University of Manitoba

DEVELOPMENT OF A FLEXIBLE, MICROCOMPUTER BASED, THREE-AXIS MACHINE TOOL CONTROLLER

by

Timothy M. Kostyniuk

A thesis
submitted to the Faculty of Graduate Studies
in partial fulfillment of the
requirements for the degree of
Masters of Science
in
Mechanical Engineering

Winnipeg, Canada, 1988

ISBN 0-315-47846-2

DEVELOPMENT OF A FLEXIBLE, MICROCOMPUTER BASED, THREE-AXIS MACHINE TOOL CONTROLLER

BY

TIMOTHY M. KOSTYNIUK

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

I hereby declare that I am the sole author of this thesis.

I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Timothy M. Kostyniuk

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Timothy M. Kostyniuk

The University of Manitoba requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

A flexible, microcomputer based, three-axis machine tool controller is developed and described. The controller consists of a microcomputer, a software control program, and a custom-made motion control card to interface the axes' motors with the microcomputer. The software based controller performs sampled-data control and it is implemented in FORTRAN and ASSEMBLER. This software provides much greater flexibility than a traditional hardwired controller. For example, software routines for multi-axis linear and circular interpolation, and backlash compensation are incorporated easily. In addition, a commercial part design system is interfaced to the machine tool controller to further demonstrate its flexibility.

A manual three-axis milling machine is modified for use as a CNC machine in order to evaluate the performance of the new machine tool controller. A software based, proportional-integral-derivative (PID) compensator is employed to control the positions of the three axes of the milling machine. The results of linear and circular accuracy tests indicate that the position controller itself achieves the desired machining accuracy of +/- 0.0005 inches. Structural characteristics such as vibrations and pitch error in the leadscrew, however, are found to degrade the attainable machining accuracy.

ACKNOWLEDGEMENTS

The author would like to thank Drs. S. Balakrishnan and N. Popplewell for acting as thesis advisors. Their advice, guidance, and dedication are appreciated gratefully.

The author would like to also thank Mr. V. Shkawrytko and Mr. J. Sill for their work on the motion control interface. In addition, special thanks go to the technical support staff of the Mechanical Engineering Department for their various contributions to the project.

The author extends thanks to Mr. C. Gadsby who introduced the thesis project. Thanks and best wishes are also extended to Mr. R. Toutant who performed the experiments on the linear encoder and who will be pursuing further research in the adaptive control of machine tools.

The donation of an IBM compatible PC/AT computer by Mind Computers of Winnipeg for the project is appreciated gratefully. Finally, the financial support of the Natural Sciences and Engineering Research Council in the form of a Postgraduate Scholarship is acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

ACL = acceleration rate (inches/second^2)

AREA1 = distance travelled during the acceleration phase of a movement (inches)

AREA2 = distance travelled at constant speed (inches)

AREA3 = distance travelled during the deceleration phase of a movement (inches)

AREA3M = modified AREA3 to account for quantization of the speed profile (inches)

$AREA_{quantized}$ = quantization error caused by the discrete nature of the acceleration and deceleration phases of the speed profile (inches)

C(s) = input to a control system

DCL = deceleration rate (inches/second^2)

DCLMOD = modified deceleration rate to account for $AREA_{quantized}$ (inches/second^2)

DCOSX = direction cosine in the X direction

DCOSY = direction cosine in the Y direction

DCOSZ = direction cosine in the Z direction

DELTA = total travel distance (inches)

DELTAX = travel distance in the X direction (inches)

DELTAY = travel distance in the Y direction (inches)

DELTAZ = travel distance in the Z direction (inches)

ζ = damping ratio

e = error input to a PID compensator

E = contour error (inches)

E_x = position following error for the X axis (inches)

E_y = position following error for the Y axis (inches)

E_z = position following error for the Z axis (inches)

f_s = sampling frequency in the position control loop (Hertz)

G(s) = open-loop transfer function of a second-order control system

$G_c(s)$ = transfer function of a PID compensator

$G_f(s)$ = machine drive's transfer function

$G_m(s)$ = servo drive's transfer function (radians/(second-volt))

$G_t(s)$ = overall transfer function of the compensated position control system

H = feedback element in a control system

I = X axis offset for circular interpolation (inches)

J = Y axis offset for circular interpolation (inches)

K = Z axis offset for circular interpolation (inches)

k = summation index

K_d = derivative gain of a PID compensator

K_{dac} = digital-to-analog converter's gain (volts/pulse error)

K_e = position encoder's gain (pulses/radian)

K_i = integral gain of a PID compensator

K_m = motor-amplifier's gain (radians/(second-volt))

K_p = proportional gain of a PID compensator

K_t = total gain of the position control loop ($second^{-1}$)

n = sample number

PT1 = number of interrupts required for acceleration

PT2 = number of interrupts at constant speed

PT3 = number of interrupts required for deceleration

PT3MOD = modified number of interrupts required for deceleration

q = output of a PID compensator

RATE = specified feedrate (inches/minute)

R(s) = output from a control system

$R0$ = position of an axis at the next part program block (inches)

$R1$ = position of an axis at the present program block (inches)

$R2$ = position of an axis at the previous part program block (inches)

s = Laplace operator ($second^{-1}$)

θ_s = angle of the position vector at the start of a sample during circular interpolation (radians)

θ_m = angle of the feed vector with respect to the angle of the position vector during circular interpolation (radians)

τ_m = servo system's mechanical time constant (seconds)

τ_e = servo system's electrical time constant (seconds)

T_s = sampling period (seconds)

$V(s)$ = amplifier's voltage input (volts)

V_x = velocity of the X axis (inches/minute)

V_y = velocity of the Y axis (inches/minute)

V_z = velocity of the Z axis (inches/minute)

$w(s)$ = motor's speed (radians/second)

ω_1 = reciprocal of the servo system's mechanical time constant (radians/second)

ω_c = crossover frequency (radians/second)

ω_n = underdamped natural frequency of a second-order control system (radians/second)

X = position of the X axis (inches)

Y = position of the Y axis (inches)

Z = position of the Z axis (inches)

CHAPTER 1
INTRODUCTION

1.1 Background

Computer numerical controlled (CNC) machine tools like milling machines and lathes offer significant cost savings over traditional, manually operated machines. A CNC system can achieve greater uniformity and higher machining accuracy than a manually operated machine [1]. In addition, a CNC system can perform accurate multi-axis linear or circular machining, which is generally not possible on a manual machine. Finally, the part design and manufacturing processes are simplified greatly when CNC technology is used in conjunction with a computer-aided design (CAD) support package that generates a part program code. Affordable microcomputer based CAD systems are now available due to recent advances in computer technology. The interactive graphics of these CAD systems permit part design, modification, and verification on a monitor screen prior to the actual machining.

The initial cost of CNC systems is high compared to that of manual machines. CNC machines can range in price from about $20,000 to several hundred thousand dollars. Small and medium sized manufacturers have a difficult time justifying the initial cost of CNC systems when their manually operated systems are still in good working condition. An economical alternative to purchasing new CNC equipment is the conversion of a manual machine to CNC. Retrofitting a manual machine for CNC can be done usually at less than 30% of the cost of comparable new equipment [2]. Therefore, many manufacturers would prefer to upgrade their manually operated machines to CNC as opposed to buying completely new CNC machines. A networking capability is also a desirable feature to incorporate in retrofit applications. This feature would enable several machines to be integrated into a more efficient manufacturing system. For example, several CNC machines could be networked into a manufacturing system with one CAD workstation used for designing parts and then downloading the part programs to the individual machines.

In recent times, hardware based, commercial motion control systems have been introduced into the marketplace. Examples include the Compumotor™ PC-23 Indexer [3-5], which is a microprocessor-based three-axis position controller, and the PRO-400 Stored Motion Profile Controller from Electro-Craft Corporation [6]. These systems have many user-friendly features, such as programming capabilities and adjustable controller gains. However, they are stand-alone or dedicated controllers which do not have a CAD system interface or a networking capability and they must be programmed by using a non-standard CNC programming language. With stand-alone controllers, it is necessary to manually enter the coordinates specifying a geometry through a keypad. If a design must be changed significantly, then this time consuming process must be repeated.

Commercial controllers are generally supplied as complete systems with hardwired control functions. A hardware based controller generally operates at higher sampling frequencies than a more flexible software based controller. Higher sampling frequencies permit higher performance operation in a wider variety of applications such as robotics. However, the form of the hardwired controller cannot be changed easily. It is desirable to have the flexibility to customize a controller to suit the requirements of a particular machine tool application. A flexible machine tool controller should allow the use of different control schemes which can vary from simple contouring to adaptive control capabilities. For example, adaptive control could be used to manipulate the cutting feedrate to maintain the maximum permissible cutting torque. Adaptive control increases the efficiency of a machine tool permitting a higher production output. Finally, it is critical in high accuracy machine tool applications to compensate for machine errors such as backlash and leadscrew pitch error in the drive system. Such features are not usually offered on commercial motion control systems.

To provide flexibility and to customize a controller to suit the requirements of a particular application, it was decided to develop a flexible, microcomputer based, motion control system. The motion controller should accept standard machine tool part program codes to facilitate a CAD interface. As well, a machine tool controller with an open controller design is a desirably powerful tool because it can be integrated easily into a computer controlled manufacturing system of the future.

A flexible motion control system consists of a microcomputer, a software control program, and a custom motion control card to interface the axes' motors with the computer. Such a flexible motion controller should use software based control algorithms which can be developed and modified for a wide variety of applications without any hardware changes. Before examining the development of such a system, the classification, structure, and requirements of a machine tool controller will be described next.

1.2 Classification of CNC Machines

CNC machines may be classified as point-to-point or contouring (continuous path) systems [7-9]. A point-to-point machine is one in which the final position of a programmed move is important, and not the complete path to that position. A drilling machine is a typical example of a point-to-point application in which the CNC controller moves the table to a specified point before drilling is started. Contouring operations, on the other hand, are performed on machine tools such as milling machines and lathes. The primary objective of any CNC contouring system is to generate a coordinated movement of separately driven axes in order to accurately machine a specified path of the cutting tool relative to the workpiece. In contouring operations, such as the machining of a circular arc, accuracy must be maintained over the whole path because material removal occurs over the complete motion.

CNC machines may be also classified according to the type of control loop employed. The position control loop can be either an open- or a closed-loop [10]. As shown in Figure 1.1, an open-loop controller generates the desired position along an axis but does not receive a feedback signal to determine the actual position. An example of such a system is a stepping motor driven, CNC machine. This type of open-loop control lowers the performance of a CNC system because a change in load, amplifier gain, or any other system variable will cause a deviation from the desired position. In

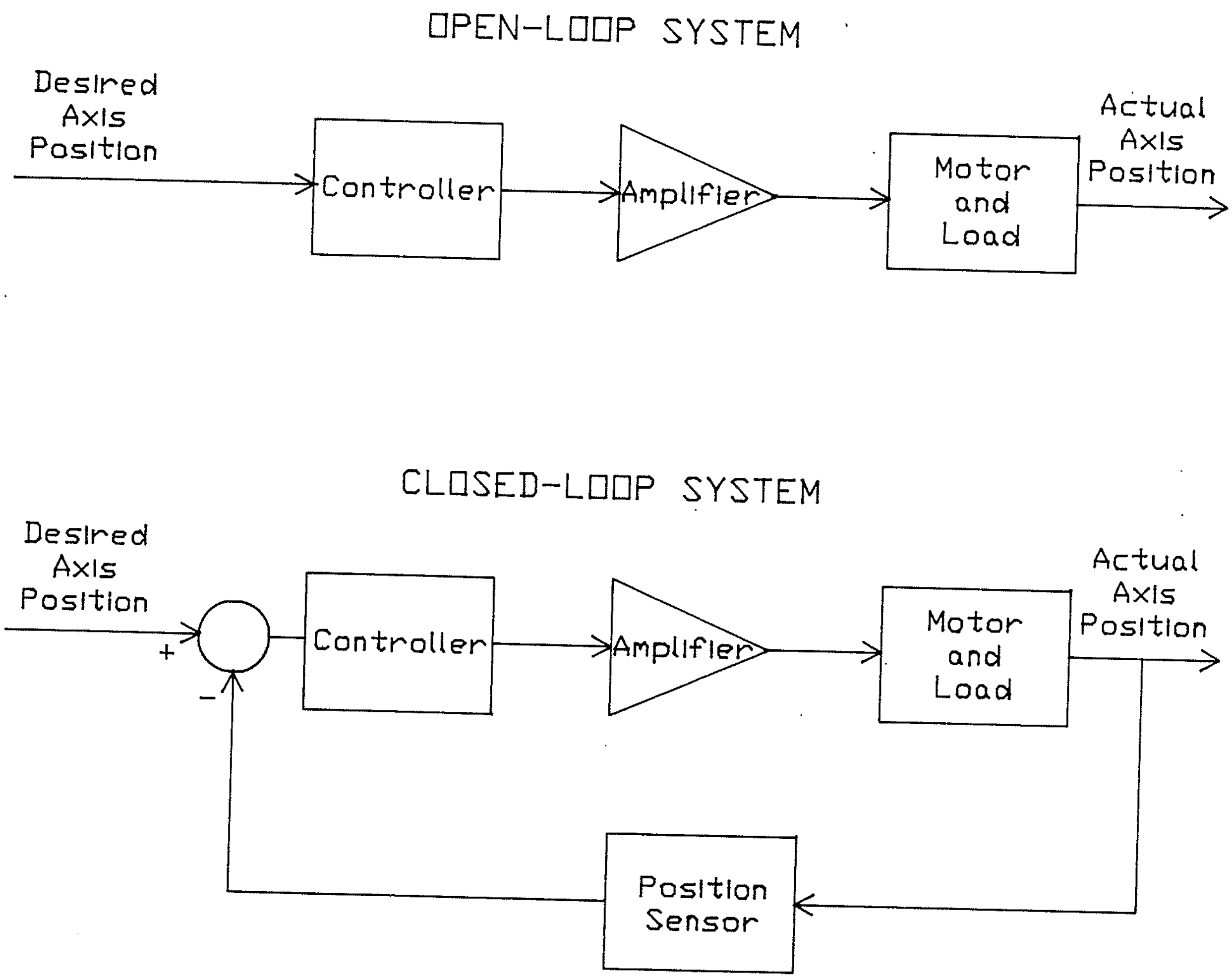


Figure 1.1 Open- and Closed-Loop Position Control Systems

contrast, closed-loop controllers use feedback of the controlled variable to achieve a more reliable and higher level of performance. This thesis will be concerned with the development of an affordable, microcomputer based, closed-loop, contouring CNC system.

1.3 Structure of a CNC Control System

A CNC machine tool consists of a machine control unit and the machine tool itself. The machine control unit can be subdivided further into two main components, namely, a data processing unit (DPU) and a control loops unit (CLU) [7]. The data processing unit decodes a part program, processes it, and supplies data to the CLU. The part program specifies the toolbit path required to produce a desired part design. Each line of the part program is called a block. A block specifies the end location, machining feedrate, and other machining parameters for the desired linear or circular toolbit path. The process of simultaneously coordinating the motion of several axes to produce a desired contour is called interpolation. The DPU generates the intermediate or interpolated positions between the endpoints of a contoured move. For example, a circular arc is approximated by many small line segments. The DPU outputs the required position changes, for each axis, to achieve the desired contour from the starting point to the end point of the movement.

The control loops unit, CLU, uses the speed and position determined by the DPU to drive the axes' motors so that the toolbit moves along the desired path. The CLU also receives feedback from each axis to determine the actual position along each axis. The difference between the actual position and the required or reference position is termed the position error. The CLU uses the position error to determine the speed signal which should be used to drive the amplifier and the motor. The control function of the CLU can range from the traditional lead-lag or proportional-integral-derivative (PID) control compensation to adaptive control [11-15]. The traditional CLU consists of hardwired circuits for position and velocity control [7]. However, it is preferable to implement the control functions of the CLU in software to provide more flexibility.

1.4 Requirements for a Flexible Machine Tool Controller

The controller in a machine tool application must satisfy several requirements with respect to machining accuracy and standard CNC features. The requirements for a cheap, flexible machine tool controller include:

a) Microcomputer Based - The availability and affordability of microcomputers makes them ideal controllers for machine tools. The microprocessor of the computer uses a software implemented, control algorithm to sample and control the positions of the axes of a CNC system.

b) Accuracy - The controller of the machine tool must accurately control the positions of the various axes to machine a specified path of the cutting tool relative to the workpiece. The machining accuracy is determined not only by the controller's ability to command the machine to follow a desired path, but also by the mechanical characteristics of the machine tool itself. Mechanical characteristics such as backlash, pitch error in the leadscrew, and machine tool vibrations can adversely affect accuracy.

c) Software Flexibility - It is necessary to have access to the software source code in order to develop custom control features. For example, suitable position and velocity control loop compensators can be implemented in software to achieve a desired machining accuracy and performance. Open software greatly increases the ability to network the CNC machine tool to other machines and to a CAD system. Furthermore, it is possible to easily add features such as backlash compensation with open software.

d) Interpolation Capabilities - A multi-axis CNC controller must be able to perform a coordinated motion of several axes. Coordinated three-dimensional linear interpolation and circular interpolation on any two axes are standard capabilities of CNC machines. Open software also permits more complex interpolation routines such as spline interpolation to be implemented. Such flexibility is usually not possible on commercial systems.

e) Part Programming Capabilities and CAD System Interface - It is desirable to have a machine tool controller which has part programming capabilities and a CAD interface. This feature reduces part design and manufacturing costs because parts can be designed and modified on the CAD system prior to machining.

1.5 The Goals of the Thesis

The goal of the work reported in this thesis was to design and construct a flexible and inexpensive microcomputer based, machine tool controller. It was desired to keep the hardware cost of the controller below $5000, a price which excludes the servo motors and amplifiers. Common off-the-shelf hardware was utilized as much as possible to minimize the cost. Existing computer languages and compilers were used to minimize the time for software development. Based on the experience gained from previous work [16], software was implemented in FORTRAN and ASSEMBLER. FORTRAN was used for its powerful mathematical capabilities and ease of programming the controller and interpolation routines. On the other hand, ASSEMBLER subroutines were employed for low level data input and output between the computer and hardware devices.

A manual three-axis milling machine was used to evaluate the performance of the flexible machine tool controller. The software controller was developed to satisfy typical performance requirements in milling operations. The target positional and contouring accuracies were +/- 0.0005 inches under loaded conditions. Software routines were developed for three-dimensional linear interpolation and circular interpolation on any two axes. As well, a software backlash takeup routine was developed to compensate for the leadscrew backlash whenever a direction reversal occurs on any axis. Finally, to achieve the full benefits from a CNC system, the developed machine tool controller was interfaced to an existing, commercial part design system.

The techniques used to model and design the drive controller will be introduced in Chapter 2. The design and implementation of both the hardware and software will be discussed in Chapter 3. Finally, the performance of the machine tool controller will be presented in Chapter 4.

CHAPTER 2

STRUCTURE AND ANALYSIS OF CONTROL SYSTEMS FOR MACHINE TOOLS

2.1 Introduction

The structure and analysis of control systems for machine tools will be introduced in this chapter. The structure of the control loop and the sampled-data control of a servo motor drive system will be described. The theoretical model for the machine drive and for the servo system compensation will be analyzed. Criteria to establish the machining accuracy required for a machine tool controller will be also presented. This chapter will include a discussion of the mathematical model and the basic hardware and software needed for the implementation of a sampled-data, machine drive controller.

The theory which will be used is based on several inherent assumptions. The most important requirement is that the motor drive can be modelled as a linear control system. The validity of a linear model must be verified by experimentally determining the speed response of the motor drive. In the mathematical analysis, it is assumed that the mechanical machine drive does not suffer the dynamic consequences of structural flexibility or leadscrew errors arising from backlash and pitch error. This assumption is generally acceptable for a good quality machine tool which is built for high precision machining. The accurate modelling of the machine drive system is crucial to the successful implementation of machine tool control.

2.2 Structure of a Microcomputer Based, Machine Tool Control System

The primary function of a machine tool control system is to accurately control the position of the tool relative to the workpiece during machining. The machine drive's control system usually consists of several control loops such as position, velocity, and sometimes motor current feedback as shown in Figure 2.1 [17-19]. The servo motor's drive system is generally an off-the-shelf, commercially available unit. It consists of a motor and a power amplifier. The compensated velocity and current feedback loops are typically implemented within the amplifier.

Velocity feedback stabilizes the servo system. It is usually generated by a tachometer attached to the motor [20]. Velocity compensation is used to obtain a stable and fast velocity response to a voltage input. The continuous velocity signal helps to stabilize the servo system when the motor rotates at low speeds. This stabilization results in a smoother motion [21]. Velocity control is essential in CNC applications in which high levels of accuracy are required for the final position, surface finish, and path tracing accuracy [22]. The innermost current feedback loop limits the motor's armature current and reduces the effect of torque disturbances on the servo system. The velocity and current feedback loops typically utilize lead-lag controllers and they are usually implemented in analog circuitry [23]. The compensator in the velocity feedback loop is tuned by using potentiometers to obtain a good output velocity response to a step voltage input. The current feedback loop potentiometers are adjusted to limit the motor's maximum armature current.

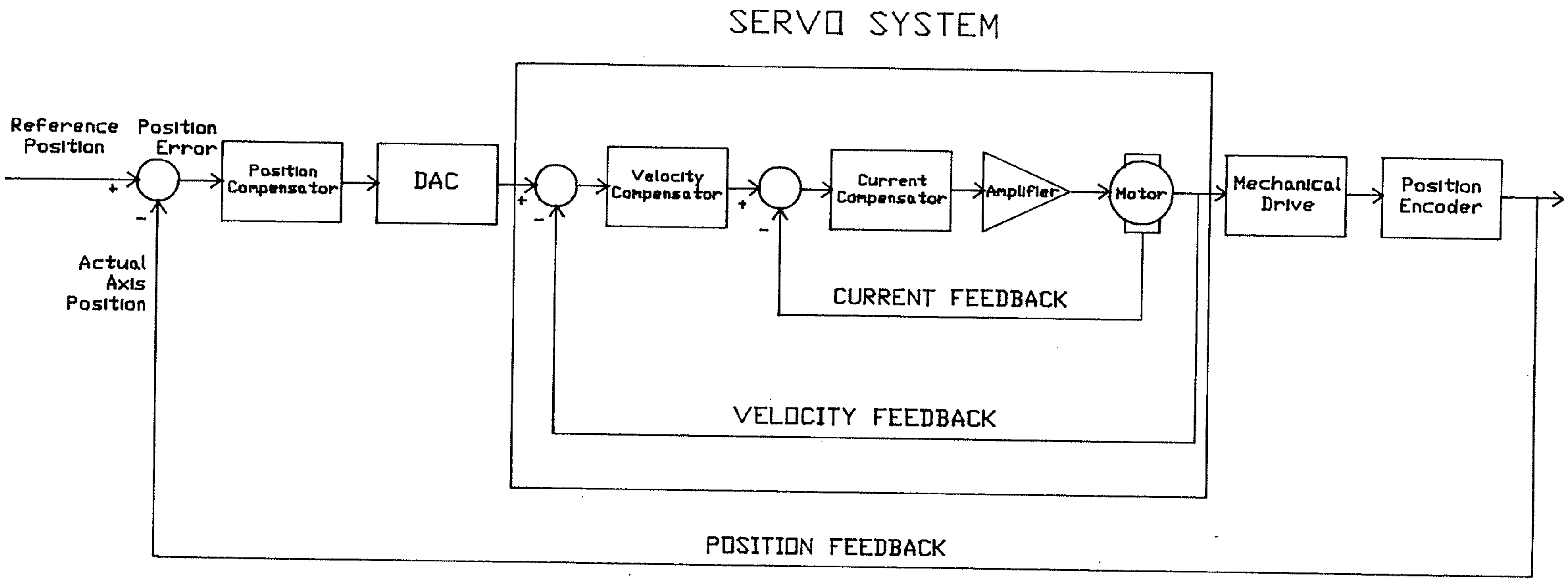


Figure 2.1 Structure of a CNC Control System

The position feedback loop illustrated in Figure 2.1 consists of a position compensator or controller, a digital-to-analog converter (DAC) , and position feedback. The DAC provides an analog speed reference to the motor's amplifier and a position encoder provides the position feedback. Position loop compensation is needed to achieve a desired positional accuracy in the CNC system. The position compensator controls the motion of the motor driven axis based on the difference, or error, between the reference position and the actual position along the axis. The position compensator can be implemented digitally by using a microcomputer. The feedback signals from the machine's axes are sampled by the microprocessor which then executes a software based, control function based on the feedback signals. The major advantage of microcomputer based control is that the software based compensation can be modified much more easily than the conventional hardwired compensation. Details of position loop compensation will be discussed later in this chapter.

