Development of an Automated Robotic Deburring Workcell

Venkateswara Rao Ayyadevara

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

December 1995

0-612-47729-0

Abstract

Development of an Automated Robotic Deburring Workcell

Venkateswara Rao Ayyadevara

This thesis deals with the development of an automated robotic deburring workcell for refurbished components with intricate geometry. Such components have incurred a great deal of cost by the time they are ready for deburring. Hence, a dedicated machine like a deburring robot becomes an ideal tool to cut down costs and scraps, and to heighten competitiveness. Deburring of a refurbished workpiece involves establishing the location of the workpiece, establishing its geometry using surface digitizing and mathematical splining algorithms, extrapolating the reconstructed surface to get the edge profile, determining and executing the tool path.

The robot chosen for the workcell is YAMAHA Zeta-1 robot designed specifically for deburring. A displacement sensor with an accuracy of 0.004 mm is interfaced to the workcell. The limited computational ability of the robot controller and its closed architecture have necessitated the development of an external controller to coordinate the activities of the workcell. The robot controller neither provides for communication with external computing systems, nor does it allow interfacing external sensors. The novelty of the designed controller is in developing a technique to provide this communication capability. This is achieved by mimicking the motion of the joystick of a Teach Unit of the robot. The workcell controller is divided into an intelligent part, comprising a PC-parallel processor network and low-level interfaces to communicate with the robot controller and other devices in the of the trajectory of the robot. The workcell controller developed has a facility for communication of tool path to the deburring robot in real-time. A sampling period of 0.004 s is used for trajectory tracking of the robot and 0.252 s for the displacement probe. An open architecture format is adopted to allow future incorporation of additional sensors or other components into the deburring workcell. Experiments for trajectory tracking of straight line paths at a typical machining speed of 5 mm/s have shown a root mean square error of 0.02mm. This is within the repeatability of the robot (±0.1 mm). The path tracking error (measured using the displacement sensor), when the robot is controlled by the PC-transputer network through the Teach Unit, is compared with the error when the robot is controlled by the YAMAHA controller itself in conventional teach and playback mode.

Acknowledgements

I dedicate this thesis to my parents, brother and sister for their love and encouragement through all these years.

I am indebted to my supervisors Drs. R. Rajagopalan and R.M.H. Cheng for their invaluable guidance and encouragement and look forward to their support in all future endeavour.

I wish to thank Mr. Thai Luong, Adjunct Faculty for helping with the conceptualization of the work. I also wish to extend my thanks to Mr. Richard Hewitt, Hammond Machinery, USA and Mr. Shinobu Kawase, YAMAHA Corporation, Japan for their continued technical support.

I am grateful to Mr. Gilles Huard for not only helping with the development work but also for the innumerable ideas he gave during the course of this work. I wish to thank Mr. Karun Thanjavur, Mr. Lim Fung Peng and Dr. Mostafa Mehrabi for helping with the software development. I would like to acknowledge all the fruitful discussions I had with Dr. Mark Temple-Raston regarding my work. I also appreciate the time spent by Mr. Bruce Roussel helping with the experiments and proof reading my thesis.

I am grateful to all my colleagues and staff at Centre for Industrial Control for their help and for making my stay at Concordia very pleasant. Thanks are also due to Mr. Mahesh Balike, Mr. Girish Mahanta, Mr. Sriram Ganapathi, Mr. G.R. Ramesh and all other friends for the great time I had in Montreal.

This work was supported by Strategic Grant STRO134360 from NSERC awarded to Drs. R.M.H. Cheng and R. Rajagopalan and fellowships from J.W. McConnell Memorial Graduate Fellowship Committee and FCAR. Thanks are due to Mr. Robert E. Hammond, Hammond Foundation for providing US $ 45,000 towards the purchase of the robot.

Venkateswara Rao Ayyadevara

Table of Contents

List of Figures

List of Tables

Nomenclature

θ, R, Z Three joints of Zeta-1 robot analogous to cylindrical coordinate system.

α, β Wrist joins of Zeta-1 robot.

$\{X, Y, Z, A, B\}$ World Coordinate system of Zeta-1 robot.

$\{\theta, R, Z, \alpha, \beta\}$ Machine Coordinate system of Zeta-1 robot.

a, b Set of pulses with a phase difference of 90° emitted by optical encoders used for position feedback.

f_{CLK} Frequency of input clock for Intel 8254 timer.

X_i Any one of the World Axes $\{X, Y, Z, A, B\}$

Δx_i Desired displacement along X_i-axis.

T Sampling period for command pulse generation.

f Frequency of pulse generated for X_i-axis.

C Number of pulses required by teach unit for unit displacement along X_i-axis.

S Scaling factor between motion of joystick of teach unit and machining point of robot.

G Number of revolutions of the shaft of optical encoder for unit displacement along X_i-axis.

$N_{Encoder}$ Number of lines of the optical encoder.

M Frequency divisor downloaded to TIMER 2 chip to reduce time taken to implement new command pulse-rate.

V_{min} Smallest axis velocity of interest.

V_{max} Largest axis velocity of interest.

E_{RV} Relative error in axis velocity.

$S_D(k)$ Desired setpoint provided to path execution routine at the instant kT.

$\{x_D(k),y_D(k),z_D(k),a_D(k),b_D(k)\}$ World-axis coordinates of $S_D(k)$.

$S_A(k)$ Actual location of machining point of robot at instant kT.

$\{x_A(k),y_A(k),z_A(k),a_A(k),b_A(k)\}$ World-axis coordinates of $S_A(k)$.

$E(z)$ Instantaneous position error for X_i-axis.

$P(z)$ Pulse rate of square wave generated for X_i-axis.

K_P, K_I Position and integral gains of PI controller for X_i-axis.

K_C, T_C Critical gain and critical time period for PI controller for a World axis.

$\{\theta(k),R(k),z(k),\alpha(k),\beta(k)\}$ Machine coordinates corresponding to $S_A(k)$.

R_V Axis parallel to World Z-axis. Motion along this axis results in displacement along R-axis.

L_{MAX} Maximum distance travelled continuously by the robot when controlled by workcell controller.

Chapter 1

Introduction

The manufacturing industry is looking increasingly towards automating the deburring process for components with intricate geometry. Such components, like an impeller blade used in an aircraft engine, have incurred a great deal of cost (added value) by the time they are ready for deburring. It is far too expensive to allow a mistake to be made, through a lapse of concentration or fatigue, on part of a human operator. Thus, a dedicated machine becomes an ideal tool to cut down costs and scraps, and to heighten competitiveness.

1.1 Deburring Operation

Deburring is a secondary machining operation created to remove burrs created by the primary machining and metal removal operations [1.1]. This operation was required due to new turning, boring, milling, shaping, drilling and other special machinery being developed. Removal of burrs is essential from safety point of view and can be critical to ensure satisfactory performance in the case of mating parts and those which are subjected to large stresses (impeller blades used in an aircraft engine).

Till the early forties, brushes, buffs and files were manually held and applied to work using human strength [1.1]. The advent of bench and pedestal floor grinders and elementary portable hand-held air and electric power tools marked the start of mechanical deburring and surface finishing. A lot of work has been done through the fifties and the sixties in the area of filler materials, such as wire, synthetics, natural fibres, cloth, and abrasive compounds.

Research has also been done on coated and flexible abrasives and special carbide tools. Deburring has come to be recognized as an operation necessary to ensure good product quality.

1.2 Robotic Deburring

Deburring can be classified into 38 basic deburring operations [1.2] including mechanical cutting, grinding, brushing or blasting. The term robotic deburring is given to automation of any of these basic deburring processes. A robot is an ideal choice for this operation due to the ease of approach offered and the small machining forces involved. It possesses the repeatability and consistency required for deburring. Further, it is more economical than a five axis CNC machining centre [1.3] offering the same tool approachability characteristics.

The tools used for robotic deburring include brushes made of abrasive material, and chamfering tools. A detailed explanation of different types of tools used for deburring and their pertinence to different geometries, materials and machining operations is given by Gillespie [1.2]. Rotating files are the most commonly used tools for robotic deburring. Carbide cutters held rigidly by a robot can produce excellent surface finish unattainable by a human operator [1.2]. Conical countersinking tools, clothespin-type tool and elliptical deburring tools are used to deburr holes. Belt-grinders are used for rapid material removal while ensuring even surfaces for edges easily accessible from outside. Compliant tool holders are required when hard tooling is used [1.3]. These tool holders are responsible for keeping the tool on the part despite robot and system inaccuracies, accommodating burr variation and eliminating chatter. The tool holder is a sophisticated piece of machinery with built in passive or active compliance (axial or radial or both) and hence could cost as high as 15% of the cost of the robot itself.

Conventional tools like those described above cannot be used for deburring components of hydraulic systems. A deburring system developed by BOSCH for this purpose makes use of the Electro Discharge Machining (EDM) concept. Conventional schemes cannot be used because of the size of these components, material hardness and possibility of fine chips in the line, which would be detrimental to the system in which the hydraulic lines and valves are used.

1.3 Previous Work in Robotic Deburring

Previous work in automating the deburring process [1.4, 1.5] has addressed several machining and control issues. The Cleaning and Deburring Workstation (CDWS) developed at National Institute of Standards and Technology [1.6-1.8] is capable of deburring parts based on their CAD description and a graphically developed process plan. The CDWS consists of a workstation controller, two robots, PUMA 760 for deburring and a hydraulic robot Unimate 2000 for buffing and part handling, various quick change deburring tools, a rotary vice for part fixturing and a part transfer station. However, the capability of this workstation is limited to roughly prismatic aluminum and brass parts.

Kramer and Shim [1.9] have developed a robotic precision deburring system using an Adept One manipulator, providing simultaneous edge following and force control. The Adept One robot is fitted with a passive compliance end-effector instrumented for force measurement along two Cartesian axes and torque measurement about the third. The compliance is provided by a remote centre-of-compliance (RCC) device and force sensing is provided by light emitting diodes focussed on a photosensitive detector. Measured force signals are used to follow the edge of the part in addition to being used to regulate the contact force. However, steady state deviations as high as 40 % from the nominal normal force are observed when the part is deburred. Further, the tracking speed is limited by the long cycle time required to obtain sensor data and move the robot to the desired setpoint.

A system for automated edge finishing has been developed by Selleck and Loucks [1.10] for automated deburring, chamfering and blending of machined edges. The workcell is comprised of a five axis Adept One manipulator, process support components, a CAD program to program robot trajectories and deburring process based on CAD database of the workpiece, a supervisory computer, a vision computer, and a structured lighting inspection system. The vision system is used to determine the orientation of the cylindrical component using a notch on the inner wall of the part, to within one half of a degree. Structured lighting is required for on-line inspection of the formed edge. The cell is capable of brushing operations and of precision chamfers of 0.3 ± 0.1 mm on aluminum.

Her and Kazerooni [1.11] have used compliance control to address the problem of robotic deburring of parts with unknown geometry. Contact forces measured by a roller bearing mounted on a force sensor are used to calculate the normal vector to the part surface and generate compliancy in the robot. Knowledge of cutting forces generated by the cutter is used by the deburring algorithm to produce a stable metal removal process. Owing to the presence of a tracking mechanism, a priori knowledge of the geometry of the part is not required. When commanded to maintain a tangential force of 5 N and a constant material removal rate, deviations of about 20% from the nominal value are observed in the tangential cutting force.

Save for a few exceptions like the above mentioned work, it has generally been assumed that the dimensions of the workpiece match closely with the CAD database. While the assumption is valid for most cases, it does not hold good for refurbishing of used components. For example, reconditioned impeller blades and gas turbines used in aero-industries may have geometrical variations as large as 2.0 mm (0.08") from the original CAD data base. Additional variations could be caused due to tolerances of the original manufactured part and fixturing errors. Therefore, a correction method is required to take care of these variations. Cheng *et al* [1.12] propose one such method to establish the distorted surface and edge geometry, using a series of probing and mathematical splining techniques. While a dedicated coordinate measurement machine is very accurate at probing, it renders the idea of an automated workcell cost prohibitive. Instead, by using a robot, with satisfactory accuracy and repeatability for probing as well as deburring, the idea of an automated workcell becomes cost effective.

1.4 Requirements of a Deburring Workcell

Deburring of a refurbished component with intricate geometry involves :

(i) Establishing Location of the Workpiece : Even when the nominal location of the workpiece is known, preliminary probing operations are required to compensate for fixturing errors and thus, establish the actual location of the workpiece.

(ii) Determining Path of the Robot for Digitizing Surface of Workpiece : Once the actual location of the workpiece is established, the CAD database of the unused component can be used to construct the path of the robot while digitizing the surface of workpiece. The reason for digitizing the workpiece is explained in the next paragraph. The path for digitizing can be determined in two ways. One method is to generate a path off-line for the component when the workpiece is located at the origin of the coordinate system being used, and then translate the path to the actual location of the workpiece. The other method is to generate the path from scratch in real-time. In both cases, use is made of the CAD database. This is justified by the fact that the refurbished workpiece has undergone small deviations (while in use) from its newly machined geometry.

(iii) Digitizing the Workpiece : Since the refurbished workpiece has undergone some deformation from its original geometry described in the CAD database, the geometry of the surface which has to be polished or edges have to be deburred has to be established. This can be accomplished by reconstructing the geometry of the surface by digitization. This operation may be performed using a contact-type or a non-contact sensor attached to the wrist of the robot, transmitting data in real-time to the workcell controller.

(iv) Surface Reconstruction : The data collected from digitization of the workpiece is used by a splining algorithm to reconstruct the digitized surface.

(v) Extrapolating to Obtain Edge Profile : Once the geometry of the digitized section of the surface of the workpiece is known, extrapolation algorithms can be used to obtain the edge profile of the surface. This eliminates the need to digitize the whole surface of interest, a task which might be rendered impossible due to geometrical constraints imposed by the surface and the sensor stylus, and further is very expensive due to large process time.

(vi) Determining Tool Path : The edge profile of the refurbished workpiece along with parameters like depth of cut, feed rate, number of passes etc. are used to generate the tool path required to deburr to the desired edge profile.

(vii) Executing Tool Path : The tool path generated as explained above is executed by a deburring robot.

In order to perform steps such as (iii), (iv) and (v), a powerful computing platform is required. Once the digitizing path or the tool path has been determined, the same has to be communicated to the controller of the deburring robot. Further, a position sensor has to be interfaced to the workcell to be able to perform operations (i) and (iii).

1.5 Objectives and Scope

This thesis details the work done from September 1993 to December 1994 by the author as a member of a team working on an on-going project at the Centre for Industrial Control, Concordia University. It deals primarily with the development of an integrated workcell controller to :

(i) provide real-time control of the deburring robot to modify the tool path on the fly, as this ability is not present in the original controller.

(ii) digitize a given surface with the help of a sensor.

The robot chosen for the automated workcell is a Zeta-1 Robot developed by Yamaha Corporation specifically for edge deburring and surface machining tasks. As the robot and its controller are intended for use as a teach and playback device, the controller of the robot lacks certain features critical for real-time control of a workcell described in Section 1.4. It is incapable of executing floating point computations. It also lacks the ability to communicate with external displacement or force sensors and interpret the data obtained from them. Further, the tool path cannot be modified in real-time while the robot is in motion. Absence of these features makes it time consuming to debur refurbished components. This is because data obtained by sensors digitizing surfaces of a workpiece has to be downloaded to a PC first and analyzed. Then, the tool path has to be generated off-line and then downloaded to the controller and executed.

Since the workcell controller partially replaces the human operator, its capabilities determine the required level of human intervention. It is clear from the above discussion that the activities of the workcell, as described in the Section 1.4, cannot be monitored by the controller of the robot. The main contribution of this thesis is the development of an external workcell controller capable of performing all the computation-intensive tasks required to debur a refurbished workpiece, and downloading commands to the robot and other devices in the workcell in real-time. Communication of the trajectory in real-time to the robot controller is achieved by mimicking the motion of joysticks of its teach unit. This novel technique enables us to retain the original controller to perform servo-level control and take advantage of all its safety features. Further, no additional error is introduced due to inverse kinematics model as the path is specified to the robot controller in its world coordinate system. This technique can be useful while trying to control any machine with a closed architecture from an external computing system, without being forced to cannibalize its vendor supplied controller. An open architecture format is adopted to allow incorporation of sensors or other components into the deburring workcell.

The workcell controller is divided into an intelligent part and low-level interfaces. The intelligent part of the controller is responsible for both supervisory and process control. Servo level control of all joints of the robot is entrusted to the robot controller. Low-level interfaces are responsible for helping the intelligent part of the controller to communicate with the robot controller and other sensors. A position sensor is interfaced with the workcell controller as part of this work.

1.6 Thesis Outline

The various components of the workcell and the rationale for their selection are described in Chapter 2. Chapter 2 also describes the hardware aspects of the controller. Chapter 3 deals with organization of the system software. The results of trajectory tracking experiments are presented in Chapter 4. The conclusions and recommendations for future work are presented in Chapter 5.

Chapter 2

Components of the Deburring Workcell

2.1 Introduction

A skilled human operator is able to debur complex edges due to his/her ability to interpret the feedback (s)he receives, especially visual and tactile sensation. A robotic workcell includes sensors to at least partially substitute for human sensing capabilities. These sensors could include position sensors to determine the location and dimensions of the workpiece and force sensors to gage the machining forces exerted on it. Hence, the deburring workcell under discussion includes in addition to a deburring robot, a displacement probe to digitize surfaces and a workcell controller to coordinate the actions of different devices of the workcell. Position feedback is available in the form of joint encoder data. The 'intelligence' required to 'interpret' data collected from the sensors is provided in the form of software as part of the workcell controller.

2.2 Deburring Robot

The robot for a deburring workcell is chosen by taking a number of factors into account. In the present workcell, the robot is not only used to perform deburring operations, but also to digitize certain portions of the workpiece. Some of the requirements of such a robot are good repeatability and accuracy [2.1], control resolution, payload capacity, an open architecture controller and ease of programming.

2.2.1 Robots Used for Deburring in the Past

For initial research in robotic deburring, general-purpose industrial robots like PUMA 560 [2.2-2.5] and GE P-50 [2.6, 2.7] were applied to deburring tasks. These robots were developed for a number of tasks like assembly and part handling and could cope with only a limited interaction with the environment. This shortcoming was circumvented with the help of active and passive tool holders compensating for the inaccuracies of the robot and the use of force sensors to implement force sensing techniques. Other robots used in the past for deburring applications include a five axis SCARA type of robot, Adept One robot [2.8, 2.9], Panasonic HZL four-axis robot [2.10], GE S-700 six axis robot [2.11] and a four degree of freedom IBM SCARA 7576 robot [2.12].

2.2.2 The YAMAHA Zeta-I Deburring Robot

Recognizing the potential market for robotic deburring, new robots have been developed by Yamaha Corporation for edge deburring and surface polishing tasks. One such robot, the YAMAHA Zeta-1 [2.13], is chosen for the deburring workcell. It is a five axis manipulator (Figure 2.1) with three joints (θ, R and Z) analogous to a cylindrical coordinate system and two wrist joints (α and β). The lambda configuration extending from the main cylindrical column (Figure 2.1) enhances the rigidity of the robot.

Another interesting feature is the location of the *machining point* (point of contact between the tool and the workpiece) at the intersection of the two wrist axes : α and β. Owing to this unique configuration of the two wrist joints, it is possible to change

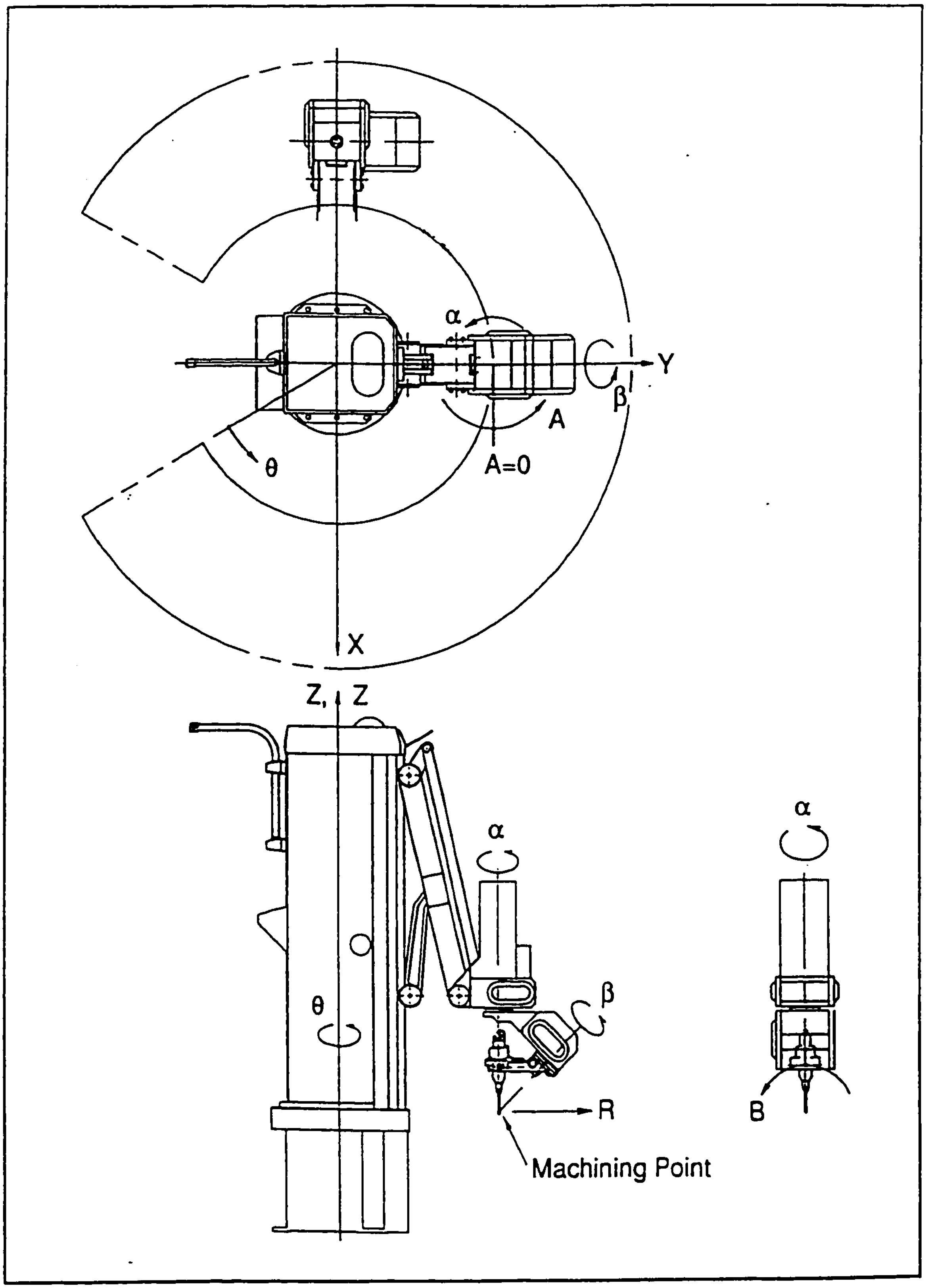

Figure 2.1 YAMAHA Zeta-1 Deburring Robot [2.13]

the orientation of the tool, while keeping the position of the *machining point* constant. This gives more freedom in choosing the angle of approach of the tool with respect to the workpiece. This robot is reported to have a repeatability of ±0.1 mm (±0.0039") and can carry a maximum movable load of 30 Kg [2.13].

