}} + z \qquad (6.6)$$

For a point {x, y, z} on the plane parallel to XZ plane, the isometric projection is the intersection point of two lines which are parallel to X and Z axes respectively. The projection of this point is:

$$x_p = x\cos 210^\circ + y\cos(-30^\circ)$$

$$y_p = x\sin 210^\circ + y\sin(-30^\circ) + z \qquad (6.7)$$

For a point {x, y, z} on the plane parallel to YZ plane, the isometric projection is the intersection point of two lines which are parallel to Y and Z axes respectively. The projection of this point is:

$$x_p = x\cos 210^\circ + y\cos(-30^\circ)$$

$$y_p = x\sin 210^\circ + y\sin(-30^\circ) + z \qquad (6.8)$$

All curves of a workpiece consist of points. For different locations of curves, use the different equations shown above.

6.7 Toolpath Transformation

Up to now, the toolpath is generated in the workpiece coordinate system. In order to transfer the toolpath from the workpiece coordinate system into the robot control system, a group of transformation equations are used. The deduction of these transformation equations are shown in Chapter 5

6.8 Output of Script Files

The output of the interface is a script data file which can be directly used by the Yamaha control workcell. The format of a script file is shown as following:

Table 6.1 Output Data File

X	Y	Z	A	B	V	ResetSignal
-707.031	50.095	270.825	-7.898	0.000	40.000	0xdd
-734.072	100.190	246.650	-15.796	0.000	40.000	
-751.113	150.285	222.475	-23.694	0.000	30.000	0xdd
-768.154	200.380	198.300	-31.592	1.080	10.000	
-785.195	250.475	174.125	-39.490	2.088	10.000	
-802.236	300.658	149.952	-47.388	3.148	5.000	
-804.236	320.568	149.952	-47.388	3.148	5.000	
-803.739	316.504	170.005	-40.728	2.006	10.000	
-769.156	211.003	198.335	-30.728	1.023	10.000	0xdd
-734.573	105.502	246.665	-10.728	0.000	30.000	0xdd
-699.999	0.000	295.000	0.000	0.000	40.000	0xdd

The first five columns are in robot coordinate system. The sixth column is the absolute velocity while the robot traves from start position to target position. The last column is optional, and when the coordinate differences, Δx, Δy, and Δz, between two point or the accumulated differences exceed the limits, the robot need to be reset. If the robot control workcell receives data "0xdd", a reset signal is sent to the robot. These limits are determined by moving the joystick of the teach units along x, y, and z axes. The hexadecimal value 0xdd equals to decimal value 221, so this value need to be avoided in coordinates of points on the tool path.

6.9 Conclusion

The interface is implemented in c. Array linked structures are used to save the related parameters for each kind entity, such as structure POINT is used to save the X, Y, and Z values of a point. Because there are many points on a workpiece, the structure POINT are defined as array object pt[].

Use flags to identify the different layers on which a certain entity lies.

In graphic model, a isometric Projection is used to represent a workpiece.

Function image is called to simulate the movement of toolpath.

CHAPTER 7 EXPERIMENTAL VERIFICATION

7.1 Introduction

This Chapter presents results from experiments carried out to verify the transformation equations derived in Chapter 5, and the tool path generated by the interface. The experiments are carried out using the Yamaha Zeta-1 Deburring robot and several measuring instruments.

Section 7.2 introduces the three methods developed to verify the transformation equations between the workpiece coordinate system and the robot coordinate system. Section 7.3 presents the methodology to verify toolpath and determine the location of the workpiece.

7.2 Experimental Verification of the Transformation Equations

The tool orientation obtained from the transformation equations are verified by measuring the angle between the axis of the tool and the machining surface.

The orientation of the axis of the end effector is calculated using the transformation equations (5.36 and 5.37). In order to verify these equations, a test setup containing a sine bar with inclination in two planes is used. Three frames are assigned. Frame {0} is fixed on the base of sine block which is attached to the machining table, frame {1} is fixed on the middle layer of the sine bar, and frame {2} is fixed on the top layer of the sine bar. Initially these three frames coincide and all Z axes are vertical and parallel to each other. To align the tool axis to be normal to the top layer of the sine bar, the sine bar is positioned and orientated by rotating frame {1} and {2} by δ_1 about Y-axis of frame {0}, and the frame {2} rotated by δ_2 about the X-axis of frame {1} as shown in Figure 7.1. The resulting orientation of the frame {2} with respect to the frame {0} is

$$^0T_2 = {}^0T_1\, {}^1T_2 = \mathrm{Rot}(Y_0, \delta_1)\mathrm{Rot}(X_1, \delta_2),$$

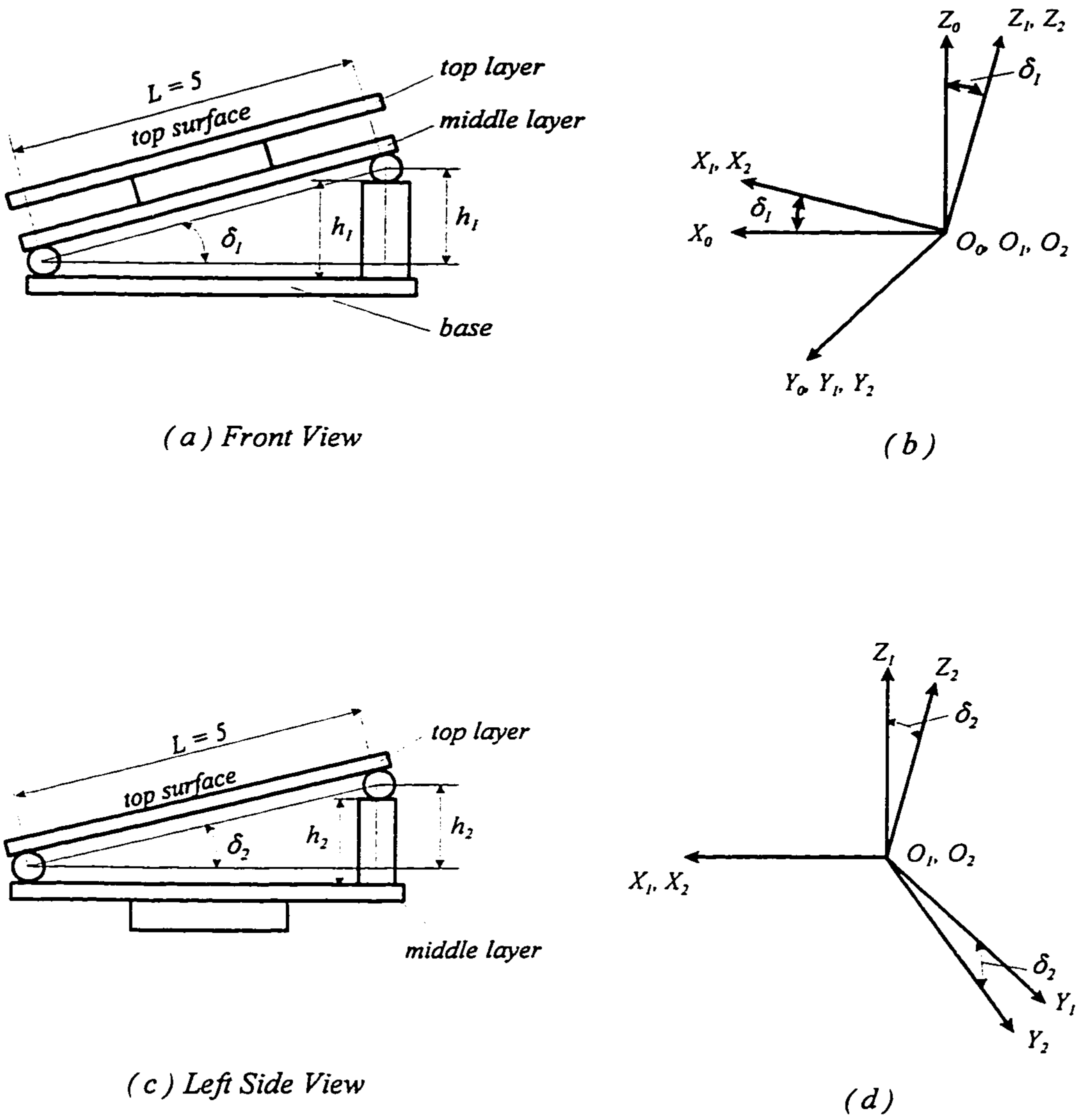
$$^0T_2 = \begin{bmatrix} \cos\delta_1 & \sin\delta_1\sin\delta_2 & -\sin\delta_1\cos\delta_2 & 0 \\ 0 & \cos\delta_2 & \sin\delta_2 & 0 \\ \sin\delta_1 & -\cos\delta_1\sin\delta_2 & \cos\delta_1\cos\delta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (7.1)$$

The three elements in the third column of the matrix $^0T_2$ are the projections of the z-axis of the frame {2} on the x-, y-, z-axis of the frame {0}. The normal of the top surface (frame{2}) with respect to the frame {0} is:

$$\begin{aligned} {}^0nx_2 &= -\sin\delta 1 \cos\delta 2 \\ {}^0ny_2 &= \sin\delta_2 \\ {}^0nz_2 &= \cos\delta_1 \cos\delta_2 \end{aligned} \qquad (7.2)$$

where $\delta_1 = \sin^{-1} [\frac{h_1}{L}]$ (7.3)

$\delta_2 = \sin^{-1} [\frac{h_2}{L}]$ (7.4)



(a) Front View

(b)

(c) Left Side View

(d)