To implement a microcomputer based, machine tool position compensator, it is necessary to approximate the continuous flow of reference and feedback signals by using signals sampled at discrete instants. The reference position in a CNC system can be transmitted either as a sequence of individual reference pulses or as a sampled binary word [24]. These control strategies are referred to as the reference-pulse technique and the sampled-data technique, respectively. Each pulse represents one basic length-unit (BLU) of axis travel in the reference pulse technique. One BLU is the position resolution of the machine axis. The major drawbacks of the reference-pulse technique are:

a) the speed of an axis is restricted by the computer's ability to produce reference pulses,

b) interpolation capabilities are limited by the high interrupt frequencies required for high axes' speeds [24], and

c) the control loop is closed outside the computer so that the actual axes' positions are not known during the motion.

The position control loop is closed within the computer when the sampled-data technique is employed. The control program compares a reference position word with the actual position feedback to determine the position error. The error signal is input at a fixed frequency to a DAC. The DAC outputs a voltage which is proportional to the required axis speed. The sampling frequency, f_S, employed in the sampled-data technique is considerably slower than the interrupt frequency of the reference-pulse technique. Indeed, the sampling frequency for sampled-data machine tool controllers typically ranges between 100 and 600 Hertz [15, 25]. Selection of the actual sampling frequency depends on the machine's bandwidth and interpolation requirements. This aspect will be discussed further in Section 2.3.4. Conversely, the sampling frequency for the reference-pulse technique depends on the maximum axis speed required. The maximum rotational speed of the motor in a sampled-data system is not limited by the sampling frequency because an entire error word and not just a single pulse is output at each sample. The resulting longer intersample period for a sampled-data system allows a more complex interpolation scheme to be implemented in real time. The functional elements of a sampled-data position control system will be discussed next.

2.2.1 Functional Elements of a Sampled-Data, Position Controlled System

Figure 2.2 represents the functional elements of a single axis CNC drive which uses sampled-data control. A sampled-data controller employs a microcomputer as part of the control loop. The CNC machine tool consists of a software based control program, interface electronics for motion control, servo system, mechanical drive, and position encoder. The motion control electronics interface the computer's control program, the servo system, and the position encoder. This electronic circuitry consists of a digital-to-analog converter (DAC) and an external counter which receives position information from the axes' encoders. The DAC provides a reference voltage signal to the servo system whilst the external counter is used to count the number of encoder pulses during a given period. A single encoder pulse represents one basic length-unit (BLU) of the CNC machine. The mechanical drive for one axis of a CNC machine typically consists of a gearbox, coupling, and leadscrew.

The microprocessor of the computer periodically samples the position of each machine axis. The number of pulses transferred from the external counter to the microcomputer represents the incremental change in position for that sample period. The microcomputer accumulates these increments to determine the actual position of each axis. It then compares the actual position with the reference position which is determined by the interpolation routine during each intersample period. The difference between the reference and the actual position represents the error in the drive's position. The position compensator uses the position error to determine the digital value which should be applied to the DAC [26]. The output from the position

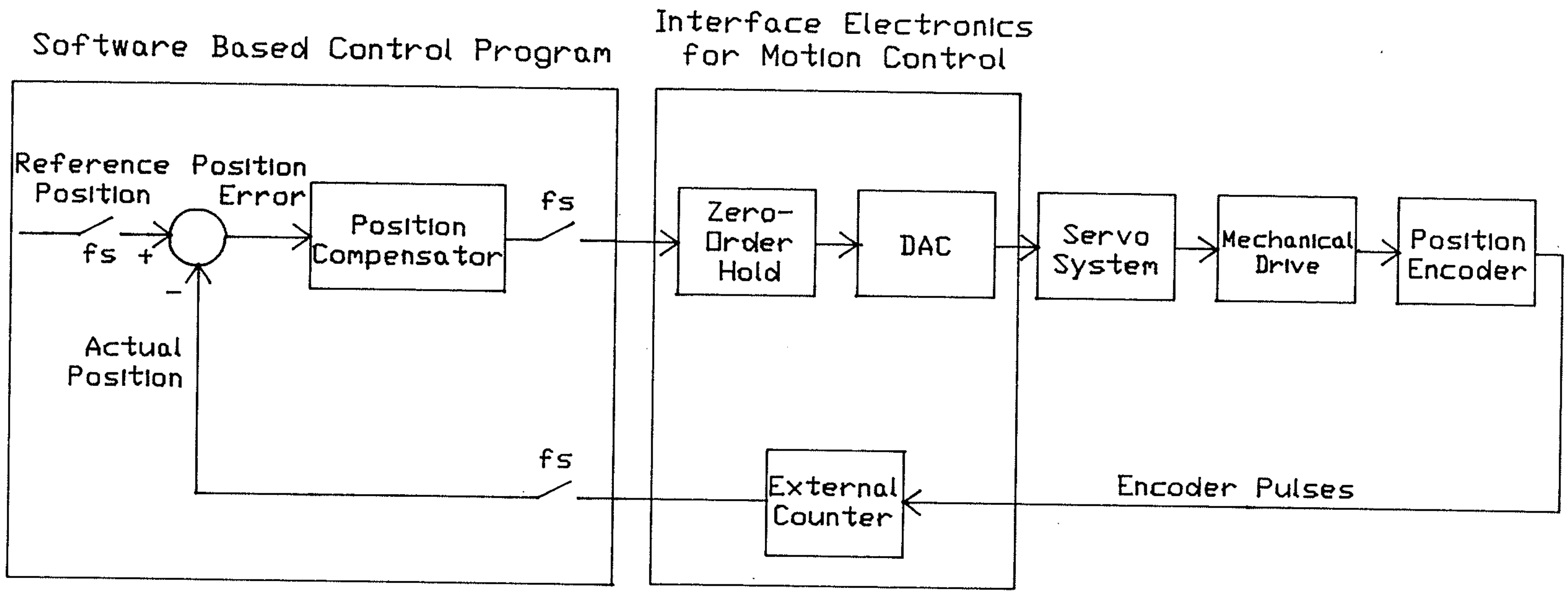


Figure 2.2 Structure of a Microcomputer Based, Sampled-Data Controller

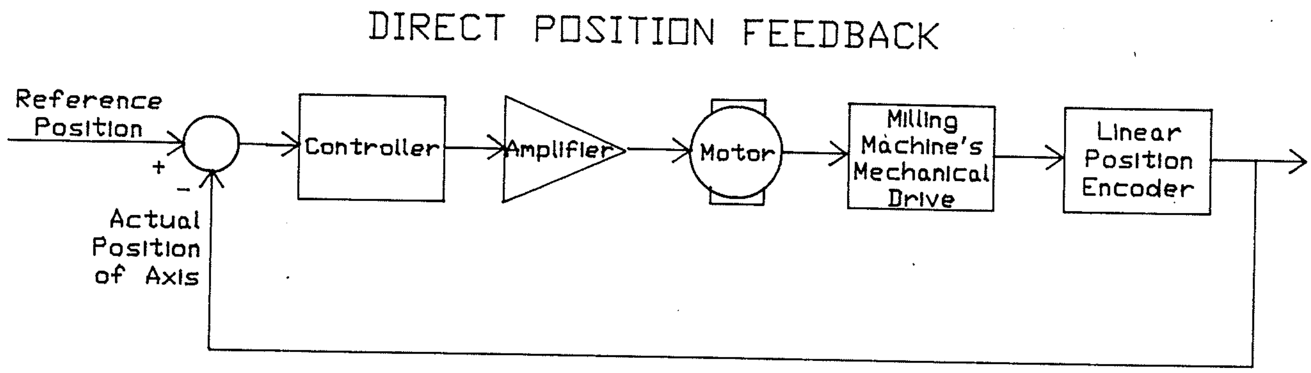
compensator is applied periodically to a zero-order hold. The zero-order hold maintains the digital input to the DAC for the duration of the sampling period. The DAC converts the digital input to a corresponding analog output which causes the motor to rotate in a direction which reduces the error. The direction of the motor's rotation is determined by the sign of the applied voltage. A positive voltage produces rotation in one direction and a negative voltage produces a rotation in the opposite direction. The external counter, reference position, actual position, position error, and DAC value are updated at a rate equal to the sampling frequency, f_S.

Implementation of a sampled-data position controller will be explained in more detail later. Before doing so, it is useful to introduce the two closed-loop feedback structures, namely, indirect and direct feedback.

2.2.2 Indirect and Direct Closed-Loop Feedback

Closed-loop control systems can use either direct or indirect feedback [19, 22]. In indirect feedback, the output of the axis motor is measured at the motor shaft as shown in Figure 2.3. An example of an indirect closed-loop feedback is that resulting from the use of a rotary position encoder mounted on the end of a milling machine's drive motor. The encoder senses the rotation of the motor shaft which is assumed to be an accurate representation of the motion of the milling machine axis itself. Thus, indirect position feedback does not account for machine inaccuracies such as leadscrew backlash and leadscrew pitch error.

The use of indirect feedback is common in CNC systems but it may not provide as high a degree of machining accuracy as the direct feedback



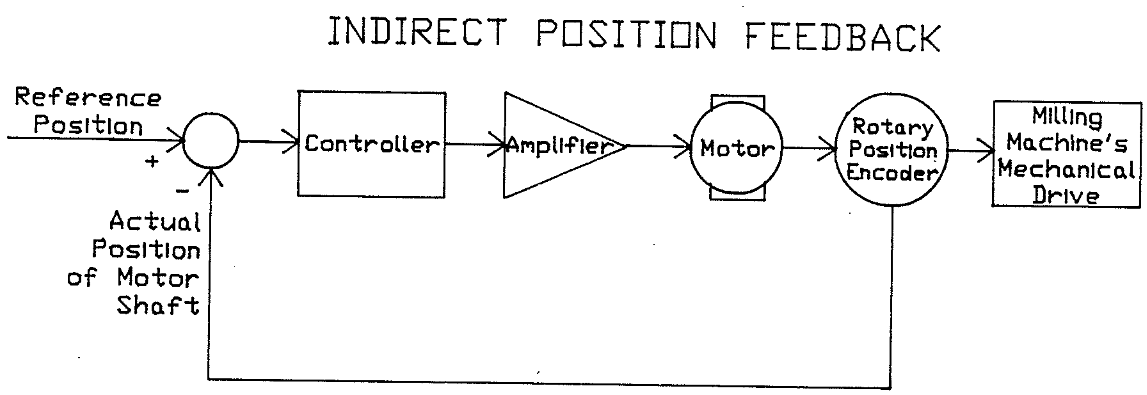


Figure 2.3 Direct and Indirect Closed-Loop Position Feedback

approach [22]. Direct closed-loop feedback includes the milling machine's mechanical drive in the feedback loop. In a closed-loop system with direct feedback, a position sensor is used to monitor the motion of the axis itself. An example of direct closed-loop feedback is the use of a linear position encoder mounted directly on the table of a milling machine as shown in Figure 2.3. In this situation, the linear encoder senses the movement of the milling machine's table. Direct position feedback from the table provides a true representation of the position of an axis which is independent of machine inaccuracies such as leadscrew errors. Thus, direct feedback is more accurate but it is also more costly to implement in machine tools than indirect feedback. The hardware cost of a linear encoder, for instance, can range from $500 to $2000 while a rotary encoder typically costs under $200 [27, 28]. Most commercial CNC systems use the cheaper rotary encoders and the effects of the machine tool's inaccuracies are minimized by using precision leadscrews with virtually no backlash.

2.3 Analysis of the Servo Motor's Drive System

A model of the servo motor's drive system must be developed in order to design a servo system to satisfy a desired level of machining accuracy. The model can be used to predict and to compensate the performance of the servo motor's drive. Figure 2.4 presents a detailed block diagram of a single axis, microcomputer based, sampled-data control system for a motor. The position encoder, which is usually attached to the end of the motor, outputs pulses with an encoder gain of K_e pulses/radian. The pulses are accumulated in an external counter during each period between the sampling instants. The microcomputer compares the reference axis position to the actual position at

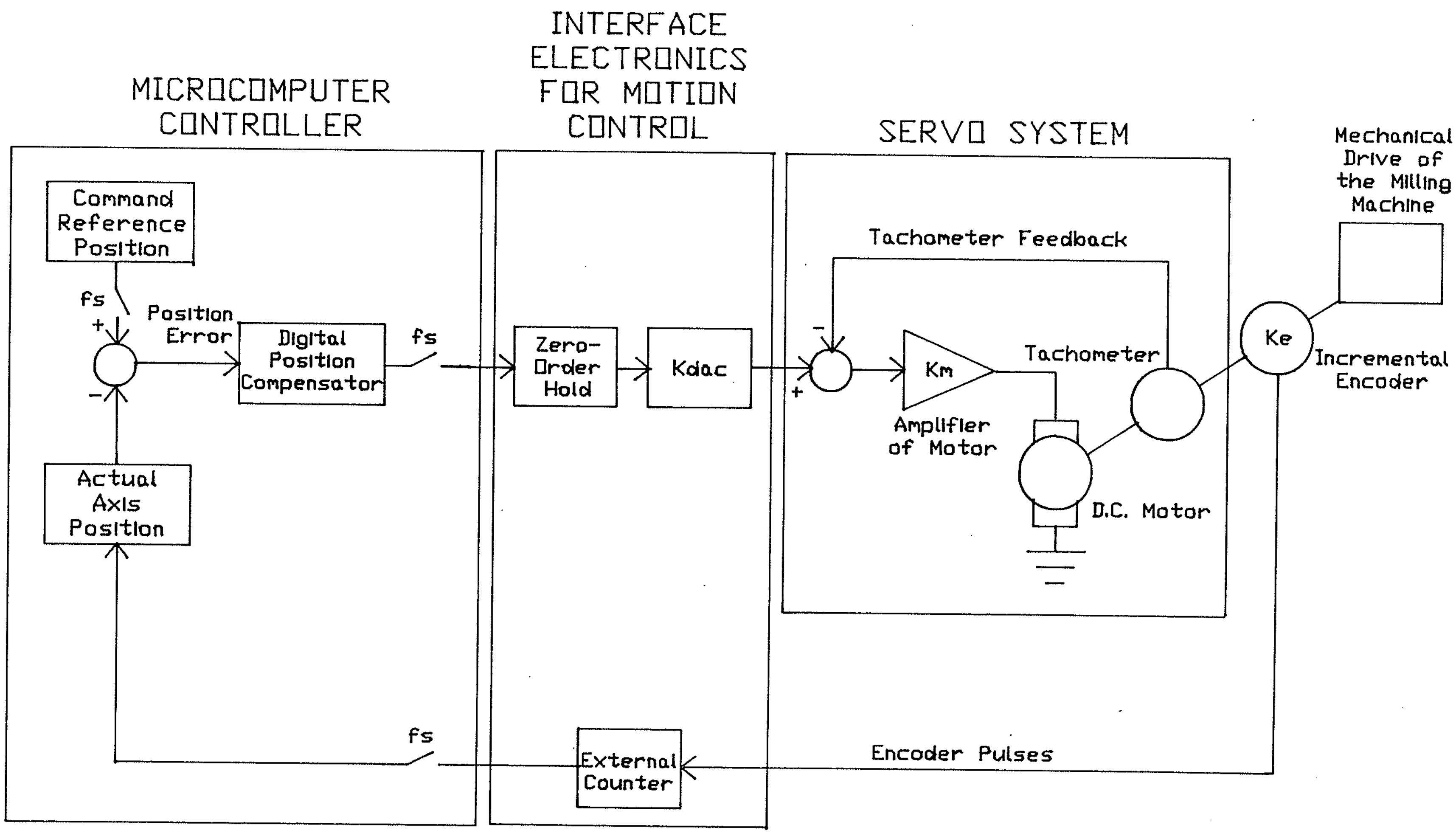


Figure 2.4 Block Diagram of a Motor's Drive System and its Microcomputer Controller

each sampling instant. The resulting position error is compensated digitally by a position loop compensator. The compensated error is passed to the digital-to-analog converter (DAC) which converts the position error to a corresponding voltage. The DAC has a gain of K_{dac} volts/pulse error. It outputs the voltage to the motor's amplifier which accepts it as a speed reference signal. The motor and amplifier drive convert this voltage input to a corresponding speed output with a gain of K_m radians/(second-volt). Thus, the voltage output of the DAC effectively reduces the error between the reference position and the actual position of an axis. In summary, the microcomputer is responsible for producing the reference position for an axis, reading the external counter to determine the actual position of that axis, and outputting the digitally compensated position error to drive the motor. Each of these functions is performed by the microcomputer at a rate equal to the sampling frequency, f_s. In this way, the position control loop is closed inside the microcomputer. The modelling and analysis of the servo motor's drive will be discussed next.

2.3.1 Model of the Servo Motor's Drive

The model of a servo motor's drive is presented in this section. In a DC servo motor-amplifier system, the transfer function between the input voltage, V(s), and the motor's speed, w(s), is represented generally as a second-order model [29, 30]. The second-order model normally provides a sufficiently accurate representation of the servo system's response. In addition, it is difficult to exactly determine an actual machine drive's transfer function due to the proprietary nature of commercial amplifier designs and the unpredictable nature of external loads on the motor. For example, large cutting load disturbances, which exceed the torque output of a motor, would undermine the validity of the motor drive's transfer function. In this situation, the output speed of the motor would be lower than that expected. Most amplifiers are designed to provide an output speed which is linearly proportional to an input voltage over the whole torque-speed range of the motor. In this situation, the total gain of the motor-amplifier drive system can be represented as K_m radian/(second-volt). The voltage input to a servo amplifier translates to a corresponding rotational shaft speed of the motor. Thus, the transfer function describes the ratio of the input voltage to the output speed of the servo motor. It may be written as:

$$G_m = \frac{w(s)}{V(s)} = \frac{K_m}{(1 + \tau_m s)(1 + \tau_e s)} \qquad (2.1)$$

where:

G_m = servo drive's transfer function (radians/(second-volt)),
w(s) = motor's speed (radians/second),
V(s) = amplifier's voltage input (volts),
s = Laplace operator (second^{-1}),
K_m = motor-amplifier's gain (radians/(second-volt)),
τ_m = servo system's mechanical time constant (seconds),
τ_e = servo system's electrical time constant (seconds).

The mechanical time constant is defined as the time for an unloaded motor to reach 63.2% of its final velocity after the application of a DC armature voltage [29]. This characterization is illustrated in Figure 2.5. Similarly, the electrical time constant is equal to the time required for the current to rise to 63.2% of its final value when a step input voltage is applied to the motor's armature with the motor's shaft locked [29]. The motor's electrical time constant is usually small compared to its mechanical time constant [31, 32]. The electrical time constant typically ranges from 0.25 to

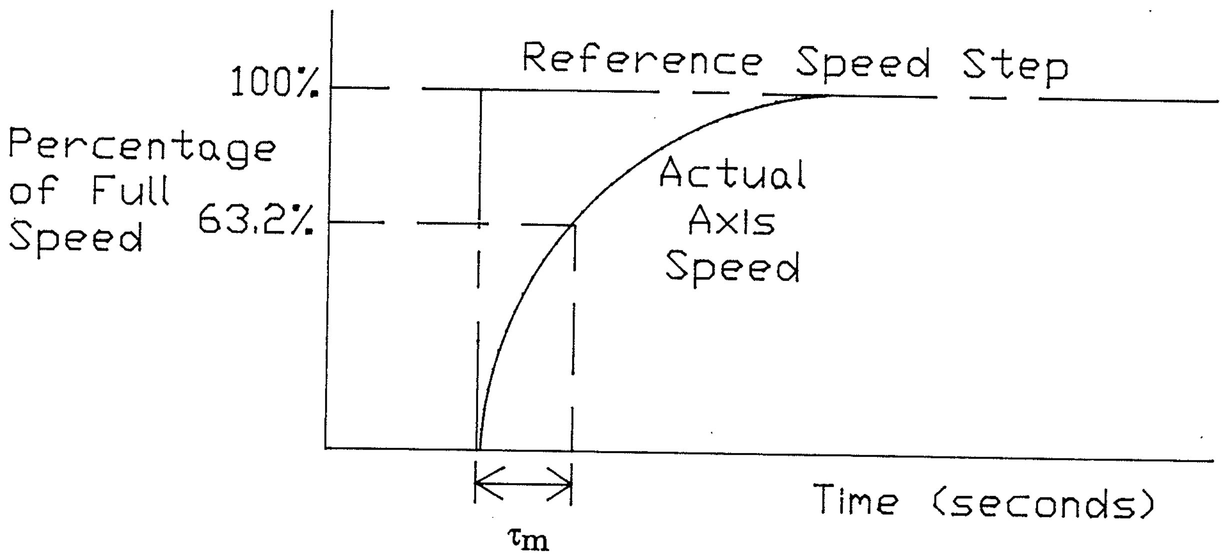


Figure 2.5 Definition of the Servo System's Mechanical Time Constant, τ_m

6.0 milliseconds [6]. The mechanical time constant, on the other hand, can range from about 10 to 50 milliseconds for motors under 1 horsepower [6]. Thus, by assuming that the electrical time constant is much smaller than the mechanical time constant, the representation of the motor's drive system which is given by equation (2.1) can be simplified to:

$$G_m = \frac{K_m}{(1 + \tau_m s)} . \qquad (2.2)$$

Hence, a complicated motor-amplifier system with internal velocity and current feedback can be represented reasonably by a simple first-order model. However, it is important to recognize that such a model is valid only as long as the amplifier operates linearly without any current or torque overload.

Figure 2.6 presents a block diagram of a simplified machine drive for one axis of a CNC system. The process of accumulating incremental position changes between sampling instants is represented essentially by an integration. Thus, the open-loop forward-path transfer function, G_f, of the machine drive system can be written as:

$$G_f = \frac{K_{dac}{*}K_m{*}K_e}{s\,(1 + \tau_m s)} = \frac{K_t}{s\,(1 + \tau_m s)} \qquad (2.3)$$

where:

$K_t = K_{dac}{*}K_m{*}K_e$ = total gain of the position control loop (second-1),
K_{dac} = gain of the DAC (volts/pulse error),
K_m = motor-amplifier's gain (radian/second-volt),
K_e = encoder's gain (pulses/radian),
s = Laplace operator (second-1),
τ_m = loaded servo system's mechanical time constant (seconds).