The controller of the robot accepts instructions via a teaching station or from a script file that is downloaded by means of an RS-232 serial communication link. As a result, the motion sequence for deburring of a component can either be taught to the robot using the teach unit, or a sequence of points computed off-line from the geometry of the component can be downloaded from a computer. Irrespective of the method of programming used, points entered in World Coordinate System {X, Y, Z, A, B} are transformed to Joint space or Machine Coordinate System {θ, R, Z, α, β} by the controller of the robot. Both these coordinate systems are shown in Figure 2.1.

Special mention should be made of the teach unit which, by minimizing the programming effort involved, helps a skilled operator make an easy transition to a robot operator/programmer [2.14}. The teach unit, shown in Figure 2.2, consists of five joysticks each of which is used to activate motion of the robot along one axis of the World Coordinate System (Figure 2.1). More than one joystick can be used in synchronization to achieve the desired motion of the *machining point* of the robot in space. The teach unit also includes a keyboard display used to write programs listing a sequence of points in space and operating commands for the tool. Facility is also provided for off-line adjustment of system parameters like *machining point* velocity, maximum acceptable load and scaling factor between joystick motion and actual robot motion. However, it should be emphasized that the programming features provided on the teach unit are rather rudimentary. Unlike the VAL-II controller of a PUMA 560 robot [2.15], there is no operating system provided. A set of user-defined parameter registers are used as variables. No facility is provided for performing floating point operations. Interaction with the environment is limited to a set of I/O ports of the ON-OFF type.

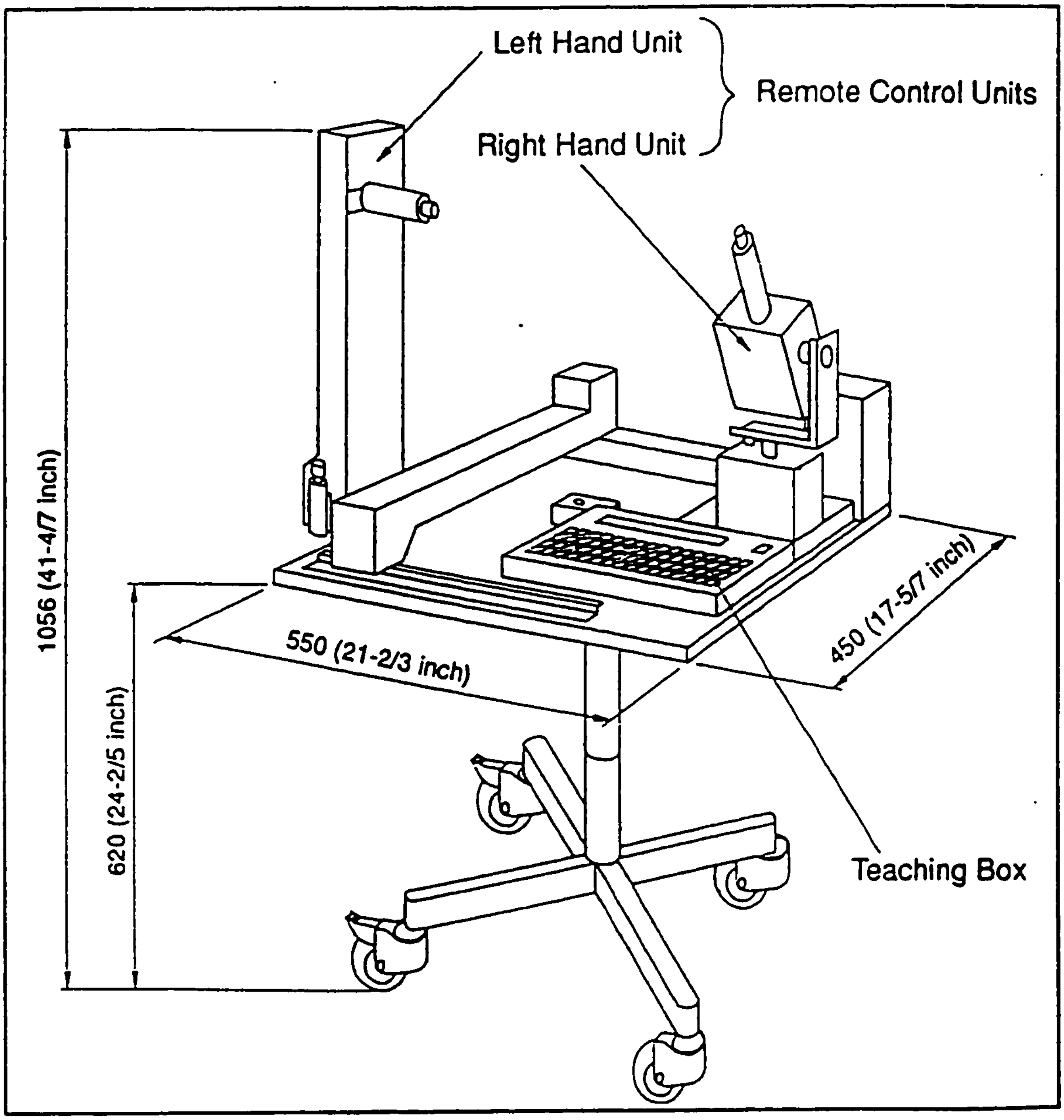

Figure 2.2 Teach Unit for Zeta-1 Deburring Robot [2.13]

For a component with complex geometry, the operator is required to record some key points from the workpiece using joysticks and send them to a computer to generate all the intermediate points of the tool path. Finally, the tool path is downloaded to the controller for execution. On the positive side, this procedure obviates the need for the operator to be an expert in programming the robot. On the other hand, limitations in programming features of the teach unit and the inability to modify the path in real-time poses some problems for deburring refurbished components. These problems are addressed in Section 2.4 while explaining the role of the workcell controller, especially the interface developed to communicate with the controller of the Zeta robot.

2.3 Surface Digitization

Digitizing of the surface of a refurbished workpiece is required to establish the actual geometry of the same before polishing it or deburring the edge. This step is necessitated by the small but significant deformation undergone by the component while in service. These deformations preclude the use of the CAD database of a newly machined component to construct the tool path. However, for digitizing, the CAD database of the original component is used. The data collected from the surface is used to establish the surface and extrapolate to obtain the desired edge profile. Further, the position sensor used for digitizing can also be used to establish the exact location of the workpiece.

Sensing devices have been used in the past to establish the geometry of the component to be deburred. These devices could be either of the contact or non-contact type. Instances of a non-contact type sensor is a laser based system, camera vision and inductance sensors. Some examples of contact type sensors are capacitive type position sensors and force sensors. Selleck and Loucks [2.9] used a structured lighting system to establish the orientation of a cylindrical workpiece of known geometry by looking for a notch in the inside diameter of the part. Her and Kazerooni [2.16] used a roller bearing mounted on a force sensor to track the contour of the workpiece while deburring two-dimensional planar parts with unknown geometry. The various factors which are taken into account while selecting a contact or non-contact position sensing device are :

(i) Resolution and Accuracy : These qualities are required to ensure that the errors due to probing operation are small enough to permit the accurate reconstruction of the surface of the workpiece. In the current application, it is desired that the accuracy of the probe should be at least an order of magnitude smaller than the repeatability of the robot which is ±0.1 mm (0.004 inches). Hence the accuracy of the probe should be as good as ±0.01 mm or better. The motivation behind using the repeatability of the robot is that the final edge profile being constructed and the tool path thus computed is to be finally executed by the robot. In the absence of any data regarding position accuracy of the robot, the repeatability of the robot is being used as a reference.

(ii) Range of measurement : The range of the probe used should be large enough to accommodate the errors experienced in execution of the path by the robot and the uncertainty in the dimensions of the used component. For example, in the case of refurbished aircraft engine impeller blades, the deviation from the CAD database can be as large as 2.0 mm (0.08 inch) and welds and pits could be present on the surface.

(iii) Accessibility : The physical dimensions of the probe determine whether the probe can access remote portions of a geometrically intricate surface. For a contact-type sensor, if the probe itself cannot access the interior of the part, an extension to the probe stylus can be fabricated. This extension will affect the accuracy of the probe.

(iv) Speed of Data Collection : For a fixed number of data points to be collected, the time taken for probing the surface is determined by the speed of data collection and the speed of the robot itself. Hence the process time could be reduced by using a probe which can collect data faster.

(v) Sensitivity to Noise : The robustness of the probe to electrical noise is very important because the workcell is being developed for a shop floor environment where it is likely to be surrounded by heavy-duty machines. Dust particles normally present in a shop floor environment can affect the performance of non-contact sensors, specially vision or laser based sensing systems.

(vi) Ease of Interfacing : In order to reduce the development time for the deburring workcell, it is necessary to choose a probe requiring minimum effort to develop an interface with the workcell controller.

(vii) Cost : The price of the probing system is very important in determining whether the workcell is cost-effective.

Two types of probes are considered. One is a contact type sensor called Digimatic

Indicator (shown in Figure 2.3) made by Mitutoyo Inc. Linear displacement of a stylus is converted by an encoder of capacitance type to a digital signal [2.17]. This signal is displayed and can be tapped for further processing. The second probe considered is a non-contact type inductive sensor (shown in Figure 2.4) made by Kaman Inc [2.18]. Impedance variation, caused by eddy currents induced in a conductive target by a coil, are translated by a signal conditioning circuit into a usable analog signal. The analog signal is then digitized and fed to the micro-controller.

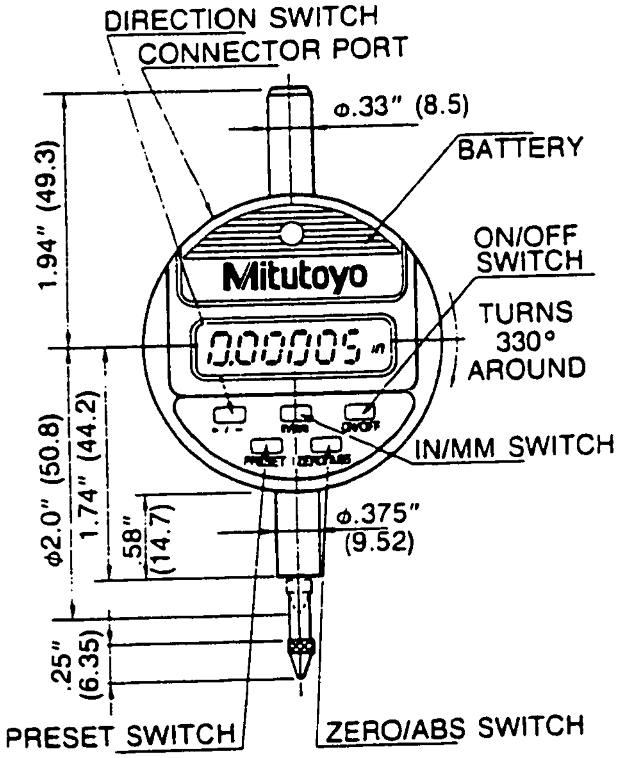

Figure 2.3 Digimatic Indicator

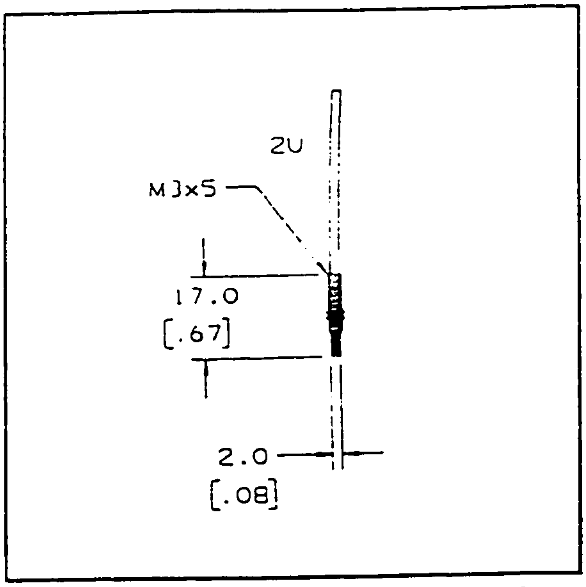

Figure 2.4 Inductive Sensor [2.18]

Both probes satisfy the accuracy and resolution requirements. The range of the inductive sensor (2.0 mm) just equals the maximum permissible deviation of 2.0 mm from the nominal CAD database for a used component. The range of the Digimatic Indicator is 12.27 mm and hence meets the range criterion comfortably. Speed of data collection for the inductive probe is 5 times as fast as the Digimatic Indicator. As far as accessibility is considered, the physical dimensions of the inductive probe are much smaller than that of the displacement probe. For probing complex surfaces like that of an impeller blade, an extension would have to be made for the contact type Digimatic Indicator. To directly collect data from the Digimatic Indicator to a computer, either a vendor-supplied interface box can be used to obtain the data through an RS-232 port or shift registers can be cascaded to directly read the data. The inductive probe is supplied with an interface software executing on the PC. However, since all the control software for the workcell is being written for INMOS transputers (explained in Section 2.4.1), this interface software is not useful. It should also be noted that the inductive probing scheme is applicable only to conducting surfaces. While a non-contact sensor does not face any wear and tear problems, the Digimatic Indicator should be fitted with a wear resistant tip for extensive use.

Finally, one of the main advantages of using the Digimatic Indicator is that the signal generated is dependent on only the movement of the stylus of the probe. In the case of the inductive probe, the high resolution and the fact that the probe is of the inductive type could introduce significant errors in the signals transmitted by it. Further, it is required that the inductive probe be normal to the surface with a maximum cosine error of 2% for reliable data measurement. This is not possible due to uncertainty in the geometry of a reconditioned workpiece. It should also be noted that the range of the Digimatic Indicator is very large enabling easier path planning for curved surfaces. Owing to the above mentioned reasons, this probe has been chosen to digitize surfaces for the robotic deburring workcell.

The displacement sensor is attached in place of the tool, acting as the end-effector of the robot. An interface has been designed to transmit data from the probe to the transputer network. The interface is explained in Section 2.4.3. Points obtained by probing the workpiece are used to reconstruct surface of the workpiece with the help of splining algorithms developed at Centre for Industrial Control [2.19]. Integration of control software with the above mentioned routines is discussed in the next chapter.

2.4 Workcell Controller

An integrated controller is essential for coordinating the activities of such a versatile workcell. At the ***supervisory level*** [2.21], the workcell controller is required to interact with the user, perform surface reconstruction and path planning for the robot. At the ***process level***, it has to control the trajectory tracking of the robot and communicate with the sensors present. At the ***servo level***, the individual joints of the robot have to be controlled.

Since the workcell controller partially replaces the human operator, its capabilities determine the required level of human intervention. As discussed in Section 2.2.3. the controller of Zeta-1 is incapable of executing floating point computations. Further it lacks the ability to communicate with external displacement or force sensors and interpret the data obtained from them. Moreover, the tool path cannot be modified in real-time while the robot is in motion. Hence the functions of the workcell controller, as described in the previous paragraph, cannot be implemented directly on the controller of the robot. Absence of means for real-time modification of the tool path makes it time consuming to debur refurbished components. This is because the digitized data has to be analyzed off line on a PC and the tool path has to be downloaded off-line to the controller and executed. Instead, by having an external workcell controller capable of performing all the computation-intensive tasks required to debur a refurbished workpiece, as well as downloading commands to the robot and other devices in the workcell, the limitations of the robot controller can be avoided.

The workcell controller, therefore, is divided into the intelligent part and the low-level interfaces. The intelligent part of the controller is responsible for both supervisory and process control. Servo level control of all joints of the robot is entrusted to the YAMAHA controller. Low-level interfaces are responsible for helping the intelligent part of the controller to communicate with the YAMAHA controller and other sensors such as the

Digimatic Indicator. A schematic drawing of the workcell controller is given in Figure 2.5. An open architecture format has been adopted to allow future incorporation of additional sensors or other components into the deburring workcell.

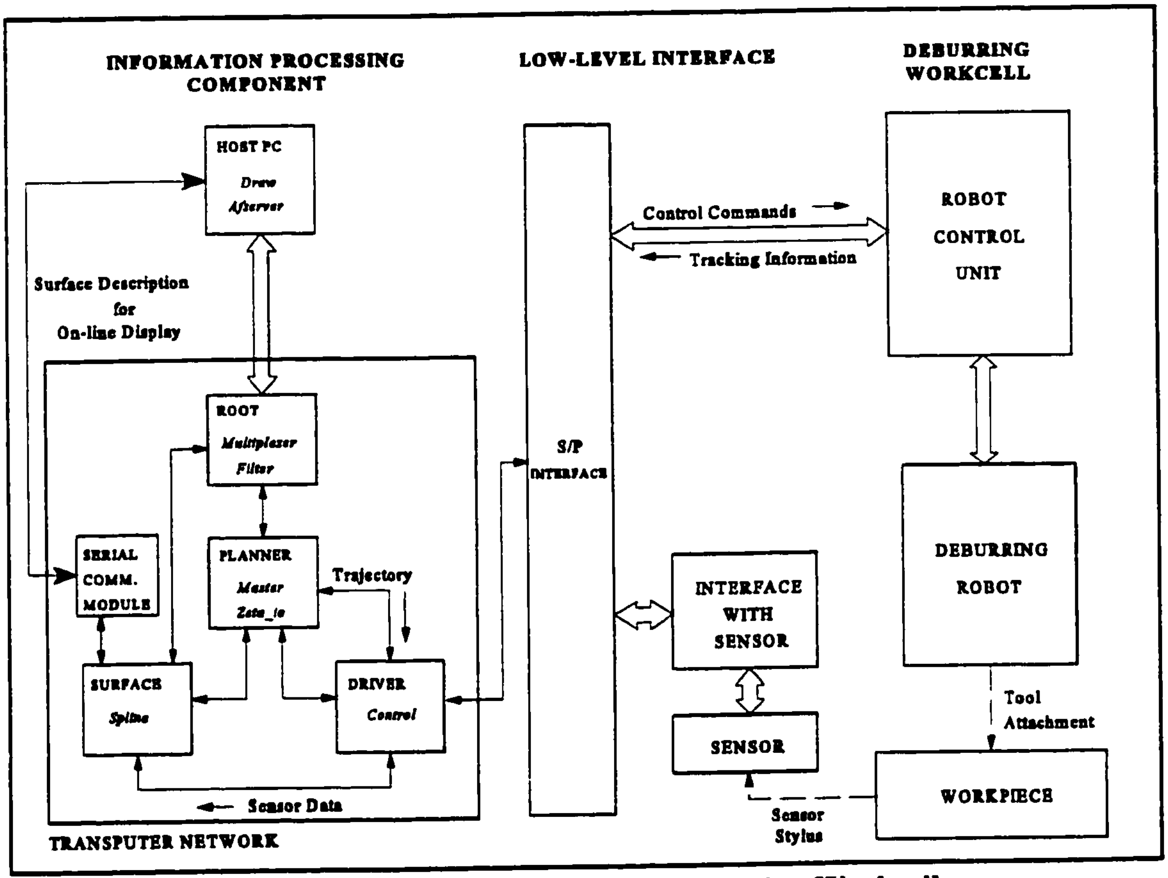

Figure 2.5 Automated Robotic Deburring Workcell

2.4.1 Information Processing Part

The information processing part of the controller is composed of a PC-Parallel Processor network. Parallel processing capability is required for three reasons. First, while surface reconstruction is being performed for one region of the workpiece, the controller can direct the robot to collect more points from some other portion of the surface. This minimizes process time. The second reason is that the multi-tasking and multi-threading ability of the processors enable the trajectory control of the robot simultaneously with the analysis of the raw data obtained from the position sensor. Finally, powerful processors capable of parallel computing offer a good platform for future implementation of a dynamics-based control scheme, to enhance the precision of the robot.

The parallel processors being used are INMOS T-800 transputers. The transputers are mounted on a TMB008 motherboard which fits into a slot on the motherboard of the host IBM PC (labelled HOST in Figure 2.5). HOST is responsible for providing the transputers access to disk and screen I/O facilities and to serve as a platform for the graphics visualization program under development by another member of the workcell development team [2.20]. The transputer labelled ROOT (Figure 2.5) is responsible for communication between the host PC and the transputers mounted on TMB008 motherboard. Path planning is performed by the transputer labelled PLANNER and trajectory control functions are performed by the transputer labelled DRIVER. The transputer labelled SPLINE in Figure 2.5 is responsible for surface reconstruction and tool path determination. Communication with any component of the workcell is channelled through DRIVER via a Serial-to-Parallel interface (S/P INTERFACE in Figure 2.5) [2.22]. This interface is required to route data transmitted serially from one of the links of the transputer to appropriate data, address or control buses connected to the chips constituting the low-level interface. The software routines which execute on the PC-transputer network are discussed in Section 3.2.

2.4.2 Control of the Deburring Robot

2.4.2.1 Options Available for Control of Robot

Three options are open for controlling the deburring robot, as shown in Figure 2.6. The first option is to replace the existing controller with a controller developed in-house. This controller would be a part of the workcell controller, receiving commands from the PC-transputer network, and performing low-level control of the servomotors powering the joints of the robot. One advantage would be the removal of a potential bottleneck caused by the slow processor used by the YAMAHA controller. However, owing to the fact that the Zeta-1 robot controller is developed specially for deburring tasks, it is not advisable to spend a considerable amount of time to develop an in-house controller replacing it.

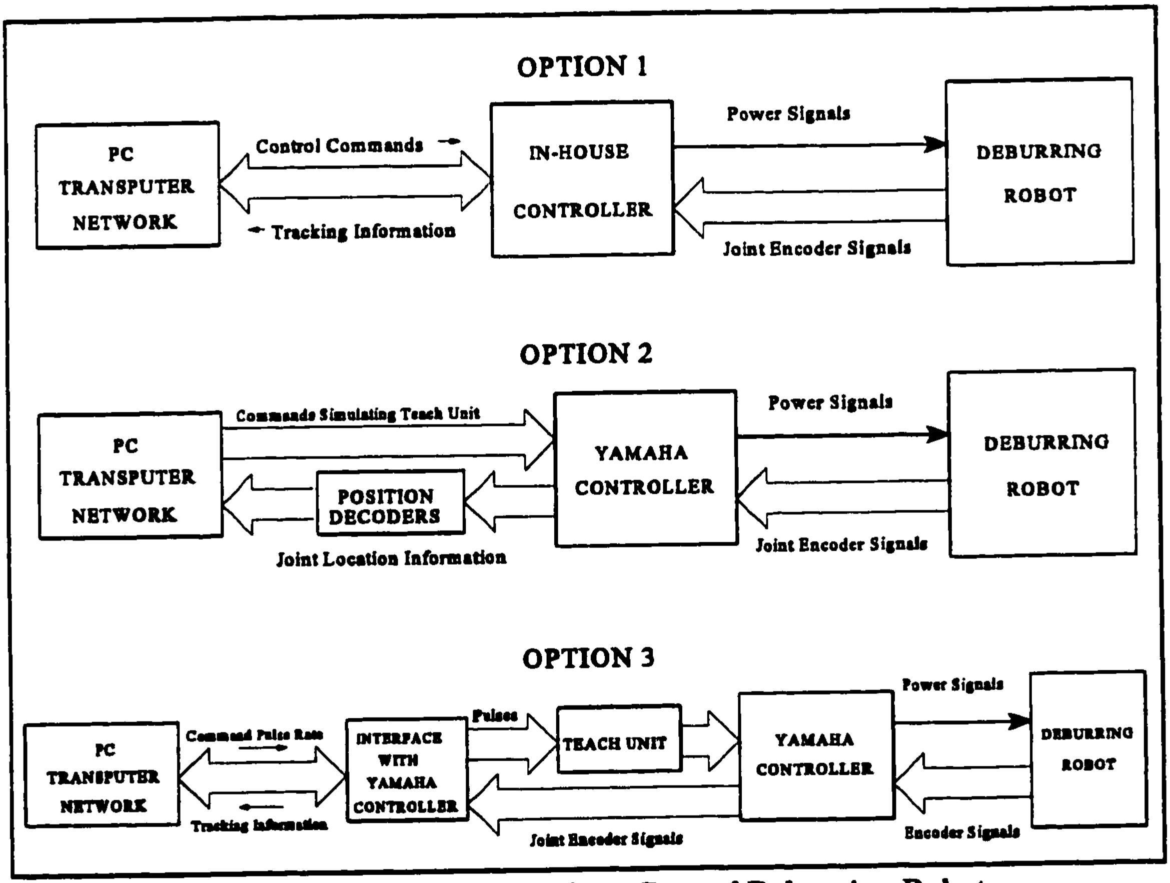

Figure 2.6 Options Available to Control Deburring Robot

Since, there is no facility provided to communicate the trajectory on-line to the YAMAHA controller from a supervisory controller, two other options are available. These are also illustrated in Figure 2.6. Option 2 requires the PC-transputer network to simulate the function of the teach unit. The teach unit is normally used by the operator, after setting the YAMAHA controller in *TEACH* mode, to teach the robot the tool path, and provides access to a number of parameters governing the performance of the robot and the tool. The learned tool path can then be executed by the robot by setting the controller in *AUTO* mode and calling the recorded sequence of points. By simulating the teach unit, trajectory and other relevant details can be directly fed to the robot controller from the Information Processing Part of the workcell controller without any manual intervention. However, this requires knowledge of the communication protocol between teach unit and robot controller. Lack of the above mentioned information has forced us to abandon this option.