Figure 7.1 Transformations Between Frames {0}, {1}, and {3}

The transformation between the base of robot and frame {0} is:

$$ {}^{r}T_{0} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & F_x \\ \sin\phi & \cos\phi & 0 & F_y \\ 0 & 0 & 1 & F_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.5} $$

so, $\quad {}^{r}nx_2 = {}^{0}nx_2 \cos\phi - {}^{0}ny_2 \sin\phi = - \sin\delta_1 \cos\delta_2 \cos\phi - \sin\delta_2 \sin\phi \qquad (7.6)$

$$ {}^{r}ny_2 = {}^{0}nx_2 \sin\phi + {}^{0}ny_2 \cos\phi = - \sin\delta_1 \cos\delta_2 \sin\phi + \sin\delta_2 \cos\phi \tag{7.7} $$

$$ {}^{r}nz_2 = {}^{0}nz_2 = \cos\delta_1 \cos\delta_2 \tag{7.8} $$

where ϕ is the rotation angle of the frame {0} with respect to the robot frame

The following transformation equations derived in Sections 5.3 and 5.4 are evaluated using equations from (7.6) - (7.8) :

$$ \theta = \tan^{-1} \frac{Y_r}{X_r} \tag{7.9} $$

$$ \alpha = \tan^{-1} [\frac{ny_r}{nx_r}] - \tan^{-1} [\frac{1-\cos\beta}{\sqrt{2}\sin\beta}] - \theta \tag{7.10} $$

$$ \beta = \cos^{-1} (2\, nz_r - 1) \tag{7.11} $$

$$ A = \tan^{-1} [\frac{ny_r}{nx_r}] \tag{7.12} $$

$$ B = \cos (\frac{1+\cos\beta}{2}) \qquad \text{if } \beta > 0 \tag{7.13a} $$

$$B = -\cos\left(\frac{1+\cos\beta}{2}\right) \qquad \text{if } \beta < 0 \tag{7.13b}$$

For a plane, the normal is constant, i.e., the value of β is also constant, and the value of A can also be assumed as a constant, so the value of $\alpha + \theta$ is a constant. Since the height of the gauge block is known, the angles δ_1 and δ_2, and the values of ${}^{r}nx_2$, ${}^{r}ny_2$, and ${}^{r}nz_2$ are calculated. Then, the value of A and B can be calculated using equations (7.12) and (7.13) respectively. Oriented the end effector using the calculated values of A and B, and move the robot to any point on the top surface of the sine bar.

Following scheme is devised to measure the offset ψ between the axis of the end-effector and the machining surface due to computing and control error when the robot is commanded to move to a location that is normal to the machining surface.

Using a height gauge and a dial gauge. The hight gauge gives the value of distance traversed along the slide bar. In order to measure the angle ψ, the height gauge should be modified. A fixture to hold the dial gauge has be designed and is firmly attached to the hight gauge as shown in Figure 7.2. A cylindrical bar is also designed to replace the probe to obtain a longer distance for measurement. The orientation error is estimated from the height gauge and dial gauge readings obtained while traversing the dial gauge along the cylindrical rod. The angular error (ψ) is obtained from

$$\tan\psi = \frac{\Delta x}{\Delta h} \tag{7.14}$$

where Δx is the change in dial gauge reading, and Δh is the change in hight gauge reading.

The records of measurements taken presented in Table 7.1. From this table, it can be concluded that the transformation equations and the parameters used are correct as the maximum angular error is about 0.21°. The small value of ψ is caused probably by many factors, namely, the initial error between the axis of the end-effector and the machining surface and the machining surface accuracy.

Table 7.1 Records of Orientation Error Measurement (Method-3)

Change in Height Gauge Reading Δh (mm)	Change in Dial Gauge Reading Δx (mm)	Error Angle ψ (°)
66.55	0.12	0.103
70.45	0.15	0.122
40.25	0.8	0.114
44.25	0.12	0.155
41.2	0.1	0.139
76.25	0.28	0.210 maximum error
73.35	0.05	0.039
70.7	0.03	0.024
68.6	0.1	0.084
50.65	0.13	0.136
79.12	0.05	0.036
65.46	0.18	0.158
47.45	0.15	0.181
56.99	0.2	0.201
46.5	0.09	0.111

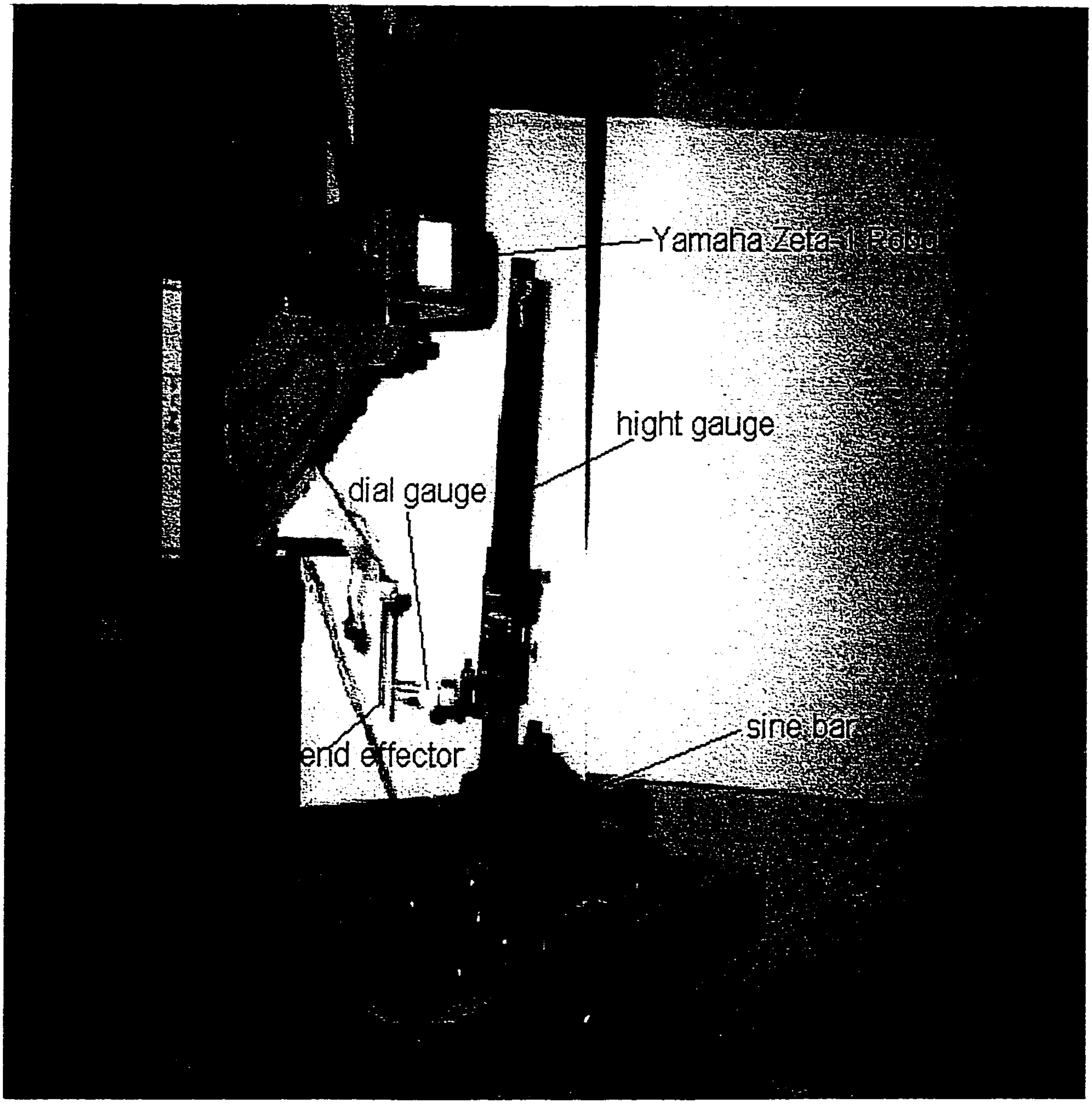


Figure 7.2 Experimental Set-Up Used to Measure the Off-Set

7.3 Experimental Verification of Toolpath Generated by the Interface

Experiments are carried out using the Yamaha Zeta-1 Deburring Robot to verify the tool path generated by the interface. For two-dimensional toolpath, a contour in the plane of the machining surface, and a trace of the actual tool path executed by the robot are obtained. The robot workcell also computes the tool path using the feedback data from the joint encoders. These records are compared with the CAD data. For a 3-D toolpath, a sample workpiece as shown in Figure 7.7 is designed and it is located in the robot coordinate system. The tool path for this workpiece is executed by the Yamaha robot which is controlled by the teach unit of the robot.

7.3.1 Verification of a Two-Dimensional Toolpath

Figure 7.3 shows the profile of the 2D workpiece generated using AutoCAD. The interface processes the DXF data file for this part, and automatically generates the toolpath. The generated toolpath is used to guide the end effector of the Yamaha robot, and a trace of the tool location is obtained.

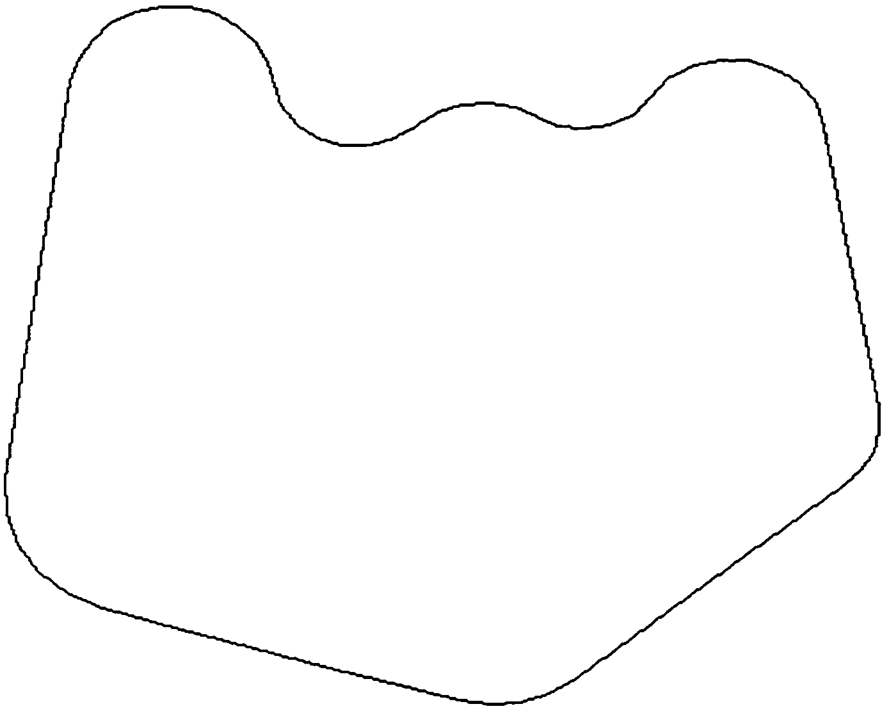


Figure 7.3 Schematic of a 2D Test Surface generated using AutoCAD

7.3.2 Verification of Three-Dimensional Toolpath

For three-dimensional toolpath, a workpiece shown in Figure 7.4 has been designed. The workpiece contains basic analytic entities: points, lines, arcs, and an intersecting curve of two cylindrical surfaces, and this intersecting curve appears in many industrial workpieces. The shapes of the intersecting curves are varied with the diameters of two intersecting cylinder. The parametric equation of the curve is derived from the parametric equations of two intersecting cylinders.