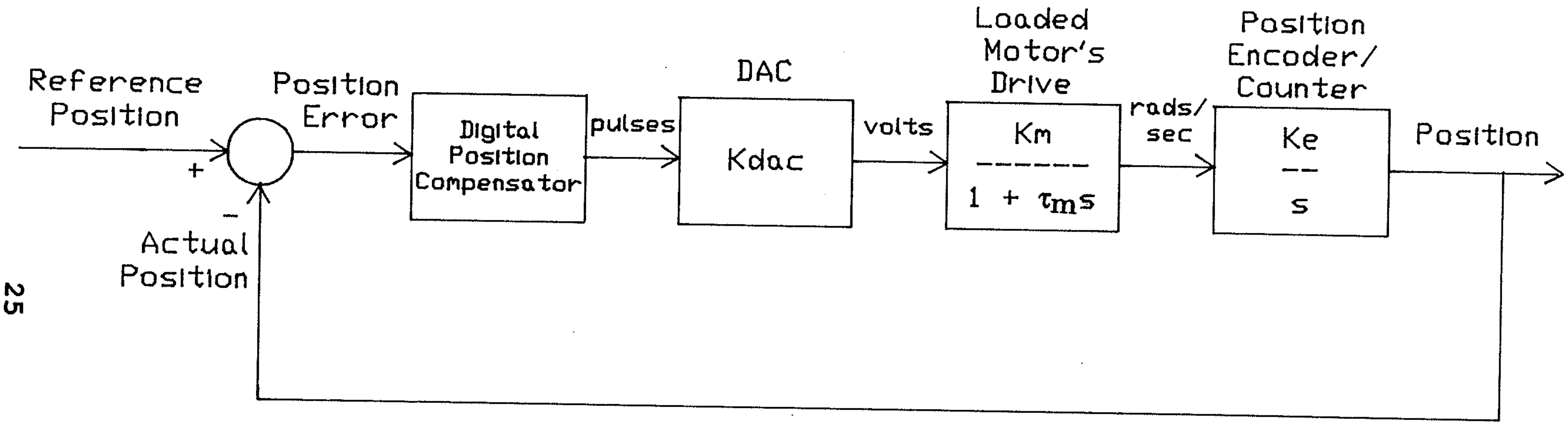


Figure 2.6 A Simplified Machine Drive

An examination of equation (2.3) indicates an s^2 term in the denominator. As a result, the forward-path transfer function of the machine drive is a second-order control system. The open-loop transfer function of a second-order system can be represented as [33]:

$$G(s) = \frac{\omega_n^2}{s\,(s + 2\zeta\omega_n)} \tag{2.4}$$

where:

$G(s)$ = open-loop transfer function,
ω_n = undamped natural frequency (radians/second),
ζ = damping ratio.

Furthermore, the closed-loop transfer function of a second-order system having unity feedback is [33]:

$$\frac{C(s)}{R(s)} = \frac{G}{1 + GH} = \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} \tag{2.5}$$

where:

H = feedback element,
C(s) = input to control system ,
R(s) = output from control system.

Since the transfer function of the machine drive is second-order, the theory for second-order linear control systems can be used to predict the performance of the servo drive system. Furthermore, this theory can be utilized beneficially in the design of a compensation scheme to produce a desired positioning accuracy in a CNC system.

The damping ratio, ζ, in equation (2.4) determines the response characteristics of a servo system to a command input [34]. The response of a servo system is usually based on a step input. A step input is easy to generate electrically and it is a quite harsh input condition which tests the stability of a servo system. The response of a servo system to a step input can be classified as underdamped ($\zeta<1$), critically damped ($\zeta=1$), overdamped ($\zeta>1$), or unstable depending on the damping ratio, ζ. The use of the damping ratio to tune the position response of a servo system will be explained later in the design of the control loop compensation.

The gain constants in a servo system are determined easily from hardware specifications. However, it is necessary to experimentally determine the loaded servo drive's time constant due to its dependence on the loading conditions. The procedure for doing this measurement will be explained next.

2.3.2 Determination of the Servo System's Time Constant

It is necessary to validate the servo motor system's mathematical model and to determine the system's mechanical time constant. The unknown time constant is found by disconnecting the motor's amplifier from the position control system so that the encoder merely measures, without correcting, the speed of the axis motor. Various step input voltages can be applied to the reference speed terminal of the motor's amplifier and the resulting motor speeds can be recorded.

Typical results from experiments performed on one of the Electro-Craft E703-MG servo motors used in this work are presented in Figure 2.7. Step voltages of increasing amplitude were applied to the loaded motor's drive system. The time required for the loaded motor to respond to 63.2% of each voltage is recorded in Figure 2.7. For small voltage inputs, corresponding to

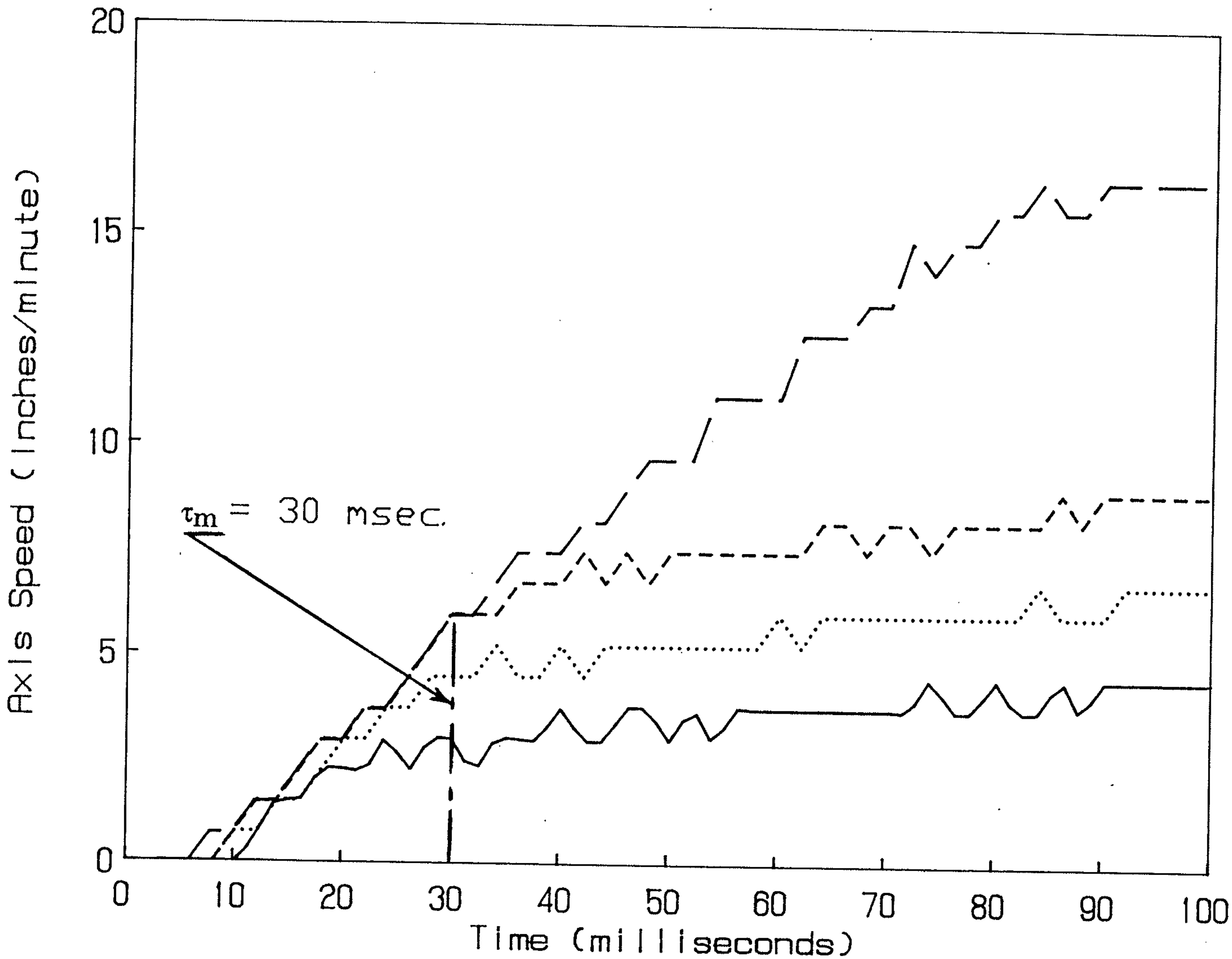


Figure 2.7 Servo System's Step Response

speeds under 8 inches/minute, the response history is approximately exponential. The mechanical time constant is 0.030 +/- 0.002 seconds for the particular motor tested. The exponential response is that expected for a first-order model so that the validity of the simplified first-order motor-drive model in equation (2.2) is also verified. Furthermore, the mechanical time constant of 0.030 seconds corresponds well to the unloaded motor's time constant of 0.028 seconds specified by the manufacturer [6].

It must be pointed out that the response of the amplifier becomes nonlinear for large voltage inputs corresponding to speed steps above about 8 inches/minute. Then the loop limiting current to the amplifier restricts the motor's current and, hence, the maximum acceleration rate to prevent damage. The motor's acceleration, which is the initial slope of the speed curves in Figure 2.7, remains approximately constant for large step changes in speed above 8 inches/minute. A truly linear behavior, on the other hand, would result in higher accelerations for larger step inputs to achieve the same time constant. However, a machine drive would not be subjected to such large step changes in speed under normal use. A motor drive is accelerated and decelerated typically by using ramp speed functions rather than large step changes. Consequently, the experimentally determined time constant of 0.030 seconds was considered quite representative of the time constant likely in practice.

2.3.3 Steady State Errors for a Type 1 Servo System

The single integration in equation (2.3) indicates that the machine drive corresponds to a type '1' position control system [34]. Knowledge of the system's type allows the position following error and the speed error to be predicted in the steady state for specified inputs. The two inputs of interest in machine tool applications are a constant speed input and a ramp speed input. Constant speed and ramp speed inputs correspond to ramp and parabolic position inputs, respectively, because position has an integral relationship to velocity. The behaviour of a type '1' system is well-known for these inputs [34] and it is summarized in Figure 2.8. Notice that the errors shown in Figure 2.8 are inversely proportional to the overall position loop gain, K_t, of the machine's drive . Consequently, a higher K_t will reduce position errors along an axis.

2.3.4 Sampling Frequency Requirements for a Servo System

The sampling frequency requirements for a sampled-data machine drive can be determined from the model of the servo motor's drive. The transfer function of a machine drive is represented customarily on a Bode plot. The Bode plot shows the gain-frequency characteristics of the servo system. Consider the Bode plot presented in Figure 2.9 which corresponds to the forward-path transfer function of the experimental machine drive. This Bode plot is typical of CNC machine drives. The ω_1 marked on the abscissa is the "break frequency" which is equal to the reciprocal of the servo system's time constant, τ_m [34]. The slope of the Bode plot changes from -1 to -2 at this frequency. The bandwidth of the servo system, on the other hand, can be estimated from the Bode plot by using the crossover frequency, ω_c, where the gain is unity [35]. Then,

$$\text{Bandwidth} = \omega_c / (2*\pi) \qquad (2.6)$$

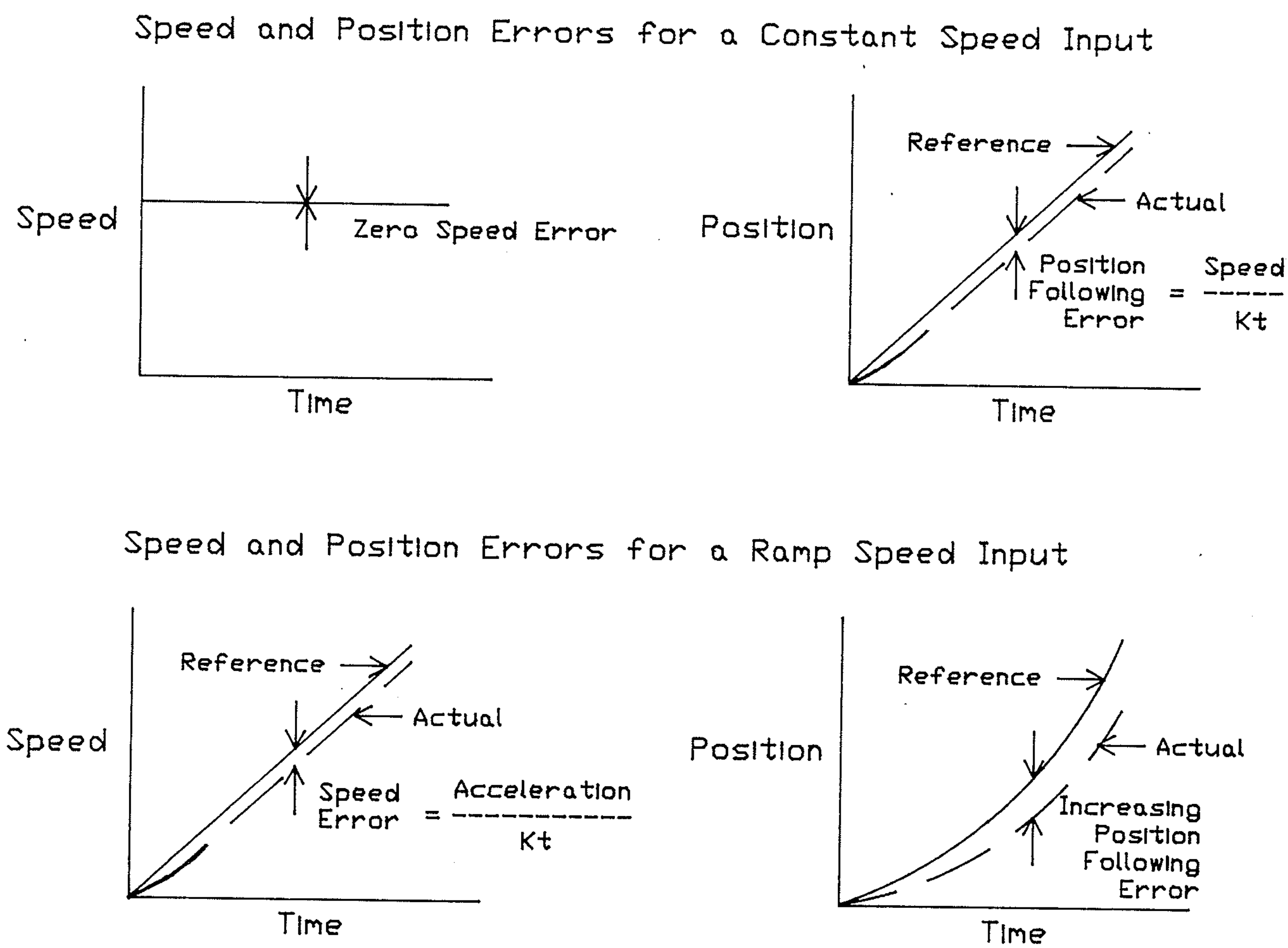


Figure 2.8 Constant Speed and Ramp Speed Responses of a Type '1' Control System

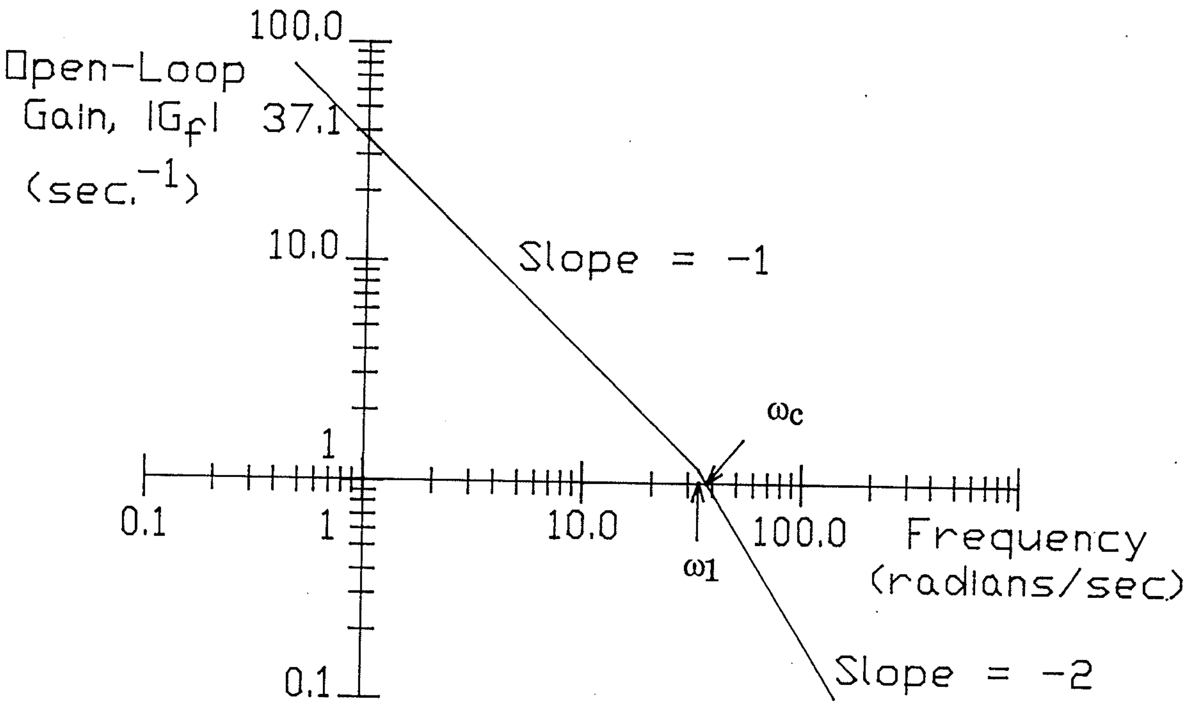


Figure 2.9 Bode Plot for a Typical CNC Machine Drive

where:

ω_c = crossover frequency (radians/second),
$\pi = 3.141592654$.

A control system can be modelled as essentially continuous if the system's inputs and outputs are sampled sufficiently quickly. Several authors [13, 36, 37] indicate that a sampling frequency of between ten and twenty times the servo system's bandwidth will adequately capture the position information for an axis. Thus, a conservative sampling frequency for a machine tool controller would be about twenty times the drive system's bandwidth for a stable servo system operation. CNC machine drives have closed-loop bandwidths in the range of 5 to 30 Hertz [15]. As a result, machine tool controllers typically employ sampling frequencies in the range of 100 to 600 Hertz.

For the experimental servo drives used in this work, the total open-loop gain, K_t, given in equation (2.3) can be calculated easily from hardware specifications for the DAC, the servo motor, and the position encoder. The DAC has 8 bits and its analog output is in the range of +/- 10 volts. As a result its gain, K_{dac}, is 10 volts for 127 pulses or 0.0787 volts/pulse. The amplifiers on the X and Y axes were tuned separately so that the motor rotated at 2850 rpm for a reference voltage of 10 volts. Thus, the motor's gain, K_m, is 29.85 rad./sec.-volt (2850 rpm / 10 volts * 2π rad./rev. * 1 min./60 sec.). A rotary encoder producing 100 pulses /revolution was used on both the X and Y axes. This value corresponds to an encoder gain, K_e, of 15.92 pulses/rad. (100 pulses/rev. * 2π rad./rev.). As a result, the total position loop gain K_t, which is the product of K_{dac}, K_m, and K_e, is 37.1 seconds^{-1}. From Figure 2.9, the crossover frequency is easily calculated to be 35.2 seconds^{-1} so that the machine drive's bandwidth is 5.6 Hertz. Now the sampling frequency of the machine tool controller should be about twenty times the drive system's bandwidth or 120 Hertz here to ensure a stable operation of the machine's drive. As result, a sampling frequency of 150 Hertz was chosen for the experimental system.

2.4 Performance Criteria for Machine Tool Controllers

A machine tool controller must satisfy several performance requirements for CNC applications. The most important requirement is the need for the controller to maintain a specified machining accuracy over the entire range of cutting speeds of the CNC machine [38, 39]. The accuracy of a machine tool controller may be described in terms of a following error, contour error, and a final position overshoot.

A following or tracking error between the reference and the actual position of an axis always exists in a closed-loop servo system. The reason for this discrepancy is that each motor of a CNC system is driven by the difference, or error, between the reference position and the actual axis position. Following errors are not critical as long as the time response characteristics of each of the axis motors are virtually identical. Then the reference position leads the actual position by a finite, constant value during a steady motion. Although the toolbit's actual position lags behind the reference position, the actual position still lies on the desired path. It is desirable, nevertheless, to have a sufficiently high servo system gain to achieve an acceptable following error. The following error in a CNC system can be reduced by increasing the position loop gain, K_t. The gain, however, cannot be increased without limit because the servo system will become unstable beyond a certain gain level. Schmitt [25] states that, due to the effects of the inertia of the motor and mechanical drive, most machine tools operate with a following error in the range of one mil (0.001 inches) for each 1.0 inch/minute of axis speed. Polad [36] indicates that the following error should be about 1 mil for each 5 inch/minute speed commanded for a high speed laser machining center. This high performance laser machine operates at feedrates up to 500 inches/minute. An appropriate following error requirement may be established for a desired feedrate range based on these suggestions. For typical milling operations at feedrates from 1 to 25 inches/minute, a suitable following error requirement would be about one mil of following error for each 2 inches/minute of axis feedrate. This design following error can be expressed in terms of the position loop gain, K_t, of the machine drive. Namely, a following error of one mil for each 2 inches/minute of axis feedrate corresponds to a gain of 33.3 seconds^{-1} (2.0 inches/minute-mil * mil/0.001 inches * minute/60 seconds). Thus, the positon loop gain should be at least 33.3 seconds^{-1} to satisfy the design following error.

In contrast to following errors, <u>path</u> or <u>contouring errors</u> are position errors where the cutter deviates from the desired path. The contour error is defined as the spatial shift of the toolbit's actual position in a direction which is perpendicular to the desired path. A typical example is shown in Figure 2.10. Contour errors can be caused by:

a) <u>Different Time Response Characteristics of the Various Motor Drives</u>. A contour error will be generated if the motor driving one axis responds more quickly to the reference speed than that for another

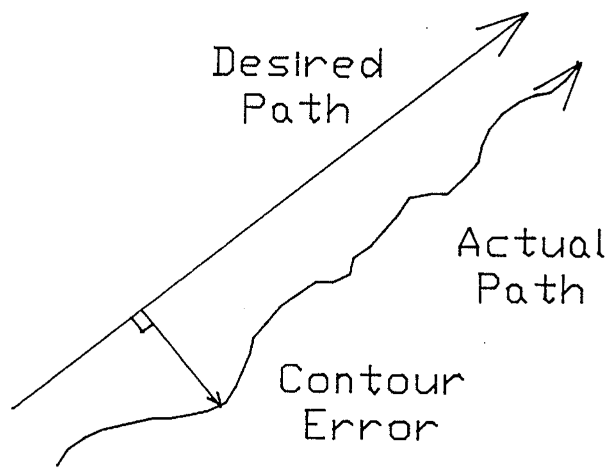


Figure 2.10 Illustration of Contour Error

axis. This situation can be caused by mismatches in the gains and the time constants of the different axes during a multi-axis motion [39]. As a result, it is desirable to have servo motor drives with well matched gains and time constants.

b) Excessive Loading On One or More Axes. A contour error will also be produced if the loading on any one axis exceeds the maximum torque permissible for the motor. Such an overload, however, indicates that the servo motor is undersized for the given application.

c) Interpolation Errors. Interpolation errors can be caused when curves such as circular arcs are approximated by line segments.

d) Quantization of the Feedback Signal. Quantization of the velocity or position feedback signals can cause contour errors. For example, the motion can become "rough" at very low speeds due to quantization of the position feedback.