Option 3, which has been implemented for the present deburring workcell, will now be discussed in detail. Motion along a desired trajectory is achieved by mimicking the motion of joysticks of the teach unit of the robot after setting the controller in *TEACH* mode. The teach unit, as explained in Section 2.2.3, has five joysticks dedicated to one World axis each (X, Y, Z, A & B in Figure 2.1). When the joystick is moved, two sets of pulses *a* and *b*, with a phase difference of 90°, are generated by a optical quadrature encoder and fed to the YAMAHA robot controller. Based on the number of pulses generated, the *machining point* of the robot is moved along the corresponding World axis with a velocity corresponding to the pulse rate. The two sets of pulses are required to distinguish the direction in which the robot is to be moved. Taking X-axis as an example, if *a* is leading *b* by 90°, the *machining point* of the robot is moved along positive X-axis. If *b* is leading *a* by 90°, the *machining point* of the robot is moved along negative X-axis. A special interface, shown in Figure 2.7, is developed to generate the set of pulses *a* and *b*, instead of the joysticks and the pulses are fed to the teach unit. Commands from the transputer CONTROL are downloaded to the special interface (in Figure 2.7) once every sampling period to make the set of pulses correspond to the desired trajectory of the robot. This interface is required as there is no provision for a supervisory controller to communicate trajectory set points to the YAMAHA controller.

2.4.2.2. Hardware for Pulse Generation

The special interface mentioned in Section 2.4.2.1 is responsible for communicating the trajectory of the machine point along all World axes. The hardware used for generation of pulses for a general World axis X_i is shown in Figure 2.8. Commands from transputer DRIVER are communicated via the Serial to Parallel (S/P) Interface shown in Figure 2.7 to the chips constituting the low-level interface. The block CHANNEL in Figure 2.8 is comprised of INTEL 8254 timers [2.23], TIMER 1 and TIMER 2, which can be programmed to operate in a variety of modes. In the present application, they are set to Square Wave Mode. In this mode, a square wave of a desired frequency f is generated by the timer chip by downloading a 16-bit *divisor word*, which is obtained by dividing the frequency of the input clock for the 8254 timer f_{CLK} by the desired frequency f. The square wave generated by CHANNEL is fed as input to a Programmable Array Logic (PAL in

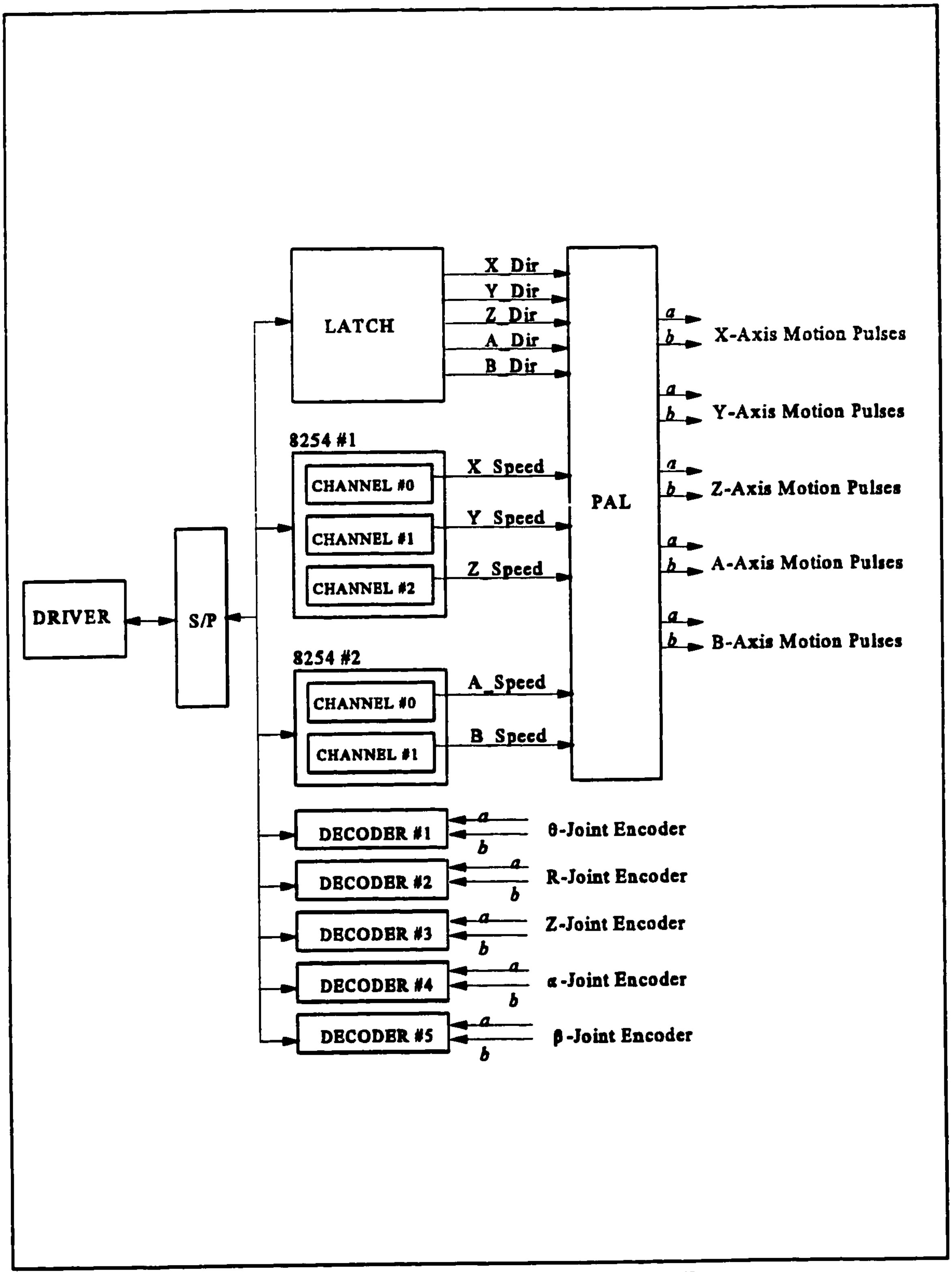

Figure 2.7 Interface with Robot Controller

Figure 2.8) device. PAL consists of a number of AND and OR gates and registers and serves as a programmable replacement for TTL circuits. They can be configured to perform logic functions [2.24]. Since two sets of pulses *a* and *b* are required with a phase difference of 90° between them, PAL is programmed to generate the two sets of pulses, using the square wave generated by CHANNEL. A latch (LATCH in Figure 2.8) is used to specify the direction of motion for the World axis. This information is used by PAL to decide whether *a* should lead *b* by 90° or vice-versa.

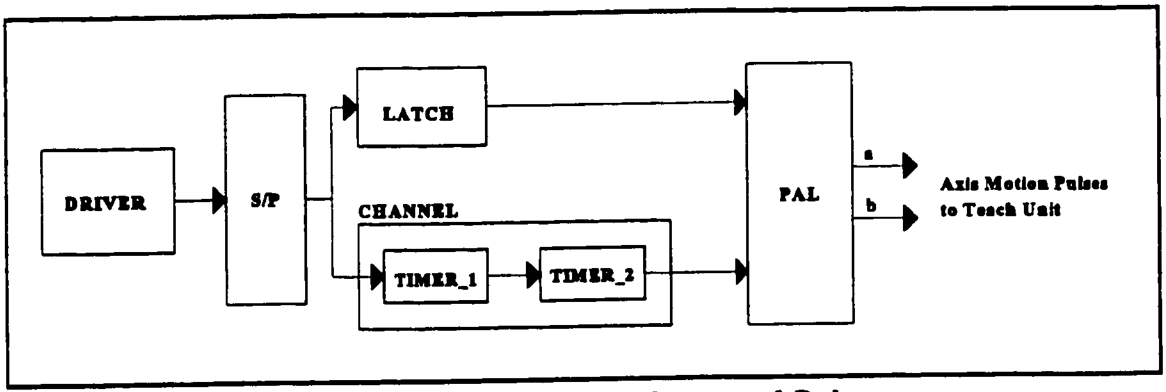

Figure 2.8 Generation of Command Pulses

Once the desired displacement Δx_i along (or about) a general World Axis X_i is determined at the beginning of a sampling period (T s), the corresponding number of pulses P to be fed to the teach unit during that sampling period, and hence the frequency of pulse generation f is computed as shown (for axis X_i) :

$$f \cdot \frac{P}{T}\ Hz \qquad (2.1)$$

where

$$P \cdot C\,(\Delta x_i) \qquad (2.2)$$

where

$$C \cdot S\,G\,N_{Encoder} \tag{2.3}$$

where

C (pulses/unit displacement), for World axis X_i, is the number of pulses required by teach unit for unit displacement. For X, Y and Z axes, unit displacement refers to 1 mm. For A and B axes, unit displacement refers to 1°.

S is the scaling factor between motion of joystick of teach unit and machining point of robot and is the same for all axes.

G (revs/unit displacement), for World axis X_i, is the number of revolutions of the shaft of the optical encoder for unit displacement.

$N_{Encoder}$ is the number of lines of the optical encoder used for axis X_i.

Every sampling period, CHANNEL is programmed to generate a square wave with a frequency 2f. This square wave is used by PAL to generate two sets of pulses of frequency f with a phase difference of 90° and the pulses are fed to the teach unit.

Every time a new command frequency is downloaded to a 8254 timer, it switches over to the new frequency at the end of the current half pulse being generated [2.23]. As a consequence, there could be a delay as large as half the time period of the current pulse being output by the 8254 timer, in responding to a command from the PC-transputer network. When CHANNEL is comprised of only one timer, for a desired velocity of 1 mm/s, with an input clock frequency of 1 MHz, it could take as long as 0.032 seconds to implement a new command. Therefore, to reduce the delay involved in implementing a new command, CHANNEL is comprised of two 8254 timers (TIMER 1 and TIMER 2) cascaded in series. The output of the TIMER 2 is fed as an input to PAL. TIMER 1 is programmed to output a square wave with a frequency equal to 2M times the desired frequency f_U, where M is a scalar. This timer receives a new command pulse-rate to implement, once every sampling period. TIMER 2 is responsible for dividing the frequency of the incoming signal from TIMER 1 by the same scalar M to produce a square wave of frequency $2f_U$, fed to PAL. Once the constant M has been downloaded to TIMER 2 at the beginning of the tool path execution, no further communication is necessary. Since TIMER 1 is producing a much faster square wave signal, it is able to respond to a new command in a shorter interval of time.

The choice of the scalar M depends on the range of velocities of interest for the World axes, the frequency of the clock input to TIMER 1 (f_{CLK}) and the permissible error allowed in implementing a desired velocity. As explained before, in order for CHANNEL to implement a desired frequency, a 16-bit divisor word is downloaded to TIMER 1. This divisor word is obtained by dividing f_{CLK} Hz by the desired frequency. Hence the smallest frequency which can be implemented by CHANNEL is limited by the fact that the largest divisor which can be downloaded to the 16-bit command register is 65536. Hence, the smallest frequency which can be implemented by TIMER 1 is (f_{CLK} / 65536) Hz and by CHANNEL as a whole is [f_{CLK} / (65536M)] Hz. Hence for a fixed f_{CLK}, M should be as large as possible to accommodate the smallest velocity of interest.

$$M \geq \frac{f_{CLK}}{(65536\ V_{min}\ C)} \tag{2.4}$$

where

V_{min} (mm/s for X, Y, Z and deg/s for A, B) is the smallest axis velocity of interest.

C is obtained from Equation 2.3.

On the other hand, in order to implement a desired frequency f for CHANNEL as a whole, TIMER 1 has to be programmed for the frequency fM Hz. Therefore, the magnitude of the 16-bit divisor, a whole number, is $[f_{CLK} / (fM)]$. Hence, the largest frequency, and thus desired axis velocity, which can be implemented is limited by the permissible relative error caused by the fact that the divisor is a whole number. In order to minimise this error, the magnitude of the divisor, corresponding to the velocities of interest, should be as large as possible. This constraint implies that f_{CLK} should be as large as possible and further, places an upper bound on M.

$$M \leq \frac{2 f_{CLK} E_{RV}}{C\, V_{max} (1 - E_{RV})} \tag{2.5}$$

The largest clock frequency, the timer chips are designed for is 8 MHz and this is the value of f_{CLK} in the present interface. From Equation 2.4, for $V_{min} = 0.1$ mm/s for X, Y, Z axes and 0.1 °/s for A, B axes, it is found that M should be at least as large as 110. Allowing for a maximum relative error in velocity of 1%, using Equation 2.5 and $V_{max} = 80$ mm/s for X, Y , Z axes and 100°/s for A, B axes, it is found that M should be at most 128. The performance of each axis can be fine-tuned by choosing a distinct value for M for every World axes. However, in this interface, an identical value of M was found to give satisfactory performance for the range of velocities tested for.

2.4.2.3 Position Feedback

The actual position of the *machining point* of the robot is required to calculate the command pulse rate for the next sampling period. The sampling time and the control law are discussed in Section 3.3.2.2. The actual position is calculated with the help of 32-bit decoders reading signals from joint encoders of the robot. Once the positions of the joints are known, it is possible to compute the location of the *machining point* in World coordinates using the kinematic model given in Chapter 3. This is necessary for incorporating feedback in the control law and computing the points probed on the surface of the workpiece.

2.4.2.4 Limitations of the Option Chosen

One drawback of the above option is that there is no access at present to several deburring parameters like maximum permissible overload and maximum number of repeat passes. These parameters enable the repetition of sections of the tool path to gradually reduce exceptionally large burrs and thus preventing any damage to the tool or the tool holder of the robot [2.13]. Hence extra care has to be taken to ensure that the control software executing on the transputers is able to handle overload situations. Further, a potential bottleneck could be caused by the slow processing speed of the processor of the teach unit and the processor in the robot controller.

2.4.3 Hardware Interface for Probe

The probe is mounted in place of the deburring tool. There are two options open while interfacing the probe to the workcell controller. One method is to communicate with the probe through a vendor-supplied interface box through a RS-232 serial port. For this option to be implemented, a special serial communication module is added to the transputer motherboard and is used to communicate with one of the transputers. This option did not work out satisfactorily owing to some problems with the serial communication module mounted on the transputer motherboard.

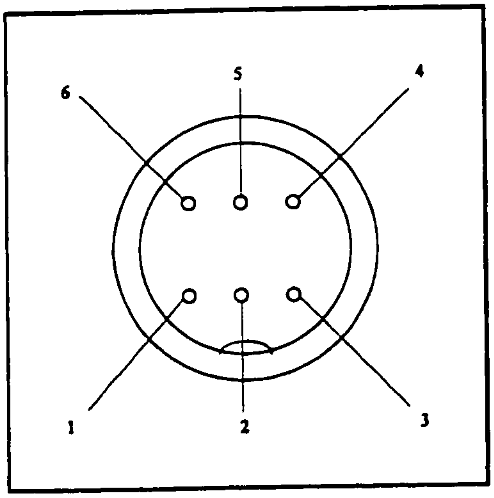

Figure 2.9 Pin Assignment

Table 2.1 Connector Pin Assignment

Pin No.	Signal	I/O	Description
1	GND	-	Signal ground
2	DATA	O	Measurement Data
3	CK	O	Clock for Data Transmission
4	N.C.	-	No Connection
5	/REQ	I	Request for Data Transmission
6	N.C.	-	No Connection

The second option, which is implemented, is to directly communicate with the Digimatic Indicator. The pin out for the probe connector is shown in Figure 2.9 and Table 2.1. The timing details are shown in Figure 2.10 and Table 2.2. Points are collected from the probe every 250 milliseconds by sending a request to the probe and using a CLOCK output to clock shift registers cascaded to serially collect data from the probe. The large time period of 250 milliseconds is required due to the initial set up time required by the probe to start the CLOCK output (up to 200 milliseconds in Figure 2.10) followed by serial transmission of 52 bits of data, for every request sent to the probe. The displacement of the probe stylus coupled with the knowledge of the actual position of the *machining point* of the robot (obtained from the forward kinematics of the robot and decoders) gives the location of the probed point. The speed of collection of the points is limited by the processing capabilities of the circuitry of the probe.

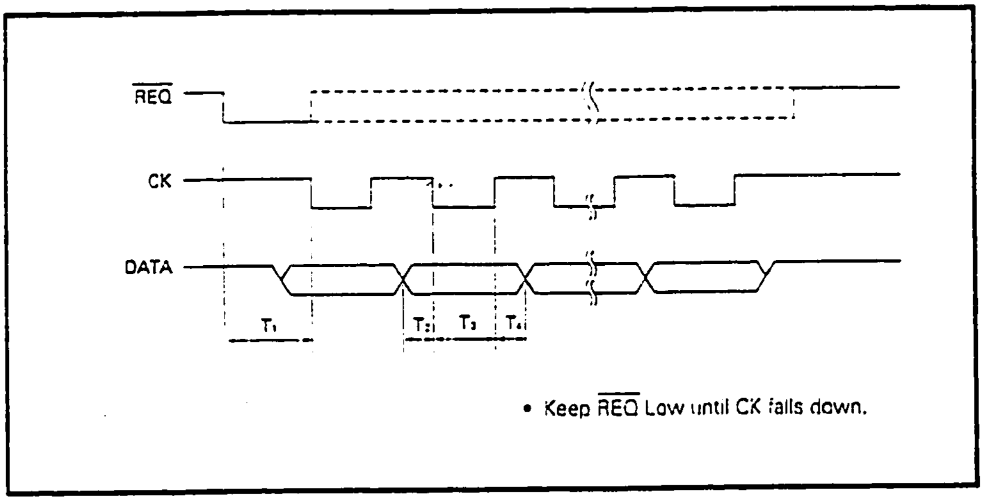

Figure 2.10 Timing Chart for Digimatic Indicator

Table 2.2 Timing Parameters

Limits	T_1	T_2	T_3	T_4
min	6 ms	200 μs	200 μs	650 μs
max	200 ms	300 μs	300 μs	750 μs

2.5 Summary

The structure of the deburring workcell and its components are discussed. The YAMAHA Zeta-1 deburring robot is chosen due to its robust mechanical design. Problems due to lack of adequate facilities on the controller of the robot to interface external sensors and modify the tool path in real time have been circumvented by implementing the workcell controller externally. A contact type displacement sensor is chosen for digitizing surfaces of refurbished workpieces and to establish the location of the workpiece before deburring. The workcell controller includes a PC-transputer functioning as the information processing component for supervisory and process control. Servo level control of the deburring robot is left to its controller. The PC-transputer network is assisted by low-level hardware interfaces to communicate with the controller of the robot and the displacement sensor.

Chapter 3

Software Architecture of Workcell Controller

3.1 Introduction

This chapter describes organization of the operating system software in the workcell controller. This component is responsible for directing the devices of the workcell, in the current application, the deburring robot and the probe, to perform various tasks such as locating the workpiece, probing the workpiece surface, performing surface reconstruction and executing a tool path. The "intelligence" required for these tasks is present in the workcell controller in the form of software. The software shell for this workcell controller is implemented as a collection of tasks running in parallel on a PC-transputer network. The parallel processing capability has helped in this real-time application by keeping the sampling time for the control loop small. Some of the tasks have been developed, and the others have been developed by other members of the Robotic Deburring Workcell team. Therefore, in the course of this chapter, all the tasks are explained and the present and intended final states of integration are described.

3.2 Parallel Processing on Transputers

As explained in Section 2.4.1, the information processing part of the workcell controller has been implemented on a PC-transputer network. All the software executing on the transputers has been written in 3L Parallel C, which is essentially standard ANSI C augmented with extra libraries enabling parallel processing and information exchange between different functions. Some of the concepts of Parallel C are paraphrased in the following lines from the 3L Parallel C manual [3.1] for convenience of the reader. This manual is an excellent reference for readers interested in implementing multi-tasking applications using transputers.

Parallel processing in transputer systems is based on the idea of *communicating sequential processes*. In this model, an application is a collection of concurrently active sequential processes which can only communicate with each other over *channels*. A channel connects exactly one process to exactly one other process and can carry messages only in one direction. For two-way communication between two processes, two channels must be used. Each process is referred to as a *task*. Each task has its own region of memory for code and data and a set of channels for communication with other tasks. Tasks can execute simultaneously on different transputers. In this case, the number of channels available for communication between different tasks is limited to the number of hardware links connecting the transputers. It is also possible to have more than one task executing on the same transputer. In such a case, each task can be assigned an URGENT or NON URGENT priority. A task with NON URGENT priority is executed only when an URGENT task is waiting for some data from another task. The CPU time-slices between tasks of the same priority. For tasks executing on the same processor, a channel for inter-task communication is implemented as a portion of the local memory on the transputer. Hence, there is virtually no limitation on the number of channels available for communication between two such processes.

Parallel processing is also possible within a single task. Parallel C allows for a function to be executed concurrently with the calling routine. Such a function is called a *thread*. It is even possible to have multiple copies of the same function execute in parallel. Like independent tasks, individual threads can be assigned an URGENT or NON URGENT priority. Similar to multiple tasks executing on the same processor, the CPU time-slices between threads of the same priority. There are, however, some differences between tasks executing on the same transputer and multiple threads existing in one task. Being constituent functions of the same program, threads, apart from having their own stack, have common access to global variables which can be used to exchange information. Tasks do not share any memory and hence, use only channels to communicate between one another. Tasks can be moved to a different transputer without any change in the source code. However, a thread has to necessarily execute on the same transputer as the parent task.

It is preferable to use tasks when the complexity of the tasks warrant executing them on different CPUs. Further, information exchange required between the tasks should be as little as possible. Multiple tasks executing on different transputers is a truly parallel process. However, if there is a large amount of data exchange between the tasks, there is a large communication overhead as a task has to wait till it receives the information it requires before proceeding. In such a case, it is better to reformulate the tasks as threads of a single program. The data exchange can be minimized using global variables.

3.3 Organization of the Workcell Tasks

The deburring application software consists of a number of tasks. Constituent tasks and their channels for interaction are shown in Figure 3.1. A symbolic name assigned to each processor is shown at the top left corner of each processor block and tasks executing in them are displayed in italics. Two tasks are executing on HOST, which is an IBM compatible PC-486. All other tasks are downloaded to the transputer network. Some of them are to facilitate screen and disk I/O, one to reconstruct surfaces from probe data and a couple of tasks to accept commands from the user and accordingly communicate with the devices of the workcell.