The workpiece is fixed on the machining table of the robot, and is located by two locating pins on the table. This ensures that the Z-axes of the workpiece and the robot control systems are parallel to each other, the Y-axis of workpiece coordinate system {W} aligns with the axis of the horizontal cylinder, and the line L connecting two locating holes is parallel to Y-axis of frame {W} and lies on the YZ plane of {W}. Assume the middle point between two aligning holes is the origin of the frame {W}, and the X-axis passes through these two holes as shown in Figure 7.5. If the locations of two aligning holes and the Z coordinate of the origin of frame {W} (machining table) in frame {R} are measured, the relationship between the workpiece coordinate system {W} and the robot control coordinate system {R} can be determined.

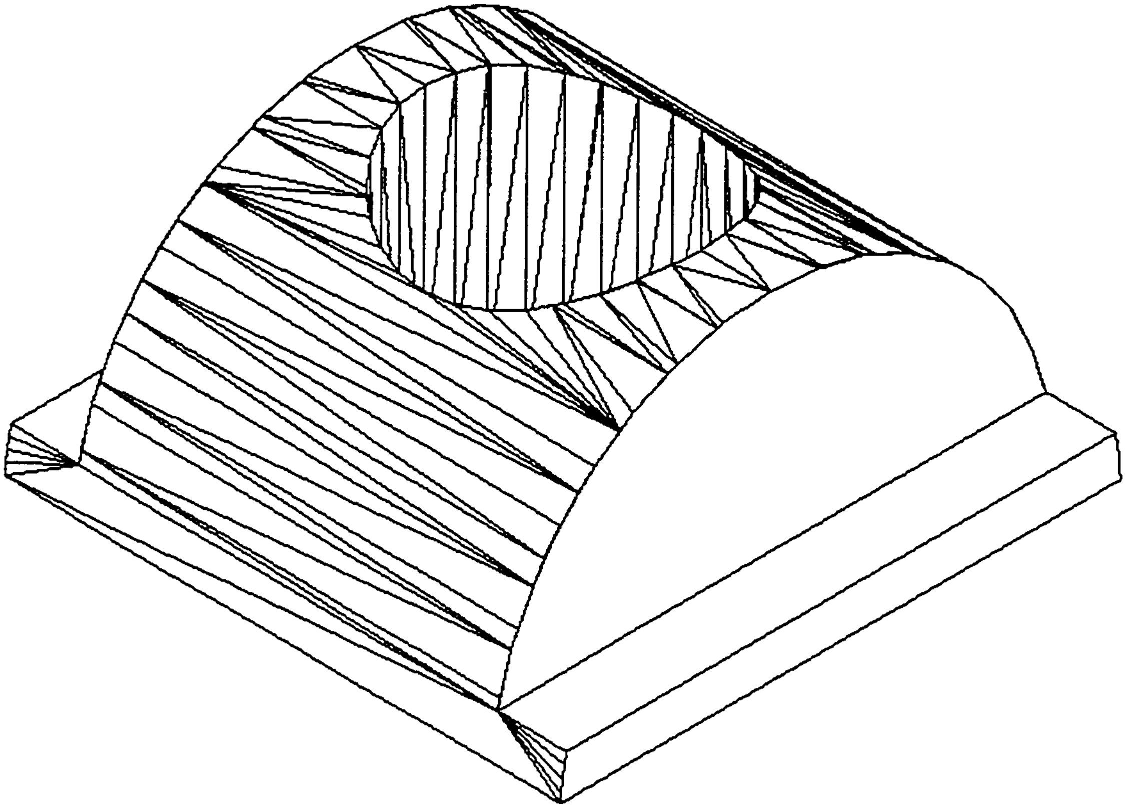


Figure 7.4 Workpiece Used for 3D Toolpath Experiments

The transformation between two systems is given by

$$ {}^{r}T_{w} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & F_x \\ \sin\phi & \cos\phi & 0 & F_y \\ 0 & 0 & 1 & F_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.15} $$

where ϕ is the angle between X_axis, of {W} and X_axis of {R}

T_x, T_y, and T_z are translations of the origin of {W} with respect to {R}. The results of the measurement are:

$\phi = 162.78°$

$T_x = -814.24$ mm

$T_y = 201.615$ mm

$T_z = 56.44$ mm

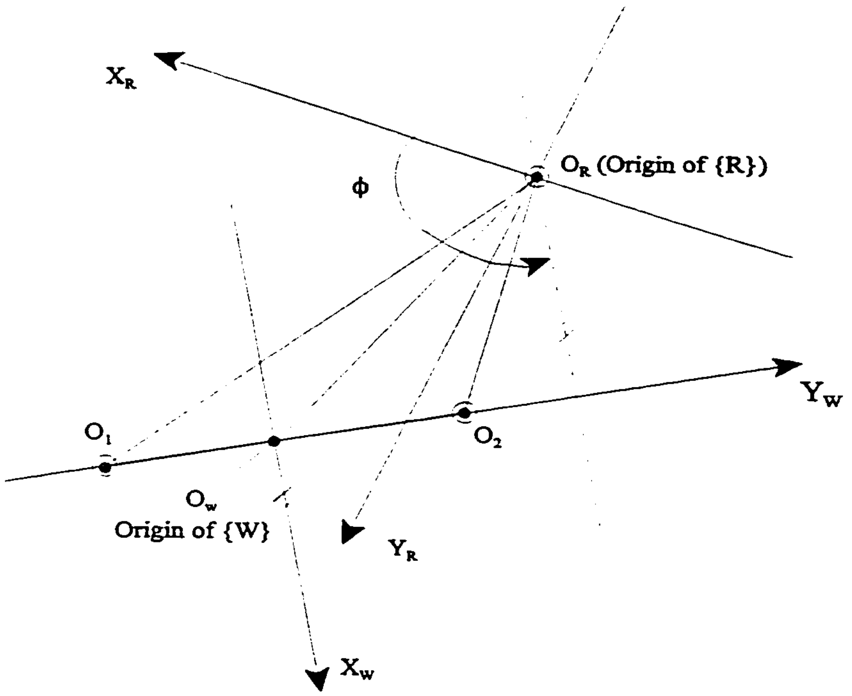


Figure 7.5 Relationship between Workpiece and the Robot Coordinates

The output of the interface is a script file which contains the values of X, Y, Z, A, B, V, and 0xdd (optional, used for reset the teach unit of the robot). This script file is read by control software of the robot workcell controller to automatically guide the tool along the intersecting curve. Figure 7.8 shows the desired intersecting curve and Figure 7.9 shows the tool ath generated by the interface, wherein the normals and the tangents at various locations are also shown.