Different time responses of the various motor drives is the major source of contour errors. It can occur due to mismatch in the gains or the time constants of the motor drives. Contour errors arising from mismatched time constants of axes' motors are most noticeable when the motors are either accelerating or decelerating. However, such a mismatch does not cause significant contour errors during a constant speed motion. In steady motion, a mismatch in gain between the servo motors for different axes is the major cause of contour errors [39]. The effects of gain mismatch will be described next for both linear and circular interpolation.

Figure 2.11 illustrates the contour error caused by a mismatch in gain during a two-axis, linear motion. The variation in position of the X and Y

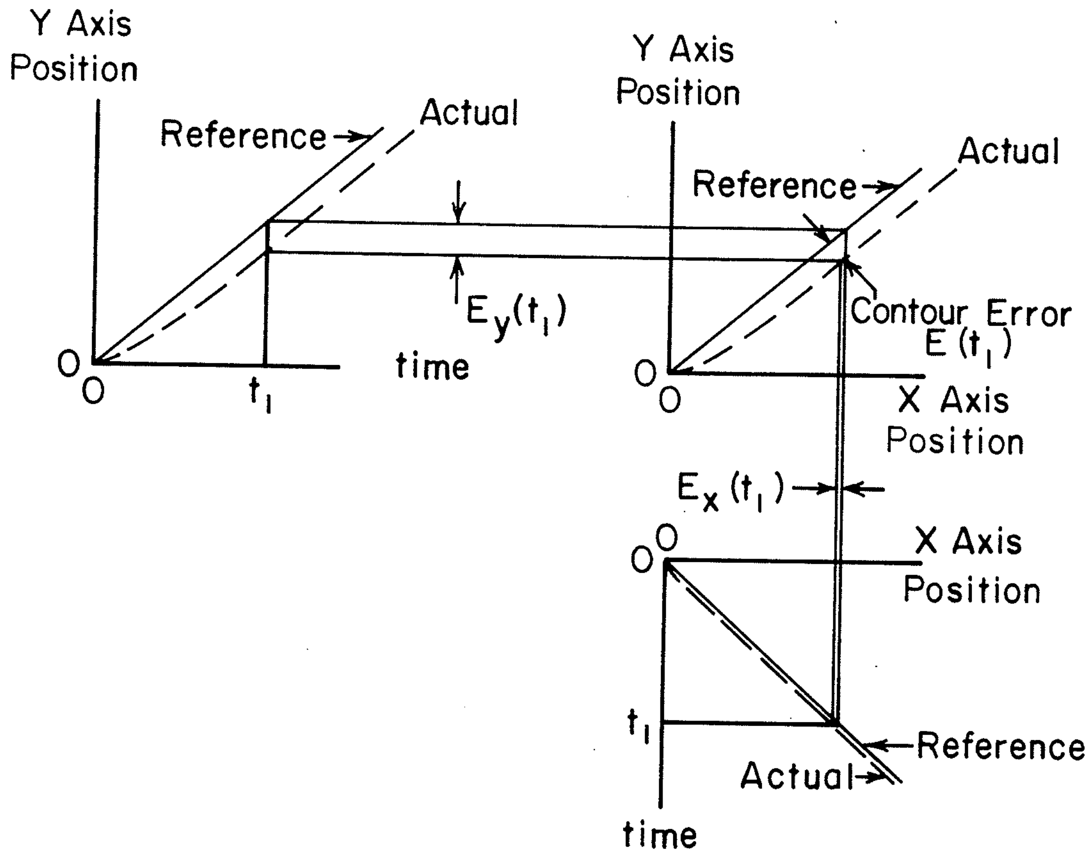


Figure 2.11 Contour Error Caused By a Mismatch in Gain During Linear Interpolation [39]

axes is shown for a constant speed motion as a function of time. Each axis has a position following error which is inversely proportional to the speed of that axis. The Y axis servo motor has a slightly lower gain than that of the X axis. As a result, the projected toolbit motion in the XY plane has a contour error. If the Y axis servo motor has the same gain as that of the X axis, then each axis would have individual position following errors but the projection would still lie on the desired path. The contour error is the offset of the toolbit's actual position in a direction perpendicular to the desired path. Poo et al [39] indicate that the maximum path error occurs at a 45 degree angle for a two-dimensional linear motion.

Figure 2.12 shows the contour error caused by a mismatch in gain during a circular motion. The variation in position of the X and Y axes is shown for a constant speed motion as a function of time. Again, the Y axis servo motor has a slightly lower gain than that of the X axis. The mismatch in gain produces different following errors in the positions along each axis. Such differences produce a circular contour which is approximately elliptical. The radial contour error is the difference between the reference radius and the actual radius. The maximum contour errors for circular interpolation occur at the angular midpoints of each quadrant of motion. The contour errors can be reduced by increasing the individual gains of the axes' motors in order to reduce the position following errors. It is necessary to ensure that the contouring error for a machine tool is lower than the machining accuracy required during contouring operations.

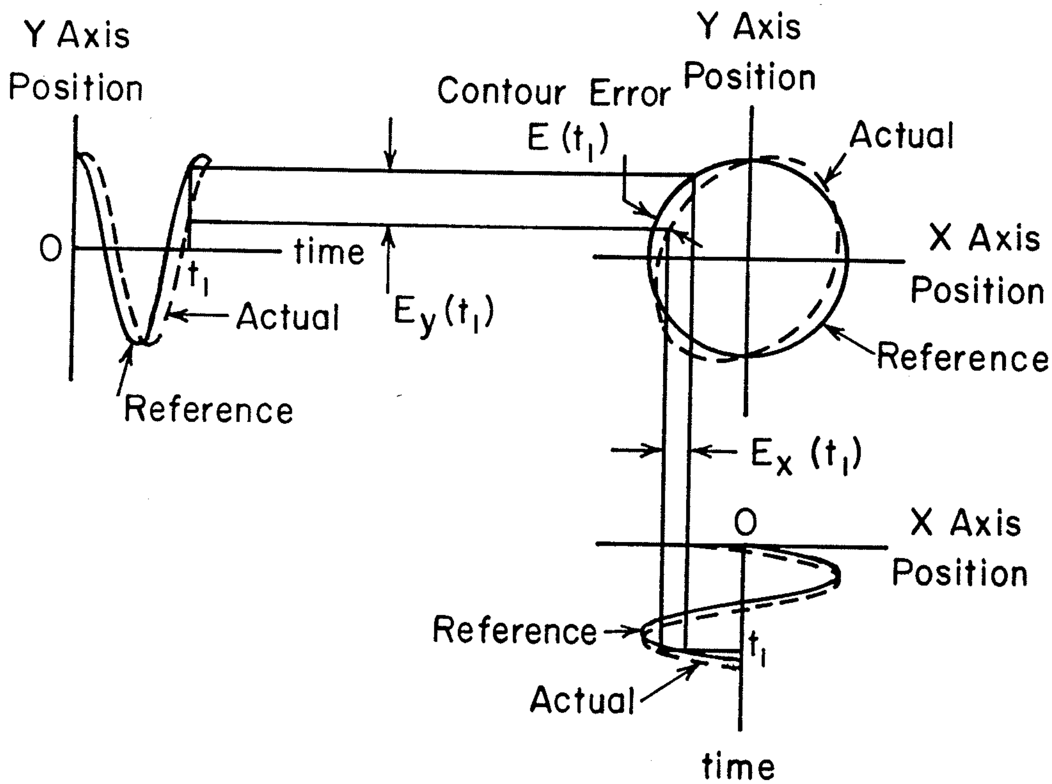


Figure 2.12 Contour Error Caused By a Mismatch in Gain During Circular Interpolation [39]

A final position overshoot is defined as the maximum allowable overshoot on the desired distance of travel. It is crucial in metal removal operations because the overshoot will result in excess material removal.

In summary, a machine tool controller must be designed to achieve the desired following error, contour error, and final position overshoot requirements. These errors can be predicted reasonably, and then compensated, by using an accurate model of the servo motor's drive system.

2.5 Design of Control Loop Compensation

The performance of a microcomputer based, servo system depends strongly on the choice of the control loop compensator. The compensator is selected according to the requirements for the system's performance as well as the microprocessor's computational capability. The control algorithm should not only satisfy the machining accuracy needed but it should be executable between consecutive sampling instants.

There is no simple rule for compensating servo control loops but several worthwhile ideas will be discussed. The designer of a servo system usually desires high gain in the open-loop transfer function. A high gain reduces the sensitivity of the system to disturbances and parameter variations and increases the system's bandwidth. For example, a position control system having a high open-loop gain can faithfully follow a command despite disturbing torques. On the other hand, the system tends to become unstable and oscillatory as the gain is increased. Thus, the requirements for a high gain and a wide bandwidth oppose the basic requirement for stability. To overcome this difficulty, compensation networks are used to change the open-loop transfer function so that a higher gain can be achieved under stable conditions [34].

There are a multitude of compensation schemes available for use in machine tool applications. For example, the compensation function can be implemented in the feedback loop as opposed to the more traditional feedforward loop [13, 14]. Adaptive and time-optimal control algorithms can be used also to optimize the machining process [40-42]. Most compensation schemes, however, use simple first-order lead and lag filters as building blocks [43-45]. The lead filter provides a phase lead to cancel the phase lag that a motor's pole introduces in the loop. The lead filter effectively extends the bandwidth of the system to give a more "responsive" system. The lag filter, conversely, introduces a dominant pole in the position control loop which allows a high DC gain to be employed. The high gain reduces the position following error. A proportional-integral-derivative (PID) filter contains the desirable features of both the lead and the lag filters [46]. PID compensation is generally more robust than most other algorithms so that it is implemented in many motion control applications [47-51]. The main features of a PID compensator will be discussed in the following section.

2.4.1 PID Compensation

Several machine tool controllers are based on the widely used proportional-integral-derivative (PID) controller [47, 52, 53]. The transfer function of a PID controller may be written as:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d * s \quad (2.7)$$

where:

G_c = transfer function of PID compensator ,
K_p = proportional gain,
K_i = integral gain,
K_d = derivative gain.
s = Laplace operator (seconds^{-1})

The design problem is to determine the values of the constants K_p, K_i, and K_d which satisfy the desired performance requirements. The proportional term increases the total gain of the servo system. The derivative control, on the other hand, is equivalent to adding a single zero at $s = -K_p/K_d$ to the open-loop transfer function as shown on the Bode plot of Figure 2.13. The derivative of a controllable variable represents the rate of change of the variable and, thus, derivative control is essentially an "anticipatory" type of control. On the other hand, the integral term of the PID controller produces a signal which is proportional to the time integral of the input to the controller. Integral control is equivalent to adding a zero at $s=-K_i/K_p$ and a pole at s=0 to the open-loop transfer function. The integral term increases the order of the system by one which reduces the steady state error. One practical problem associated with using the integral term is the accumulation of the integral of the error. Consequently, the integral term can "windup" for moves over long distances [52]. A very large accumulated integral term can

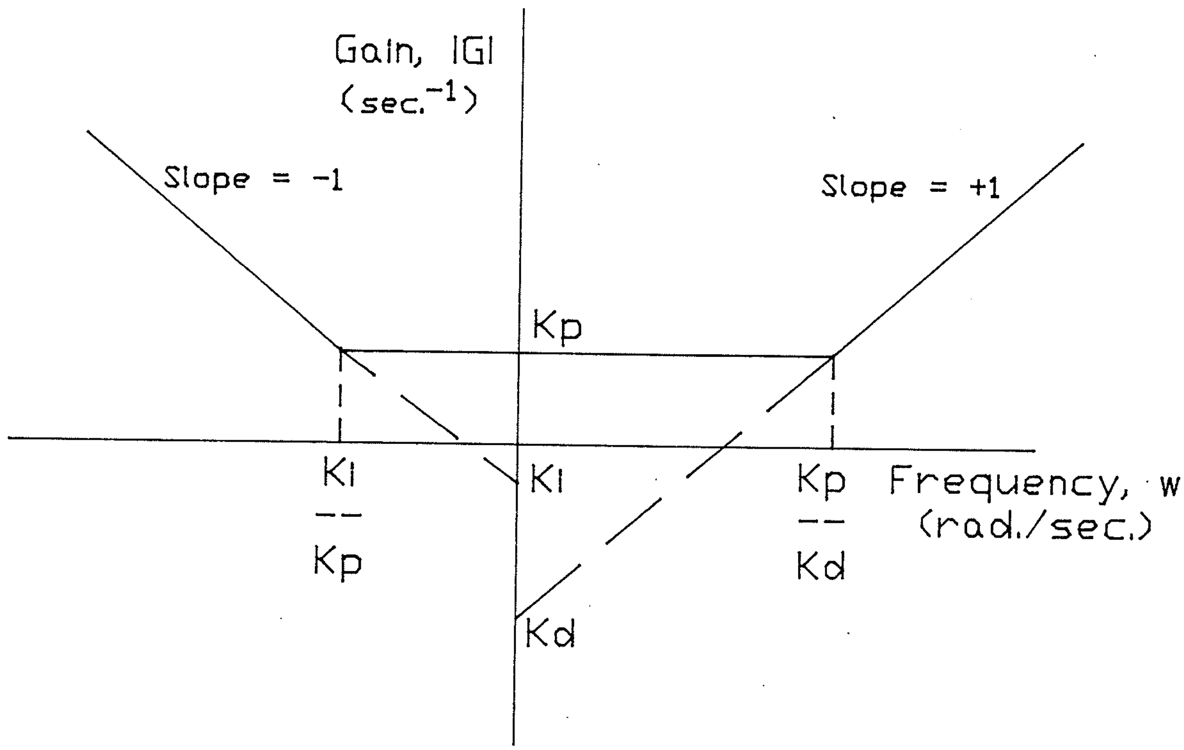


Figure 2.13 Typical Bode Plot of a PID Compensator

dominate the compensator and cause the motor to overshoot the final position.

The PID compensator discussed above can be implemented on a microcomputer as a difference equation [54]. This difference equation is given by [55]:

$$q(n) = K_p*e(n) + K_i*T_s*[\sum_{k=1}^{n} e(k)] + K_d/T_s*[e(n) - e(n-1)] \qquad (2.8)$$

where:

n = sample number,
q(n) = nth compensator output,
e(n) = nth position error input to the compensator,
k = a summation index,
T_s = sampling period (seconds).

Real-time calculations are required to implement the sampled-data PID controller. The machine drive's actual position error for a sample must be available for the calculations of equation (2.8). As well, floating decimal point calculations should be used in order to avoid quantization problems associated with a finite computer word length.

Now that the basic control theory and software requirements have been examined, the determination of the coefficients of the PID compensator for a machine tool controller will be discussed.

2.5.2 Determination of the PID Gain Coefficients

There is no simple rule for selecting the gains of the PID compensator [33, 56]. To determine the gain coefficients, it is necessary to analyze the requirements for machining accuracy. As discussed in Section 2.4, the accuracy of a machine tool controller may be described in terms of a following error, contour error, and a final position overshoot. The following error and the final position overshoot can be translated directly into control system requirements. The following error can be expressed in terms of the servo system's total gain. As established in Section 2.4, its value should be about one mil for each 2.0 inches/minute of axis speed for typical milling operations. This following error corresponds to an open-loop machine drive gain of 33.3 seconds^{-1} (2.0 inches/minute-mil * mil/0.001 inches * minute/60 seconds) . For the experimental system considered, the machine drive's gain, without compensation, is 37.1 as determined in Section 2.3.4. This gain indicates that the following error will be about 1 mil for each 2.2 inches/minute of axis speed. Thus, the following error specification is satisfied and the open-loop gain does not need to be increased for the experimental system. Consequently, the proportional gain of the PID compensator can be unity for the experimental system.

It is critical in metal removal that the toolbit does not overshoot the desired final position. An overshoot results in excessive material removal. The zero overshoot requirement indicates that the position control loop should be critically damped for operations involving contoured metal removal.

For a contouring machine tool, the integral gain reduces the following error but increases the final position overshoot. As a result, if the following error for the machine is acceptable, then the integral gain can be set to zero. In this situation, the PID position compensator is reduced to a proportional-derivative (PD) compensator. The only gain which must be determined still is the derivative gain. The derivative gain must be selected to achieve critical damping of the position control loop.

The forward path transfer function of the machine drive is obtained from equation (2.3). With a PD position compensator, the overall open-loop transfer function of the compensated position control system is then:

$$G_t(s) = G_f(s) * G_c(s) = \frac{K_t (K_p + K_d s)}{s (1 + \tau_m s)} \qquad (2.9)$$

and, by factoring K_p from the numerator,

$$G_t(s) = \frac{K_p K_t (1 + (K_d/K_p)s)}{s (1 + \tau_m s)}. \qquad (2.10)$$

Equation (2.10) indicates that the appropriate selection of the ratio of K_d/K_p will result in a pole-zero cancellation in the machine drive's transfer function. If the ratio K_d/K_p equals the time constant, τ_m, of the servo system, then the PD compensator's zero effectively cancels the motor's pole. Now the K_p value for the experimental system is unity so that the K_d value should equal to τ_m, or 0.030 seconds here, in order to achieve the pole-zero cancellation. In this situation, the behaviour of the machine drive is equivalent to a control system with a transfer function of $G(s) = K_p K_t/s$. Such a control system has a first-order exponential response which is critically damped. A closed-loop position control system with this transfer function will move to a desired position without overshoot. This principle will be utilized in Chapter 3 to design a critically damped position control system.

It must be noted that the pole-zero cancellation in the transfer function of the machine drive is accurate only if the motor's time constant does not change. If the response of the motor was slowed, for example, due to torque overloads, then the accuracy of the pole-zero cancellation would be degraded. Under these circumstances, the positioning performance of the servo drive would suffer too. However, under normal operating conditions for a CNC machine, it is reasonable to assume that the drive motors will not be operated beyond their torque limits. Hence, the pole-zero cancellation in the transfer function of the machine drive is valid.

2.6 Chapter Summary

The structure and analysis of control systems for machine tools has been introduced in this chapter. The functional elements of a sampled-data position control system have been discussed. The theory for the machine's drive system and for the compensation of the position control loop have been explained. Finally, the design and implementation of a PID position compensator to achieve a desired level of machining accuracy have been overviewed. The next chapter will provide the details of an experimental control system for a machine tool which was designed by using the theory of this chapter.

CHAPTER 3

OVERVIEW OF EXPERIMENTAL SET-UP

3.1 Introduction

A manual three-axis milling machine was modified for use as a CNC machine in order to evaluate the performance of the developed, microcomputer based, flexible machine tool controller. The hardware requirements for the CNC conversion will be discussed in this chapter. The software structure of the data processing unit and the control loops unit will be given too. The implementation of a backlash takeup routine as part of the control loops unit will be included in this discussion. Finally, the interface between the machine tool controller and a commercial part design system will be presented. It will be shown that the open software structure of the machine tool controller greatly increases the flexibility of the CNC system.

3.2 CNC Machine Tool Hardware

The modified milling machine is shown in Figure 3.1. A microcomputer was used to control the three axes of this milling machine. Commercial servo motors and amplifiers were purchased to drive each individual axis. As well, rotary incremental optical encoders were mounted on the ends of the motor shafts for indirect position feedback. A linear optical encoder was mounted on one table axis of the milling machine so that direct position feedback could be also obtained for that axis. This duplication allowed a comparison of the performance resulting from indirect and direct

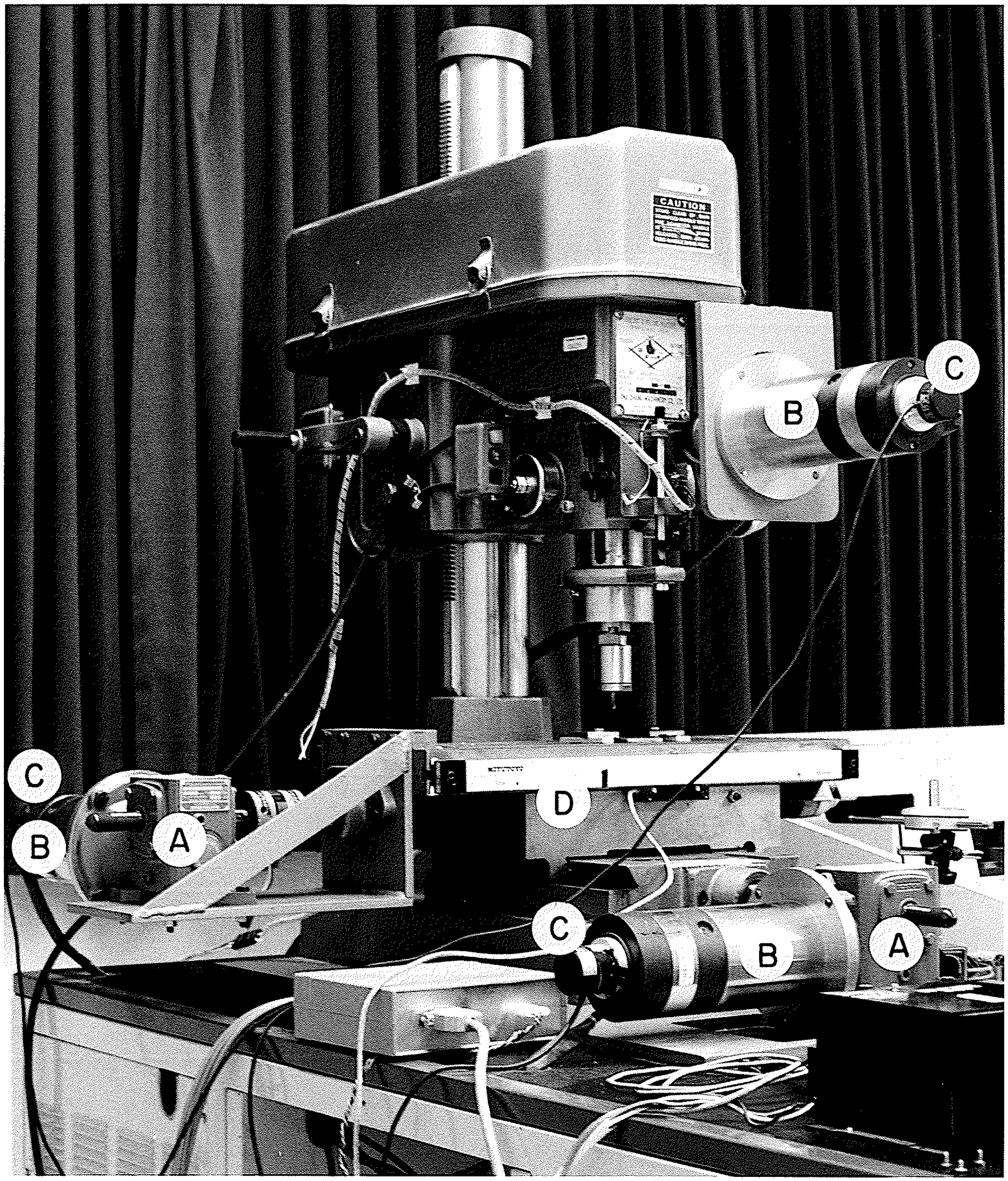


Figure 3.1 The Modified Milling Machine

closed-loop position feedback. Finally, custom developed electronics for motion control were used to interface the microcomputer, position encoders, and the servo motor system. This electronic circuitry will be discussed further in Section 3.2.4.