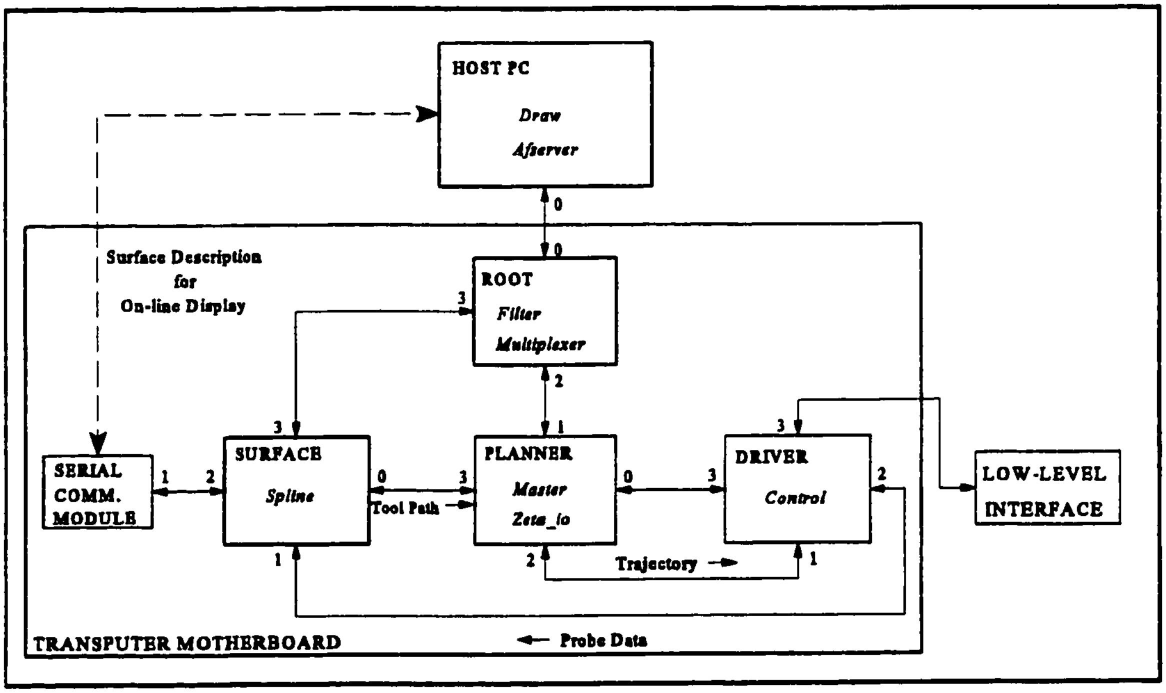

Figure 3.1 Software Layout for Workcell Controller

Although there are two tasks shown for HOST in Figure 3.1, there is no multi-tasking taking place on the PC. One of the programs, *Draw*, is a program for on-line graphics display to enable the user to visualize the digitized and reconstructed surface (Peng [3.2]). Another task executing on the PC is the transputer file server *Afserver*. At present, this is the only task executing on HOST. In the final implementation, *Afserver* would be executed first and terminated. The graphics program *Draw* would then be run on HOST. The reason for executing both tasks on HOST is explained in Section 3.3.1, where the role of *Afserver* is described.

3.3.1 *Master* Task

The task *Master*, executing on PLANNER, is responsible for interacting with the user and communicating his/her choice to the task *Control*. Depending on the option chosen by the user, an appropriate function is called by the main routine of *Master*. *Master* does not have any access to the devices of the workcell. Hence, an appropriate message is sent from *Master* to *Control* which in turn communicates with the devices of the workcell. Some of the functions accessible to the user are shown in Figure 3.2 and their implementation is described below.

3.3.1.1 Initialization Routine

This option must be chosen first when the application is started. A command is sent to *Control* to reset the chips constituting the Low-Level Interface shown in Figure 3.1 to ensure that no command pulses are being fed to the teach unit of the robot and the registers of the position decoders used to compute the actual location of the robot are set to zero.

3.3.1.2 Path Execution Routine

This routine enables execution of any path given as a sequence of points in World Coordinate System {X, Y, Z, A, B} of the robot at a desired speed. This sequence of points could come from two sources. After the workpiece has been located accurately, the CAD base of the newly machined component and the knowledge of location of the workpiece can be used to generate the path to be followed while probing. The digitized points are then used by *Spline* (explained in Section 3.3.3) to reconstruct the surface and extrapolate to obtain the edge profile. This edge profile is used to compute the tool path for deburring and passed on to the path execution routine.

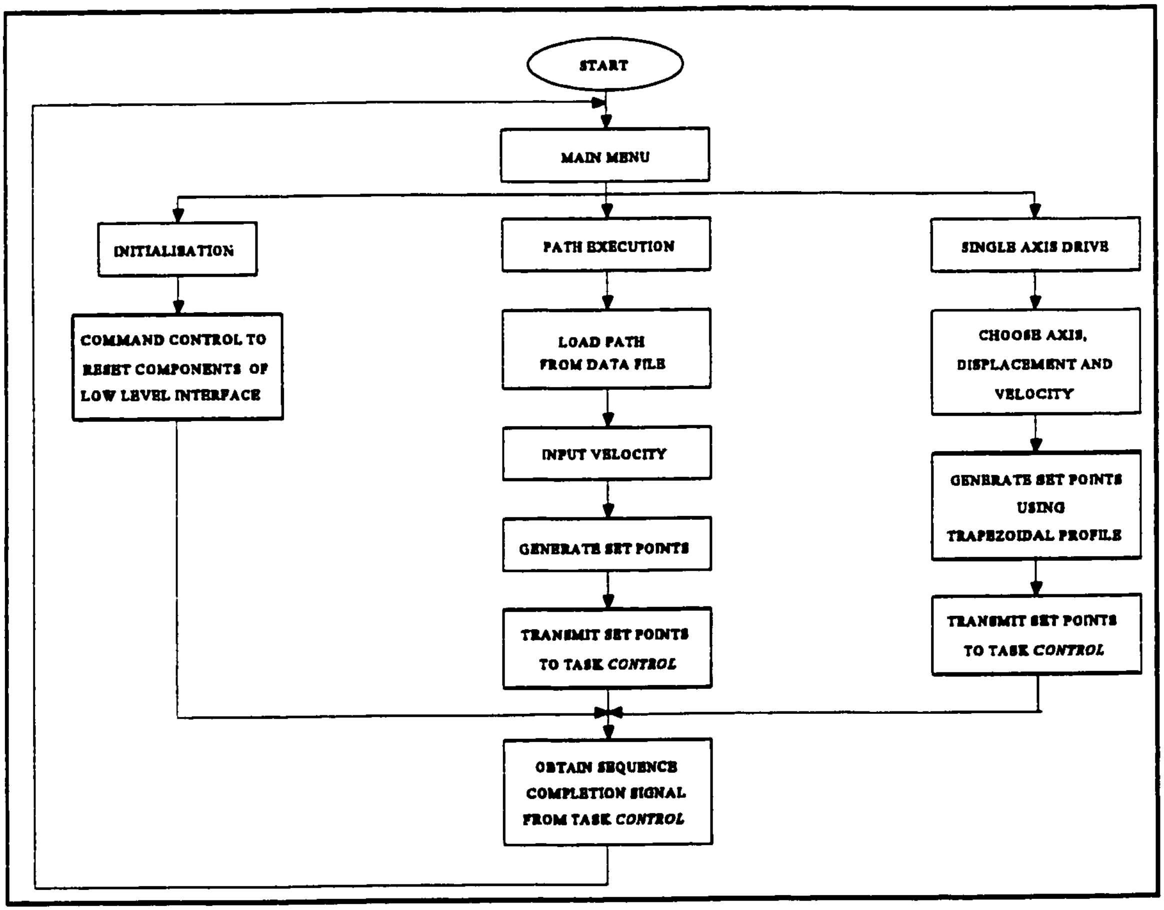
START
MAIN MENU
INITIALISATION
PATH EXECUTION
SINGLE AXIS DRIVE
COMMAND CONTROL TO RESET COMPONENTS OF LOW LEVEL INTERFACE
LOAD PATH FROM DATA FILE
CHOOSE AXIS, DISPLACEMENT AND VELOCITY
INPUT VELOCITY
GENERATE SET POINTS USING TRAPEZOIDAL PROFILE
GENERATE SET POINTS
TRANSMIT SET POINTS TO TASK *CONTROL*
TRANSMIT SET POINTS TO TASK *CONTROL*
OBTAIN SEQUENCE COMPLETION SIGNAL FROM TASK *CONTROL*

Figure 3.2 Flow Chart for task *MASTER*

Once the sequence of points and the desired speed are known, intermediate set points are calculated for every sampling period (0.004 s) and transmitted to *Control* to be communicated to an equivalent number of pulses. At present, *Spline* has not been integrated with *Master*. Therefore, the sequence of points is read from a file on the hard disk. Each point is checked to verify if it lies within the workspace of the robot. Once the points have been verified, a trapezoidal velocity profile is used to accelerate to a pre-specified velocity. If a change in direction (greater than 45°) in space is observed, the controller decreases the velocity to 2 mm/s at the point of change of direction and accelerates back to the desired peak velocity.

3.3.1.3 Digitizing Routine

This option (not shown in Figure 3.2) is used when data has to be collected from the Digimatic Indicator (displacement probe) continuously. This routine is intended for development purposes to test the interface communicating with the probe and is also used for experiments described in Section 4.2.

3.3.1.4 Single Axis Motion Routine

This routine is available primarily for debugging the hardware interface of the workcell controller. Motion along any of the five axes of the robot can be achieved. The desired acceleration, peak velocity, and displacement can be specified. These parameters are used to calculate the desired set-point for the machining point of the robot every sampling period and transmitted to *Control*. This function can also be used to move the robot without supplying a path in terms of World coordinates.

3.3.2 *Control* Task

Control is the only task with access to hardware interfaces used to communicate with the devices of the workcell. As seen in Figure 3.1, Link #3 of the transputer **DRIVER** is connected to the low-level interface and is used for communication between *Control* and the chips constituting the hardware interface. The advantage of restricting hardware access to one task is in the resulting ease of debugging the system during development. since any faulty instructions can be traced back to a single task. Further, during operation, from a safety point of view, it is preferable to give access to hardware to a single process at a time to avoid sending conflicting instructions to devices of the workcell. If any system related information (for example, present location of machining point) is required by any task constituting the workcell controller, it would have to send a request to *Control.* Hence *Control* has to continuously monitor all its channels to check if any request has been received. This is accomplished by using a Parallel C function called *alt vec wait()* [3.1] used to check all the channels connected to *Control* and indicate if a request is received on any channel.

There are a number of process routines corresponding to the different processes requested by the calling task. Some of them are shown in Figure 3.3. There are separate routines for initialising the devices of the workcell, testing the probe, moving the robot along a single World-axis, and execution of a path. These middle-level routines make use of a number of low-level routines developed for communicating with different chips constituting the low-level interface. Examples are a routine to program a 8254 timer chip to generate a pulse-rate of a desired frequency and reading the joint encoder count from a position decoder chip. Some of the low-level routines have been taken from the work of Kabra [3.3]. The important process routines are explained below.

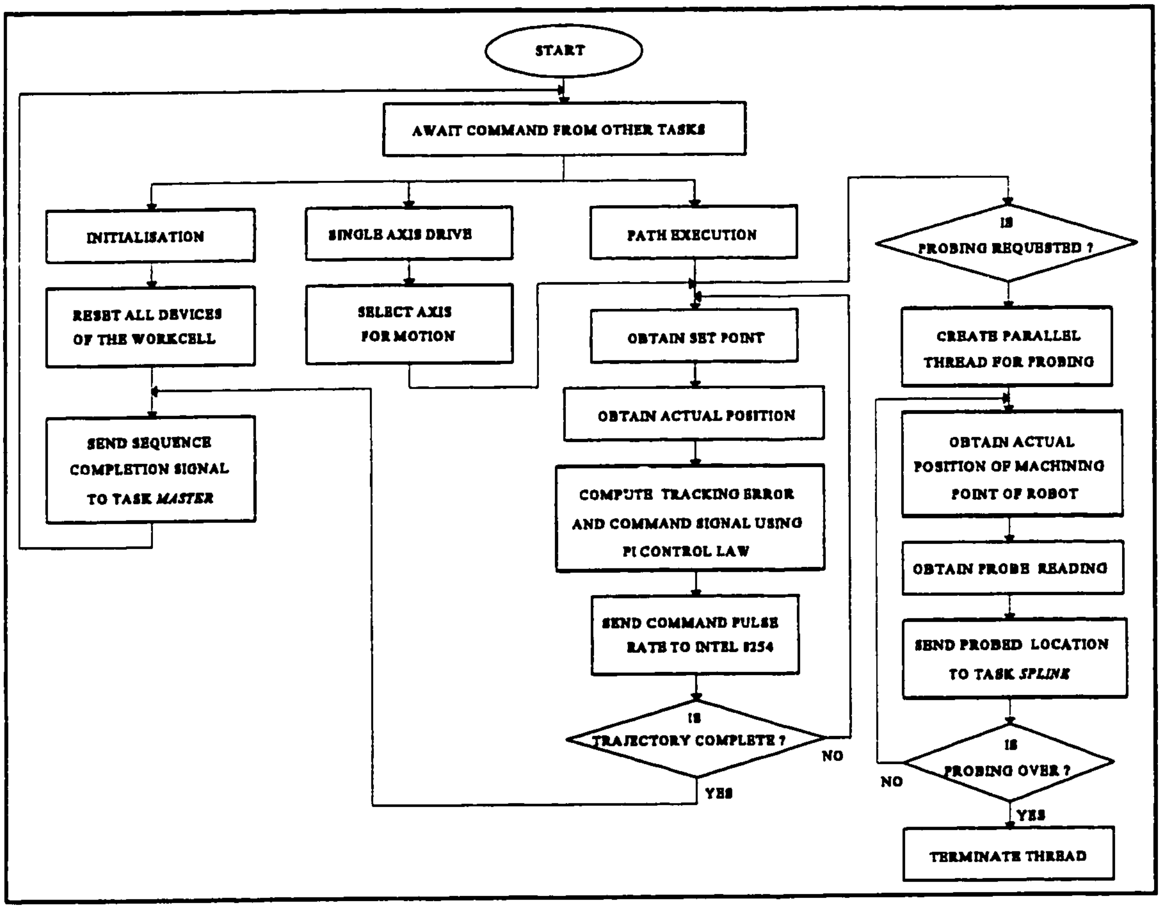
START
AWAIT COMMAND FROM OTHER TASKS
INITIALISATION
SINGLE AXIS DRIVE
PATH EXECUTION
IS PROBING REQUESTED ?
RESET ALL DEVICES OF THE WORKCELL
SELECT AXIS FOR MOTION
OBTAIN SET POINT
CREATE PARALLEL THREAD FOR PROBING
OBTAIN ACTUAL POSITION
SEND SEQUENCE COMPLETION SIGNAL TO TASK MASTER
OBTAIN ACTUAL POSITION OF MACHINING POINT OF ROBOT
COMPUTE TRACKING ERROR AND COMMAND SIGNAL USING PI CONTROL LAW
OBTAIN PROBE READING
SEND COMMAND PULSE RATE TO INTEL 8254
SEND PROBED LOCATION TO TASK SPLINE
IS TRAJECTORY COMPLETE ?
NO
YES
NO
IS PROBING OVER ?
YES
TERMINATE THREAD

Figure 3.3 Flow Chart for task *CONTROL*

3.3.2.1 Initialization Routine

This function is called by the main routine of *Control* when the user chooses the Initialisation option (Section 3.3.1.1) from the main menu displayed by *Master* and an appropriate command is received by *Control*. All the components of the low-level interface are reset. The 8254 timers, responsible for generating pulses for motion along a World axis, are reset to stop any pulses being generated and are kept ready to accept a new frequency command to be downloaded from *Control*. Shift registers used to obtain data from

Digimatic Indicator are initialised to zero.

The 32-bit position decoders, which keep track of the pulses being generated by the encoders of the joints of the robot, are set to zero. The joint encoders are of the incremental type. Hence, to be able to keep track of the joints of the robot, a reference location in Machine Coordinates $\{\theta, R, Z, \alpha, \beta\}$ has to be assumed for the machining point when all the position decoders are set to zero. The reference location is known as the READY position, which in Machine Coordinates is the location $\{180°, 700, 295, 0, 0\}$. It corresponds to the location $\{-700, 0, 295, 0, 0\}$ in World Coordinates. This reference position has been chosen because the deburring robot after being switched on is automatically moved to the READY position. Therefore, any time this routine is performed, it should be ensured that the machining point of the robot is located at the READY point.

3.3.2.2 Path Execution Routine

A path to be executed is supplied to *Control* as a sequence of set-points in World Space. If the objective is to locate the workpiece or probe a surface of the workpiece, data from the Digimatic Indicator could also be collected. The parallel thread created for probing is described in Section 3.3.2.3. A new set-point is received from *Master* at the beginning of every sampling interval. These set-points are converted to equivalent pulses for each World axis of the robot by programming the 8254 timer allotted for each axis. Let $S_D(k) = \{x_D(k), y_D(k), z_D(k), a_D(k), b_D(k)\}$ be the desired set-point sent at the beginning of the sampling interval from kT to [k+1]T, where T is the sampling period. Further, at the instant kT, let $S_A(k) = \{x_A(k), y_A(k), z_A(k), a_A(k), b_A(k)\}$ be the actual location of the machining point of the robot as computed from the data from position encoders of the joints of the deburring robot. The superscript in the notation is used to denote whether it is the desired set-point (D) or the actual set-point (A). The subscript is used to denote the sampling period. The pulse-rate computed for every World axis and is downloaded to the 8254 timer dedicated to that axis. A PI control law is used for every World axis and its digital form for a general World axis X_i is shown in Figure 3.4. The procedure used for computing the pulse rate is shown below for the general X_i-axis. The instantaneous axis position error is given by :

$$e(k) \cdot x_{iD}(k) - x_{iA}(k) \tag{3.1}$$

and the PI law used is in the form of

$$P(z) \cdot C\left[K_P \cdot K_I \frac{z}{z-1}\right]E(z) \tag{3.2}$$

where

E(z) = Z-transform of e(k)

P(z) = Pulse-rate (in Hz) for square wave to be generated by 8254 timer for X_i-axis.

C = Conversion factor (in Hz/mm) from control signal in mm to pulse rate in Hz (Refer Section 2.4.2).

K_P = Position gain.

K_I = Integral gain.

Equation 3.2 is implemented as follows :

$$P(k) \cdot C\left[K_p e(k) \cdot K_I \sum_{j \cdot 0}^{k} e(j)\right] \tag{3.3}$$

The sampling interval T is constant and is chosen as 0.004 seconds because the time taken for calculating the pulse-rate for all axes and downloading them to the 8254 timer chips is approximately 0.002 seconds if no probing is done and almost 0.004 seconds if data has to be collected from the Digimatic Indicator. The sampling time is monitored with the help of a Parallel C function called *timer wait()* [3.1]. The function has a resolution of 1 micro-second.

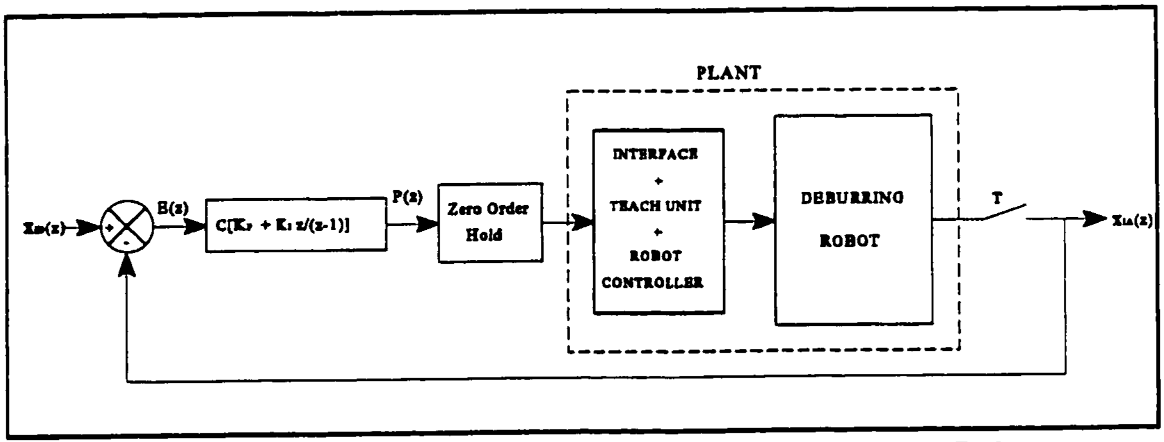

Figure 3.4 Control Loop for a General World Axis X_i of the Robot

The gains K_P and K_I for each axis have been chosen using the Ziegler-Nichols method [3.4]. The critical gain, K_C and critical period T_C and the gains for every axis are shown in Table 3.1. The position and integral gains are chosen using the criteria :

$$K_P \cdot 0.45\,K_C$$
$$K_I \cdot \frac{K_P}{0.85\,T_C} \tag{3.4}$$

It is to be noted that the critical gain and period are more or less the same for all axes irrespective of whether they correspond to position (X, Y, Z) or orientation (A, B).