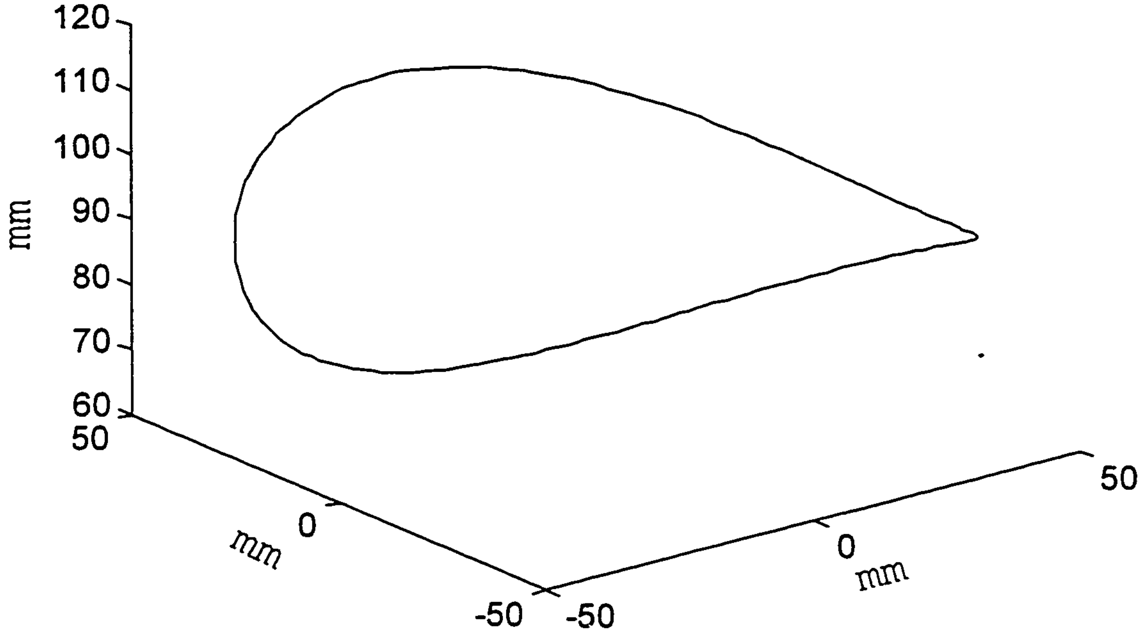


Figure 7.6 Intersecting Curve Designed by AutoCAD

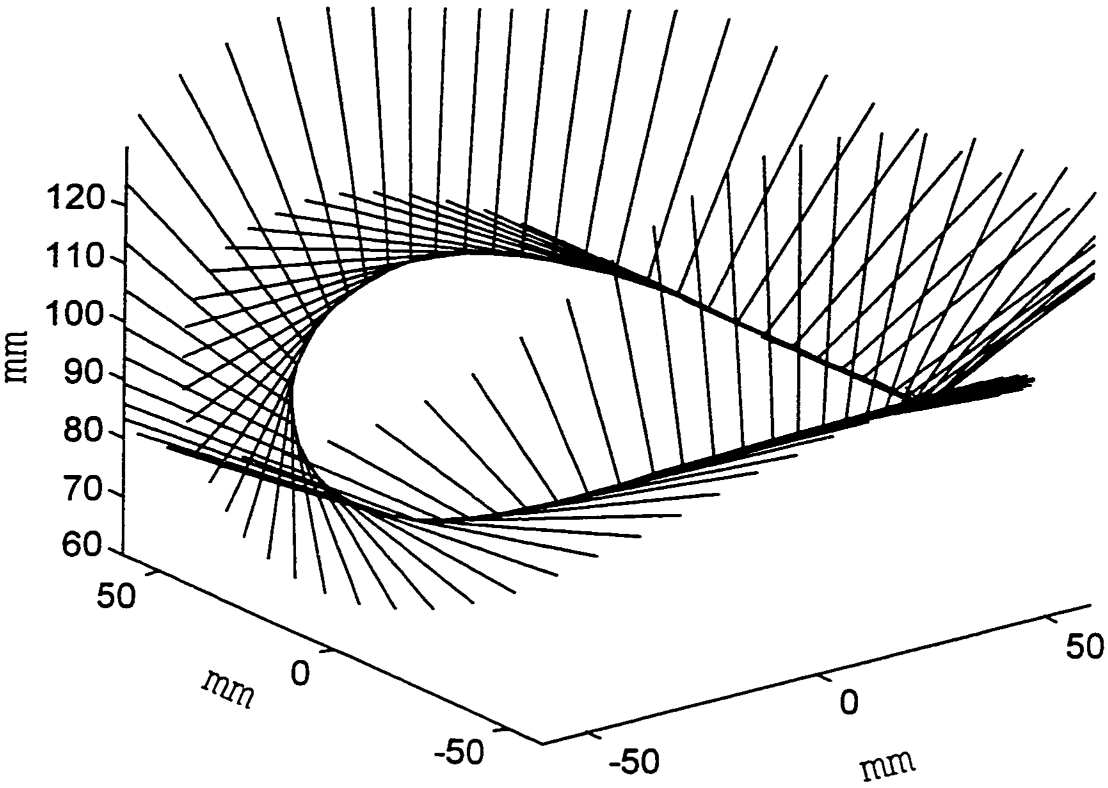


Figure 7.7 Intersecting Curve And its Tangent and Normal generated by the Interface

The designed toolpath is tested by using the teach unit, and the movement of the tool is observed. All machining points are located on the intersection curve, and the change in orientation of the tool is expected according to the calculation of the normals of the curve.

CHAPTER 8 CONCLUSION AND FUTURE WORK

8.1 Conclusion

An interface to generate toolpath for robot manipulators from CAD database has been developed in this thesis. The developed interface enables the user to design complex components using a standard off-the-shelf CAD package while machining can be carried out using robot manipulators designed for machining. The interface can also be used to generate toolpath for other machine tools by suitably modifying the sections related to the kinematics that relates the tool location to joint coordinates of the machine tool. A similar method of translating the CAD data to tool path is commercially available for numerically controlled (NC) machines.

In this work, Yamaha Zeta-1 deburring robot is chosen as it is designed specifically to debur complex engineering components and is also used by the aerospace industries to edge debur and polish surfaces of impellers and bearing races in vane rings used in gas turbine engines. Further, the Centre for Industrial Control has one such robot acquired as a part of the NSERC strategic grant. Experimental verification of the interface has been carried out using this robot. While there exist many CAD packages, the AutoCAD is chosen in this thesis as it is commonly used in small to medium-size manufacturing industries. The part database, DXF and ACIS formats generated by AutoCAD can be read by other CAD packages, such as CADKey, etc.

8.1.1 Research Focus

The thesis focused on developing an interface that automatically translates CAD information to toolpath for machining, in particular, for robot manipulators. The main contribution of this thesis lies in developing the interface to translate an industry standard CAD database to toolpath for machining robots. The various tasks carried out by the interface are

- Extract the geometric data from the database and re-organize the data so that connectivity between adjacent geometric entities can be established
- Generate parametric representation of 2D and 3D curves and extracting geometric shape at the points of intersection of solids in different planes
- Carry out kinematic transformation between workpiece coordinates and machining robot tool and world coordinates to generate the tool path
- Generate the toolpath based on the geometric data of the component, machining data (tool velocity, cutting depth, tool geometries) and robot-workpiece transformations

8.1.2 Geometric Data Processing

The geometric data processing is carried out on the database DXF and ACIS format files. The ACIS format saves solid model data while the DXF saves all the other model data. The DXF files are standard ASCII text files, providing information on the various entities in a particular CAD drawing and their properties. ACIS is an object-oriented geometric modeling toolkit designed for use as a geometry engine within 3-D modelling applications. The data format is open and can be read easily by external data processing utilities. It should be noted that these files do not provide the information concerning the connectivity between entities. The interface is capable of reading these two formats.

The data read is stored based on the entity, for example, a line structure stores all information (end points and direction cosines) concerning a line in Cartesian space.

8.1.3 Parametric Representation

Three types of geometric models are in use, namely, the wireframe, surface and solids. This thesis addresses only the wireframe as it is easy to represent them mathematically and is the most common method used in CAD systems.

All existing CAD systems provide users with basic wireframe entities which can be divided into analytic and synthetic entities. Analytic entities are points, lines, arcs, circles, and conics (ellipses, parabolas, and hyperbolas). Synthetic entities include various types of spline (cubic spline, B-spline, and so on) and Bezier curves. Only the analytic entities are considered in this thesis. Mathematical formulations for the above mentioned entities are described in Chapter 3.

8.1.4 Toolpath Generation

A workpiece is machined to the finished shape by allowing a relative motion between the workpiece and the cutting tool. Such relative motion can be provided by holding the workpiece stationary and moving the cutting tool, or by moving the cutting tool and the workpiece simultaneously. In the Yamaha Zeta-1 deburring robot, the relative motion is achieved by the motion of the end effector (tool tip). An axis of motion is defined as an axis where relative motion between the cutting tool and workpiece occurs. The primary three axes of motion are referred to as the X, Y, and Z axes and from the workpiece coordinate system. The XYZ system is a right-hand system and the location of its origin is fixed. The Yamaha Zeta-1 deburring robot is a special machine, except three motions along X, Y, and Z, it also has two more rotary motions, so machining a workpiece involves five motion axes. In order for the cutting tool to produce the required finished workpiece, its path must guide it properly, and transitional points must be calculated properly.

The complete toolpath generation should include four steps: recognition of machined edges, tool path generation, tool path verification, and collision detection. This thesis consider the cutting free-form edges, so it deals with the problem of accuracy. The method used for generating and programming a toolpath is called cutter center programming. It considers the actual diameter of the cutter during the toolpath generation. In this method, the program guides the tool around the workpiece contour. The cutter will have to follow the path at a set distance away from the workpiece, at each point on the path, corresponding to the cutter radius.

8.1.5 Transformations and Experimentation

The toolpath is generated in workpiece coordinate system. A kinematic model of the Yamaha Zeta-1 robot has been developed to transform the tool path from workpiece coordinates to robot coordinates. The forward and inverse kinematic models are developed using the transformation approach (presented in Chapter 5). The generated equations are verified by developing a geometric approach (described in Appendix B) and experimental results from the Yamaha robot. Since actual machining is beyond the scope of this thesis. In order to verify the generated tool path through actual machining one need to address the following issues as they would contribute to inaccuracies:

- Robot trajectory tracking accuracy
- Kinematic model accuracy as it depends on knowing accurately the kinematic parameters of the robot and assumptions (robot joints are either perpendicular or parallel to each other) in

transformation and geometric approaches

- Workcell toolpath transformation accuracy
- Component accuracy
- Accuracy with which the component can be located with respect to the robot
- Interface tool path generation accuracy

Hence, actual machining are not carried out as they would not result in additional information on the correctness of the interface. Instead, the tool is moved in the air without actual machining. Visual checks are made with regard to the path followed and tool orientation (for any abrupt large changes in orientation -- due to singularity).

8.2 Future Work

The work presented in this thesis can be enhanced by focusing on

- Processing other CAD database files
- Analyzing synthetical entities and surfaces
- Generating toolpath for other robot manipulators used for machining and machine tools
- Addressing the issues related to experimental validation in order to transfer the knowledge to manufacturing industries

One of the standard CAD/CAM packages adopted by large industries is CATIA. The interface should be modified to include CATIA database as well. This would serve the aerospace industries as the use CATIA and Yamaha robot for machining. To machine products such as car bodies, ship hulls, airplane fuselage and wings, and propeller blades, synthetical entity (Hermite cubic spline, Bezier and B-spline) and surface representations are needed. The interface can be modified to include these shapes by incorporating the parametric form of these shapes. As indicated in section 8.1.4, it is necessary to address many issues related to machining accuracy before attempting to carry out validation experiments. Future work should focus on addressing accuracy issues and experimental validation of the interface.

REFERENCES

1 Chris McMathon, Jimmie Browne, *CADCAM, from principles to practice*, Addison-wesley Publishing Company, 1993

2 Ayyadevara, V. R., *Development of an Automated Robotic Deburring Workcell*, M.A.Sc. Thesis, Department of Mechanical Engineering, Concordia University, Dec.1995.

3 Bruce J. Roussel, *Development of a Methodology for Reconditioning Complex Precision-Machined components Using a Robotic Deburring Workcell*, M.A.Sc. Thesis, Department of Mechanical Engineering, Concordia University, July 1996.

4 *YAMAHA Zeta-1 Deburring Robot User's Manual*, v.1.2, YAMAHA Corporation, Jan.1990.

5 T. M. Stepien, el, *Control of Tool/Workpiece Contact Force with Application to Robotic Deburring*, IEEE J. of Robotics and Automation, Vol. RA-3, No.1, pp.7-18, Feb.1989.

6 H. Kazerooni, J. J. Bausch, and B. M. Kramer, *An Approach to Automated Deburring by Robot Manipulator*, ASME J. of Dynamic System, Measurement, and Control, Vol.108, pp.354-359, Dec. 1986.

7 S. Liu and H. MAsada, *Adaptive Control of Deburring Robots Basaed on Human*

Demonstration Data, Proceedings of the 1992 IEEE Conference on Industrial Electronics, Control, Instrumentation and Automation, Vol. 2, pp.927-933, 1992

8 M. G. Her, H. Kazerooni, *Automated Robotic Deburring of Parts Using Compliance Control*, ASME Journal of Dynamics Systems, Measurement and Control, Vol.113, No.1,pp60-66, 1991.