3.2.1 Milling Machine

The manual milling machine modified for use as a CNC machine is a Model LC30A manufactured by the Long Chang Machinery Company Ltd. The milling machine has dovetail slides on each of the table axes. The original handwheels for the three axes of the mill were replaced with couplers. The couplers provide a connection between the motor shafts and the leadscrews of the milling machine. A complete rotation of each leadscrew was found to produce 0.1 inches of linear motion. As a result, the milling machine's leadscrews have a pitch of 0.1.

The effect of external loads on the motors must be considered when designing a CNC machine tool. Cutting forces and frictional forces in the drive system produce loads on the motors. The effect of these loads can be reduced by using a gear reducer between each drive motor and leadscrew driving the CNC machine. Indeed, for a gear reduction ratio of N, the cutting loads are reduced by a factor of N. In addition, the gear reducer also increases the inertia of the motor drive by N^2 [57, 58]. Thus, the overall drive system is less sensitive to changes in load inertia (which depend on the weight of the workpiece being machined on the machine's table). Any change in the load inertia will be a smaller multiple of the motor's inertia with the use of a gear reducer [7, 57]. The gear reducer also converts the high rotational speeds of the motor into useful cutting speeds. For the experimental system discussed here, a gear ratio of ten-to-one was employed on each of the X and Y table axes as identified by Label 'A' in Figure 3.1. On the other hand, an external gear reducer was not used on the vertical Z axis because a worm gear reduction was present already.

3.2.2 Servo Motors and Amplifiers

The motor-amplifier combination selected for this work was an Electro-Craft, Model E703-MG, servo motor with a SA-9000 Series Pulse Width Modulated (PWM) amplifier. The servo motors are identified by Label 'B' in Figure 3.1. The amplifier electrically interfaces the controller and the motor and it has built-in velocity and current feedback loops. The velocity feedback stabilizes the speed response of the motor whilst the current feedback limits the motor's current to prevent damage. This dual feedback scheme improves the dynamic performance of the servo system. The amplifier also has a dynamic braking feature in which the motor is braked when a request is made for it to slow down. During dynamic braking, the kinetic energy of the motor is dissipated as heat through a resistance. Dynamic braking enables the motor system to be decelerated at a faster rate than that due to load friction alone. This feature linearizes the response of the motor for both acceleration and deceleration phases [29].

The presence of the velocity feedback inside the amplifier prevents complete control of the motor by using a microcomputer. As a result, the amplifier was tuned by appropriately adjusting potentiometers for a stable fast speed response to a voltage step input [17]. The position control loop, however, was closed inside the microcomputer itself. Details of the position control loop will be discussed later in this chapter.

3.2.3 Position Encoders

Rotary shaft encoders were used for position feedback from the axes of the milling machine. The three rotary encoders were mounted on the end of each shaft of the servo motors to provide indirect position feedback. The rotary encoders are identified by Label 'C' in Figure 3.1. A linear encoder was also mounted on the X axis of the milling machine to assess the accuracy of the corresponding leadscrew and, thus, to determine the reliability of the feedback from the rotary encoder. The linear encoder is identified by Label 'D' in Figure 3.1.

The position encoders used in a CNC system must have a higher resolution than the position resolution desired in the CNC system. The desired resolution was 0.0001 inches on each axis of the experimental system. This value should ensure that the target position and contouring accuracies of +/- 0.0005 inches are achieved. Three rotary incremental optical encoders, model REX-44 manufactured by SUNX, were mounted on the ends of the motor shafts [28]. Each of these digital encoders outputs two pulse streams which have a 90 degrees phase difference as shown in Figure 3.2. Channel B leads channel A for a clockwise encoder rotation and channel A leads channel B for a counter clockwise rotation. The rotary encoders also have an index pulse which occurs once every complete revolution of the encoder's shaft. This pulse can be used as a reference or home position. The effective resolution of the encoder can be quadrupled by using "quadrature decoding"

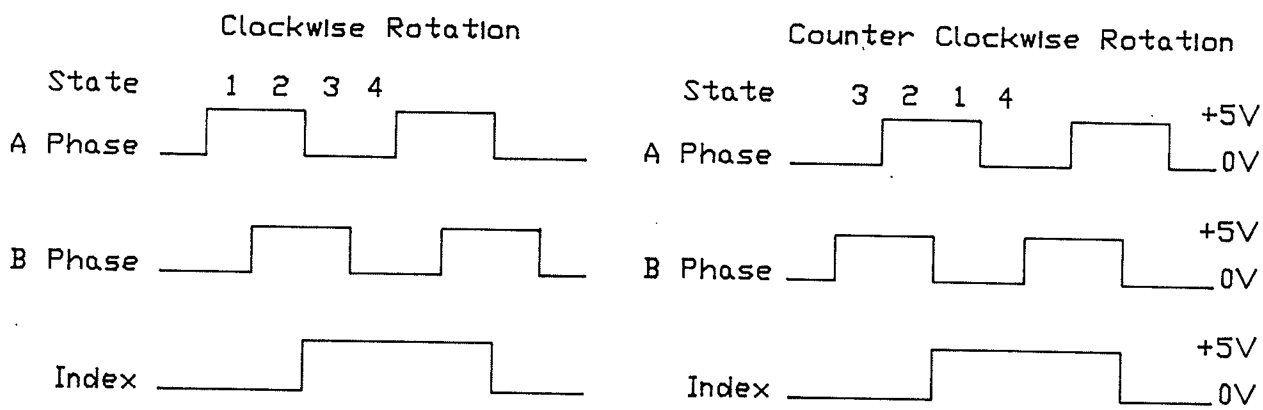
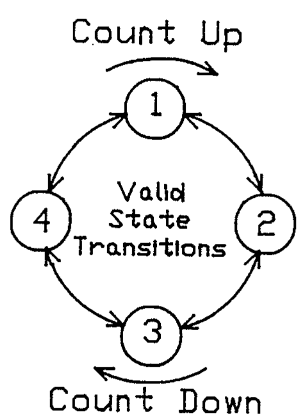


Figure 3.2 Pulse Outputs from an Optical Encoder



Ch. A	Ch. B	State
1	0	1
1	1	2
0	1	3
0	0	4

Figure 3.3 State Transition Diagram for Four Times Quadrature Decoding

circuitry [59, 60]. Direction information and a fourfold increase in resolution may be obtained by decoding the four unique states of channels A and B indicated in Figure 3.2. A state diagram which depicts the four states and possible transitions between them is presented in Figure 3.3. The state transition logic may be implemented by using flip-flops, a Programmable Logic Array, or a Read Only Memory [59, 60]. Thus, each pulse from channels A and B can be essentially quadrature decoded into four counts. The encoders on the X and Y axes of the machine's table produced 100 pulses per revolution of the motor shaft. Considering the ten-to-one gear reduction on these axes, 1000 pulses were generated for each revolution of the gear reducer's output shaft. With a leadscrew pitch of 0.1 inches/revolution, the machine drive's basic resolution on the X and Y axes was 0.0001 inches/pulse (0.1 inches/rev. * 1/1000 rev./pulse). The basic resolution can be increased by a factor of four to 0.000025 inches/count through the use of quadrature decoding. The quadrature decoded signals from the encoders were not actually used on the X and Y axes because the resolution of the position feedback corresponded already to the desired 0.0001 inches. On the other hand, the rotary encoder on the vertical Z axis produced 500 pulses per revolution of the motor shaft. With one revolution of the motor shaft producing 0.1 inches of vertical displacement, this drive's basic resolution was 0.0002 inches/pulse (0.1 inches/rev. * 1/500 rev./pulse). This resolution is lower than the desired 0.0001 inches, so that the quadrature decoding was used on the Z axis to improve the resolution to the desired 0.0001 inches.

The linear encoder on the X axis was manufactured by Mitutoyo [27]. The linear encoder is a model AT2-FN and it required a pulse signal interpolation unit PSU2-1 to provide a resolution of 2 micrometers or

0.0000787 inches/pulse. This resolution can be increased by a factor of four to 0.0000197 inches through the use of quadrature decoding. The high resolution of the position feedback from the linear encoder can be employed to assess the accuracy of the milling machine's leadscrew.

3.2.4 Interface for Motion Control

Figure 3.4 shows the custom developed interface card for motion control. This interface card allows the microcomputer to monitor and control each axis of motion. The electronic circuitry was designed by Mr. V. Shkawrytko and debugged by Mr. T. Kostyniuk. Mr. J. Sill of the Electrical Engineering Department, at the University of Manitoba, built the wire wrapped circuit card. The card was designed to be IBM PC data bus compatible and included an encoder interface section, a digital-to-analog converter, and a microcomputer data bus interface. Circuit diagrams are included in Reference [61].

An encoder interface is needed to determine the positions of the milling machine's axes. A commercially available, integrated circuit chip, was used for the encoder interfacing. The HCTL-2000 chip from Hewlett-Packard receives the feedback pulses from the optical encoder and performs quadrature decoding [62]. The HCTL-2000 chip has a 12 bit binary up/down counter and an output data latch. The up/down counter is used to distinguish positive from negative movements. The value in the counter can be transferred to the output data latch for further processing by a microprocessor. The HCTL-2000 chip samples the two encoder channels A and B at a selectable clock frequency and then filters the encoder pulses to

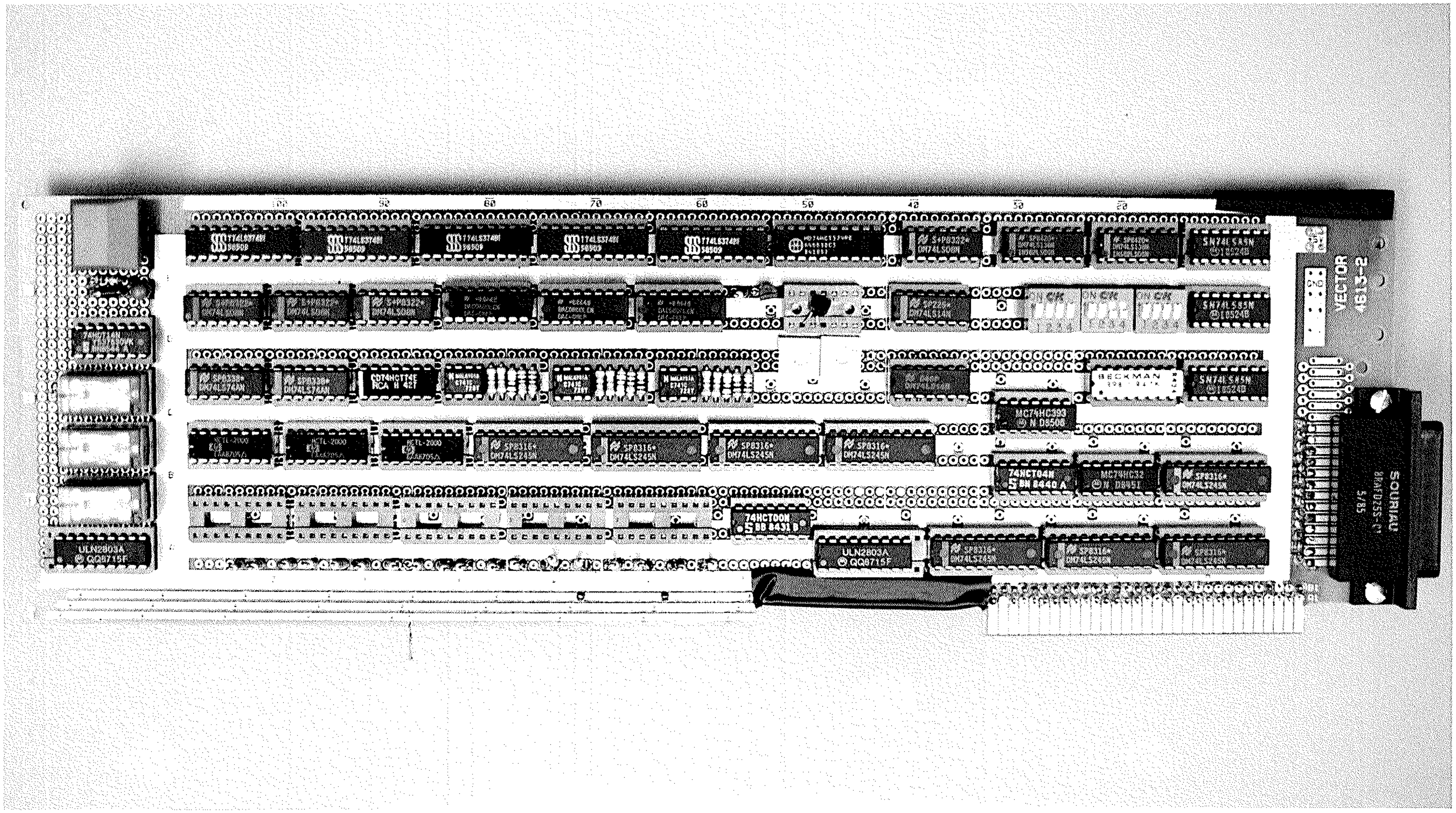


Figure 3.4 Custom Developed Interface Card for Motion Control

prevent extraneous noise from producing false counts. The filter section rejects noise on the incoming quadrature signals by requiring the same input level for each channel on three consecutive, rising clock edges. Provision was also made in the encoder interface circuitry to use the index pulse of an encoder as a home position reference. This feature is useful in CNC applications where it is desired to start a machining operation at a fixed starting position.

An axis interface must also contain a digital-to-analog converter (DAC) to provide a voltage to the drive's amplifier which corresponds to the speed reference. The output of the DAC should be a linear function of the digital input. A commercially available DAC-08 made by Precision Monolithics Incorporated was used to convert the digital compensator's output to a corresponding analog value [63]. The output voltage of this 8-bit DAC is applied to the motor's amplifier which, in turn, powers the motor. The voltage applied to the motor's amplifier reduces the error between the reference position and the actual position of an axis as discussed in Section 2.3.

The microcomputer's data bus interface enabled communication and data transfer to be made between the custom motion control electronics and the microcomputer. Information about each encoder's position and the outputs of the digital compensators were transferred over the computer's data bus. Details of the microcomputer hardware will be detailed next.

3.2.5 Microcomputer

The machine tool controller consists of a custom three-axis interface card together with an IBM compatible PC/AT computer. The computer is supported by an Intel 80286 microprocessor and must have an Intel 80287 math coprocessor. The 80286 is a 16-bit processor which uses a 12 MHz clock. The 80287 math coprocessor works in parallel with the 80286 microprocessor. It performs high speed arithmetic and trigonometric operations on floating point numbers [64]. FORTRAN allows the control program to employ the real time, floating point arithmetic support of the 80287. The single processor of the computer performs all the position control tasks and it interfaces to all three axes of the motion control card over the data bus of the computer.

The focus of this work was to implement an inexpensive, software based, machine tool controller. The PC/AT computer with an 80287 coprocessor costs about $3500, while the hardware for the interface card for motion control is about $300. Thus, hardware for the microcomputer based motion controller basically costs around $3800. This cost figure excludes development costs and the price of servo motors, servo amplifiers, and position encoders for the axes of the machine tool. The next two sections will discuss the design and implementation of the data processing unit and the control loops unit.

3.3 Data Processing Unit

As described in Section 1.3, the purpose of the data processing unit (DPU) is to provide speed reference signals to the control loops unit (CLU).

The DPU consists of a part program decoding subroutine and interpolation subroutines. The DPU reads the part program block, decodes it, and generates the interpolated positions between the end points of a contoured move. The software for the DPU and the CLU is implemented in FORTRAN and ASSEMBLER. FORTRAN provides powerful mathematical capabilities for programming the controller and the interpolation routines. On the other hand, ASSEMBLER subroutines are employed for the low level data input and output between the computer and hardware devices. Program listings for the DPU and CLU are included in Reference [61].

Three-dimensional linear interpolation and circular interpolation on any two axes are implemented as part of the DPU. Other interpolation routines, such as spline interpolation, could be also implemented due to the open software of the DPU. The interpolation routines provide a series of reference speeds to the CLU for each axis. The CLU uses these speeds to calculate the reference positions of the axes. For each axis, the difference between the actual and reference positions is used by the CLU to determine the motor's speed needed for the next sample period. Details of the operation of the CLU will be presented in Section 3.4.

The process of interpolation involves the approximation of a desired curve by many line segments. This segmentation is an inherent product of the sampled-data control scheme because the speed of the axes' motors can be changed only at the sampling instants. Thus, a circular arc is approximated as a series of line segments; each segment length represents the distance that the toolbit must travel in a single sample period. The segment length is proportional to the feedrate of the axis but inversely proportional to the sampling frequency. Thus, a contour such as a circular arc can be approximated more accurately by using a lower feedrate or a higher sampling frequency. This segmentation will be discussed later. The function of the part program decoder will be explained first.

3.3.1 Part Program Decoder

The part program decoder reads each block of the part program and interprets the actions required for each block. The part program decoder should correspond to the Electronics Industries Association Standard RS-274-D for numerically controlled machines which perform positioning and contouring [65]. Functions performed by the part program decoder include:

a) invoking the appropriate interpolation subroutine (e.g. G01 for linear interpolation, G02 for clockwise circular interpolation, and G03 for counterclockwise circular interpolation),

b) selecting the plane for two-dimensional circular interpolation (e.g. G17 for XY plane, G18 for XZ plane, and G19 for YZ plane), and

c) miscellaneous functions (e.g. M02 to signal the end of the part program, M03 to turn on the spindle motor , and M05 to turn off the spindle motor).

Other part program codes can be added to the DPU as required because of the open software. This capability is usually not available on commercial systems. The speed profile employed in the DPU will be discussed next.

3.3.2 Trapezoidal Speed Profile

A mismatch in the response times of the servo drives for the axes is the major source of contour errors as indicated in Section 2.4. It is preferable, therefore, to request gradual speed changes as opposed to large step changes in order to minimize the contour error. For this reason, ramp speed changes are normally used at the beginning and end of each part program move, respectively.

A trapezoidal speed profile is used for all linear and circular motions. To determine the speed profile, the absolute value of the travel distance along an axis, DELTA, must be calculated for a given motion. For linear interpolation, DELTA is the incremental distance between the initial and final points of the axis motion. For circular interpolation, DELTA is the incremental arc distance between the initial and final points of the arc. The area under the trapezoidal speed profile represents the total distance to be travelled by an axis. This area can be subdivided into three regions which represent acceleration, constant feedrate, and deceleration phases as shown in Figure 3.5. The variables PT1, PT2, and PT3 given in Figure 3.5 represent the number of sample periods that an axis must accelerate, travel at a constant feedrate, and decelerate, respectively.

For a given motion, the feedrate and toolbit's final position are specified by the part program. The incremental position change, DELTA, is simply the difference between the present toolbit position and the next specified position. After determining DELTA, the interpolator must find the number of sample periods, PT1, PT2, and PT3, that each axis must accelerate, travel at the specified feedrate, and then decelerate. The number

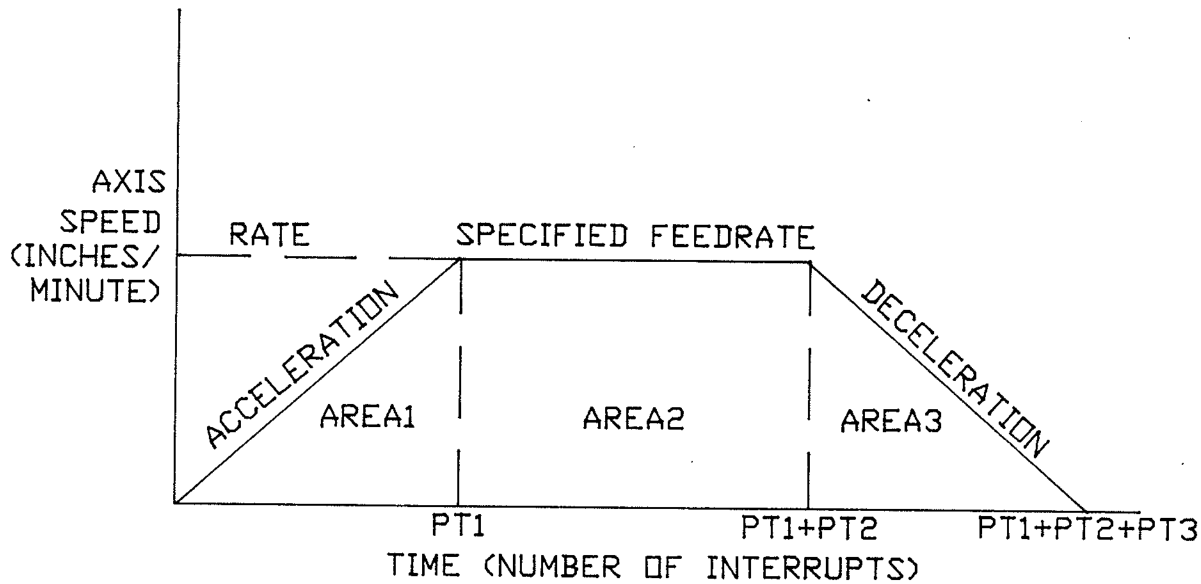


Figure 3.5 The Trapezoidal Speed Profile

of sample periods or interrupts of the microprocessor for the acceleration and deceleration portions of the typical speed profile shown in Figure 3.5 are calculated as:

$$PT1 = RATE / ACL$$
$$PT3 = RATE / DCL \qquad (3.1)$$

where:

RATE = specified feedrate (BLUs/interrupt),
ACL = acceleration rate ($BLUs/interrupt^2$),
DCL = deceleration rate ($BLUs/interrupt^2$),
PT1 = time to accelerate (interrupts),
PT3 = time to decelerate (interrupts).

The data processing unit (DPU) converts the feedrate from the typical units used in a part program of inches/minute to BLUs/interrupt. It also converts the acceleration and deceleration rates from $inches/second^2$ to $BLUs/interrupt^2$. The acceleration and deceleration rates are not specified in the part program but they can be set to suitable values in the DPU. The acceleration and deceleration rates were chosen as 1.0 $inches/second^2$. This acceleration and deceleration value provided a fast speed response without producing a current overload in the motor's amplifier.