Table 3.1 Critical Parameters for Tuning of PI Gains for World Axes

World Axis	K_C	T_C (s)	K_P	K_I (s^{-1})
X	0.04320	0.6827	0.019440	0.03350
Y	0.04500	0.6827	0.020250	0.03489
Z	0.04410	0.6827	0.019845	0.03419
A	0.04458	0.6827	0.020059	0.03457
B	0.04467	0.6827	0.020102	0.03464

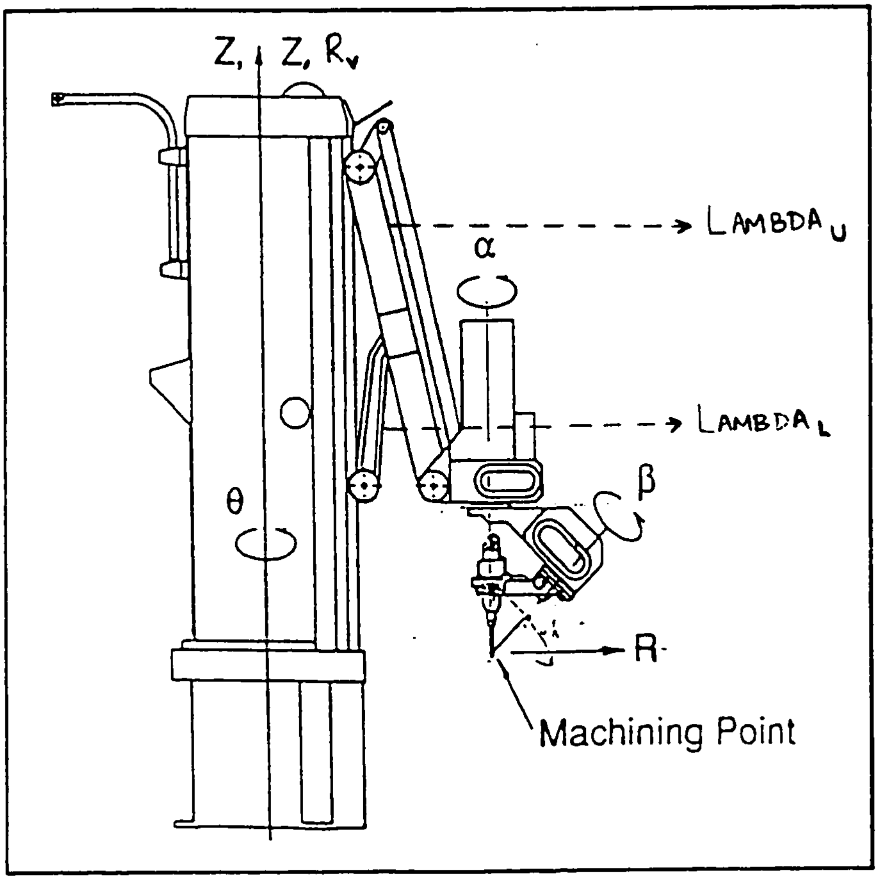

Figure 3.5 Motion Along R Axis [3.5]

In order to compute $S_A(k)$, the location of the five Machine Coordinates $\{\theta(k), R(k), z(k), \alpha(k), \beta(k)\}$ are required. These axes are shown in Figure 2.1. The coordinates θ(k), z(k), α(k) and β(k) are computed from the pulse count obtained by tapping the position encoders of the joints of the robot. However, calculation of R(k) is not straightforward. Movement along R-axis (Figure 3.5) is achieved by an equivalent motion along R_V axis, parallel to Z-axis. When motion along Z-axis is required, both links Lambda_U and Lambda_L (shown in Figure 3.5) are translated in the vertical direction by an equal amount. Hence the machining point is translated along Z-axis. When motion along R-axis is required, Lambda_L remains stationary and Lambda_U is moved along R_V axis, parallel to Z-axis by an amount given by the relation:

$$R(K) \cdot \sqrt{700^2 - [R_V(k)]^2} \cdot 355 \ mm \tag{3.5}$$

Once the five Machine Coordinates are known, the location of the machining point of the robot in World Space is obtained using the kinematic formulation :

$$\begin{aligned}
x(k) &- R(k)\cos\theta(k) \\
y(k) &- R(k)\sin\theta(k) \\
z(k) &\cdot z(k) \\
a(k) &\cdot \theta(k) \cdot \alpha(k) \cdot \arctan\left[\frac{1 - \cos\beta(k)}{\sqrt{2}\sin\beta(k)}\right] \\
b(k) &- \arccos[0.5\,(1 \cdot \cos\beta(k))]
\end{aligned} \tag{3.6}$$

In the above equations, x(k), y(k), z(k) and R(k) are expressed in mm while a(k), b(k), θ(k), α(k) and β(k) are expressed in radians. This kinematic model is derived in Appendix A. If the position of the robot is found to be near the limits of the workspace, the robot is stopped immediately and execution of the path is aborted. Similarly, if the position tracking error becomes excessively large and the command pulse-rate exceeds 1800 Hz, execution of the path is aborted. The reason for choosing the threshold of 1800 Hz is that the teach unit of the deburring robot can accept a maximum pulse-rate of 2000 Hz. The smallest displacement which can be obtained is 0.063 mm for X, Y and Z axis and 0.09° for A and B axes.

3.3.2.3 Probe Data Processing Thread

When probing is required, a parallel probing thread is started by the path execution function (previous section). The probing thread has a lower priority (NON URGENT) as compared to the path execution function, which has URGENT priority. While probing, the probe is moved continuously moved over the surface of interest with the tip of the stylus in contact with the surface and oriented normal to it. Raw data from the probe is collected every 0.252 seconds by the path execution function and passed on to the probing thread. The speed of the robot while probing is determined by the desired grid size. That is, if the grid size is Δx mm, the speed of the robot is ($\Delta x/0.252$) mm/s. The raw data is processed by the probing thread and the value of displacement of the probe stylus in mm is returned to the parent function. The reason for assigning a separate thread for processing of raw data is that the time taken for this process is greater than 0.002 seconds. Hence, with the computation of command pulse-rates and programming of the 8254 timers taking around 0.002 seconds, it would not be possible to maintain a sampling period of 0.004 seconds. Owing to NON URGENT priority of the probing thread, processing of raw data is done only when the path execution function has finished downloading the command pulse-rates to the 8254 timers and is waiting for the start of the next sampling interval.

The reading of the probe and the location of the machining point of the robot at the instant when the data is obtained from the probe are used to compose the coordinate of the probed location and can be transmitted to *Spline*. The probed locations will be used by *Spline* to reconstruct the surface of the workpiece.

3.3.2.4 Probing Routine

This function is activated when any task on the transputer network requires data from the probe to be taken continuously. For example, when the option from the main menu discussed in Section 3.3.1.4 is chosen by the user, an appropriate command from *Master* is sent to *Control* to access this function. Apart from helping test the interface with the probe, this function is also used in experiments conducted to compare the trajectory tracking performance of the robot when controlled only by the controller of the robot and when it is controlled through the Teach Unit by the workcell controller. The experiments are discussed in Section 4.2.

Raw data is collected from the shift registers interfaced with the probe every 0.252 seconds. The positions of the joints of the robot are also recorded from the position decoders. This information is used to compute the probed locations. These locations are saved as a file on the hard disk of HOST. Since no trajectory control is being performed and the probe is sampled only once in 0.252 seconds, all the operations are performed sequentially and no separate thread (as in Section 3.3.2.3) is created for data processing.

3.3.2.5 Single Axis Motion Routine

This function is developed for two reasons. One is to help test the low-level interface while developing the workcell controller. The other reason is to have the option of moving the robot without necessarily specifying the path in the form of a data file as required for the path execution function. This option is useful for unplanned situations. For example, when a path execution is aborted for some reason, and the robot has to be taken out of the vicinity of the workpiece safely, it can be carried out by specifying successive motions along individual axes of the robot to take it to a safe location (like a joystick).

This routine is called when the user selects the Single Axis Drive option (Section 3.3.1.5) from the main menu displayed by *Master* and an appropriate command is sent to *Control*. An axis set-point is received for the axis chosen every sampling period and converted to equivalent number of pulses by the 8254 timer dedicated to that particular axis. Motion along a single axis can also be achieved using the path execution function by varying a single world coordinate only in the sequence of points stored in the input data file.

3.3.3 *Spline* task

Once a workpiece has been probed, the data is sent to the task *Spline* executing on SURFACE for surface reconstruction and extrapolation to obtain the edge profile of the surface of the workpiece. The edge profile is sent to the task *Master* for generating the tool path. The data regarding the digitized surface and its reconstruction can be sent to *Draw* using an RS-232 serial link between HOST and the transputer motherboard. The serial link is required because there exists no facility for transferring data from the local memory of any transputer to the memory of the PC. Details regarding the algorithm (for the task *Spline*) used for reconstruction of the surface using bicubic splines and extrapolation to obtain the edge can be found in [3.6]. This task has not been integrated with *Master* and *Draw*. Transfer of probed data from *Control* to *Spline* is implemented successfully. Probed locations are sent in the form of a data structure to *Spline* every 0.252 seconds.

3.3.4 Routines for Access to I/O Facilities of HOST

One of the vendor-supplied tasks is a file server called *Afserver* [3.1] executing on HOST. It loads all the programs onto the transputers and processes requests from them for access to disk and user I/O facilities of the host. Owing to *Afserver* being of the polling type, the HOST is completely tied up with checking if any of the tasks executing on the transputers need any action to be carried out. This is not acceptable as it is planned to have the graphics software *Draw* execute on HOST.

To prevent the host PC from being preoccupied by the polling-type file server, Cheng et al [3.7] developed an Interrupt Driven File Server (IDFS). After loading all the tasks on to the transputer network, IDFS terminates but stays resident in the memory of the host PC. This file server is activated only when a request for I/O from the transputer is received. Once the request has been processed, IDFS once again goes back to Terminate-Stay-Resident (TSR) mode and the PC is free to execute any other program. However, the source code for IDFS has a number of system based parameters which had been tuned for a PC 286 processor. Since, HOST in Figure 3.1 is a PC 486 processor, the parameters do not hold good and tuning of these system parameters is required to make the file server work for

HOST. Hence, IDFS is not used for the deburring application, as the portability of the workcell controller software is affected by choosing this file server.

Another option is to use the vendor-supplied *Afserver* to load all the application tasks on to the transputers and then terminate the server. In the absence of a file server, *Draw* would have to assume the additional role of processing screen and disk I/O access requests from the transputer tasks, along with its primary function as a graphics visualization package. For example, if a task on the transputer needs to have a message displayed on the screen of HOST, it could send a command accordingly with the relevant data to *Draw*. This command is then processed by *Draw* and the message is displayed on the screen. The command and relevant data could be sent from transputers to HOST using the RS-232 serial port of the computer. Therefore, when any task on the transputer network needs to access the facilities of the host, a request is sent to a task called *Zeta io*, executing on the transputer PLANNER, which is responsible for communicating the request to *Draw* through the serial port. As explained before, communication through RS-232 serial port is required because there is no facility for data transfer from local RAM of a transputer to that of HOST. At present, *Draw* has not been integrated with the other routines. Hence, *Zeta io* still uses *Afserver* to access the facilities of the host. Since, *Zeta io* is the only task which communicates with *Afserver* during integration of *Draw* into the workcell controller, other tasks will not have to be revised.

Another vendor-supplied task, executing on ROOT, is *Filter* [3.1]. This task is required because *Afserver* and user tasks executing on transputers use different message protocols while communicating through channels. The task *Multiplexer* [3.1] is required only if more than one task is trying to access the I/O facilities of HOST through *Afserver*. *Multiplexer* is used because *Spline* has not been integrated with *Zeta io* and hence is still required to communicate directly with *Afserver*. Once *Spline* has been integrated with *Zeta io*, *Multiplexer* will not be necessary as *Zeta io* will be the only task communicating with *Afserver*.

3.4 Summary

The software architecture of the information processing part of the workcell controller is discussed. The various tasks constituting the workcell controller software are presented and the present state of system integration is discussed. *Draw* is a software under development, to be run on HOST, for on-line graphics display to enable the user to visualize the digitized and reconstructed surfaces of the workpiece. *Master*, executing on PLANNER, is the task responsible for accepting commands from the user and sensing appropriate instructions to *Control*. It is also responsible for performing path planning for the deburring robot and transmitting set points every sampling interval to *Control*. The task *Control* is the only task with access to the hardware interface used to communicate with the devices of the workcell. It is capable of resetting the chips constituting the hardware interface, execute a path supplied in the form of set-points and collect information from the probe at the same time and testing the interface for the probe and the interface used to send commands to the controller of the robot. The sampling time used for path execution is 0.004 seconds and a PI control law is used for each axis. *Afserver*, *Filter* and *Multiplexer* are vendor-supplied tasks to enable the tasks executing on the transputers to access the screen and disk I/O facilities of HOST. The task *Zeta io*, executing on PLANNER, is also developed for the same purpose to function in the absence of *Afserver*. The integration of all the tasks constituting the software shell for the workcell controller is not yet complete.

Chapter 4

Experimental Results

4.1 Introduction

The previous two chapters discussed the hardware and software aspects of the deburring workcell and the extent of integration of all its components. A lot of emphasis has been placed on the development of a workcell controller enabling the control of the robot from a PC-transputer network. The experiments described in this chapter are intended to show that the workcell in place is capable of executing straight-line trajectories supplied from the workcell controller with a reasonable accuracy and can also obtain data from the probe in real-time while the probe is continuously moving along the surface of the workpiece. It should be noted that the controller is not limited to straight-line trajectories. Performance evaluation of tracking of curvi-linear trajectories is discussed in Rajagopalan et al [4.1]. First, experiments are conducted to show the performance of the robot when moved along each World axis of the robot. Next, the trajectory tracking of the robot (for X and Y axes) is evaluated in seven different locations of the workspace. The first set of tests has been conducted to determine the performance of the control loop implemented for each World axis and the second set of tests is required to observe whether the performance of the robot varied significantly over the workspace thus requiring variable control gains. Finally, in Section 4.4, a comparison between two methods of directing the robot along a straight line path is provided. This is to demonstrate the differences in using the workcell controller to direct the robot and using the YAMAHA controller alone in *AUTO* mode to direct the robot.

4.2 Trajectory Tracking Performance of Each Axis

The first set of tests is carried out to evaluate the trajectory tracking performance of the robot when commanded to move along a World axis of the robot at a constant velocity. The experimental set-up is shown in Figure 4.1. The trajectory of the robot is computed in the PC-transputer network and downloaded to the controller of the robot via the teach pendant using a special interface. The optical joint encoders of the robot are read from the YAMAHA controller. These readings are used to compute the actual location of the machining point of the robot in World space.

For the axes X, Y and Z, the robot is commanded to move by a distance of 100 mm. Two speeds are tried out : 5 and 10 mm/s. These speeds are chosen as they correspond to typical values chosen for machining and free travel of the tool respectively. The robot is accelerated from rest at the rate of 10 mm/s^2 to the desired velocity and decelerates to rest at the end of the trajectory. These experiments are carried out without any machining being performed. All the locations mentioned in the following lines are described in World Coordinate System {X, Y, Z, A, B}. In every figure except Figure 4.1 (explained in the next paragraph), there are two graphs. The desired trajectory for the axis is shown in the top graph and the instantaneous position error for the same axis is shown in the bottom one. The desired trajectory corresponds to the sequence of set-points locations (refer Section 3.3.1.3) sent from the task *Master* to the task *Control*. The instantaneous position error for a sampling interval is computed as the difference between the desired set point for that interval and the actual set point for the same interval. The actual set-point is computed from the positions of the joint encoders of the robot. The sampling time for the control loop is 0.004

seconds.

Due to memory constraint, data collected every 10 sampling intervals (every 0.04 seconds) are used for Figures 4.1-4.3. This memory constraint is caused by the limited on-board memory of the transputer modules used. The small sampling time implies that data being collected cannot be downloaded to the hard disk during the course of the execution of the path. Hence, the amount of data which can be collected is limited by the RAM available for the transputer DRIVER. Currently, DRIVER has 4 Megabytes of RAM and by using a Transputer module with a larger RAM, this memory problem can be overcome.

For X-axis, the robot is commanded to move from the location {-700, 0, 295, 0, 0} to the location {-600, 0, 295, 0, 0}. Generally, the joints θ and R of the robot (Figure 2.1) are used to achieve motion along X-axis. But for this particular path, only motion along R-axis is required. The results for the speed 5 mm/s is shown in Figure 4.1(a)-(c). Two horizontal lines at X = ±0.063 mm are drawn in Figure 4.1 (b). The lines represent the displacement of the machining point of the robot for one pulse fed to the Teach Unit of the robot (refer Figure 2.5). It is seen that for most part of the trajectory, the trajectory tracking errors lie in the band defined by the two command resolution lines at X = ±0.063 mm. This value corresponds to the parameter K_X calculated in Chapter 2. One possible method to improve the command resolution is to increase the scaling factor between motion of the robot and corresponding motion of the joysticks of the teach unit. However, the larger the scaling factor, the smaller is the magnitude of continuous displacement the robot can undergo. The trajectory tracking errors are larger in the beginning and at the end of the

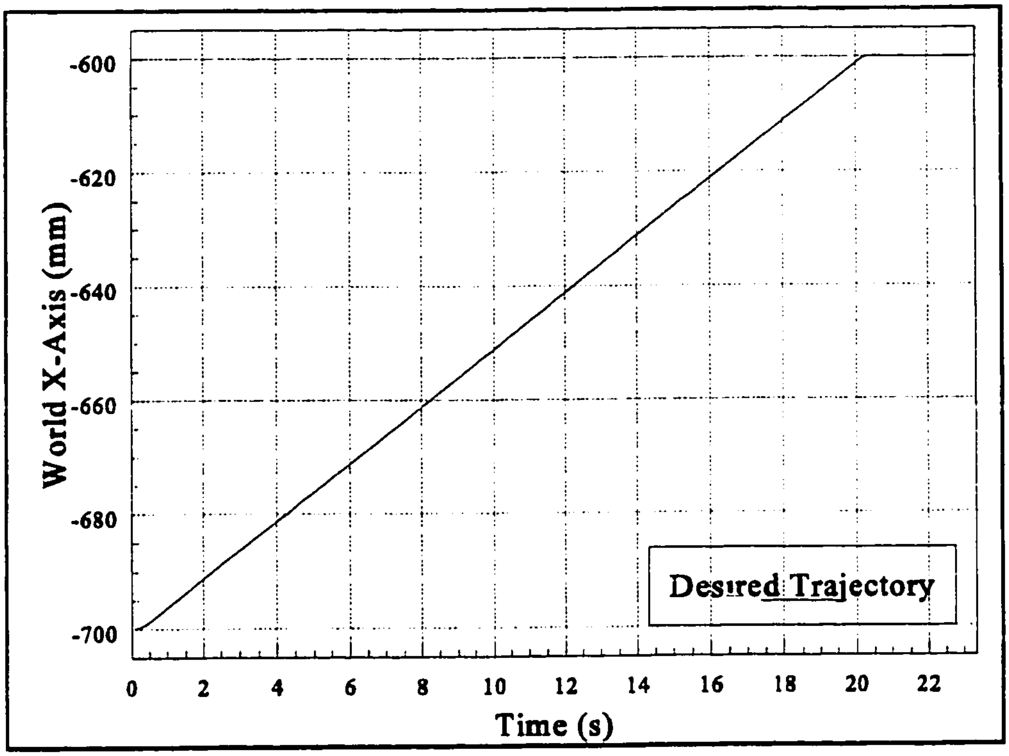

Figure 4.1(a) Desired Trajectory for World X-axis : Speed 5 mm/s

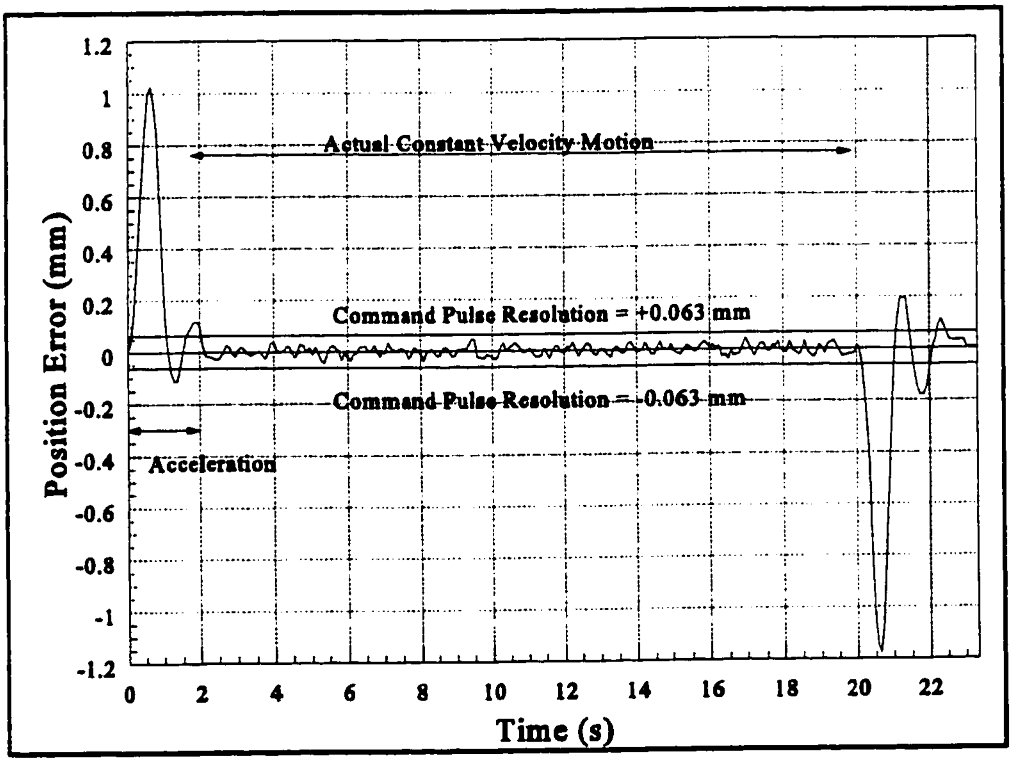

Figure 4.1(b) Position Error for World X-axis : Speed 5 mm/s

trajectory as the robot tries to keep up to the commanded acceleration. The same trend is observed for the results shown for the other speeds as well. These large magnitudes can be reduced by implementing algorithms to smoothen the trajectory to enable a gradual change in acceleration. A magnified view is given of the trajectory tracking error corresponding to Figure 4.1 (a) in Figure 4.1 (c). The phenomenon of hunting is observed at a frequency much smaller than the sampling frequency (250 Hz). The hunting is due to the fact that the resolution of the command pulse is X = ±0.063 mm. The root mean square error (excluding the initial peak observed in the error graph) is 0.0191 mm.

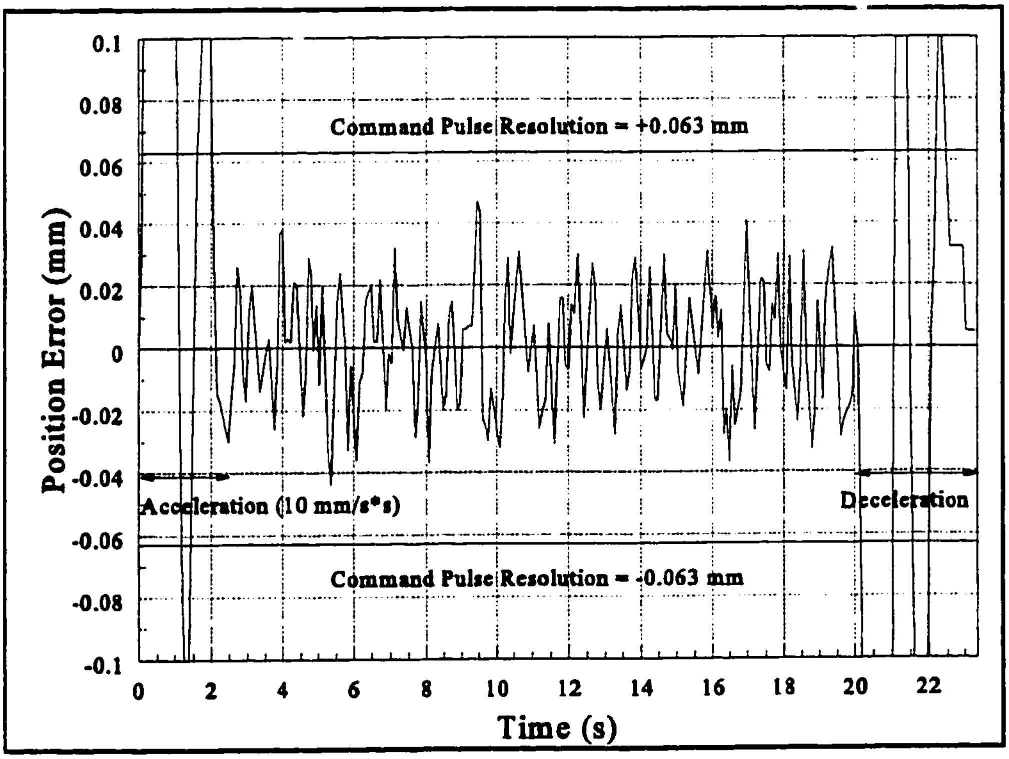

Figure 4.1(c) Magnified View of Position Error for World X-axis

Motion along Y-axis involves the joints R and θ, as in the case of X-axis. When commanded to move from the location {-700, 0, 295, 0, 0} to the location {-700, 100, 295, 0, 0}, the movement of the θ-joint is far more than the movement of R-joint. The results for the trajectory executed at 5 mm/s are shown in Figure 4.2. The band described by the command pulse resolution is shown in Figure 4.2(b) at Y = ±0.063 mm. The performance for this axis is quite similar to the one observed for motion along X-axis. The root mean square error (excluding the initial peak observed in the error graph) is 0.0202 mm.