9 R. Rajagopalan, R. M. H. Cheng, V. R. Ayyadevara, and G. Huard, *Design and Performance of an Open architecture Controller for Robotic Deburring*, International Journal of Intelligent Control and Systems, Vol. 1. No. 4, 1996

10 R. Rajagopalan, R. M. H. Cheng, V. R. Ayyadevara, and G. Huard, *System Architecture and Edge Tracking Performance of an Automated Robotic Deburring Workcell*, Proceedings of the 1995 IEEE International Symposium on Intelligent Control, Monterey, California, pp. 351-356, August, 1995

11 M. Temple-Raston, R. M. H. Cheng and R.Rajagopalan, *Automated Digitization and Geometric Modelling of Refurbished Components for Deburring*, 1994 IEEE, pp. 345-349

12 R. M. H. Cheng, R.Rajagopalan, and M. Temple-Raston, *The Differential Geometric Modelling of Compressor Blades,* Proceedings of the American Control Conference, Maryland, June 1994

13 J. Canny, *A New Algebraic Method for Robot Motion Planning and Real Geometry*, IEEE 28th Annual Symposium on Foundations of Computer Science Conference Record, Oct. 1987.

14 Q. J. Ge, J. M. McCarthy, *An Algebraic Formulation of Configuration-Space Obstacles for Spatial Robots*, IEEE International Conference on Robotics and Automation, May 1990.

15 J. F. Canny, M. C. Lin, *An Opportunistic Global Path Planner*, IEEE International Conference on Robotics an Automation, May 1990.

16 Tsai-Yen Li, Jean-Claude Latombe, *On-Line Manipulation Planning for Two Robot Arms in a Dynamic Environment*, International Journal of Robotics Research, Volume 16, Number 2, April 1997.

17 Abhay V. Trivedi, *CAD/Robotics in the CIM Environment*, Fourth World Conference on Robotics Research, Sept., 1991.

18 Bopaya Bidanda, Vivek Narayanan, *Automatic CAD-Based Off-Line Programming of a Glaze Spraying Robot*, Fourth World Conference on Robotics Research, Sept., 1991.

19 A. Vishnu, M. Cutkosky, *Off-Line Programming for Automated Robotic Deburring*, Proceed of the USA-Japan Symposium on Flexable automation, Minnesota, July 1998.

20 *AutoCAD Release 11 Reference Manual*, Autodesk, Inc., Aug., 1990

21 Frederic H. Jones, and Lioyd Martin, "The AutoCAD Database Book---Accessing and Managing CAD Drawing Information Third Edition", Ventana Press, Inc.

22 ACIS Save File Format Manual, Spatial Technology, Inc.

23. Ibrahim Zeid, *CAD/CAM Theory and Practice,* McGraw-Hill, Inc. 1991

24 Faux, I. D, and Pratt, M. J. *Computational Geometry for Design and Manufacturing*, Halsted Press, 1979.

25 Irvin H. Kral, *Numerical Control Programming in APT*, Prentice-Hall, 1986

26 R. Paul, *Robot Manipulators*, MIT Press 1981

27 John J. Craig, Introduction to Robotics – Mechanics and Control, Addison-Weley Publishing Company, 1989.

APPENDIX A Spatial Descriptions of Positions and Orientations

1. Introduction

For robotic manipulation, workpieces and the tools will be moved around in space by some sort of mechanism, so it is needed to represent the positions and orientations of the workpieces, tools, and the mechanism itself. To define and manipulate mathematical quantities which represent position and orientation coordinate systems must be defined.

The philosophy that somewhere there is a universal system to which everything discussed can be referenced is adopted. All positions and orientations will be described with respect to the universal coordinate system or with respect to other Cartesian systems which are defined relative to the universal systems.

2. Description of a Position

Once a coordinate system is established, any point in the universe with a 3×1 position vector can be located, such as the machining point of the Yamaha Zeta-1 robot. Because there are many coordinate systems in addition to the universe coordinate system, vectors must be tagged with information identifying which coordinate system they are defined within. In this thesis, vectors are written with a leading superscript indicating the coordinate system to which they are referenced, for example, ^{A}P. This means that the components of ^{A}P have numerical values which indicate distances along the axes of a coordinate system {A} as shown in Figure 1 Each of these distances along an axis can be thought of as the result of projecting the vector onto the corresponding axis. Figure A1 pictorially represents a coordinate system, { A }, A point ^{A}P is represented with a vector and can equivalently be thought of as a position in space.

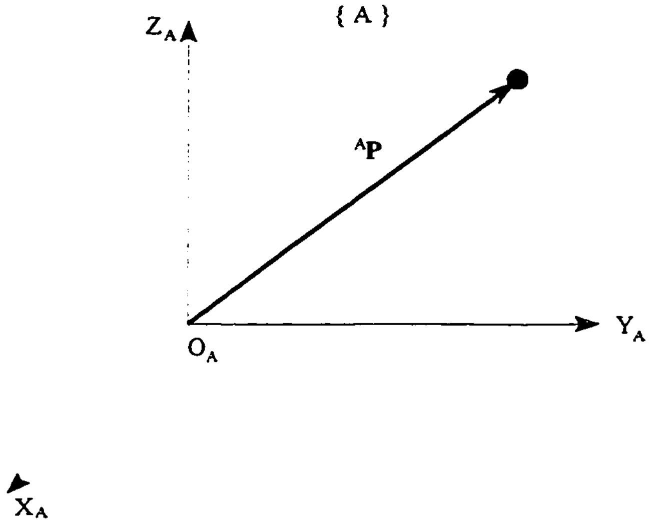


Figure A-1. Point, Vector, and a Coordinate System

$$^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad (A\text{-}1)$$

where p_x, p_y and p_z are projecting the vector onto the corresponding axis.

3. Description of an Orientation

It is necessary not only to represent a point in space but also to describe the orientation of a body in space. When machining a surface or an edge, we need to position the tool at the machining point, and also need to orient the tool with the normals of the surface or the edge. Because the YAMAHA robot has five active joints, the tool could be oriented arbitrarily while keeping at the same machining point in space. In order to describe the orientation of a body, a coordinate system need to be attached to the body and then give a description of this coordinate system relative to the reference system. In Figure 2,coordinate system { B } has been attached to a body in a known way. A description of {B} relative to {A} suffices to give the orientation of the body.

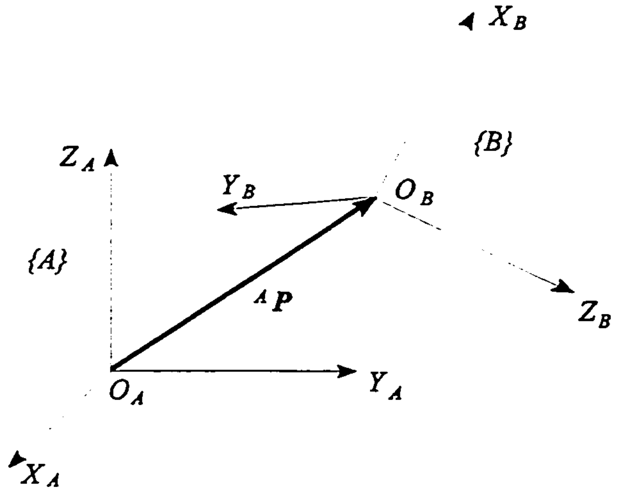


Figure A-2. Relationship between Two Coordinate Systems

Thus, positions of points are described with vectors and orientations of bodies are described with an attached coordinate system. To describe the body-attached coordinate system, {B}, is to write the unit vectors of its three principal axes in terms of the coordinate system {A}. The unit vectors giving the principal directions of coordinate system {B} as , $\bar{\boldsymbol{Y}}_{B}$, and $\bar{\boldsymbol{Z}}_{B}$. When written in terms of coordinate system {A} they are called ${}^{A}\bar{\boldsymbol{X}}_{B}$, ${}^{A}\bar{\boldsymbol{Y}}_{B}$, and ${}^{A}\bar{\boldsymbol{Z}}_{B}$. To stack these three unit vectors together as the columns of a 3 × 3 matrix, in the order ${}^{A}\bar{\boldsymbol{X}}_{B}$, ${}^{A}\bar{\boldsymbol{Y}}_{B}$, and ${}^{A}\bar{\boldsymbol{Z}}_{B}$, this matrix is called a 3 × 3 rotation matrix, and named with the notation ${}^{A}R_{B}$.

$$ {}^{A}R_{B} = \begin{bmatrix} {}^{A}\bar{\boldsymbol{X}}_{B} & {}^{A}\bar{\boldsymbol{Y}}_{B} & {}^{A}\bar{\boldsymbol{Z}}_{B} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad \text{(A-2)} $$

The scalars r_{ij} in the equation 2 are the components which are the projections of relative vectors onto the unit directions of its reference frame. Hence, each component can be written as the dot product of a pair of unit vectors as

$$ {}^{A}R_{B} = \begin{bmatrix} {}^{A}\bar{\boldsymbol{X}}_{B} & {}^{A}\bar{\boldsymbol{Y}}_{B} & {}^{A}\bar{\boldsymbol{Z}}_{B} \end{bmatrix} = \begin{bmatrix} \bar{X}_{B}\bar{X}_{A} & \bar{Y}_{B}\bar{X}_{A} & \bar{Z}_{B}\bar{X}_{A} \\ \bar{X}_{B}\bar{Y}_{A} & \bar{Y}_{B}\bar{Y}_{A} & \bar{Z}_{B}\bar{Y}_{A} \\ \bar{X}_{B}\bar{Z}_{A} & \bar{Y}_{B}\bar{Z}_{A} & \bar{Z}_{B}\bar{Z}_{A} \end{bmatrix} \qquad \text{(A-3)} $$

Since the dot product of two unit vectors yields the cosine of the angle between them, the components r_{ij} of rotation matrix are referred to as direction cosines.