The total area under the speed profile is equal to the distance travelled during a motion because the distance is simply the integral of the speed. Consequently, the distances travelled while accelerating and decelerating equal to the areas under the corresponding portions of the speed profile. By using the values of PT1 and PT3 obtained from equation (3.1), the distances travelled while accelerating and decelerating are given, respectively, by:

$$\text{AREA1} = 0.5 * \text{PT1} * \text{RATE}$$
$$\text{AREA3} = 0.5 * \text{PT3} * \text{RATE}, \tag{3.2}$$

while the area under the constant speed portion of the speed profile is:

$$\text{AREA2} = \text{DELTA} - \text{AREA1} - \text{AREA3} \tag{3.3}$$

where:

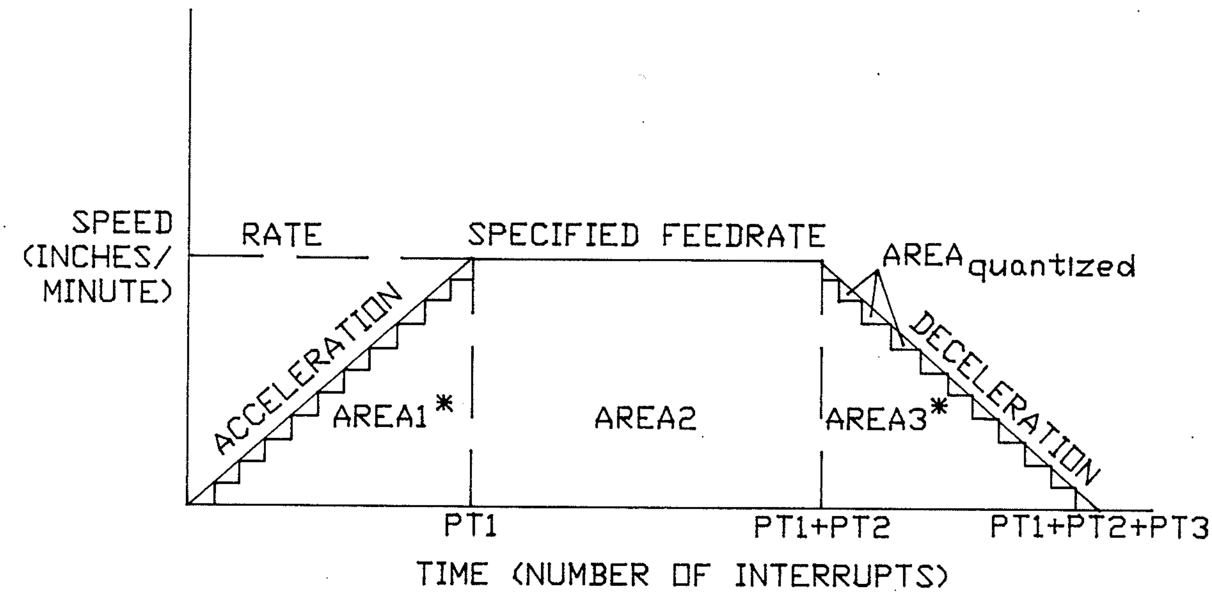
DELTA = total travel distance (BLUs),
AREA1 = distance travelled while accelerating (BLUs),
AREA2 = distance travelled at constant speed (BLUs),
AREA3 = distance travelled while decelerating (BLUs).

AREA2 is used to calculate the number of interrupts, PT2, required for the constant feedrate portion of the speed profile. This number is given by:

$$\text{PT2} = \text{AREA2} / \text{RATE} . \tag{3.4}$$

Now AREA2 can sometimes be negative if the feedrate specified by a part program is too high for the required distance of travel. Under these circumstances, the DPU program automatically reduces the feedrate to 90% of the present feedrate and recalculates the three areas. This procedure continues until a feedrate is found that can be reached for at least a single interrupt period.

The discrete nature of the acceleration and deceleration phases results in a quantization error which must be considered when stopping an axis. The acceleration and deceleration profiles are represented as linearly varying changes in velocity. The deceleration profile is recalculated to compensate for the reduction in area caused by quantization of the velocity. The quantization error during the acceleration or deceleration portion of the speed profile illustrated in Figure 3.6 is:



* AREA1 and AREA3 refer to the total areas under the acceleration and deceleration portions of the speed profile.

Figure 3.6 The Quantization of the Speed Profile

$$AREA_{quantized} = 0.5 * RATE . \quad (3.5)$$

Thus, the area under the deceleration portion of the speed profile is recalculated to account for the quantization occurring during the acceleration and deceleration phases. Consequently, the modified area, AREA3M, is calculated as :

$$AREA3M = AREA3 + 2*AREA_{quantized} = AREA3 + RATE. \quad (3.6)$$

The deceleration is then modified to reflect the modified distance to reach the final desired position. The corresponding deceleration, DCLMOD, is given by:

$$DCLMOD = 0.5 * RATE^2 / AREA3M \quad (3.7)$$

so that the corresponding modified number of interrupt periods, PT3MOD, for deceleration is:

$$PT3MOD = RATE / DCLMOD . \quad (3.8)$$

It must be noted that the number of interrupt periods calculated by using equation (3.8) must be an integer. If the PT3MOD value has a fractional remainder, then this fraction is multiplied by the modified deceleration, DCLMOD, to obtain the scaled deceleration for the very last interrupt period before stopping the motor. In this way, the reference speed profile ends with a zero velocity at the specified final position of travel.

3.3.3 Three-Dimensional Linear Interpolation

Three-dimensional linear interpolation is implemented as part of the data processing unit of the machine tool controller. The linear interpolation subroutine receives the final desired position of the X, Y, and Z axes as well as the cutting feedrate from the decoding subroutine. The present positions of the axes are known to the microcomputer. The linear interpolator calculates the axial velocities such that:

$$RATE = \sqrt{(V_x^2 + V_y^2 + V_z^2)} \qquad (3.9)$$

where:

V_x = velocity of the X axis (BLUs/interrupt),
V_y = velocity of the Y axis (BLUs/interrupt),
V_z = velocity of the Z axis (BLUs/interrupt).

To determine the axial velocities, the total distance for a three-dimensional move must be calculated first as:

$$DELTA = \sqrt{(DELTAX^2 + DELTAY^2 + DELTAZ^2)} \qquad (3.10)$$

where:

DELTAX = total travel distance in the X direction (BLUs),
DELTAY = total travel distance in the Y direction (BLUs),
DELTAZ = total travel distance in the Z direction (BLUs).

The direction cosines, which represent the projections of the total movement on each of the axes, are determined as:

$$\begin{aligned} DCOSX &= DELTAX / DELTA, \\ DCOSY &= DELTAY / DELTA, \\ DCOSZ &= DELTAZ / DELTA \end{aligned} \qquad (3.11)$$

where:

DCOSX = direction cosine in the X direction,
DCOSY = direction cosine in the Y direction,
DCOSZ = direction cosine in the Z direction.

By using the direction cosines, the velocity components for each axis during the constant speed portion of a movement are:

$$V_x = DCOSX * RATE$$
$$V_y = DCOSY * RATE \qquad (3.12)$$
$$V_z = DCOSZ * RATE \, .$$

The linear interpolation routine uses a moving feed vector to move the toolbit along the desired path from the starting to the end position. This feed vector changes both in magnitude and position. It points in the direction from the toolbit's present, instantaneous reference position to the desired final position. Its magnitude is provided by the previously described trapezoidal speed profile. At each sampling instant, the 'tail' of the feed vector advances to the previous reference position. Then the axial components of the feed vector are utilized as the speed inputs to the control loops unit. These speed inputs are used by the CLU to calculate the new reference position for each axis. Thus, the reference feed vector advances along the desired path by the distance specified by the feed vector of the previous sample. The feed vector moves along the path is this manner until the end of the path is reached.

3.3.4 Two-Dimensional Circular Interpolation

Circular interpolation in the XY, XZ, and YZ planes is also part of the data processing unit. As mentioned in Section 3.3, the process of interpolation involves the approximation of a curve by a series of line segments. At each sample, the interpolator specifies the next reference point to which the axes should move during the upcoming sampling period. The CLU drives the axes' motors so that the toolbit moves along the desired path. However, the actual axes' positions lag behind the reference positions due to the error driven nature of the position control system which was discussed in Chapter 2.

The segmentation of a line is not of concern in linear interpolation because each reference position and each feed vector lies exactly upon the desired line. However, during circular interpolation, the line segments approximating the desired arc do not lie exactly on the arc. The contouring error introduced by the linearization can be calculated for the worst case condition, which corresponds to the maximum feedrate. For the experimental servo system, the maximum segment length is 0.00267 inches at the maximum feedrate of 24 inches/minute and the chosen 150 Hertz sampling frequency. Figure 3.7 shows a conceptual diagram of the line segments which form a circular arc. The small line segments, which approximate the circular arc, change in magnitude, direction, and position as the motors accelerate, travel at constant speed, and then decelerate. The circular arc separates gradually from each tangential line segment as shown in Figure 3.7. Thus, the maximum contour error during each sample occurs at the end of the line. Application of Pythagoreus' theorem indicates that the contour error, E, will be less than the axes' resolution of 0.0001 inches for arc radii which are greater than 0.040 inches at the maximum feedrate of 24 inches/minute. This small contour error indicates that, for the normal operating ranges of the experimental milling machine, the linearization of

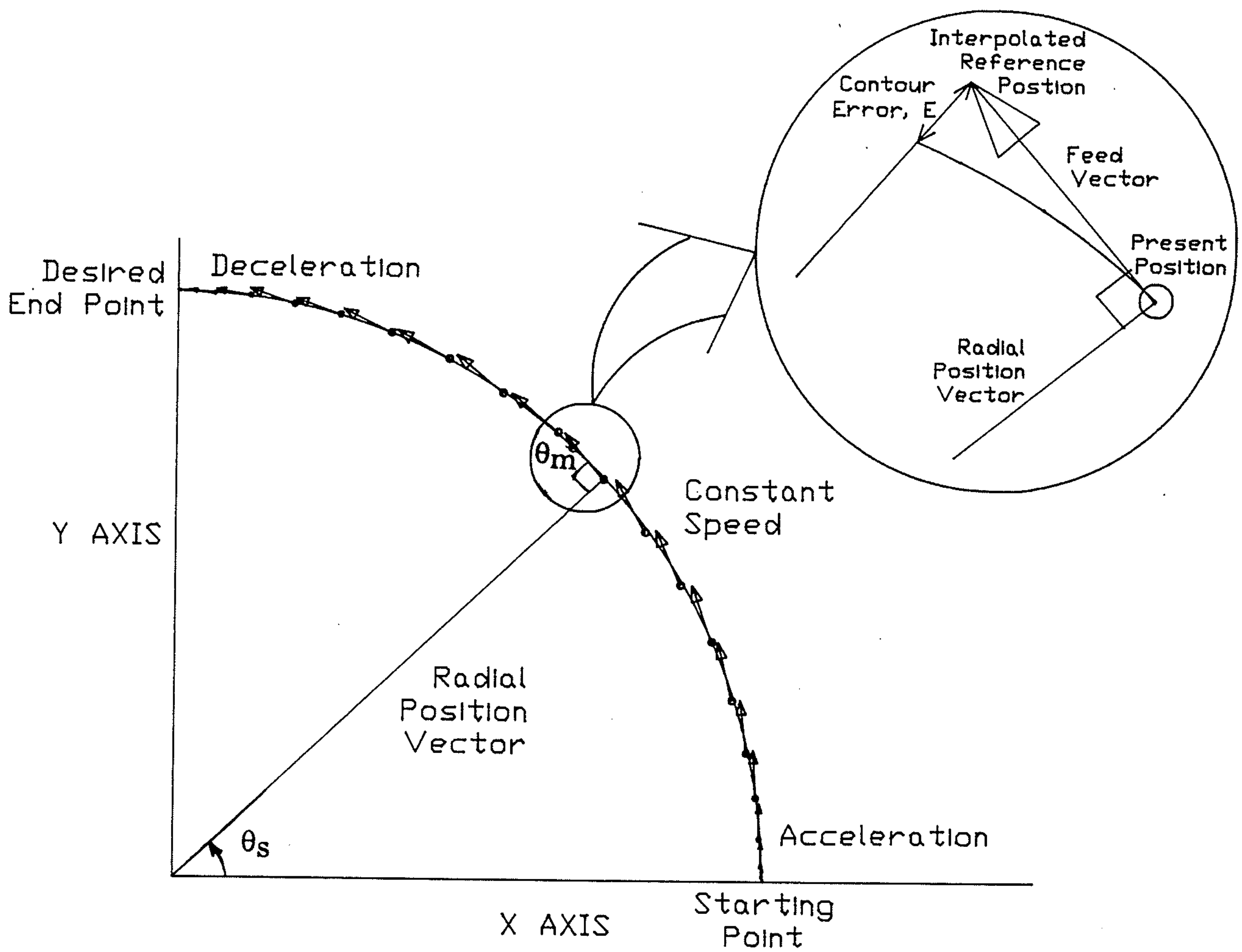


Figure 3.7 Conceptual Diagram of the Segmentation of a Circular Arc

circular arcs does not contribute significant contour errors. Details of the two-dimensional circular interpolator will be presented now.

A circular interpolator was designed to accept standard part program codes [65]. These codes require that all circular arcs start and finish in the same quadrant. The codes also require:

a) a code which defines the plane of the circular interpolation,

b) coordinates which give the final toolbit position,

c) two values which define the absolute offsets of the centerpoint of the arc from the present toolbit position. (The X, Y, and Z axis offsets are designated by the letters I, J, and K, respectively.),

d) a code which indicates if the toolbit's movement is clockwise or counter clockwise, and

e) the desired feedrate along the path.

The information provided in the part program, along with the toolbit's present position, is the only information needed to perform the circular interpolation. The basic information that is required to begin the interpolation process is:

a) the arc's starting point,

b) the arc's final point,

c) the arc's centerpoint, and

d) the desired feedrate.

The arc's starting point is available within the computer and the final toolbit position and feedrate desired are given by the part program. The arc's centerpoint is calculated from the offset values. For example, in the XY plane, the I and J codes represent the absolute values of the X and Y component offsets of the arc's centerpoint from the present position of the toolbit. The interpolation subroutine must determine the signs of the offsets. This is accomplished by finding the I and J sign combination which produces an arc's centerpoint that is equidistant from the toolbit's present position and the required final position. The circular interpolation routine checks the four possible I and J sign combinations to determine the correct offset signs. After finding the offset signs, the circular interpolator determines the starting angle and the final angle for the circular motion. These angles and the arc's radius are used to calculate the total movement, DELTA, used in the speed profile calculations described in Section 3.3.2.

The circular interpolation routine uses a moving feed vector to move the toolbit along the desired arc. The magnitude, direction, and position of this vector change at each sampling instant. The magnitude of the feed vector is provided by the trapezoidal speed profile. The feed vector moves at right angles to the radial position vector as shown in Figure 3.7. Thus, the direction of the feed vector is always tangent to the desired reference arc. The reference position vector is updated at each sampling instant as it moves around the desired reference arc.

For two-dimensional circular interpolation, the reference positions of the two axes vary in a sinusoidal fashion [66, 67]. For instance, the velocity components for the X and Y axes during the constant speed portion of a circular movement are:

$$V_x = RATE * COS(\theta_s + \theta_m) \qquad (3.13)$$
$$V_y = RATE * SIN(\theta_s + \theta_m) \qquad (3.14)$$

where:

V_x = velocity of the X axis (BLUs/interrupt),
V_y = velocity of the Y axis (BLUs/interrupt),
RATE = specified feedrate (BLUs/interrupt),
θ_s = angle of the position vector at the start of a sample during circular interpolation (radians),
θ_m = angle of the feed vector with respect to the angle of the position vector (radians).

The θ_m value is $\pi/2$ for clockwise circular interpolation and $-\pi/2$ for counter clockwise circular interpolation.

The time varying axial velocities of the feed vector represent the component speeds for the motor drives. These components are used as speed references by the CLU which then updates the reference positions for the axes. The operation of the CLU will be discussed next.

3.4 Control Loops Unit

The control loops unit (CLU) drives the axes' motors according to the speed references generated by the interpolation routines of the data processing unit (DPU). The CLU subroutine samples the digital-to-analog converter (DAC) and the HCTL-2000 encoder interface. The HCTL-2000 encoder interface is used as an absolute counter to eliminate the possibility of missing encoder counts. The CLU receives the required speeds of the axes from the interpolation subroutine. It then waits for an interrupt from an ASSEMBLER subroutine which indicates that the axes have been sampled. The period between consecutive samples allows an interpolation routine to be implemented in real time. After the interrupt has occurred, the computer uses the counter values of the new position to determine the absolute positions of the axes. A new reference position is determined from the speed reference so that the error in the current position can be calculated. The position error is used by the digital compensator to find the new digital input to the DAC. The resulting proportional analog output of the DAC is applied to the amplifier which powers the motor.

Sampling performed by the CLU subroutine is controlled by one of the interrupt timers of the computer's 8254 timer chip [64]. The time of day clock of this chip is reprogrammed to interrupt the microprocessor at the desired sampling frequency. A sampling frequency of 150 Hertz was chosen for the experimental system, as discussed in Section 2.3.4. This sampling frequency is more than twenty times the bandwidth of the experimental servo system.

The microprocessor's idle time was monitored between sampling instants to determine the microcomputer's utilization at the chosen 150 Hertz sampling frequency. This measurement was accomplished by setting a bit high whenever the control program was waiting for an interrupt from the timer chip. The bit was monitored by using an oscilloscope. It was found that the computer was waiting approximately half the time between interrupts for a linear three-dimensional movement. This observation indicates that the computer's utilization was about 50% for this test. The computer's utilization ultimately depends, of course, on the complexity of the interpolation routine used.

Details of the absolute position counting scheme, the position loop compensation, and the backlash compensation which was implemented in the CLU will be discussed next.

3.4.1 Counting Scheme for Determining Absolute Position

A counting scheme to determine the absolute position of an axis is used with the HCTL-2000 encoder interface. The 12 bit binary up/down counter of the HCTL-2000 chip is reset at the start of a motor's operation. The counter is sampled at the sampling frequency of the position control loop. At each sampling instant, the counter's value is read by the computer's control program to determine the actual position of an axis. The counter is used as an absolute counter as opposed to resetting it after each sampling instant. An absolute counting scheme eliminates the possibility of missing an encoder count during the instant that the counter is being read or reset. To implement this absolute counting scheme, provision must be made for rollover of the 12 bit counter which can count from 0 to 4095. The algorithm which performs the absolute counting compares the change in the counter's value between samples. At the 24 inches/minute maximum operating speed of the experimental system, the counter's value should not change between samples by more than about 107 counts (24 inches/min. * 1 min./60 sec. * 40000 quadrature counts/inch / 150 Hertz). Thus, if the absolute value of the change in the counter's position is more than, say, 1000 between samples, then this situation indicates that the counter has rolled over. Consequently, 2^{12} must be subtracted from or added to the absolute position in order to determine the actual position of an axis. If the change in position is greater than 1000, then 2^{12} or 4096 is subtracted from the absolute position. Conversely, if the change in position is less than -1000, then 4096 is added to the absolute position.

3.4.2 Position Loop Compensation

The position loop compensation used in the experimental system is a proportional-integral-derivative (PID) type. The PID compensator is implemented in the form of difference equation (2.8). Its gains are set to achieve a critically damped response in the position control loop along with a following error which meets the specifications discussed in Section 2.4.2. With critical damping in the position control loop, the toolbit should move to the desired final position without overshoot. As well, the position loop gain should be at least 33.3 $seconds^{-1}$ to satisfy the following error specification of 1 mil for each 2 inches/minute of axis feedrate established in Section 2.4. For the experimental system considered in this work, the machine drive's gain is 37.1 $second^{-1}$ without compensation so that the proportional gain may be set to unity. As determined in Section 2.5.2, the PID gains which produce critical damping are $K_p = 1.0$, $K_i = 0.0$, and $K_d = 0.030$. Thus, the position loop compensator is reduced to a proportional-derivative (PD) controller. The zero of the PD compensator at $K_d/K_p = 0.030$ effectively cancels the motor's pole at $\tau_m = 0.030$ seconds. As a result, the behaviour of each machine drive is equivalent to a first-order control system with a critically damped response. The gains of the PD controller are programmed in software so that the controller could be customized easily to suit this particular machine tool application.

The open software structure of the machine tool controller permits the implementation of additional CNC features. One such feature is backlash compensation which will be discussed next.

3.4.3 Backlash Compensation

A software based, backlash takeup subroutine was developed as part of the control loops unit. This backlash routine compensates for the backlash between the leadscrew and the table's nut whenever a direction reversal occurs on an axis. To use the routine, it is necessary to experimentally determine, for each axis, the amount of lost motion due to backlash. These values can be determined to the nearest ten thousandths of an inch for the axes by using a precision dial gauge. The probe of the dial gauge is placed against the table and the encoder's output is monitored while an axis is moved manually. The motion direction of the axis is reversed and encoder's pulses are counted until the dial gauge registers the movement of the table. The number of encoder pulses counted from the start of motion to the first movement of the dial gauge indicator gives the backlash for that axis. The backlash values of the experimental milling machine were measured to be 0.0410, 0.0360, and 0.0010 inches on the X, Y, and Z axes, respectively. These values are programmed as constants in the CLU.

The algorithm for the backlash compensation uses three position variables R0, R1, and R2 for each axis. These variables represent the position of each axis at the next, the present, and the previous part program moves. They can be used to determine the direction of motion of an axis between consecutive movements which, in turn, dictates if backlash compensation is needed. The backlash compensation algorithm works as follows. It is assumed initially in the algorithm that the backlash is taken up on each axis in the positive direction. After each part program move, the R0, R1, and R2 values are updated whenever the position for an axis has changed between consecutive moves. Otherwise, these values remain unchanged. If the value of R1 is smaller than R2 but larger than R0 for a particular axis, then the axis has been moving in the negative direction and it will be moving positively now. Thus, the backlash must be compensated before this next move of the axis in the positive direction. Similarly, if R1 is larger than R2 but smaller than R0, then the axis has been moving in the positive direction and will be moving negatively now. Thus, the backlash must be compensated before this next move of the axis in the negative direction. The backlash routine determines if there will be a direction reversal on an axis before each part program move. If such is the case, then the routine moves all axes requiring backlash takeup before the start of the next move.

The backlash compensation routine works in conjunction with the part program decoder of the data processing unit. The DPU calls the backlash takeup routine before executing the next move of a part program to check if any axis will reverse direction on the next move. In machining practice, however, it is necessary sometimes to look two moves ahead to determine if backlash compensation is needed. An example of this type of situation is illustrated in the part program of Figure 3.8. In typical machining practice, the toolbit rapidly traverses to a check height at the surface of the workpiece before cutting into the material. The toolbit is then commanded to machine the workpiece. The backlash on the axes must be taken up before the toolbit moves down into the workpiece to proceed to machine. For such a situation, the part program decoder must "look" at the X and Y axis coordinates for move N05 before drilling into the material specified by move N04.

```
N01 G00 X+00000 Y+00000 Z+00500 F0020
N02 G00 X+01000 Y+02000 Z+00500 F0020
N03 G00 X+01000 Y+02000 Z+00000 F0020
N04 G01 X+01000 Y+02000 Z-00050 F0005
N05 G01 X+02000 Y+01000 Z-00050 F0005
```

Figure 3.8 Part Program Code Illustrating a Situation Requiring Special Backlash Compensation

The CAD system interface to the developed machine tool controller will be presented now.

3.5 CAD System Interface

One of the major disadvantages of stand-alone or dedicated controllers is that they cannot be interfaced easily to CAD systems. These stand-alone controllers often must be programmed by using a non-standard CNC programming language. Thus, it is usually necessary to manually enter the coordinates specifying the geometry through a keypad. If a design must be changed significantly, then this time consuming process must be repeated. A machine tool controller with an open software, on the other hand, can be interfaced quite easily to a CAD system.

The developed machine tool controller was interfaced to a commercial CAD system in order to achieve the full benefits of a CNC system. There are many CAD systems that could be used for this purpose. Examples include SmartCAM™ by Point Control Co. [68] and AUTO-CAM™ by ICAM

Technologies Corporation [69]. The interactive graphics of these CAD packages permits part design, modification, and verification prior to actual machining. These CAD systems also have many time-saving features such as canned pocket milling cycles and toolbit offset compensation.