The trajectory tracking results obtained when the robot is moved from the location {-700, 0, 295, 0, 0} to the location {-700, 0, 195, 0, 0} are displayed in Figure 4.3. Motion along World Z-axis is achieved by actuating only Z-joint of the deburring robot. Similar to the axes X and Y, the command resolution pulse is shown in the error graph. The performance for this axis is shown to be much better than the other two position axes. The reason is perhaps that only one joint is involved and the relation between the World axis and the joint of the robot is linear (1:1). The root mean square error in this case is observed to be 0.0149 mm.

The trajectory tracking performance for the two orientation axes A and B are shown in Figures 4.4 and 4.5. For A-axis, the robot is commanded to move from the location {-700, 0, 295, 0, 0} to the location {-700, 0, 295, 100, 0}. Two speeds are tried out. They are 10°/s, and 20°/s. The robot starts from rest, accelerates to the desired velocity at the rate of 20°/s^2, completes the trajectory and finally decelerates to a stop at the rate of 20°/s^2. A displacement of 0.09° is obtained when one pulse is fed to the Teach Unit for these axes. Therefore, in the error graphs for the two axes, the command resolution pulse is shown at A = ±0.09° and B = ±0.09° respectively. As in the case of the position axes, there is a large build up of error (up to 2°) as the robot tries to reach the desired velocity and then the position error stays within the band bounded by the command pulse resolution lines.

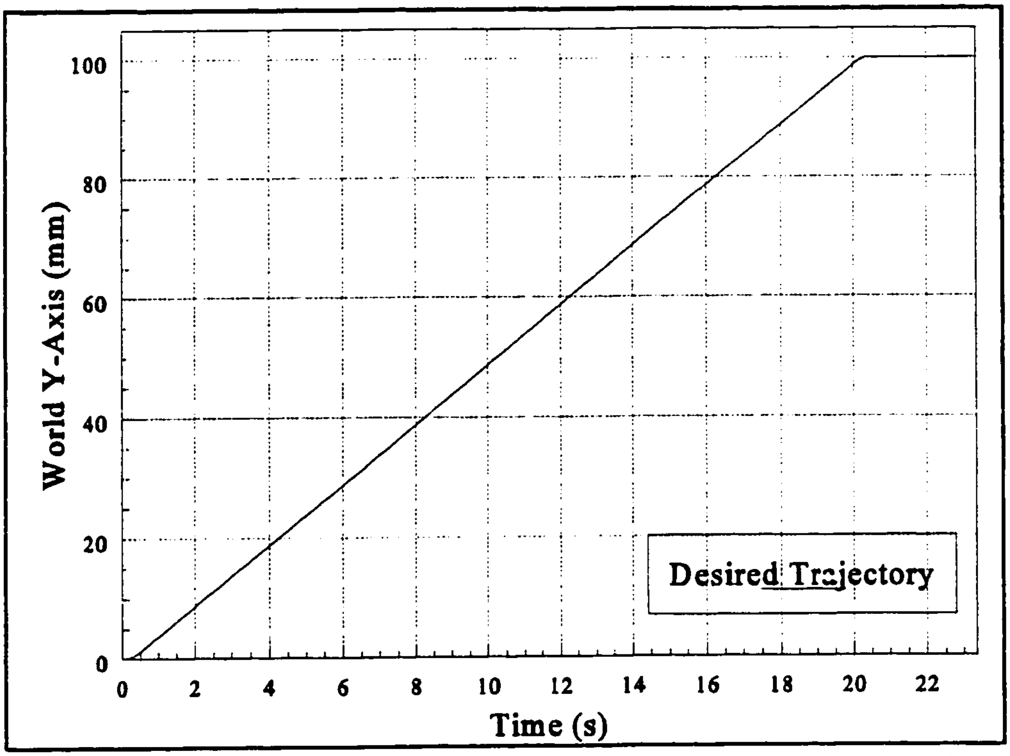

Figure 4.2(a) Desired Trajectory for World Y-axis : Speed 5 mm/s

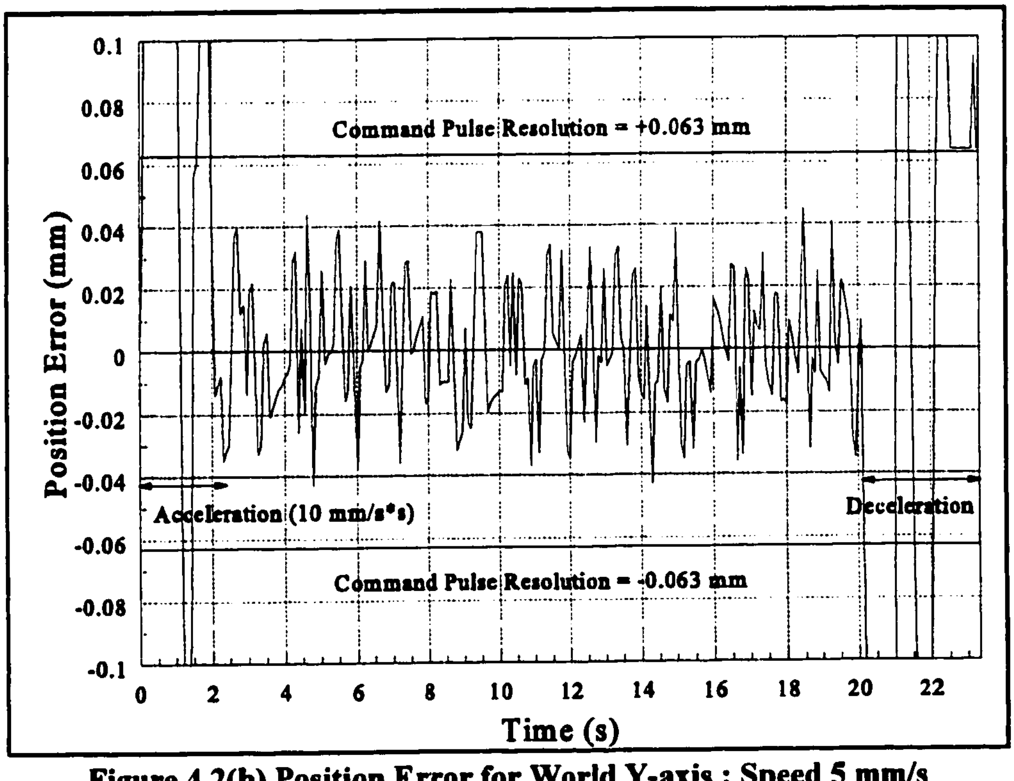

Figure 4.2(b) Position Error for World Y-axis : Speed 5 mm/s

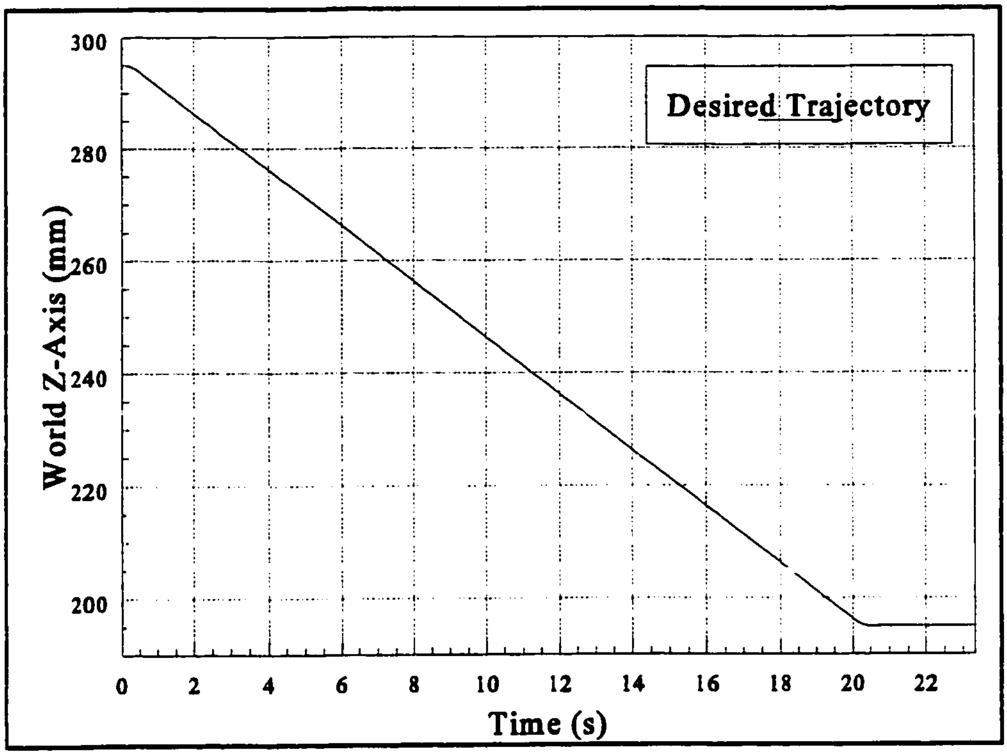

Figure 4.3(a) Desired Trajectory for World Z-axis : Speed 5 mm/s

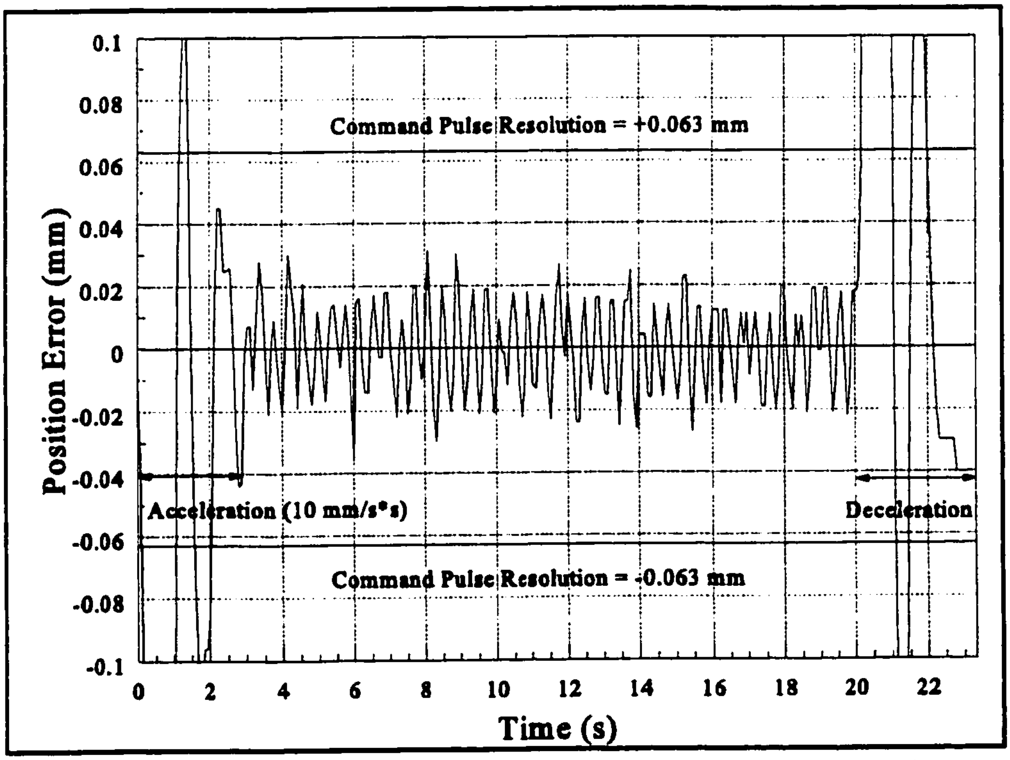

Figure 4.3(b) Position Error for World Z-axis : Speed 5 mm/s

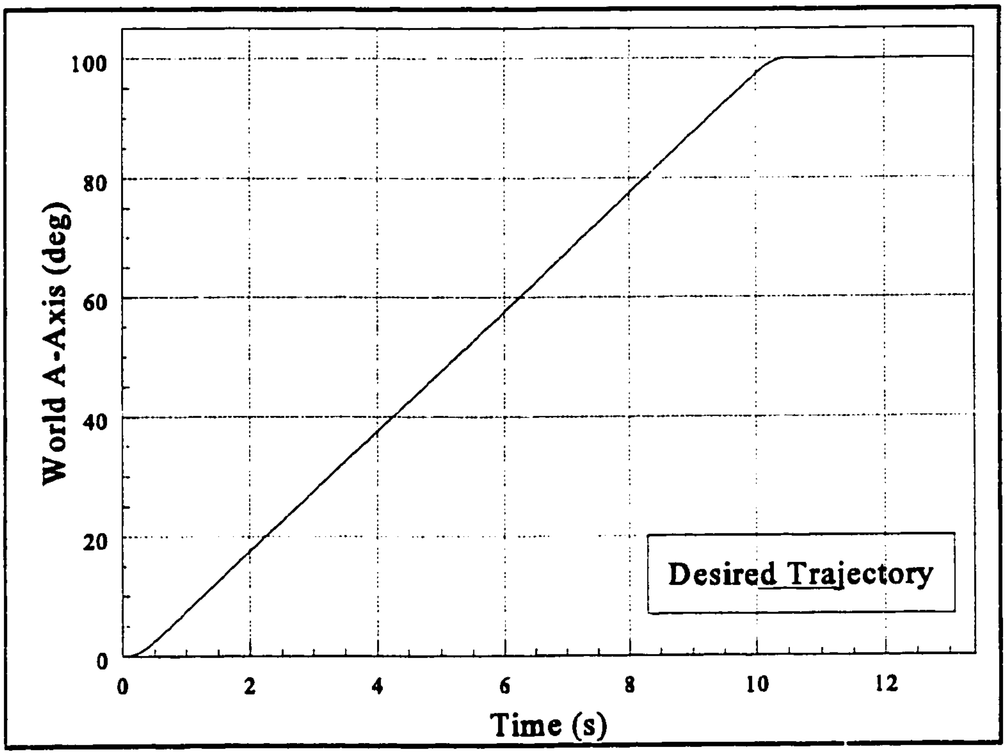

Figure 4.4(a) Desired Trajectory for World A-axis : Speed 10 deg/s

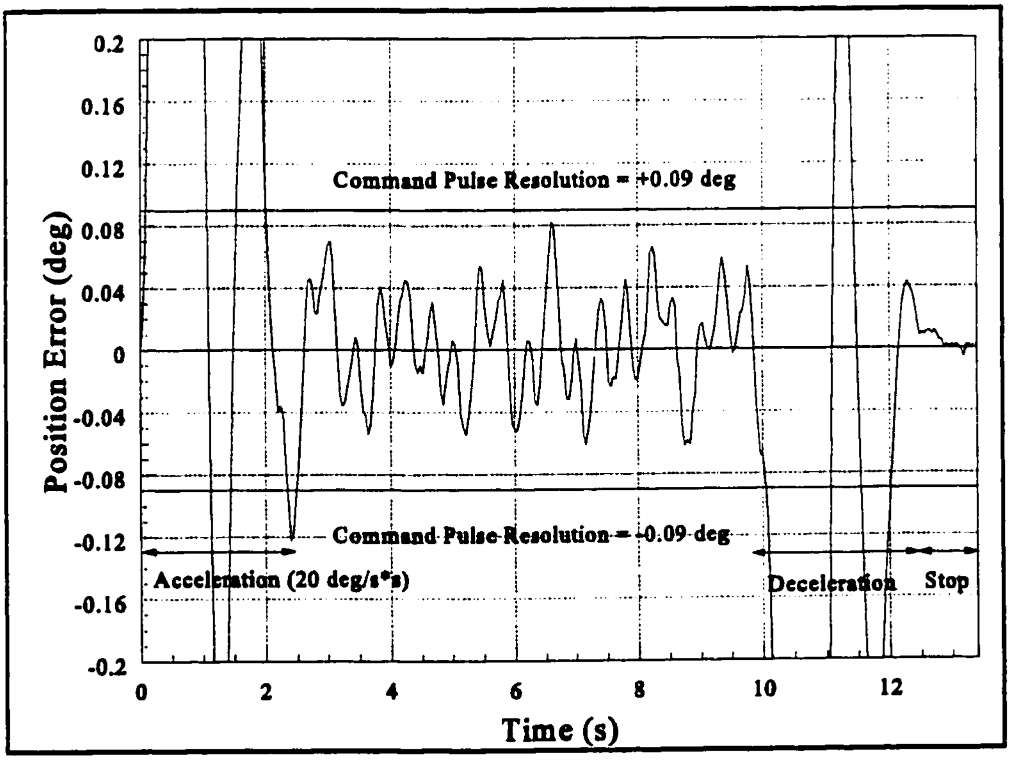

Figure 4.4(b) Position Error for World A-axis : Speed 10 deg/s

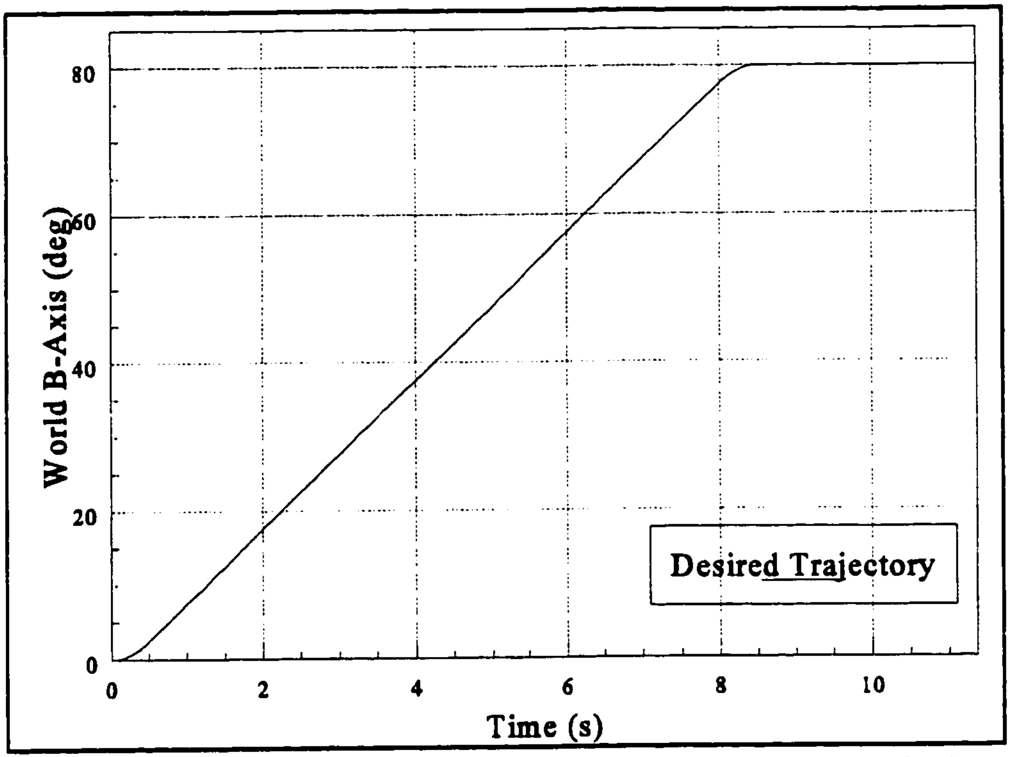

Figure 4.5(a) Desired Trajectory for World B-axis : Speed 10 deg/s

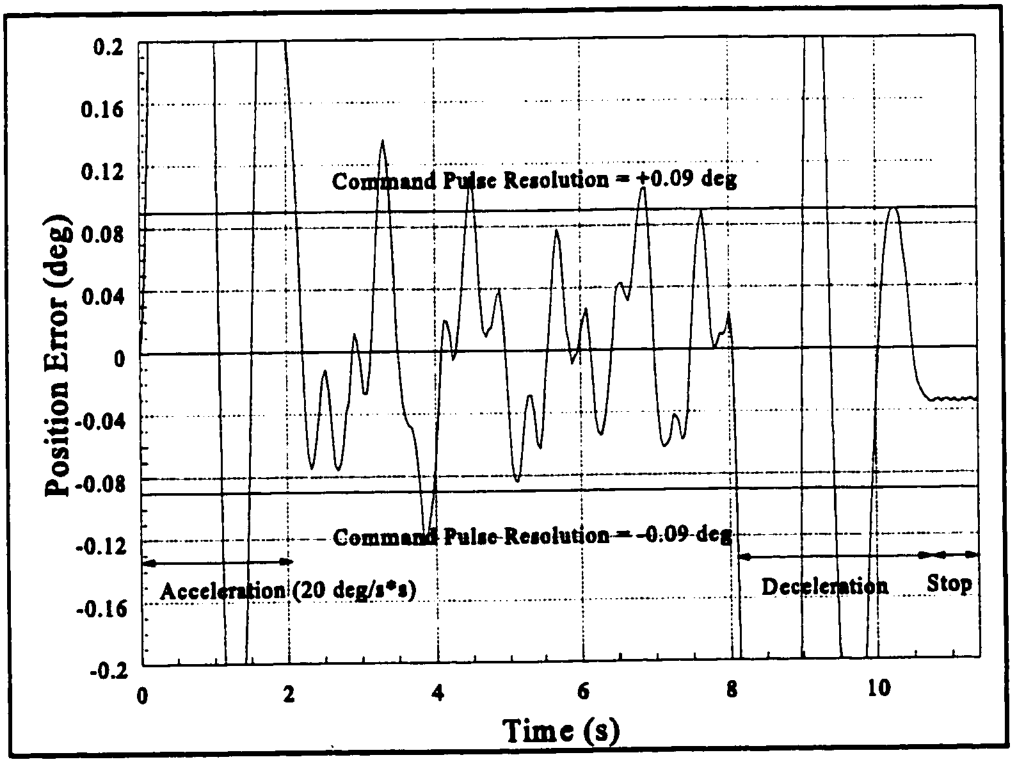

Figure 4.5(b) Position Error for World B-axis : Speed 10 deg/s

The trajectory tracking performance for higher speeds for all the five axes shows the same trend as those displayed for the speeds displayed in Figures 4.1-4.5. The root mean square error (RMSE) for the trials at all speeds are tabulated in Table 4.1 for X, Y and Z axes and Table 4.2 for A and B axes. While calculating the RMSE values, the initial and the final peaks in the error graphs have been omitted. It can be seen that the RMSE for each axis increases with increase in the desired speed.

Table 4.1 Root Mean Square Error (RMSE) for X, Y and Z Axes

Axis	Speed 5 mm/s	Speed 10 mm/s
X	0.0191 mm	0.0369 mm
Y	0.0202 mm	0.0345 mm
Z	0.0149 mm	0.0343 mm

Table 4.2 Root Mean Square Error for A and B axes

Axis	Speed 10°/s	Speed 20°/s
A	0.0384°	0.0981 °
B	0.0554°	0.0585°

4.3 Trajectory Tracking Performance over the Workspace

The two World axes X and Y involve actuation of θ and R joints of the deburring robot. In some areas of the workspace, for moving a certain distance along X-axis, the motion of θ-joint might be predominant. In other areas of the workspace, to move the same distance along X-axis, motion of R-joint might be more predominant. Therefore, depending on where exactly in workspace, a path involving X and Y axes is executed, the effect of these joints on X or Y axes varies. However, the control law and the control gains remain the same for X-axis and Y-axis, irrespective of which joint is being actuated. The following experiments are meant to verify that keeping the control gains for X and Y axes constant does not produce distinctly different results in various areas of the workspace of the robot.