4. Description of a Frame

The information needed to completely specify the whereabouts of the tool of the YAMAHA robot is a position and an orientation. For convenience, the point whose position will be described is chosen as the origin of the body-attached frame. Equivalently, the description of a frame can be thought of as a position vector and a rotation matrix.

$$\{B\} = \{ {}^{A}R_{B}, {}^{A}\mathbf{P}_{BORG}\} \qquad \text{(A-4)}$$

5. Changing Descriptions from Frame to Frame

The previous section having introduced descriptions of positions, orientations, and frames, this section considers the mathematics of mapping in order to change descriptions from frame to frame.

In Figure A-3, a position is defined by the vector ${}^{A}\mathbf{P}$. To express this point in space in terms of frame {A}, the relationship between two frames {A} and {B} need to be analysed. There are three cases: only translated frames are involved, only the rotated frames are involved, or translated and rotated frames are involved as shown in the Figure A-3. In the case, {B} differs from {A} only by a translation which is given by ${}^{A}\mathbf{P}_{BORG}$ as in the Figure A-3 (a), a vector which locates the origin of {B} relative to {A}.

$${}^{A}\mathbf{P} = {}^{B}\mathbf{P} + {}^{A}\mathbf{P}_{BORG} \qquad \text{(A-5)}$$

If the mapping involving rotated frames as in the Figure A-3 (b), as mentioned early, use a 3 × 3 rotation matrix ${}^{A}R_{B}$ to describe {B} relative to {A}.

$${}^{A}\mathbf{P} = {}^{A}R_{B}\, {}^{B}\mathbf{P} \qquad \text{(A-6)}$$

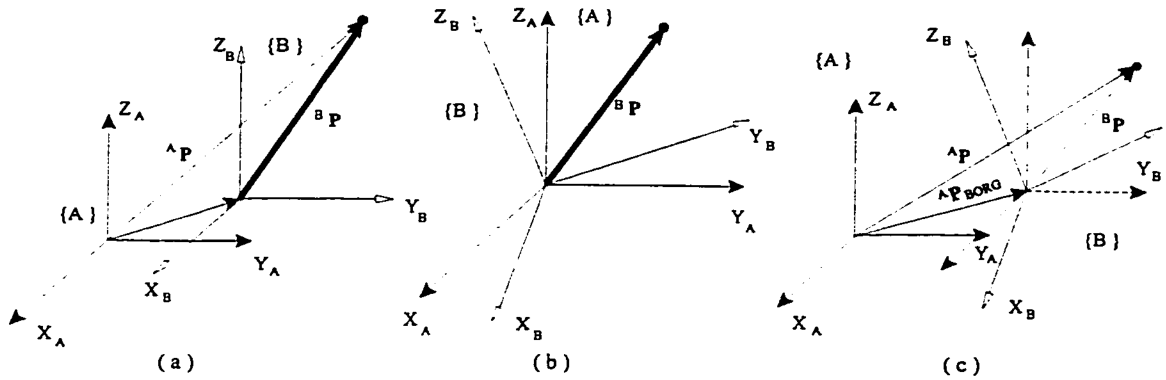


Figure A-3 Definition of the Position of a Point

If the translated and rotated frames are involved as in the Figure 4 (c), at first change ${}^{B}\mathbf{P}$ to its description relative to an intermediate frame which has the same orientation as {A}, but whose origin is coincident with the origin of {B}. This is done by premultiplying by ${}^{A}R_{B}$ as in the previous section. Then account for the translation between origins by simple vector addition, yielding

$$ {}^{A}\mathbf{P} = {}^{A}R_{B}\,{}^{B}\mathbf{P} + {}^{A}\mathbf{P}_{BORG} \qquad \text{(A-7)} $$

Equation 7 describes a general transformation mapping of a vector from its description in one frame to a description in a second frame. It can be written in the matrix form:

$$\begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \begin{bmatrix} & {}^{A}R_{B} & & {}^{A}P_{BORG} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} = {}^{A}T_{B} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix}$$

where ${}^{A}T_{B}$ is a 4 × 4 matrix and called a **homogeneous transform**.

APPENDIX B   The Relationship Between Two Planar Entities

The relationship between two entities is complex, and all possibilities should be considered in order to meet different cases. The entities considered in this section are regular planar curves, lines, circles or circular arcs, ellipse arc.

1. Two Lines

There are three possible relationships between two lines: parallel to each other, intersecting each other, and neither parallel to nor intersecting each other.

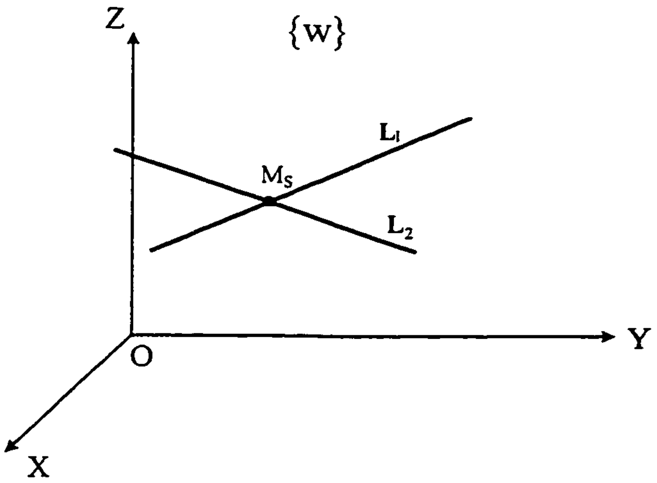


Figure B-1   Representation of Two Intersecting Lines

Lines $\mathbf{L}_1$ $(x_1 + p_1t, y_1 + q_1t, z_1 + r_1t)$ and $\mathbf{L}_2$ $(x_2 + p_2t, y_2 + q_2t, z_2 + r_2t)$ intersect at point M_s as shown in the Figure B-1. The coordinates of the intersection point is given by (x_s, y_s, z_s)

$$\left.\begin{aligned} x_s &= \frac{x_1p_2 - x_2p_1}{p_2 - p_1} \\ y_s &= \frac{y_1q_2 - y_2q_1}{q_2 - q_1} \\ z_s &= \frac{z_1r_2 - z_2r_1}{r_2 - r_1} \end{aligned}\right\} \qquad \text{(B-1)}$$

2. A Line and A Circle

A line may be (i) tangent to the circle; (2) intersecting the circle; (3) neither tangent to nor intersecting the circle. The derivation for the first two cases are presented below.

A line $\mathbf{L}$ $(x_1 + p_1t, y_1 + q_1t, z_1 + r_1t)$ is tangent to a circle $\mathbf{C}(x_c + R\cos(u), y_c + R\sin(u), z_c)$ as shown in Figure B-2. In the local coordinate system{L}, the tangent point is given by (x_t, y_t, z_t).

$x_t = H_2 / H_1$

$y_t = k\,x + b$

$z_t = z_1$

where $k = q / p$ (if $p \neq 0$)

$b = y_1 - k\,x_1$

$H_1 = 1 + k^2$

$H_2 = x_c + y_c\,k - b\,k$

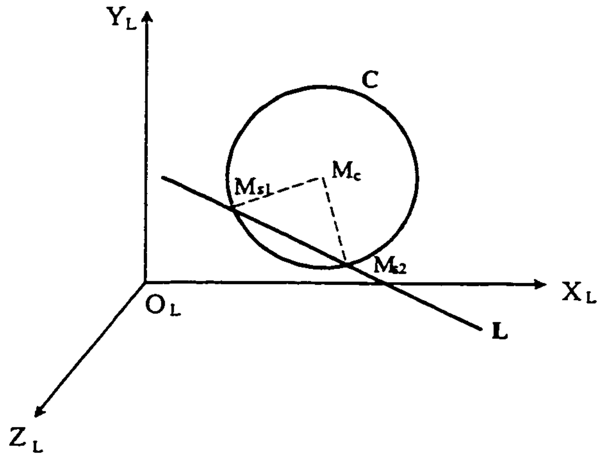


Figure B-2 Representation of a Line Tangent to a Circle

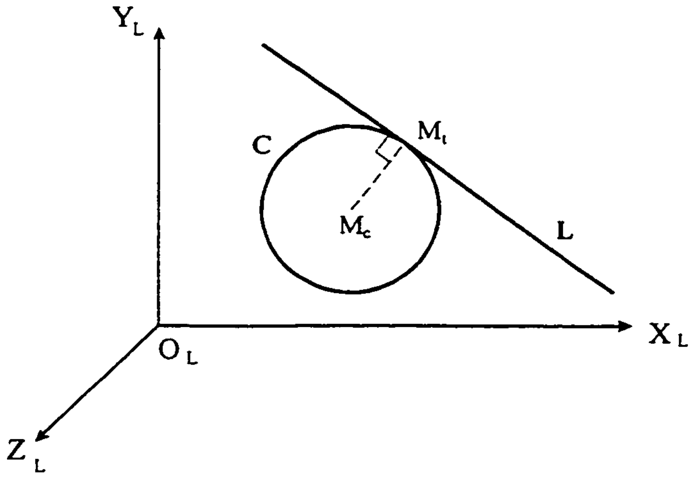


Figure B-3 Representation of a Line Intersecting a Circle

The coordinates of a tangent point can be expressed in the world coordinate system.