The CAD system used in this thesis is SmartCAM™. SmartCAM™ allows the user to customize the output of the part program post-processor to the requirements of the machine tool controller. This customization involves developing a post-processor for the specific machine tool controller. The post-processor is created by answering a set of questions about the machine tool and the controller. For example, the user must define the number of axes, the desired system of units for part design, and the position resolution on the axes. A summary of the post-processor creation for SmartCAM™ is included in Reference [61].

The part design process consists of the following steps:

a) defining the workpiece material, cutting tools required, and the appropriate cutting speeds,

b) designing the shape of the part by using the CAD system,

c) entering the desired machining tool path,

d) verifying the completed tool path,

e) generating the part program code.

Figure 3.9 shows the path of the tool for a typical geometry which was designed by using SmartCAM™. The geometry is the logo of the University of Manitoba. It should demonstrate the interpolation capabilities of the machine tool controller because it contains both linear and circular toolbit

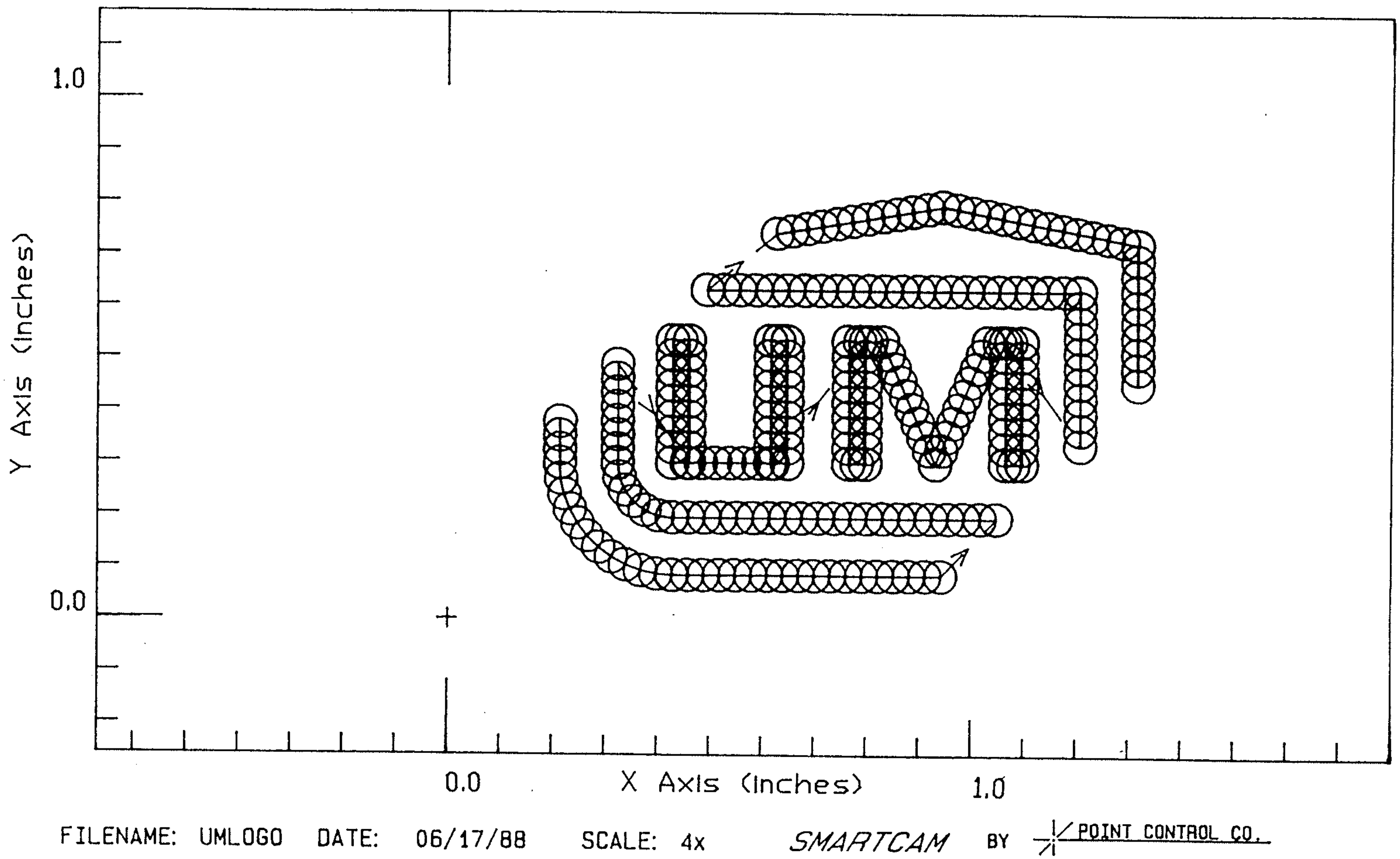


Figure 3.9 Tool Path of the University of Manitoba Logo

```
N001 G00 X+00000 Y+00000 Z+00500 F0018 M03
N002 M00
N003 G00 X+00214 Y+00378 Z+00500 F0018
N004 G00 X+00214 Y+00378 Z+00500
N005 G00 Z+00000
N006 G01 Z-00020 F0004
N007 G01 X+00214 Y+00300 Z-00020
N008 G03 X+00428 Y+00086 I+00214 J+00000
N009 G01 X+00941 Y+00086 Z-00020
N010 G00 X+00941 Y+00086 Z+00500
N011 G00 X+01047 Y+00196 Z+00500 F0018
N012 G00 Z+00000
N013 G01 Z-00020 F0004
N014 G01 X+00428 Y+00196 Z-00020
N015 G02 X+00324 Y+00300 I+00000 J+00104
N016 G01 X+00324 Y+00488 Z-00020
N017 G00 X+00324 Y+00488 Z+00500
N018 G00 X+00461 Y+00301 Z+00500 F0018
N019 G00 Z+00000
N020 G01 Z-00020 F0004
N021 G01 X+00461 Y+00535 Z-00020
N022 G01 X+00428 Y+00535 Z-00020
N023 G01 X+00428 Y+00301 Z-00020
N024 G01 X+00648 Y+00301 Z-00020
N025 G01 X+00648 Y+00535 Z-00020
N026 G01 X+00615 Y+00535 Z-00020
N027 G01 X+00615 Y+00301 Z-00020
N028 G00 X+00615 Y+00301 Z+00500
N029 G00 X+00797 Y+00535 Z+00500 F0018
N030 G00 Z+00000
N031 G01 Z-00020 F0004
N032 G01 X+00797 Y+00301 Z-00020
N033 G01 X+00765 Y+00301 Z-00020
N034 G01 X+00765 Y+00535 Z-00020
N035 G01 X+00830 Y+00535 Z-00020
N036 G01 X+00930 Y+00301 Z-00020
N037 G01 X+01031 Y+00535 Z-00020
N038 G01 X+01096 Y+00535 Z-00020
N039 G01 X+01096 Y+00301 Z-00020
N040 G01 X+01064 Y+00301 Z-00020
N041 G01 X+01064 Y+00535 Z-00020
N042 G00 X+01064 Y+00535 Z+00500
N043 G00 X+01208 Y+00337 Z+00500 F0018
N044 G00 Z+00000
N045 G01 Z-00020 F0004
N046 G01 X+01208 Y+00632 Z-00020
N047 G01 X+00495 Y+00632 Z-00020
N048 G00 X+00495 Y+00632 Z+00500
N049 G00 X+00628 Y+00742 Z+00500 F0018
N050 G00 Z+00000
N051 G01 Z-00020 F0004
N052 G01 X+00945 Y+00795 Z-00020
N053 G01 X+01317 Y+00723 Z-00020
N054 G01 X+01317 Y+00453 Z-00020
N055 G00 X+01317 Y+00453 Z+00500
N056 G00 Z+00500
N057 G00 X+00000 Y+00000 M05
```

Figure 3.10 Part Program from SmartCAM™ for the University of Manitoba Logo movements. Figure 3.10 shows the post-processed part program obtained from SmartCAM™. The use of a CAD package with a customized post-processor to generate part program code greatly simplifies the part design and manufacture processes.

3.6 Chapter Summary

The hardware and software structure developed for a machine tool controller have been detailed in this chapter. The implementation of the data processing unit and control loops unit of the machine tool controller was presented. The flexibility of the software based, machine tool controller was highlighted by the ability to select the position loop compensator and to add backlash compensation. Finally, the machine tool controller was interfaced to a commercial part design system to demonstrate its flexibility. The next chapter will provide results from an evaluation of the performance of this machine tool controller.

CHAPTER 4

PERFORMANCE OF THE MACHINE TOOL CONTROLLER

4.1 Introduction

The most important performance criterion for a CNC machine tool controller is machining accuracy. The machine tool controller must maintain a specified machining accuracy over the entire range of cutting speeds of the CNC machine. As described in Section 2.4, the accuracy of a machine tool may be described in terms of a following error, contour error, and a final position overshoot. Each of these criteria will be analyzed in this chapter for the experimental system. The theoretical and actual following errors of the milling machine's axes will be compared. The theoretical following error is based on the mathematical model developed in Chapter 2, while the actual following error is obtained from experimental tests. The effectiveness of the position loop compensator in eliminating a final position overshoot will be demonstrated. Contour errors resulting from mismatches in the time response characteristics of the axes' motors will be also discussed.

4.2 Performance of the Control Loops Unit

The performance of the control loops unit (CLU) can be described for a single machine axis in terms of the position following error and the overshoot of the final position. The mathematical analysis given in Section 2.3.1 indicated that the servo motor's drive system can be modelled as a type '1' control system. It was shown in Figure 2.8 that there is a position following error equal to speed/K_t for a constant speed input. Thus, the actual position of the toolbit lags behind the reference position by a constant distance which is inversely proportional to the gain of the motor's drive system. Figure 4.1 illustrates a typical position lag. Now the standard trapezoidal speed profile uses ramp speed inputs to accelerate and decelerate the motors. The response of a type '1' control system to a ramp speed input was shown in Figure 2.8. The speed error is equal to acceleration/K_t. Thus, the speed error is inversely proportional to the gain of the motor drive, K_t. Figure 4.2 presents typical reference and actual speed profiles for an axis speed of 24 inches/minute. It is desirable to have a high gain in the machine drive's control loop in order to reduce the position following error and the speed error. The variation of the position following error with speed will be presented next for the experimental system.

4.2.1 Position Following Error

The variation of the position following error is tabulated in Table 4.1 for various speeds. Table 4.1 compares the predicted following error to that obtained from experimental measurements. The following errors are tabulated up to the top speed of 24 inches/minute for the experimental system. The close agreement between the theoretical and actual position following errors verifies the model of the motor's drive system developed in Section 2.3.1.

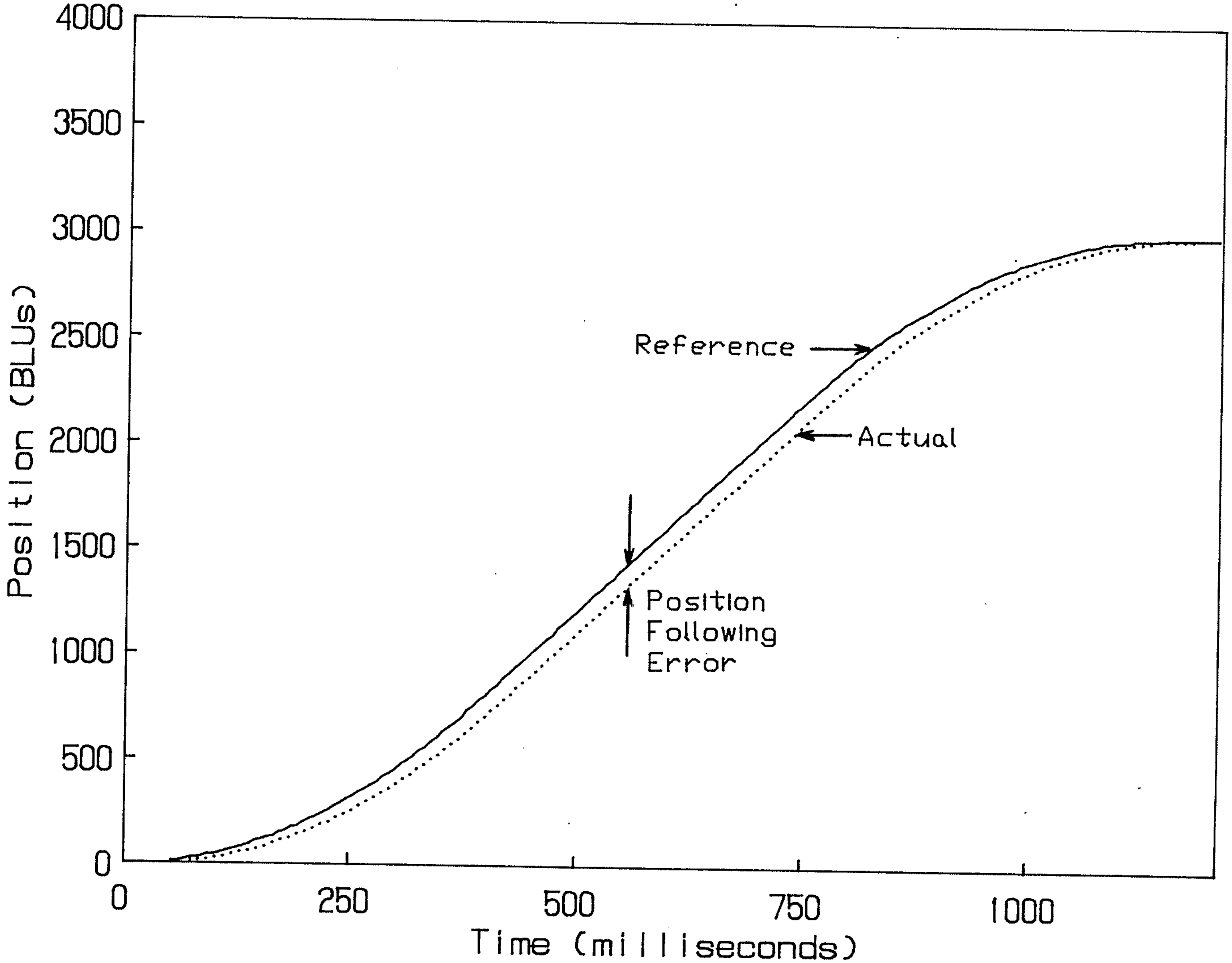


Figure 4.1 A Typical Position Following Error

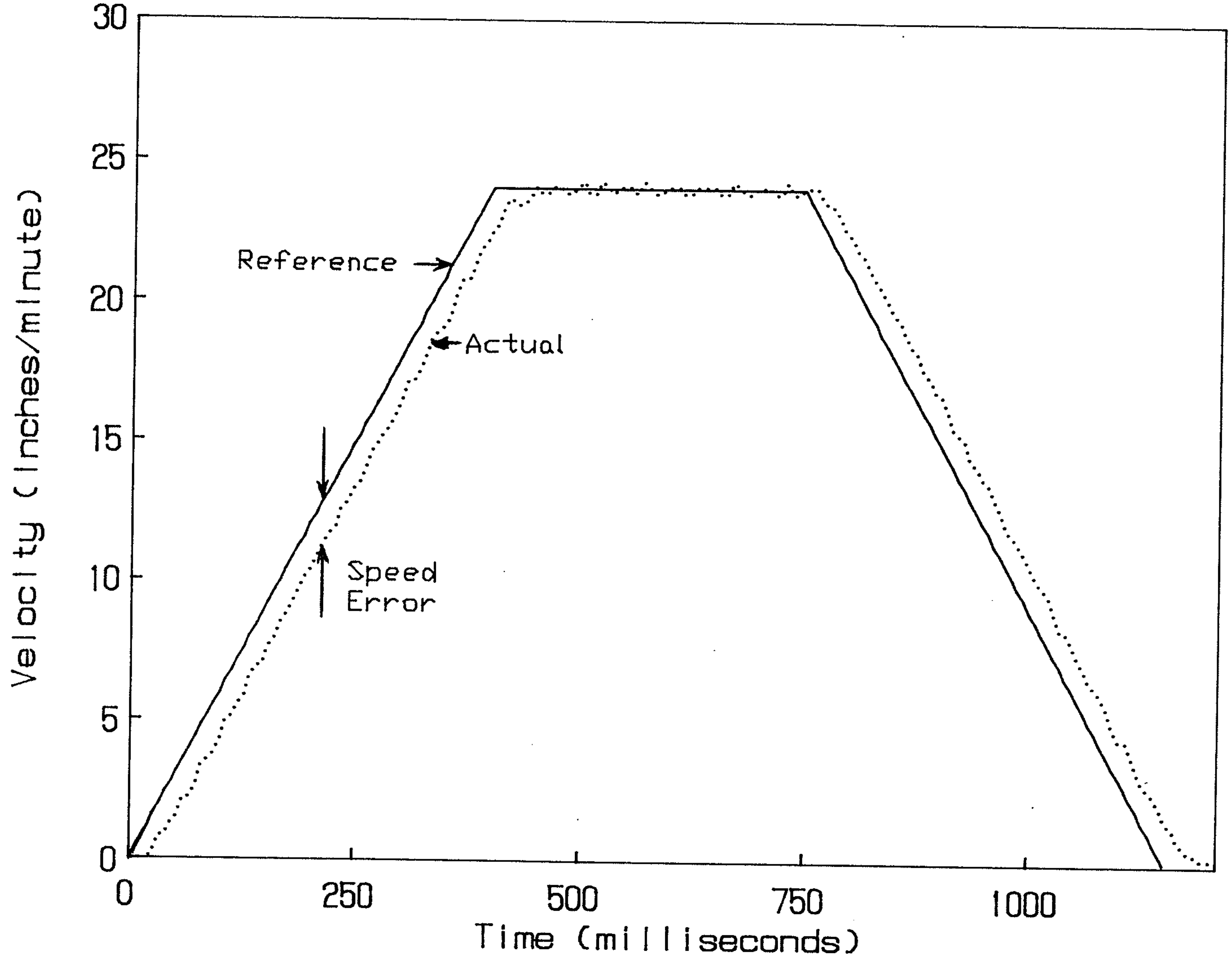


Figure 4.2 A Typical Speed Error

Table 4.1 Comparison of the Motor Drive's Theoretical and Actual Position Following Errors at Various Speeds

Speed (inches/min.)	Theoretical Error (inches*10-4)	Actual Error (inches*10-4)
6	27	27
12	54	54
18	81	81
24	108	108

4.2.2 Final Position Overshoot

It is critical in metal removal operations that the cutter does not overshoot the final position desired for the toolbit. An overshoot results in excess material removal. As a result, the position loop compensator was designed in Section 2.5.2 to achieve a critical damping of the position control loop. Table 4.2 presents the variation of overshoot on the final position for different speeds. This data verifies that the PD compensator successfully satisfies the requirement for zero overshoot. Indeed, the largest final position overshoot is 0.0001 inches at speeds above 18 inches/minute. This small overshoot is well below the desired machining accuracy level of +/- 0.0005 inches. It is important to realize that the overshoot of the final position should not accumulate with consecutive moves. The use of the absolute position counting scheme presented in Section 3.4.1 ensures that the final position overshoot is non-cumulative.

Table 4.2 Variation of Final Position Overshoot with Speed

Speed (inches/min.)	Final Position Overshoot (inches*10-4)
6	0
12	0
18	1
24	1

4.3 Contouring Accuracy

The contouring accuracy of a machine tool controller is a crucial performance measure. This section will detail the contouring accuracy of the machine tool controller for both linear and circular contours. Section 2.4 introduced the concepts of position following errors and contour errors. Contouring errors were described as position errors in which the toolbit's actual position was off the desired reference path. The position error tests described previously in this chapter focussed upon the errors produced by a single axis movement. However, the final accuracy and the overall system's performance are determined by the combined effect of all the axes of the CNC machine. Contour error tests, which will be described next, were used to determine the errors resulting from a slight mismatch between the time responses of the machine's axes.

The final contouring accuracy of a CNC machine is tested typically under loaded cutting conditions. However, the machine builder can either cut and evaluate a linear or circular contour [70,71] or follow a template with appropriate instrumentation to estimate the accuracy of the CNC controller. The template test has the advantage that the effects of toolbit vibration will be excluded from the measurements [39]. In this way, the accuracy of the machine tool controller alone may be evaluated. For the experimental system, a software based test was used to determine the contouring accuracy of the machine tool controller itself. The contour errors were calculated at each sampling instant and then stored in a software file for further analysis. The software based contour test is analagous to a template following test because the effects of toolbit vibrations and machine inaccuracies are excluded from the contour error measurements. The detrimental effects of the dynamic structural characteristics of the milling machine on accuracy will be discussed later in this chapter.

Figure 4.3 compares the contour errors produced by the machine tool controller for cutting and non-cutting tests. A two flute end mill with a 0.25 inch diameter was used to mill a 0.04 inch deep slot in a mild steel block. The toolbit was programmed to move at an angle of 45 degrees in the XY plane with an axis speed of 12 inches/minute. A test performed at the same speed, but without metal-cutting, is also shown in Figure 4.3. The close agreement between these curves indicates that metal-cutting seems to have very little influence on the contour error for this test. This result is expected as long as the CNC machine operates within the torque limits of the motors. Consequently, the remaining contour error tests were performed without metal-cutting.

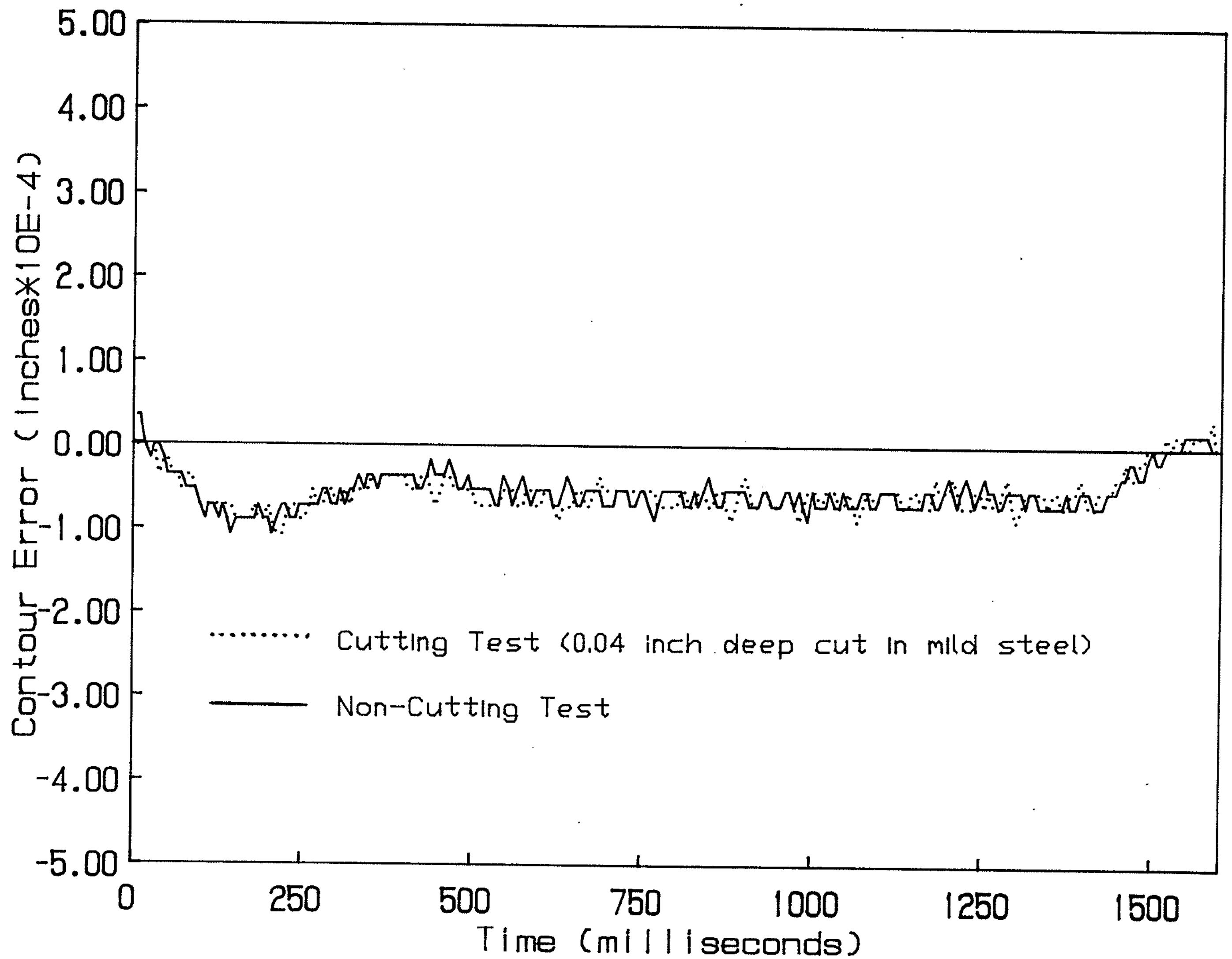


Figure 4.3 Comparison of Contour Errors for Cutting and Non-Cutting Tests at a Speed of 12 inches/minute

4.3.1 Accuracy of Linear Contouring

The microcomputer based, machine tool controller can perform linear interpolation simultaneously in three-dimensions. Figure 4.4 shows the absolute value of the contour errors for a three-dimensional linear motion at the maximum 24 inch/minute speed. The three-dimensional motion is at 45 degree angles to each of the XY, XZ, and YZ planes. This motion should represent the worst case for a contour error because the maximum contour error generally occurs at a 45 degree angle in a plane [39]. The maximum contour error is about $2.00 * 10^{-4}$ inches which is less than half the target accuracy of $5.0 * 10^{-4}$ inches for the controller.