Seven paths are selected for analysis of X and Y axes all over the workspace. The paths are shown in Figure 4.6. Each path involves tracing the edge of a square with a side of 138 mm and aligned with the World X and Y axes of the robot. All the trials are conducted at the speed 10 mm/s. The robot accelerates from rest at the rate of 10 mm/s^2 to the desired velocity at the beginning. At the corners, the robot slows down to a speed of 2 mm/s. Finally, after completing all the four sides of the square path, the robot comes to a rest by decelerating at the rate of 10 mm/s^2.

The root mean square error (RMSE) and Standard Deviation (SD) for X and Y axes for each case are tabulated in Table 4.3. It can be seen that in spite of two joints being involved to a varying extent, the RMSE is roughly the same all over the workspace for X and Y axes. This could be attributed to the controller of the robot.

4.4 Trajectory Tracking Comparison with the Robot Controller

The path for experimentation involves tracking the edge of the top face of a rectangular piece of dimensions 138 x 138 x 70 mm shown in Figure 4.7. Two speeds 5 and 10 mm/s are tried out. It is to be noted that no machining is taking place in course of any of the experiments described in this chapter. This is because the tool has not yet been interfaced to the workcell controller.

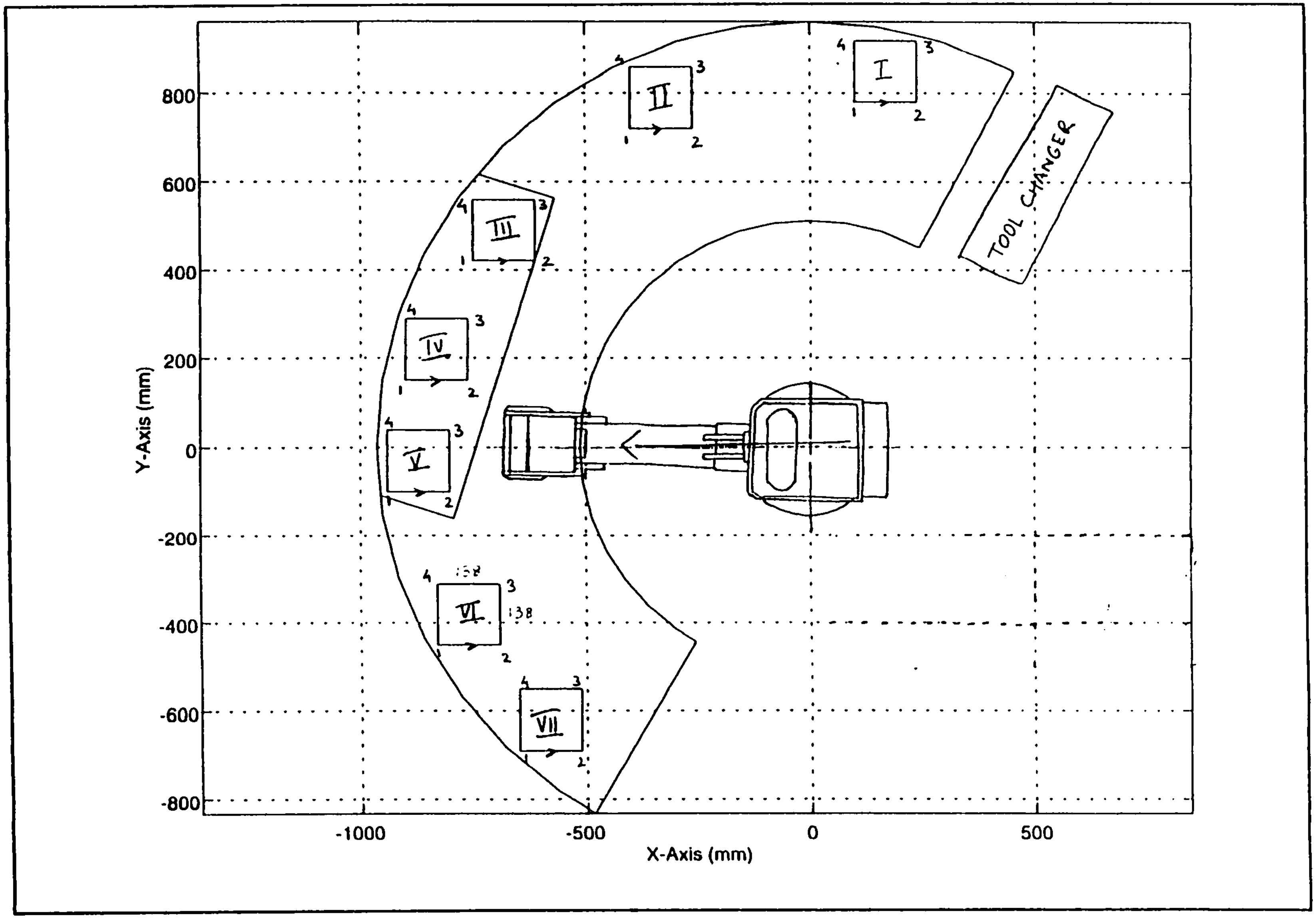

Figure 4.6 Top View of the Workspace of the Robot

Table 4.3 RMSE and SD Data for X and Y Axes

PATH	Parameter	X axis (mm)	Y axis (mm)
I	RMSE	0.0437	0.0440
	SD	0.0592	0.0580
II	RMSE	0.0437	0.0479
	SD	0.0613	0.0640
III	RMSE	0.0434	0.0460
	SD	0.0620	0.0628
IV	RMSE	0.0440	0.0480
	SD	0.0634	0.0606
V	RMSE	0.0387	0.0476
	SD	0.0558	0.0610
VI	RMSE	0.0403	0.0501
	SD	0.0596	0.0684
VII	RMSE	0.0391	0.0442
	SD	0.0593	0.0569

As explained in Chapter 2, there are two possible ways of directing the deburring robot along a desired path. One is to program the path into the controller of the robot off-line, set the controller to *AUTO* mode and execute the path, without using the workcell controller developed in this thesis. This provision is supplied by the manufacturer of the robot and is the conventional method of directing the robot. The other method is to use the workcell controller, developed as part of research done for this thesis, to supply the path to the controller of the robot on-line and execute the path.

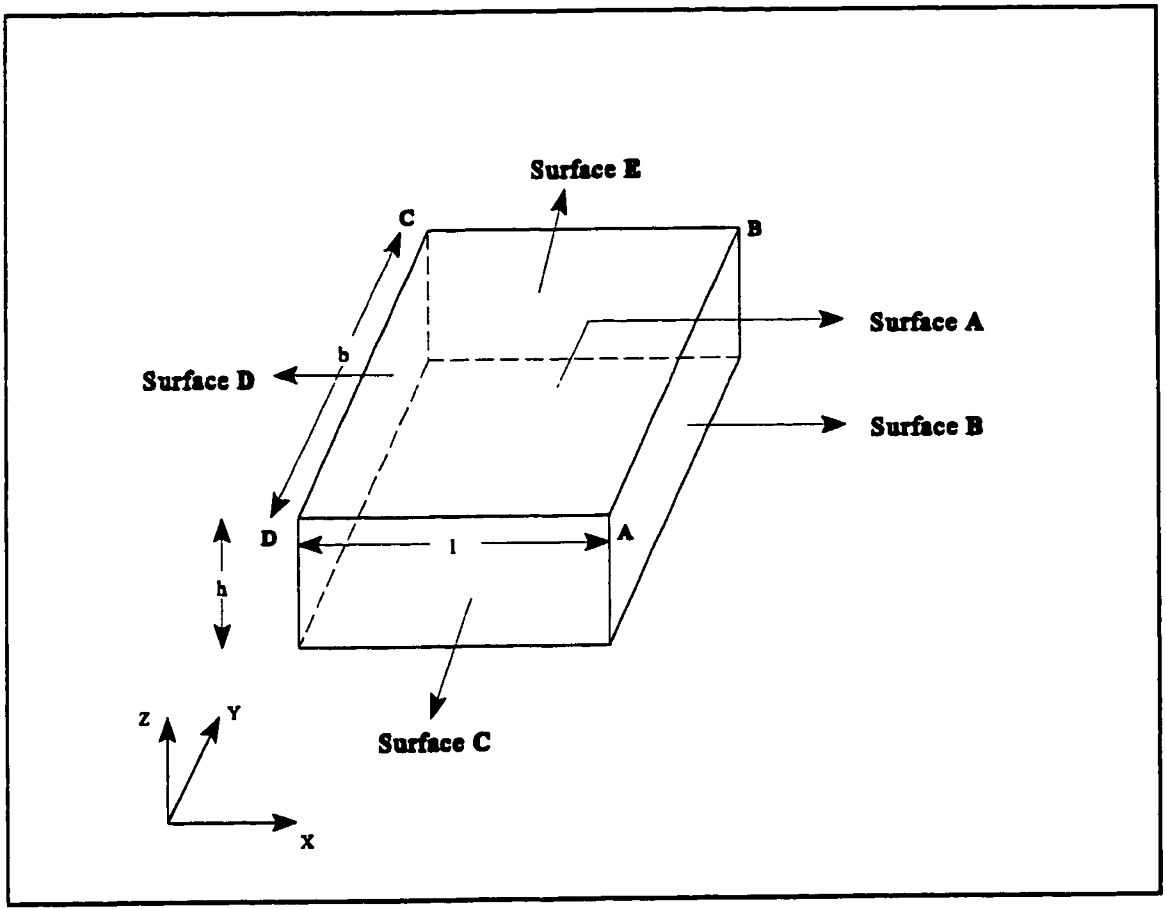

Figure 4.7 Workpiece for Comparison Experiments

In the experiment described in this section, two points from the side (say AB in Figure 4.7) are chosen. The two locations in World Coordinates are stored in a data file on the hard disk of HOST (Figure 3.1). The data file is then used by the workcell controller to direct the robot. While the path is being executed by the robot, data from the probe are collected continuously. The stylus of the probe is pressed against the face of the workpiece (Face B in the case of side AB) and is oriented normally with respect to the same face. Hence any deviation from the straight line path (say AB) requested will result in displacement of the stylus of the probe. The probed locations are computed using the location of the machining point of the robot (computed from joint encoder information) and the displacement of the stylus of the probe. These probed locations can be used to give a measure of the accuracy with which the desired path is being tracked by the robot. The same tool path is repeated by executing it directly using the controller of the robot by setting it in *AUTO* mode. Data are collected from the probe simultaneously as in the above case. It should be noted in both cases that only joint encoder information, in conjunction with the inverse kinematics model of the robot, is used as feedback while computing the command pulse rate. The probed locations are only used to evaluate the path tracking performance of the robot off-line.

The results of the experiments are shown in Figures 4.8-4.11. Each figure corresponds to one side of the rectangular piece in Figure 4.7. The reference path, the path traced by the robot when directed by the workcell controller and the path traced by the robot when directed in *AUTO* mode by its own controller only are shown. For all the trials, a bias of the order of 0.1 mm is noted. This is most likely due to error in computing the actual location of the tip of the probe stylus from the probe reading and the joint encoder data. In order to calculate the same, the reading displayed by the probe when its stylus coincided with the machining point of the robot was noted. The reading displayed by the probe kept fluctuating and hence an error could have been made while noting down the displacement of the stylus of the probe by the author.

It is observed in Figure 4.8 for the side AB that the maximum error in both cases is 0.15 mm (0.006"). This error can probably be attributed to compliance of the arm. The repeatability of the robot is only ±0.004". Hence, with the bias discussed in the above paragraph accounted for, the position accuracy of the robot is of the same order as the repeatability. Generally, the accuracy of a robot is an order of magnitude larger than its repeatability. It is seen that the amplitude of the vibrations observed when the robot is directed by the workcell controller are much larger than the other case, where only the controller of the robot is involved. This is because one command pulse fed to the Teach Unit results in a displacement of 0.063 mm. On the other hand, the resolution of the commands from the servo-controller on board the controller of the robot seems to be much smaller. However, it is also noted that the drift (dc offset) is lower for the workcell controller compared to that of the *AUTO* mode. The same phenomenon is observed for the other three sides.

The amplitude of the error in path tracking is observed to be high (up to 0.325 mm) for the sides BC and AD (Figure 4.11). For these sides, the displacement of the probe reflects primarily motion of θ-joint of the robot. The length of the radius arm from the centre of the main column of the robot (refer Figure 2.1) to the machining point is of the order of 700 mm. Therefore, even an error of 0.01 ° in θ-joint results in a position error as large as 0.12 mm in Y-axis (for this particular path). Therefore, it is extremely important that the error in this joint be as small as possible. One of the limitations of the current method used to direct the robot from the workcell controller is that no access is provided to the individual joints of the robot. Hence, direct control of θ-joint is not possible. The other limitation of the system is the command resolution of 0.063 mm and the processing time involved for the teach unit and controller to interpret a new command pulse rate and act accordingly.

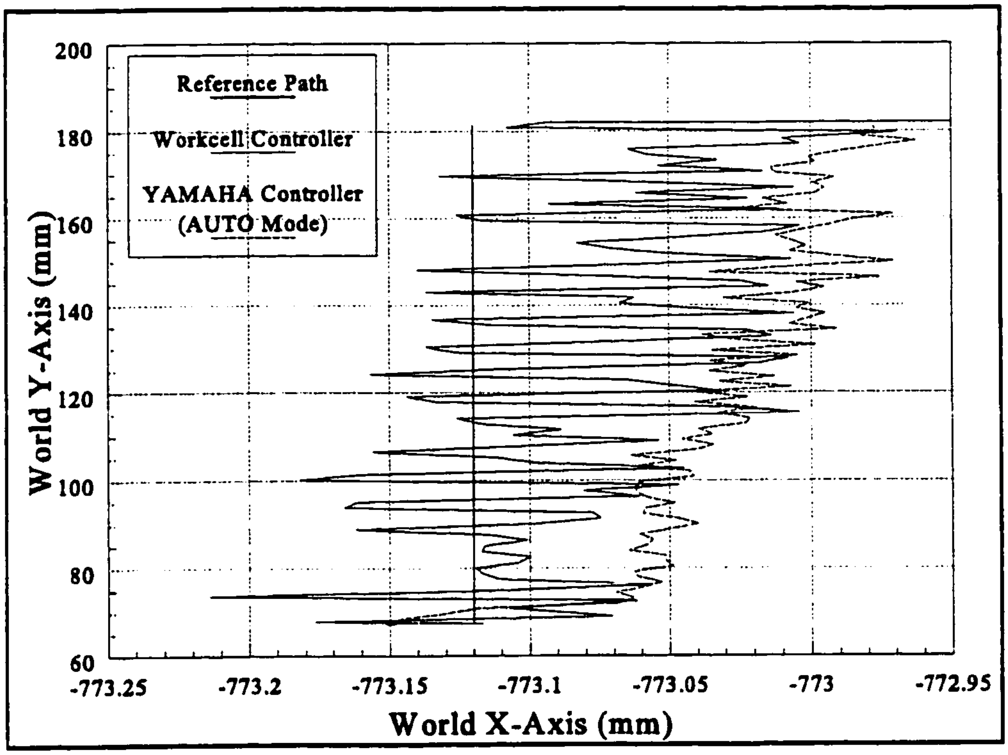

Figure 4.8 Path Tracking for Side AB : Speed 5 mm/s

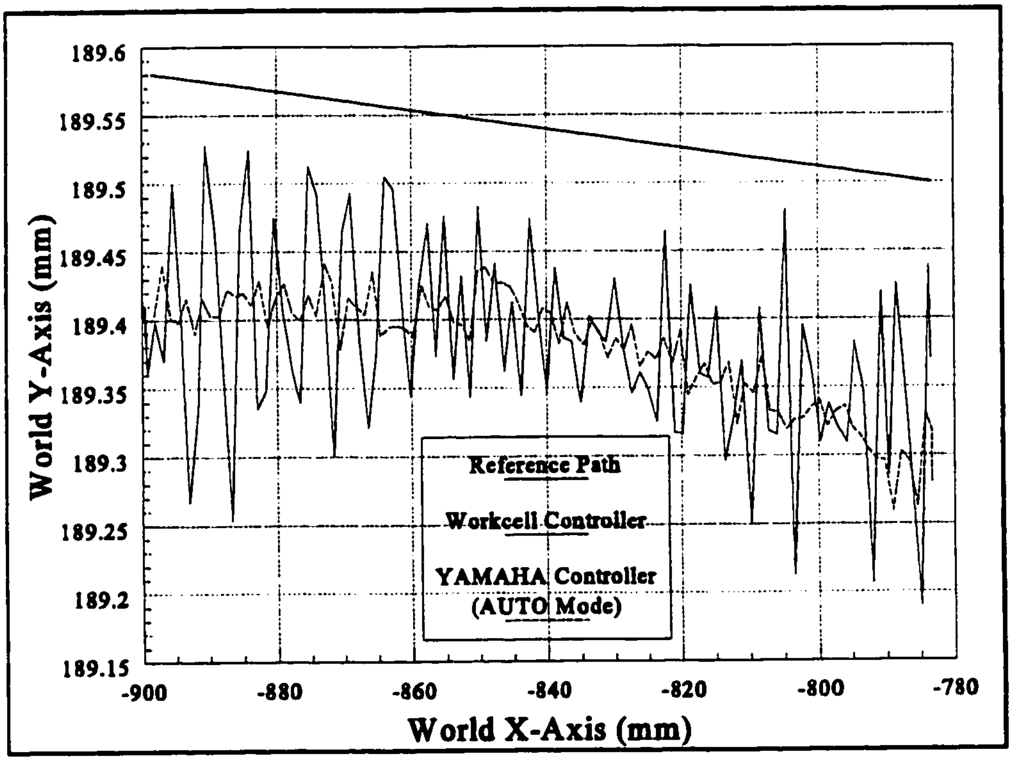

Figure 4.9 Path Tracking for Side BC : Speed 5 mm/s

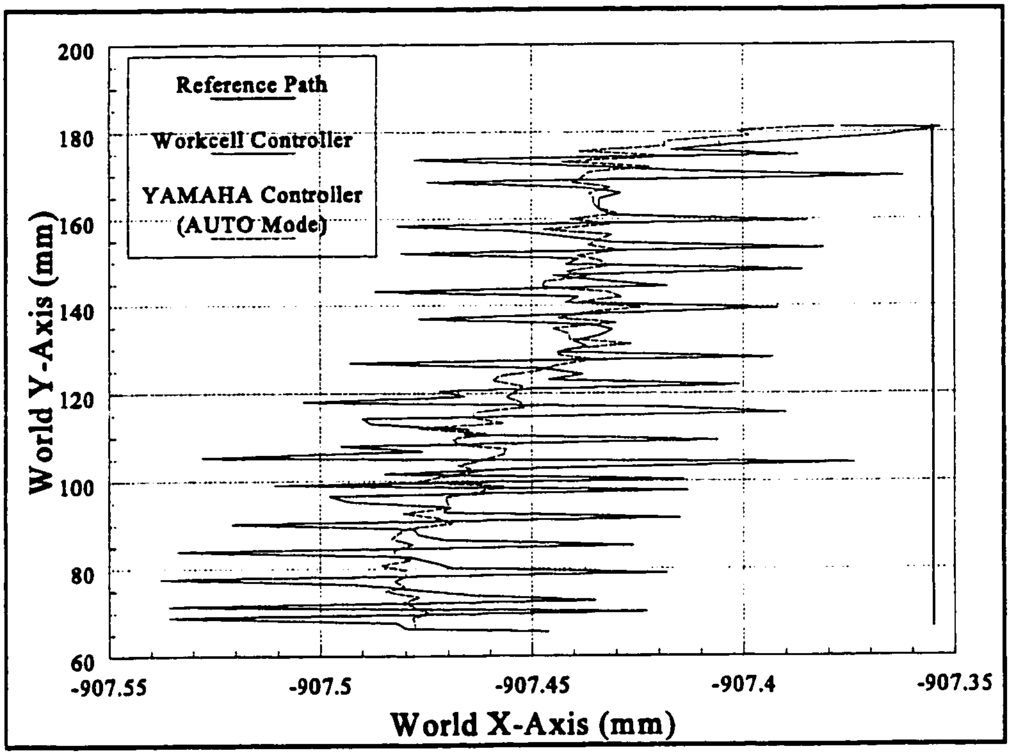

Figure 4.10 Path Tracking for Side CD : Speed 5 mm/s

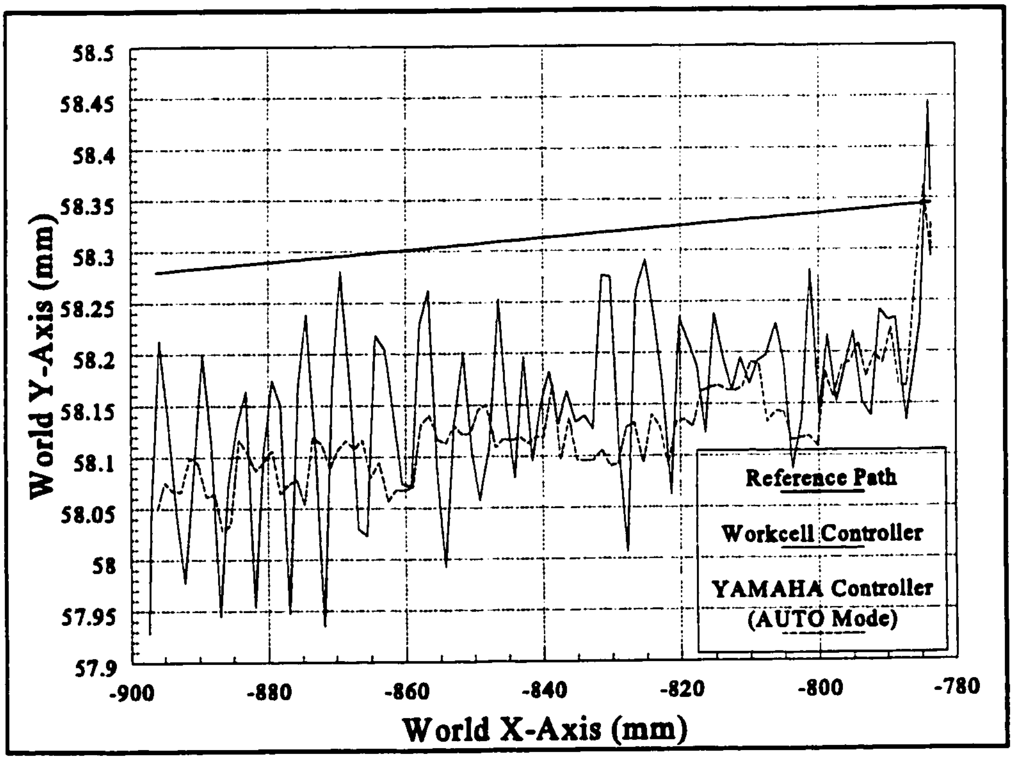

Figure 4.11 Path Tracking for Side AD : Speed 5 mm/s

However, while the magnitude of the oscillations (max 0.325 mm) observed in the case of the workcell controller is much larger, the system has now obtained the capability of communication and modification of the tool path in real-time. These oscillations appear to be a function of the dynamics of the robot and the command resolution. One way to increase the resolution of the command pulse is to decrease the scaling factor of the teach unit (ratio between displacement of joystick of teach unit and actual displacement of robot *machining point*). As seen from Equation 2.3, the resolution of the command pulse is directly proportional to the scaling factor S. The total distance travelled by the robot when commanded by a continuous stream of command pulses is limited by the scaling factor chosen. The higher the scaling factor, the larger the distance that can be travelled. At present, the scaling factor (Range: 0.01 to 1.00) is set as 0.5 to obtain as good a resolution as possible and still be able to move continuously through a distance of about 600 mm. For a smaller scaling factor, the maximum continuous distance which can be travelled would be proportionally smaller (say L_{MAX}). When a joystick of the teach unit is used to generate the command pulses, the joystick can move through only a finite distance in one direction. Hence, it is our understanding that the controller of the robot has been programmed to be able to process only a finite number of continuous stream of pulses. When the joystick is being used, a Synchronization button has to be kept pressed for the robot to follow the motion of the joystick. When the joystick has reached its limit, the Synchronization button is released. At this time the controller resets its registers used to keep track of the pulses being sent to it. If the robot is to be still advanced in the same direction as before, the joystick is brought back to its starting position and is moved again with the Synchronization button pressed. However, when the pulses are being generated by the low-level interface of the workcell controller, the registers of the robot controller are not being reset periodically. If the task *Control* can be modified to interrupt a continuous stream of command pulses, the robot controller can reset its count of command pulses. This can be accomplished by "switching off" the Synchronization button for a short time interval (0.2 s) whenever the robot has been commanded to move through a distance close to L_{MAX}. The robot can then be moved through the desired distance for smaller scaling factors leading to smaller command resolutions and thus smaller vibrations. Care has to be taken to "switch off" the Synchronization button only when the robot is moving in free space and not while it is machining. However, the lower the scaling factor, the lower is the value of the maximum velocity with which displacement in any World axis can be achieved. This constraint arises out of the fact that the teach unit does not accept command pulses of frequency higher than 2000 Hz. Hence, the magnitude of the scaling factor is then determined by the axis velocities involved in the desired trajectory.