When a line **L** intersects a circle **C**, two intersection points result, M_{s1} (x_{s1}, y_{s1}, z_{s1}) and Ms_2 (x_{s2}, $y_{s2,}$ z_{s2}) as shown in Figure B-3.

$$x_{s1} = \frac{H_2 + \sqrt{\ } H_2^2 - H_1 H_3}{H_1}$$

$$y_{s1} = k\,x + b$$

$$z_{s1} = z_1$$

$$x_{s2} = \frac{H_2 + \sqrt{\ } H_2^2 - H_1 H_3}{H_1}$$

$$y_{s2} = k\,x + b$$

$$z_{s2} = z_1$$

where $k = q / p$

$$b = y_1 - k\,x_1$$

$$H_1 = 1 + k^2$$

$$H_2 = x_c + y_c\,k - b\,k$$

$$H_3 = x_c^2 + (b - y_c)^2 - R^2$$

3. Two Circles

Two circles may be tangent to each other and may intersect each other. One of circles may be inside the other, or may be outside the other as shown in Figure B-4. All possible cases are considered in this thesis.

$$\mathbf{C_1} \qquad \left.\begin{aligned} x &= x_{c1} + R_1 \cos(u) \\ y &= y_{c1} + R_1 \sin(u) \\ z &= z_{c1} \end{aligned}\right\} \qquad \text{(B-4)}$$

$$\mathbf{C_2} \qquad \left.\begin{aligned} x &= x_{c1} + R_1 \cos(u) \\ y &= y_{c1} + R_1 \sin(u) \\ z &= z_{c1} \end{aligned}\right\} \qquad \text{(B-3)}$$

Solving the above equations simultaneously results in the coordinates of the tangent point or intersection points. If two circles are tangent to each other, the tangent point is given by:

$x_t = D_2 / D_1$

$y_t = H_1 x_L + H_2$

$z_t = z_c$

If two circles intersect each other, the intersection points are:

$$x_{s1} = \frac{D_2 + \sqrt{D_2^2 - D_1 D_3}}{D_1}$$

$y_{s1} = H_1 x_{L1} + H_2$

$z_{s1} = z_{c1}$

$$x_{s2} = \frac{D_2 - \sqrt{D_2^2 - D_1 D_3}}{D_1}$$

$y_{s2} = H_1 x_{L2} + H_2$

$z_{s2} = z_{c2}$

where $H_1 = \dfrac{x_{c1} - x_{c2}}{y_{c2} - y_{c1}}$

$$H_2 = \frac{R_1^2 - R_2^2 + y_{c1} - y_{c2} + x_{c1} - x_{c2}}{2\,(y_{c2} - y_{c1})}$$

$D_1 = 1 + H_1^2$

$D_2 = x_{c1} + y_{c1}H_1 - H_1 H_2$

$D_3 = (H_2 - y_{c1})^2 + x_{c1}^2 - R_1^2$

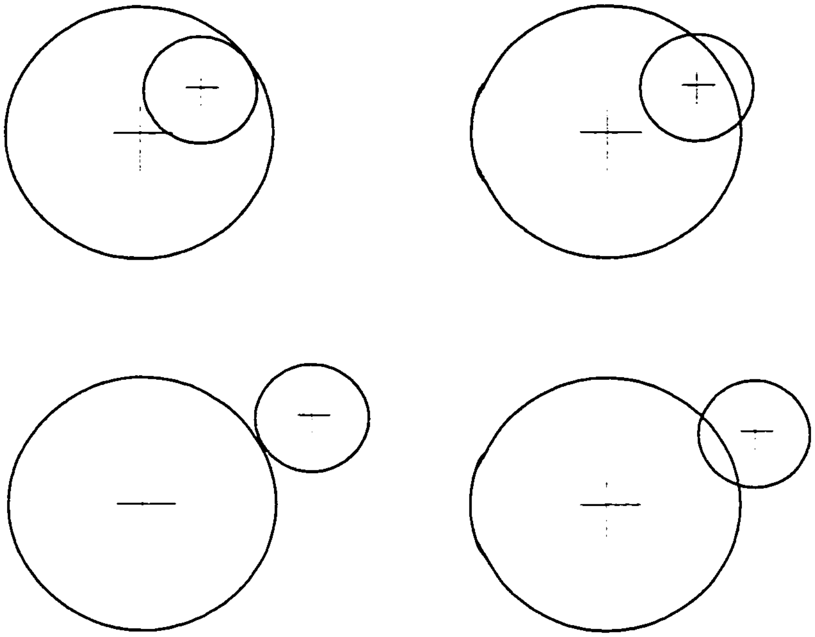


Figure B-4   The Relationship between Two Circles

APPENDIX C Geometric Approach to Solve Kinematic Equations for the Yamaha Zeta-1 Deburring Robot

1. Introduction

In Chapter 5 the algebraic approach is developed to solving kinematic equations. Here a geometric approach will be discussed to finding a manipulator's solution. In the geometric method the spatial geometry of the arm is decomposed into several plane geometry problems. Joint angles can be solved for using the tools of plane geometry.

2. Geometric Analysis of the Yamaha Robot

This section deals with the geometric analysis of the Yamaha Zeta-1 deburring robot to solve the transformation between θ, α, β and the normal components nx, ny, nz.

The relationship between the normal components nx, ny, nz and the rotation angles θ, α and β is complex. If $\beta = 0$, there is no effect of the angles θ and α on the orientation of the tool because the tool remains vertical. The angle θ is determined by the coordinate values of the machining point, and the angle α can assume any value. When $\beta \neq 0$, all three joints θ, α and β affect the orientation of the tool. Figure B-2 and B-3 show the relationship between them. The tumbling surface is the plane which passes through the tool and α axis, and when $\beta = 0$, assume it passes through the tool, α axis and β axis. After moving the machining point of the robot to the machining point on the surface or edge from the ready position, the tumbling surface, is located at P_θ.

Assume an angle $\delta = \tan^{-1} \dfrac{ny_r}{nx_r}$, -90° ≤ δ ≤ 90°. The plane containing the normal of the surface or edge is located at P_n. Assume the tool turns about α axis by -θ + δ degrees, the tumbling surface is located at the plane which is perpendicular to the plane P_n, see Figure B-2.

$\alpha_1 = -\theta + \delta.$ (B-4)

Figure B-3 shows the relationship between β angle and the normal component nz. γ is a angle between the tool and a vertical line, and $\gamma = \cos^{-1}(nz_r)$. Point M_1 is a point on the tool, and when the tool turns about β axis by β degrees, it moves to the point M_2. Assume the radius of tracing circle of point M_1 is R. In the right triangle OO_1M_1, the length of line OM_1 is r. In triangle M_1M_2O1, the length of the line M_1M_2 is:

$$|M_1M_2| = r^2 + r^2 - 2r^2 \cos\beta = 2r^2 (1 - \cos\beta)$$

In the triangle M_1M_2O, the length of the line M_1M_2 is:

$$|M_1M_2| = 2r^2 + 2r^2 - 4r^2 \cos\gamma = 4r^2 (1 - \cos\gamma)$$

$$2r^2 (1 - \cos\beta) = 4r^2 (1 - \cos\gamma)$$

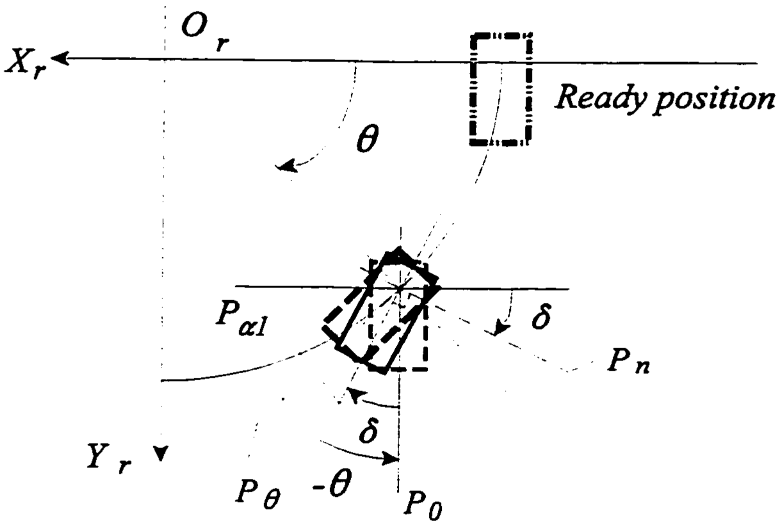


Figure B-1 The Relationship Between Planes P_n, P_0, P_θ, and $P_{\alpha 1}$

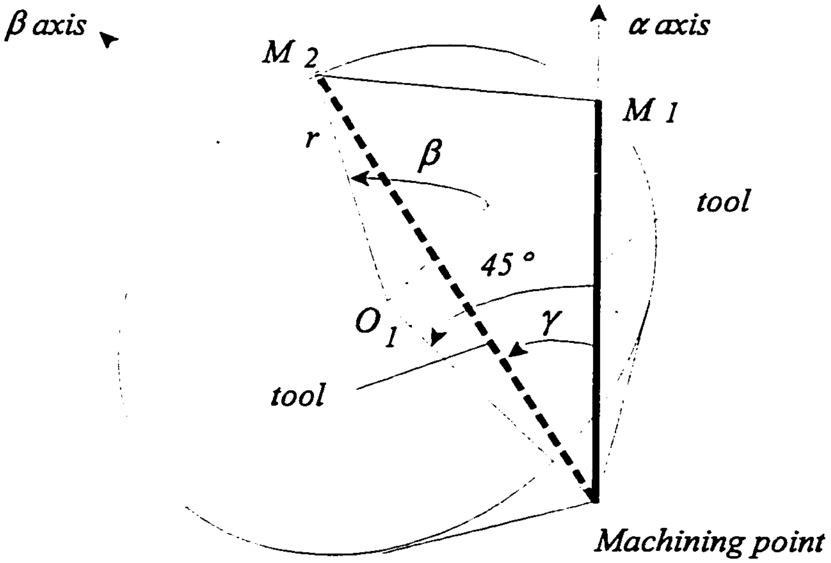


Figure B-2 The Relationship Between the Angles β and γ

$$\beta = \cos^{-1}(2\cos\gamma - 1) = \cos^{-1}(2nz_T - 1) \qquad \text{(B-5)}$$

$$\cos\gamma = \frac{\cos\beta + 1}{2} \qquad \text{(B-6)}$$

After the tool turning about β axis by β degrees, the tool tumbling surface is moved to the plane P_β. The sign of β depends on the angle value of nx. If nx >0, the value of β should be positive, else, the value of β should be negative. The plane P_T is coincident with the tracing circle of the point M_1. Assume a local coordinate system {L}, which Z axis always is vertical, ans which X axis always is parallel to the plane P_n. Project the tracing circle of the point M_1 and the point M_2 onto the plane P_T, X_LY_L and Y_LZ_L, and as shown in the Figure B- 3 and B-4. The plane P_T is inclined by 45 degrees according to the structure of the robot. The tracing circle is with its center at point O_1. Connecting O_1M_1 and O_1M_1, and from point M_2 drawing a line M_2M_3 perpendicular to the line O_1M_1 with the intersection point M_3. In order to align the tool to coincident to the normal of the surface or edge, the tool needs to be rotated about α axis by α_2 degrees. α_2 is the angle between the plane P_n and P_β.