In the long term, it is hoped that the performance of the robot would be considerably improved by incorporating a dynamics based control scheme with feedback from a force sensor mounted on the wrist of the robot. Such an algorithm will be very effective when the workcell is used for light machining applications. Tests are also being conducted by Mehrabi [4.2] to estimate the contribution of each joint of the robot to the instantaneous position error.

4.5 Summary

The trajectory tracking performance of the robot when directed by the workcell controller is tested for movement along individual axes. It is observed that, save for the period of acceleration and deceleration, the instantaneous trajectory tracking error for each axis is within the resolution of the command pulse fed to the Teach Unit of the deburring robot. It is also found that variations in the axis position error in World X and Y axes are less than 0.0061 mm for constant velocity paths over the workspace of the robot. These variations are much smaller than the smallest standard deviation experienced for either X or Y axis for a straight line path (0.0558 mm). When compared to path execution using the *AUTO* feature of the controller of the robot, it is found that the magnitude of the variations experienced when directed using the *AUTO* feature are 16% of the variations experienced when the robot is directed by the workcell controller. This is probably due to the difference between the resolution of the command pulse of the workcell controller and the resolution of the command pulse of the low-level servo-controller of the controller of the robot. One way to overcome this problem is to decrease the scaling factor of the teach unit from its present value of 0.5. The resolution of the command pulse is directly proportional to the scaling factor and hence, the performance of the robot will be improved. The interface with the displacement sensor has been tested succesfully, as it is used in the experiments described in Section 4.4.

Chapter 5

Conclusions and Recommendations for Future Work

5.1 Conclusions

An automated robotic deburring workcell has been developed. The five-axis robot chosen for the workcell is designed specifically for deburring and has a repeatability of ±0.1 mm. A displacement sensor, with an accuracy of 0.004 mm, is provided to enable the robot to probe the workpiece.

A workcell controller has been developed for coordinating the activities of the components of the workcell. The information processing part of the controller is implemented on a PC-Parallel Processor network. The network enables path planning of the robot, trajectory control of the robot, communication with the probe and surface reconstruction at the same time. A sampling period of 0.004 s is used for trajectory tracking of the robot and 0.252 s for the displacement probe.

Communication with the controller of the robot in real-time is achieved by mimicking the motion of the joysticks of the teach unit. This novel technique has eliminated the need for cannibalizing the controller and hence take advantage of its servo control utility and safety features. Since the path is generated and communicated in World coordinates, formulation of the inverse kinematic model of the robot is not required. This technique of mimicking the teach unit functions can be used to overcome the limitations of a closed architecture controller for other robots also.

Experiments for trajectory tracking of straight line paths have shown that the root mean square trajectory tracking error for any axis, at a typical machining speed of 5 mm/s, is 0.02 mm. In general, the trajectory tracking error, as computed from the joint encoder data, is less than the resolution of the command pulse fed to the Teach Unit.

The path tracking error (measured using the displacement probe), when the robot is controlled by the PC-transputer network through the Teach Unit, is compared with the error when the robot is controlled by the YAMAHA controller itself in *AUTO* mode. Owing to the command resolution being finer in the latter case, it is seen that the magnitude of deviations of the machining point of the robot from the desired path is smaller. It is also observed that, irrespective of whether the robot is controlled in *AUTO* mode or by the workcell controller, the path tracking errors, as seen from the probe readings, are much larger than those reflected in the joint encoder data. This could be possibly due to the compliance of the robot and could be rectified by implementing a dynamics-based control scheme for the deburring robot.

5.2 Recommendations for Future Work

The resolution of the command pulse generated by the workcell controller is ±0.063 mm for X, Y and Z axes and ±0.09° for A and B axes. As discussed in pages 78 and 79, the commands resolution can be increased by reducing the scaling factor S from its present value of 0.5. The maximum distance travelled by the robot when fed a continuous stream of pulses is limited by the magnitude of S. This limitation can be overcome by "switching" off and on the synchronization button available on the teach unit to reset some registers in the robot controller. Thus both an improved resolution and the ability to debur edges over reasonably large distances can be achieved. Further, a finer command pulse resolution will reduce the magnitude of the hunting observed ofthe machining point leading to a better finish while deburring.

At present, generation of intermediate set points (to be sent to the task *Control*) by the path planning module of the task *Master* is implemented only for straight line paths. Tracking edge of a biquadratic saddle surface has been performed successfully [4.1] with errors less than the command pulse resolution. But the orientation of the tool was kept constant during this procedure. In order to maintain the orientation of the tool perpendicular to the surface at all times, the path planning module of the task *Master* shculd incorporate splining of A and B axes. There should be an option for switching off the splining operation, if the user feels that the path primarily comprises of straight line segments. For every set point obtained in Cartesian space from splining, the correct A and B coordinates have to be computed. Computation of the correct orientation for every intermediate set point is discussed by Kabra [3.3].

One of the highlights of this thesis is the successful implementation of a channel for trajectory control of the robot from an external workcell controller. The next step, to make the deburring workcell completely functional, is to enable the control of the tool from the PC-transputer network. Two motion controllers (LM 629) have been provided on the interface card used by the PC-transputer network. The servo motor for the electric tool can be controlled by the workcell controller through one of the motion controllers.

A sixth degree of freedom is provided with the deburring robot in the form of an NC Table on which the workpiece is mounted. This table can be indexed with a resolution of

0.01 °. The second additional motion controller provided on the interface card can be used to directly control the servo motor of the NC-Table from the PC-transputer network.

Experiments conducted for straight line paths with no machining show the performance of the PI control law to be satisfactory. Once, the deburring tool is interfaced with the workcell controller, experiments should be conducted to evaluate the performance of the controller while machining and the control law might need to be changed.

The sensing of the machining forces involved in deburring and the on-line modification of the control signal based on this information will enhance the performance of the deburring workcell. To enable sensing of the machining forces, a force sensor should be incorporated into the workcell. A force sensor is also necessary to detect extremely large burrs requiring large machining forces. When such burrs are detected, the tool path could be modified, as done by the vendor-supplied robot controller, to execute multiple passes with reduced depth of cut for that portion of the tool path.

In the long term, a dynamics based trajectory controller for the robot might be a solution to the path tracking errors discussed in the previous section. The future implementation of such a computation-intensive algorithm has been one of the motivations behind developing a supervisory controller with parallel processing capability. However, this would require replacing the existing controller interface and the Yamaha controller, with an in-house controller capable of controlling the torque exerted by the joints of the robot.

References

Chapter 1

[1.1] Scheider, A.F., *Mechanical Deburring and Surface Finishing Technology*, Marcel Dekker Inc., New York, 1990.

[1.2] Gillespie, L.K., *Robotic Deburring Handbook*, SME Publication, 1987.

[1.3] Proctor, F.M., Murphy, K.N., *Keynote Address : Advanced Deburring System Technology*, Mechanics of Deburring and Surface Finishing Processes : ASME Winter Annual Meeting, San Francisco, California, pp. 1-12, December 1989.

[1.4] Hollowell, R., Guile, R., *Analysis of Robotic Chamfering and Deburring*, Proc. ASME Winter Annual Meeting, Boston, Massachusetts, December 1987.

[1.5] Bone, G.M., Elbestawi, M.A., *Sensing and Control for Automated Robotic Edge Deburring*, Proc. 1992 International Conference on Industrial Electronics, Control, Instrumentation and Automation (IECON), San Diego, California, November 1992.

[1.6] Norcross, R.J., *A Control Structure for Multi-tasking Workstations*, IEEE Conf. Robotics and Automation, Philadelphia, Pennsylvania, April 1988.

[1.7] Murphy, K.N., Norcross, R.J., Proctor, F.M., *CAD Directed Robotic Probing*, Second International Symp. on Robotics and Manufacturing Research, Education, and Applications, Albuquerque, November 1988.

[1.8] Proctor, F.M., Murphy, K.N., Norcross, R.J., *Automating Robot Programming in the Cleaning and Deburring Workstation of the AMRF*, SME Deburring and Surface Conditioning, San Diego, California, February 1989.

[1.9] Kramer, B.M., Shim, S.S., *Development of a System for Robotic Deburring*, Robotics and Computer-Integrated Manufacturing, Vol. 7, No. 3/4, pp. 291-295, 1990.

[1.10] Selleck, C.B., Loucks, C.S., *A System for Automated Edge Finishing*, Proc. 1992 IEEE International Conference on Systems Engineering, Pittsburgh, pp. 423-429, 1990.

[1.11] Her, M.G., Kazerooni, K., *Automated Robotic Deburring of Parts Using Compliance Control*, ASME Journal of Dynamics Systems, Measurement and Control, Vol. 113, No. 1, pp. 60-66, 1991.

[1.12] Cheng, R.M.H., Rajagopalan, R., Temple-Raston, M., *The Differential Geometric Modelling of Compressor Blades*, Proc. American Control Conference, Baltimore, Maryland, pp. 1913-1917, June 1994.

Chapter 2

[2.1] Groover, M.P., Weiss, M., Nagel, R.M., Odrey, N.G., *Industrial Robotics : Technology, Programming, and Applications*, McGraw-Hill, New York, 1986.

[2.2] Murphy, K.N., Norcross, R.J., Proctor, F.M., *CAD Directed Robotic Probing*, Second International Symp. on Robotics and Manufacturing Research, Education, and Applications, Albuquerque, November 1988.

[2.3] Bone, G.M., Elbestawi, M.A., *Sensing and Control for Automated Robotic Edge Deburring*, Proc. 1992 International Conference on Industrial Electronics, Control, Instrumentation and Automation (IECON 92).

[2.4] Nicoletti, G.M., *Adaptive Control for Robotic Deburring*, Technical Paper No.

MR91-132, Society of Manufacturing Engineers (SME), Dearbom Michigan, March 1991.

[2.5] Whitney, D.E., Edsall, A.C., Todtenkopf, A.B., Kurfess, T.R., Tate, A.R., *Development and Control of an Automated Robotic Weld Bead Grinding System*, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 112, pp. 166-176, June 1990.

[2.6] Kramer, B.M., Bausch, J.J., Gott, R.L., Dombrowski, D.M., *Robotic Deburring*, Robotics & Computer-Integrated Manufacturing, Vol. 1, No. 3/4, pp. 365-374, 1984.

[2.7] Haefner, K.B., Houpt, P.K., Baker, T.E., Dausch, M.E., *Real Time Robotic Position/Force Control for Deburring*, Proc. ASME Winter Annual Meeting, Dynamic Systems and Control Division, Califomia, pp. 73-78, December 7-12, 1986.

[2.8] Kramer, B.M., Shim, S.S., *Development of a System for Robotic Deburring*, Robotics and Computer-Integrated Manufacturing, Vol. 7, No. 3/4, pp. 291-295, 1990.

[2.9] Selleck, C.B., Loucks, C.S., *A System for Automated Edge Finishing*, Proc. 1992 IEEE International Conference on Systems Engineering, Pittsburgh, pp. 423-429, 1990.

[2.10] Stauffer, R.N., *Sensor Simplifies Changeover in Deburring Operation*, Robotics Today, Vol. 9, No. 4, August 1987.

[2.11] DeCamp, W.H., *Breaking the Edge*, Manufacturing Engineering, July 1989.

[2.12] Shoham, M., Srivatsan, R., *Automation of Surface Finishing Processes*, Robotics and Computer-Integrated Manufacturing, Vol. 9, No. 3, pp. 219-226, 1992.

[2.13] *YAMAHA Zeta-1 Deburring Robot : User's Manual*, Version 1.2, YAMAHA

Corporation, Hamamatsu, January 1990.

[2.14] Hewitt, R.J., *New Concepts in Robotic Deburring & Polishing*, Hammond Machinery Inc., Kalamazoo, Michigan, 1989.

[2.15] *Programming Manual : User's Guide to VAL II*, Unimation Inc., Danbury, Connecticut, August 1984.

[2.16] Her, M.G., Kazerooni, K., *Automated Robotic Deburring of Parts Using Compliance Control*, ASME Journal of Dynamics Systems, Measurement and Control, Vol. 113, No. 1, pp. 60-66, 1991.

[2.17] *Operation Manual for IDC Series 543 Digimatic Indicator*, Manual No. 3041, Mitutoyo Corporation, Tokyo, Japan.222

[2.18] *KAMAN Instrumentation Manual*, KAMAN Sciences Corporation, Colorado Springs, Colorado 80933, 1994.

[2.19] Cheng, R.M.H., Rajagopalan, R., Temple-Raston, M., *The Differential Geometric Modelling of Compressor Blades*, Proc. American Control Conference, Baltimore, Maryland, pp. 1913-1917, June 1994.

[2.20] Lim, F.P., *The DRAW Utility for Real-time Display of Robotic Probing and Deburring*, Internal Report # CIC-0054, Centre for Industrial Control, Concordia University, Montreal, September 1993.

[2.21] Ulsoy, A.G., Koren, Y., *Control of Machining Processes*, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 115, No. 2(B), pp. 301-308, June 1993.

[2.22] Poon, S.C.L., Huard, G., *Design of a General Purpose Interface between a Serial Device and a 8-bit Parallel Device*, CIC Report # 0028, Centre for Industrial

Control, Concordia University, Montreal, November 1989.

[2.23] *Microprocessor and Peripheral Handbook : Volume II Peripheral*, Intel Inc., Santa Clara, California, 1988.

[2.24] *PAL Device Handbook*, Advanced Micro Devices Inc., Sunnyvale, California, 1988.

<u>*Chapter 3*</u>

[3.1] *Parallel C User Guide*, 3L Ltd., Edinburgh, Scotland, July 26, 1991.

[3.2] Lim, F.P., *The DRAW Utility for Real-time Display of Robotic Probing and Deburring*, Internal Report # CIC-0054, Centre for Industrial Control, Concordia University, Montreal, September 1993.

[3.3] Kabra S., *The Development and Analysis of an Integrated Workcell Controller for Multiple-Robot Synchronisation*, M.A.Sc Thesis, Department of Mechanical Engineering, Concordia University, July 1992.

[3.4] Ziegler, J.G., Nichols, N.B., *Optimum Settings for Automatic Controllers*, Transactions of the ASME, pp. 759-768, November 1942.

[3.5] *YAMAHA Zeta-1 Deburring Robot : User's Manual*, Version 1.2, YAMAHA Corporation, Hamamatsu, January 1990.

[3.6] Cheng, R.M.H., Rajagopalan, R., Temple-Raston, M., *The Differential Geometric Modelling of Compressor Blades*, Proc. American Control Conference, Baltimore, Maryland, pp. 1913-1917, June 1994.

[3.7] Cheng, R.M.H., Poon, S.C.L., Rajagopalan, R., *Interrupt Driven Transputer File Server*, Proc. Fourth Conference of the North American Transputers Users Group, Ithaca, pp. 119-130, 1990.

Chapter 4

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

[4.2] Mehrabi, M., *Preliminary Test Results for the Accuracy of YAMAHA Zeta-1 Robot*, CIC Internal Report, Centre for Industrial Control, Concordia University, Montreal, 1994.

Appendix A

Forward Kinematic Model of YAMAHA Zeta-1 Robot

The derivation of the forward kinematic model (Equation 3.6 on page 49) of Zeta-1 robot is given in this appendix. The model gives the transformation from machine coordinates space {θ, R, z, α, β} to World coordinate space {x.y, z, a, b}. Both coordinate systems are shown in Figure 2.1 (on page 12). The joints θ, R and Z affect the position coordinates {x, y, z} of the machining point. The joint θ and the two wrist joints α and β affect the orientation of the tool specified by the coordinates {a,b}. The effect of θ, R and Z on {x, y, z} is given by the transformation from a cylindrical coordinate system to a Cartesian coordinate system as given below :

$$\begin{aligned} x &\cdot R\cos\theta \\ y &\cdot R\sin\theta \\ z &\cdot z \end{aligned} \qquad \text{(A1)}$$

It can be seen from Figure 2.1 that the axes for machine coordinates θ and α and World coordinate a are parallel and oriented in the same direction. Hence contribution of the two above mentioned joints to World coordinate a is just the algebraic sum of rotations about the two joints. The contribution of the joint β to World coordinates a and b is not straightforward. The effect can be derived by finding out the change in coordinates of point P in Figure A1 of a rotation about axis β. Only coordinate frames which have been used for

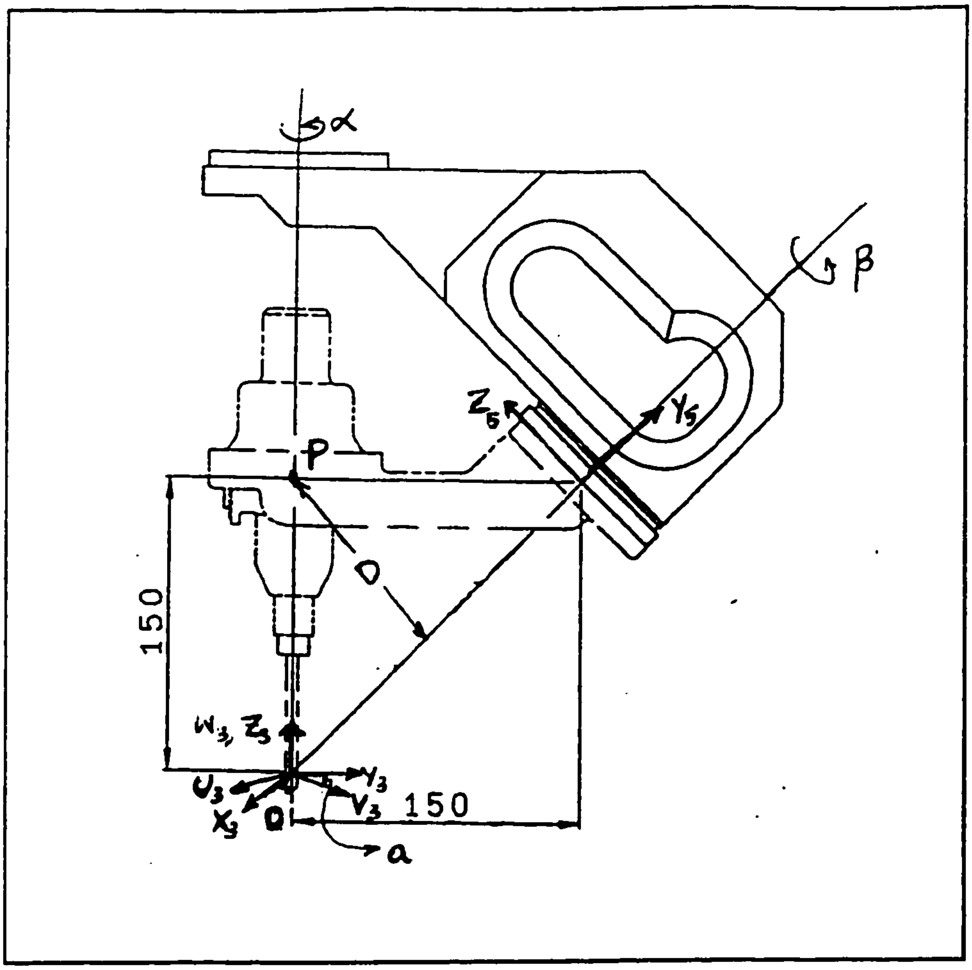

Figure A1 Wrist and Tool Holder of YAMAHA Zeta-1 Robot

the analysis given below are shown in Figure A1. The frame $\{x_3, y_3, z_3\}$ is stationary with respect to the joint α and has its origin at the machining point. The frame $\{x_5, y_5, z_5\}$ is stationary with respect to the tool. The numbering of coordinate frames is in accordance with the convention followed by Fu et al[1]. The angle a is defined as the angle between y_3 axis and the projection of the line OP in Figure A1 on x_3y_3 plane. When $\beta=0$, it is seen that the coordinates in frame $\{x_5, y_5, z_5\}$ are given by $\{0, -D, D\}^T$ where D is shown in Figure A1.

[1] Fu, K.S., Gonzalez, R.C., Lee, C.S.G., *Robotics : Control, Sensing, Vision and Intelligence*, Mc-Graw Hill International Editions, 1987.

When α=0, rotation about β-axis implies :

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 45 & -\sin 45 & \sqrt{2}D \\ 0 & \sin 45 & \cos 45 & \sqrt{2}D \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -D \\ D \\ 1 \end{bmatrix} \tag{A2}$$

After simplifying Equation A2, the following equation is obtained :

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} D \sin\beta \\ \frac{D}{\sqrt{2}}(1 - \cos\beta) \\ \frac{D}{\sqrt{2}}(1 \cdot \cos\beta) \\ 1 \end{bmatrix} \tag{A3}$$

The change in coordinate a due to β-joint is given by the following equation :

$$a \cdot \arctan\left(\frac{y_3}{x_3}\right) \cdot \arctan\left(\frac{1 - \cos\beta}{\sqrt{2}\sin\beta}\right) \tag{A4}$$

Hence, the effect on World coordinate a due to joints θ, α and β is described as follows :

$$a \cdot \theta \cdot \alpha \cdot \arctan\left(\frac{1 - \cos\beta}{\sqrt{2}\sin\beta}\right) \tag{A5}$$

The effect of β-joint on World coordinate B is obtained by describing the coordinates of point P in terms of coordinate system $\{u_3, v_3, w_3\}$. This coordinate system is obtained by rotating P about z_3 by the angle -a calculated in Equation A4. Hence, we have the following equation :

$$\begin{bmatrix} u_3 \\ v_3 \\ w_3 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \cos a & \sin a & 0 & 0 \\ -\sin a & \cos a & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D \sin \beta \\ \frac{D}{\sqrt{2}}(1 - \cos \beta) \\ \frac{D}{\sqrt{2}}(1 \cdot \cos \beta) \\ 1 \end{bmatrix} \tag{A6}$$

Equation A6 is simplified to obtain :

$$\begin{bmatrix} u_3 \\ v_3 \\ w_3 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{D}{\sqrt{2}} \sqrt{2 \sin \beta^2 \cdot (1 - \cos \beta)^2} \\ 0 \\ \frac{D}{\sqrt{2}}(1 \cdot \cos \beta) \\ 1 \end{bmatrix} \tag{A7}$$

The change in b-coordinate to motion of β-joint is then obtained as :

$$b \cdot \arccos\left(\frac{1 \cdot \cos \beta}{2} \right) \tag{A8}$$

Equations A1, A5 and A8 together give the forward kinematic model of the robot.