$$|\alpha_2| = \angle M_1''M_2''M_3'' = \tan^{-1}\frac{M_1''M_3''}{M_2''M_3''}$$

In the tracing circle, assume its radius is R. The Line M_2M_3 is parallel to the X_L, so the length of $M_2''M_3''$ is equal to the length M_2M_3, i.e.,

$$|M_2''M_3''| = |M_2M_3| = r\sin\beta.$$

The length of M_1M_3 is

$$|M_1M_3| = r - r\cos\beta.$$

The line M_1M_3 is parallel to the YZ plane, so

$$|M_1'M_3'| = |M_1M_3| = r - r\cos\beta$$

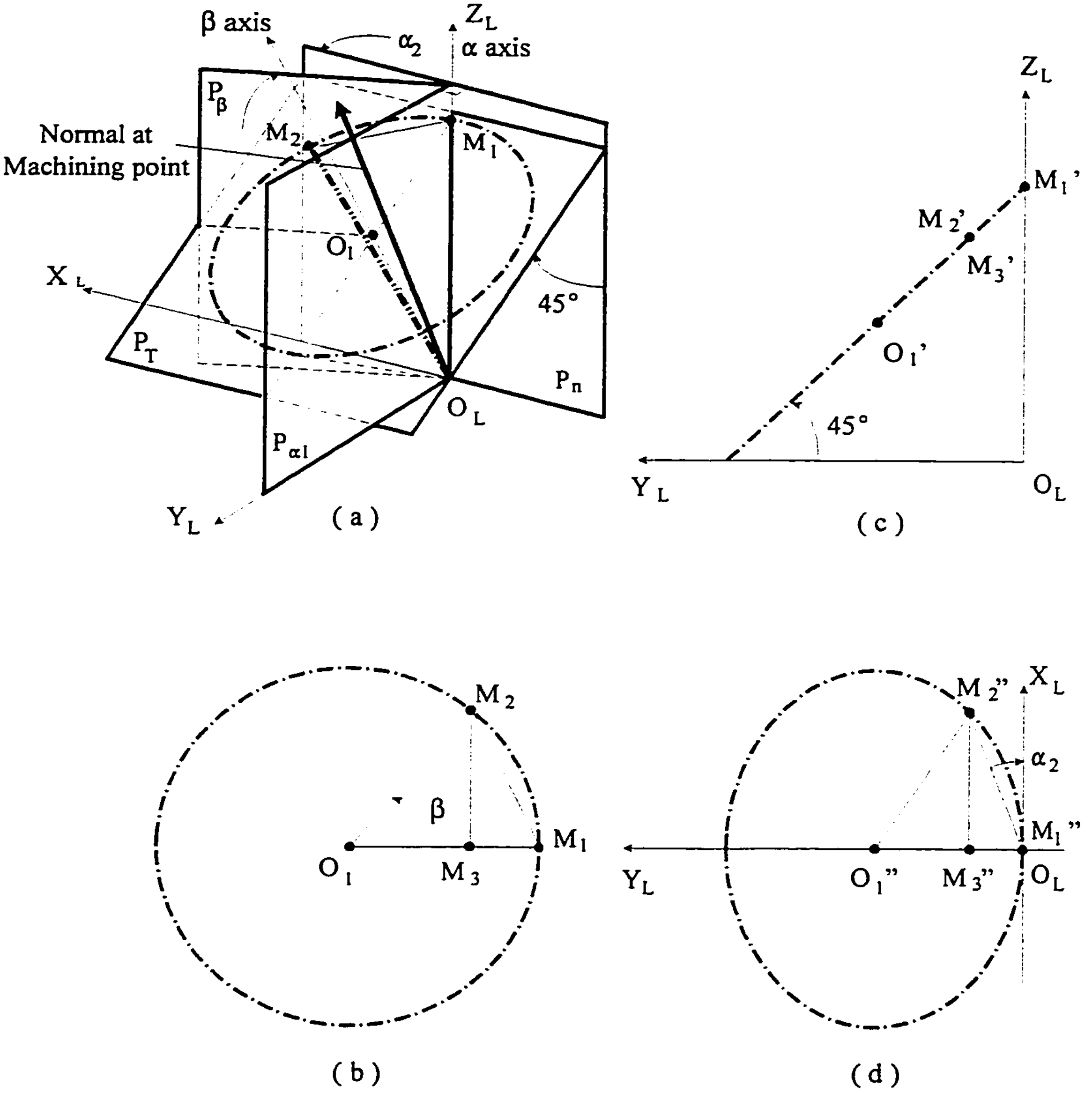


Figure B-3 Definition of the Angle α_2 When $\beta < 0$

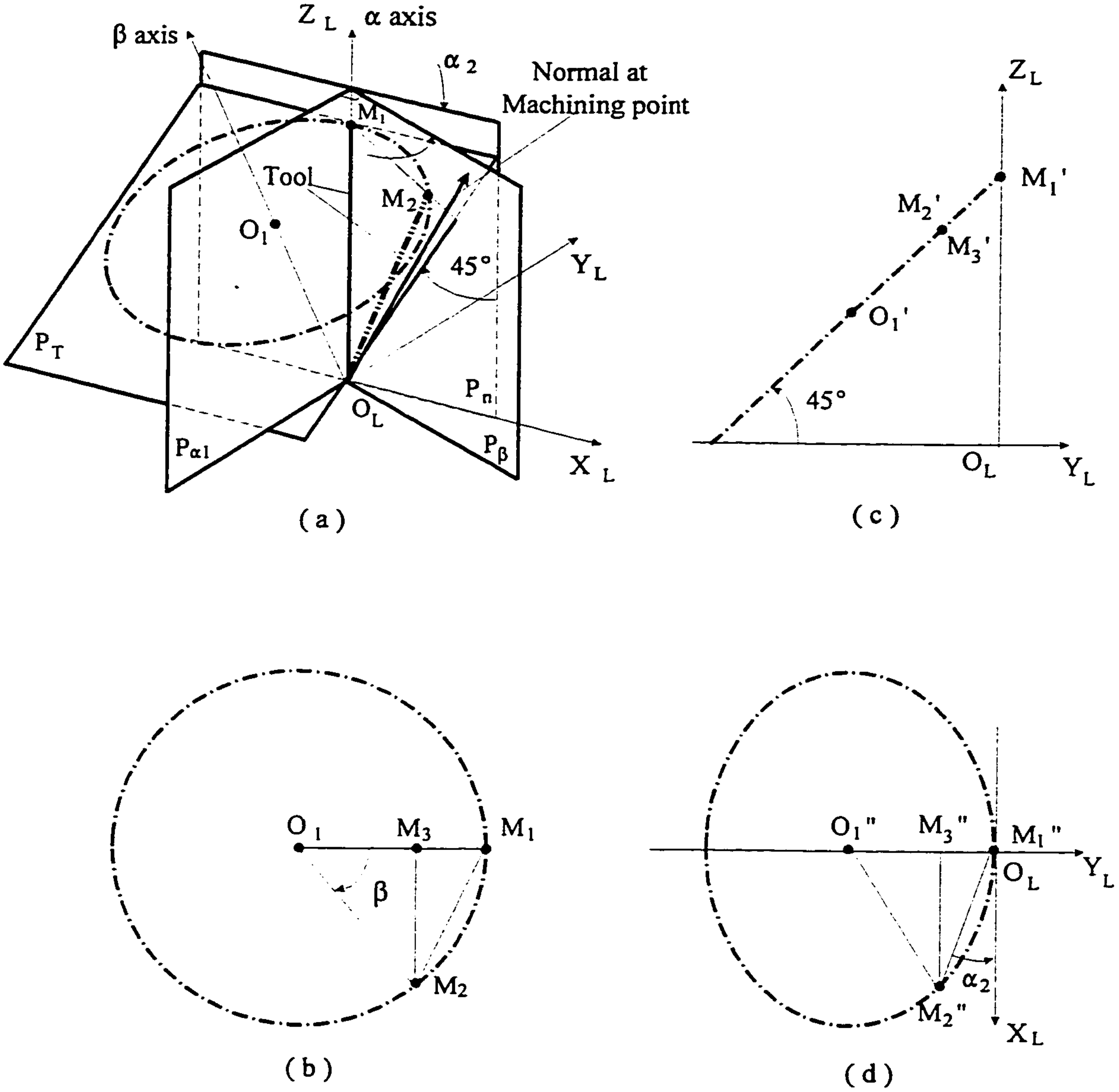


Figure B-4 Definition of the Angle α_2 When $\beta > 0$

Because the trace circle plane is inclined 45°, the length of the line M$_1$"M$_3$" is

$$|M_1''M_3''| = |M_1'M_3'| \cos45° = \frac{\sqrt{2}}{2} r (1 - \cos\beta)$$

$$|\alpha_2| = | \tan^{-1} \frac{M_1''M_3''}{M_2''M_3''} | = | \tan^{-1} \frac{1-\cos\beta}{\sqrt{2}\sin\beta} | \qquad \text{(B-7)}$$

The sign of α_2 depends on the sign of β. From the figure 5.11 and 5.12, if $\beta > 0$, $\alpha_2 < 0$, else,$\alpha_2 > 0$, i.e., α_2 and β always have the opposite signs. So,

$$\alpha_2 = - \tan^{-1} \frac{1-\cos\beta}{\sqrt{2}\sin\beta}$$

The total rotation of the tool about α axis is

$$\alpha = \alpha_1 + \alpha_2 = \theta + \tan^{-1} \frac{ny_r}{nx_r} - \tan^{-1} \frac{1-\cos\beta}{\sqrt{2}\sin\beta} \qquad \text{(B-8)}$$

3. Kinematic Equations

From the geometric analysis above, the final kinematic equations are:

$$R = \sqrt{X_r^2 + Y_r^2} \qquad \text{(B-1)}$$

$$\theta = 180° + \tan^{-1} \frac{Y_r}{X_r} \qquad \text{(B-2)}$$

$$Z = Z_r \qquad \text{(B-3)}$$

$$\beta = \cos^{-1} (2 \cos\gamma - 1) = \cos^{-1} (2nz_r - 1) \qquad \text{(B-5)}$$

$$\alpha = - \theta + \tan^{-1} \frac{ny_r}{nx_r} - \tan^{-1} \frac{1-\cos\beta}{\sqrt{2}\sin\beta} \qquad \text{(B-8)}$$

These results are the same as those from the algebraic approach discussed in Chapter 5.