Most machining is typically planar in practise. A typical example is the machining of a contour in the XY plane at a constant depth to provide a flat machined surface, say, on a cast part. Consequently, detailed contour error tests were performed in planes as opposed to a truly three-dimensional space. The toolbit was programmed to move at angles of 0, 30, 45, 60, and 90 degrees in each of the XY, XZ, and YZ planes. The tests were performed over a representative speed range of the milling machine from 6 inches/minute to the maximum speed of 24 inches/minute. Tables 4.3, 4.4, and 4.5 show the range of contour errors for the various speed-angle combinations in the XY, XZ, and YZ planes, respectively. The range of contour errors designates the total fluctuation in the contour error for a given axis speed and angle of movement. An examination of the data reveals that all the contour errors were less than half the target accuracy of +/- $5.0 * 10^{-4}$ inches. As expected, the contour errors are largest for movements with components along both

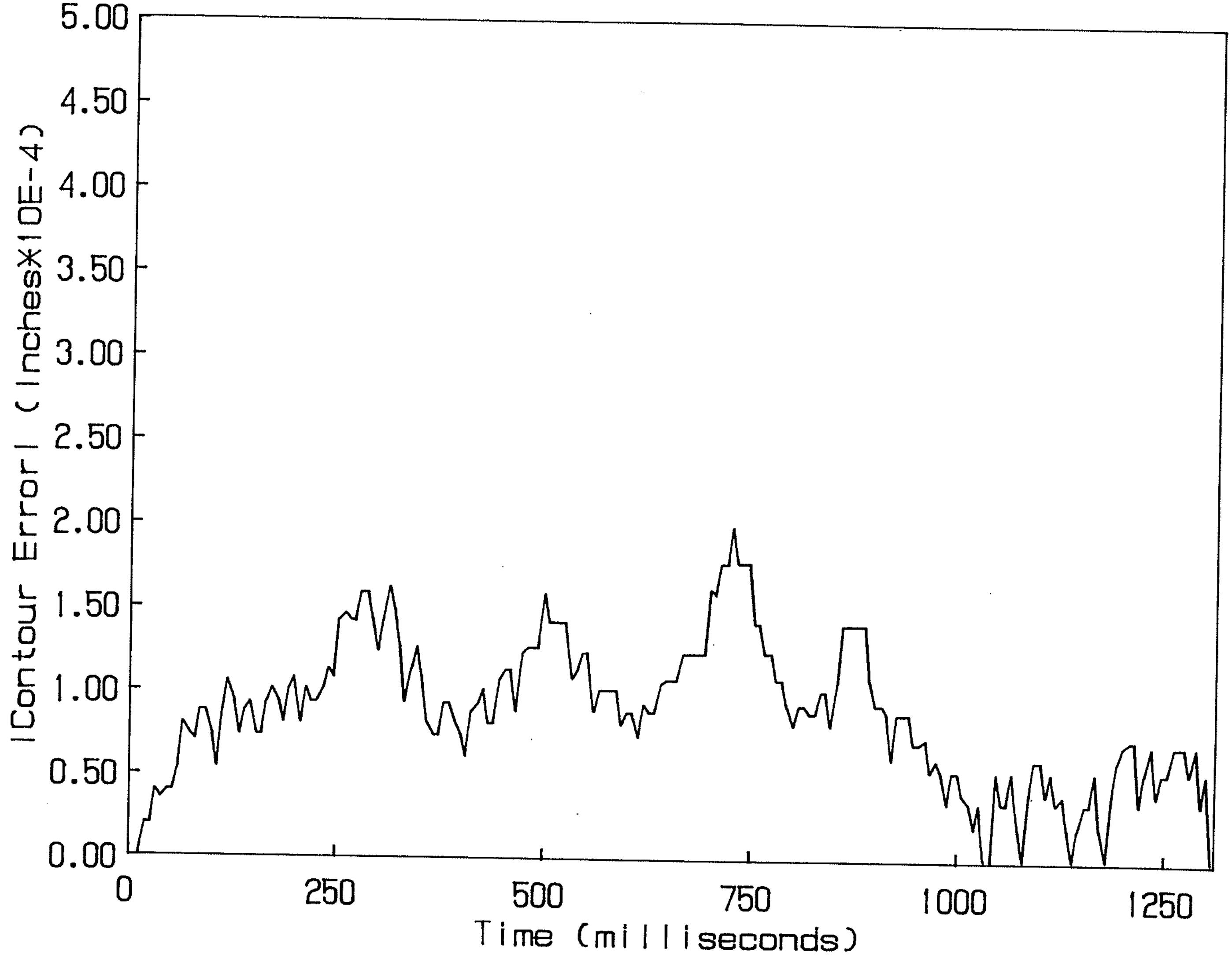


Figure 4.4 Contour Errors for a Sample Three-Dimensional Movement at a Speed of 24 inches/minute

axes, particularly at the highest speed. The contour errors are caused primarily by the slight mismatch between the gains of the servo motor drives.

Table 4.3 Range of Contour Errors for Linear Interpolation in the XY Plane (inches * 10^{-4})

Speed (inches/min.)	Angle (degrees) 0	30	45	60	90
6.0	0 to 1	-1 to 1	-1 to 1	-1 to 0	-1 to 0
12.0	0 to 1	-1 to 1	-1 to 1	-1 to 0	-1 to 0
18.0	0 to 1	-1 to 1	-1 to 1	-1 to 0	-1 to 0
24.0	0 to 1	-1 to 1	-1 to 1	-1 to 0	-1 to 0

Table 4.4 Range of Contour Errors for Linear Interpolation in the XZ Plane (inches * 10^{-4})

Speed (inches/min.)	Angle (degrees) 0	30	45	60	90
6.0	0 to 1	-1 to 1	-1 to 1	0 to 1	-1 to 0
12.0	0 to 1	-1 to 1	-1 to 1	-1 to 1	-1 to 0
18.0	0 to 1	-1 to 2	-1 to 1	-1 to 1	-1 to 0
24.0	0 to 1	-1 to 2	-1 to 2	-1 to 1	-1 to 0

Table 4.5 Range of Contour Errors for Linear Interpolation in the YZ Plane (inches * 10^{-4})

Speed (inches/min.)	Angle (degrees) 0	30	45	60	90
6.0	0 to 1	0 to 1	-1 to 1	-1 to 1	-1 to 0
12.0	0 to 1	-1 to 2	-1 to 2	-1 to 1	-1 to 0
18.0	0 to 1	-1 to 2	-1 to 2	-1 to 2	-1 to 0
24.0	0 to 1	-1 to 2	-1 to 2	-1 to 2	-1 to 0

4.3.2 Accuracy of Circular Contouring

Circular contour tests were performed in a similar manner to that of the linear contour tests. Figure 4.5 shows the contour errors for a circular motion in the XY plane at the maximum 24 inches/minute speed. The contour errors fluctuated between positive and negative values for the circular tests as demonstrated in Figure 4.5. As a result, the ranges of contour errors rather than the errors themselves were tabulated for a combination of axis speeds and circular radii. The range of contour errors designates the total fluctuation in the radial contour error for a given axis speed and radius. Tables 4.6, 4.7, and 4.8 show the ranges of the contour errors in the XY, XZ, and YZ planes, respectively. The circular contour tests were performed for axis speeds of 6, 12, 18, and 24 inches/minute at radii of 0.25, 0.50, 1.00, and 2.00 inches. An examination of the contour errors reveals that they are always less than the desired accuracy of +/- 5.0 * 10^{-4} inches.

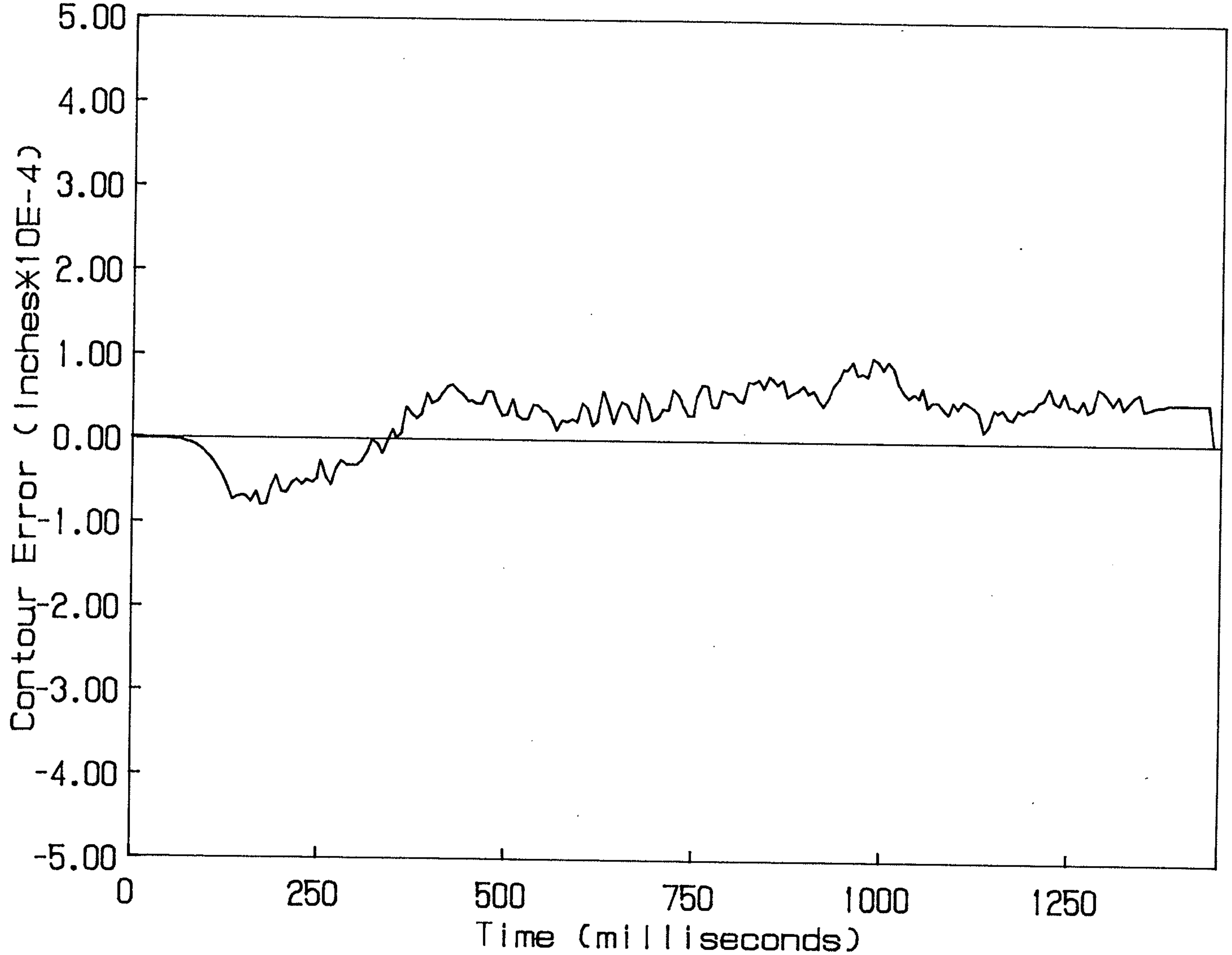


Figure 4.5 Contour Errors for a Circular Movement in the XY Plane at a Speed of 24 inches/minute

Table 4.6 Range of Contour Errors for Circular Interpolation in the XY Plane (inches * 10^{-4})

Speed (inches/min.)	Radius (inches) 0.25	0.50	1.00	2.00
6.0	-1 to 1	-1 to 1	-1 to 1	-1 to 1
12.0	-1 to 1	-1 to 1	-1 to 1	-1 to 1
18.0	-1 to 1	-1 to 1	-1 to 1	-1 to 1
24.0	-1 to 1	-1 to 1	-1 to 1	-1 to 1

Table 4.7 Range of Contour Errors for Circular Interpolation in the XZ Plane (inches * 10^{-4})

Speed (inches/min.)	Radius (inches) 0.25	0.50	1.00	2.00
6.0	-1 to 1	-1 to 1	-1 to 1	-1 to 1
12.0	-1 to 1	-1 to 1	-1 to 1	-1 to 1
18.0	-1 to 2	-1 to 2	-1 to 1	-1 to 1
24.0	-1 to 2	-1 to 2	-1 to 1	-1 to 1

Table 4.8 Range of Contour Errors for Circular Interpolation in the YZ Plane (inches * 10^{-4})

Speed (inches/min.)	Radius (inches) 0.25	0.50	1.00	2.00
6.0	-2 to 1	-2 to 1	-2 to 1	-2 to 1
12.0	-2 to 2	-2 to 2	-2 to 2	-2 to 2
18.0	-2 to 2	-2 to 2	-2 to 2	-2 to 2
24.0	-2 to 2	-2 to 2	-2 to 2	-2 to 2

4.4 Effect of a Milling Machine's Structural Characteristics on its Performance

The effects of a milling machine's structural characteristics were not included in the contour tests. These characteristics were excluded in order to assess the accuracy of the machine tool controller itself. The effects of a machine tool's structural characteristics must be considered, however, in the total machine tool design [72]. Structural characteristics such as vibrations and leadscrew pitch error degrade machining accuracy. They must be evaluated in a machine tool application and, if necessary, corrected to ensure the desired machining accuracy.

Excessive vibrations of a machine tool produce poor surface finish, increased machine and tool wear, and the loss of control over critical tolerances. Vibrations in milling operations essentially result from three basic sources; the spindle motor, the drive systems for the axes, and from actual metal cutting. A vibration analysis of the experimental milling machine was undertaken in [73]. The vibration measurements were performed without metal-cutting in order to isolate the vibrational characteristics of the structure itself. Furthermore, the dynamic characteristics of a machine tool depend primarily on its static stiffness [72]. This analysis revealed that the main source of vibrations was the electric motor driving the spindle and its associated pulley and belt system.

The relative displacement between the cutting and the workpiece determines the quality of the machined surface finish. Reference [74] suggests that, for high precision machine tools, a general guideline for peak-to-peak vibration displacement is 0.1 mils. Using the techniques outlined in [73], the maximum relative displacement between the cutting tool and workpiece was found to be 0.41 mils. This value was reduced to 0.23 mils by balancing the pulley system of the spindle motor and by isolating the spindle motor from its mounting plate [73]. The maximum vibration specification of 0.1 mils, however, was still exceeded. The basic problem with the experimental milling machine is that it was not designed for high precision machining. The poor structural stiffness of this particular, inexpensive milling machine indicates that the minimization of vibrations was not a principle concern in its design. In summary, it is important to ensure that the vibrations of a machine tool being considered for CNC retrofitting do not conflict with the desired machining accuracy.

A second structural characteristic that can degrade machining accuracy is the pitch error in the leadscrews. If there is a variation in the leadscrew pitch, and rotary encoders are used for indirect position feedback, then the actual positions of the machine's axes will be in error. The direct position feedback obtained from the linear encoder on the X axis of the experimental milling machine was used to assess the pitch error of the corresponding leadscrew. The results of these tests are detailed in [75]. The variation in leadscrew pitch was found to be as high as +/- 7%. This means that the leadscrew's pitch on the X axis varied between 0.093 and 0.107 with a mean of 0.100 inches/rotation. This condition is unacceptable for high precision machining. As a result, the leadscrews of the milling machine should be replaced with high precision ballscrews if the experimental milling machine is required for high precision machining.

4.5 Summary of the Overall Controller Performance

The performance of the machine tool controller has been presented in this chapter. The performance of the control loops unit was found to meet the position following error and the final position overshoot requirements. Metal-cutting seemed to have very little influence on the contour errors for the experimental system as long as the motors operated within their torque limits. The contouring accuracy of the machine tool controller was determined during multi-axis tests without metal-cutting. It was found to be within the target accuracy of +/- 0.0005 inches for both linear and circular contouring. Thus, the machine tool controller achieved the desired accuracy. The effects of the structural characteristics of the milling machine were then considered. Vibrations and pitch errors in the leadscrews degraded the accuracy achievable during actual machining. These sources of error would have to be considered to guarantee a desired machining accuracy.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

A flexible, microcomputer based, three-axis machine tool controller was developed. The controller consisted of a microcomputer, a software control program, and a custom motion control card to interface the axes' motors with the microcomputer. The sampled-data controller was implemented by using FORTRAN and ASSEMBLER. FORTRAN provided the mathematical capabilities for programming the controller and the interpolation routines. On the other hand, ASSEMBLER performed the low level data input and output between the computer and the custom interface card. The software control program provided much greater flexibility than a hardwired controller. This flexibility permitted the development of routines for multi-axis linear and circular interpolation. Furthermore, a position loop compensator and a backlash compensation routine were implemented in software. This software implementation permitted a free selection of the compensator's structure and its gain coefficients. The machine tool controller was interfaced to a commercial CAD system called SmartCAM™ to achieve the full benefits of a CNC system.

A manual three-axis milling machine was used to evaluate the performance of the flexible machine tool controller. A software based, proportional-integral-derivative (PID) compensator was implemented to meet typical performance requirements in milling operations. The target positional and contouring accuracies for the controller were +/- 0.0005 inches and they were achieved. The position compensator was successful in critically damping the position control loop to eliminate overshoot on the final position. Thus, the developed controller satisfied the desired following error, contour error, and final position overshoot requirements. The major source of contour errors was found to be the slightly different response characteristics of the various motor drives. This difference was caused mostly by slight mismatches in the gains of the different motor drives.

The effects of the structural characteristics of the experimental milling machine on accuracy must be considered in a total machine tool design. Structural characteristics such as vibrations and leadscrew pitch error degrade machining accuracy. For instance, inexpensive rotary encoders were used in the experimental system for indirect position feedback from the axes. However, a more expensive linear encoder mounted on one axis of the milling machine's table provided more accurate direct position feedback. The linear encoder allowed a determination of the variation in pitch error of the leadscrew, which was found to be as high as +/- 7%. In summary, the position errors caused by the machine's vibrations and pitch errors in the leadscrew must be evaluated in a machine tool application and, if necessary, they should be compensated to achieve a desired machining accuracy.

5.2 Recommendations

The flexibility of the new controller makes it attractive for the machine retrofit and custom machine tool market. The hardware cost of a microcomputer and the custom interface card for motion control is about $3800, which makes the controller seem economically feasible. The control programs are written in standard programming languages. Thus, they are largely transportable to more powerful microcomputers such as those with an Intel 80386 microprocessor. Certain enhancements would be necessary, however, if it is desired to market the new controller. For example, one or several microprocessors could be included on the motion control interface card itself to accommodate more complex interpolation routines. This enhancement would also allow operation of the controller at higher sampling frequencies and would extend the potential applications to areas such as robotics. In addition, the microcomputer could potentially supervise several machines in a computer controlled manufacturing system. Another area of improvement would be the development of a user friendly interface. A menu driven system could be developed to allow a user to customize the software to their particular machine application. It is also desirable to convert the program source code to the C programming language for increased transportability between different computers.

Considering the importance of machining accuracy, it would be useful to develop an automated system to match the gains of the various motor drives in order to minimize contour errors. The computer would apply a known voltage to each servo motor drive and then monitor the actual speed of each axis. The software gains of each motor could be adjusted until the speed responses of the motors are identical.

Another possible area of development would involve the addition of an adaptive or optimal machine tool control strategy to maximize the machining rate. This adaptive control strategy could be used to generate a knowledge base of recommended machining speeds for various materials.

REFERENCES

[1] Asfahl, C. R., Robots and Manufacturing Automation, John Wiley and Sons, Inc., Toronto, 1985.

[2] Gayman, D. J., "Calibration, Compensation, and CNC", Manufacturing Engineering, pp. 34-39, January, 1987.

[3] Compumotor PC-23 Indexer Operator's Manual, Bulletin OM-8300-PC-23, Parker Hannifin Corporation, Petaluma, California, 1987.

[4] Bailey, S., "Servomotor and Stepper: Key Elements of Motion Control Design", Control Engineering, pp. 55-59, February, 1986.

[5] Bailey, S., "Servo Vs. Stepper - Motion Control Design Decision With Dynamic Overtones", Control Engineering, pp. 68-72, May, 1987.

[6] Electro-Craft Speed and Position Control Systems Distributor Catalog, Robbins Myers Motor and Control Systems Division, Hopkins, Minnesota, January, 1988.

[7] Koren, Y., Computer Control of Manufacturing Systems, McGraw-Hill, Inc., New York, 1983.

[8] Pak, H., and L. Daneshmend, "Microprocessor Control of Multiaxes Continuous Path Motion", Microprocessors and Microcomputers, pp. 519-527, Vol. 6, No. 10, December, 1982.

[9] Seim, T. A., "Applying Microprocessors to Machine Tool Controller Design, Part 1", Computer Design, pp. 86-97, March, 1980.

[10] VanLandingham, H., Introduction to Digital Control Systems, Macmillan Publishing Company, New York, 1985.

[11] Gauen, K., "Designing a DC Servo Position Control Using a Microcomputer", Control Engineering, pp. 80-83, July, 1983.

[12] Abdel-Azim, S., and J. A. Dreiling, "Digital Control of a Brushless DC Motor", IEEE Control Systems Magazine, Vol. 7, No. 5, pp. 31-35, October, 1987.

[13] Meshkat, S., "Feasible Control Algorithms for DC Motors with Wide Bandwidths", Motor-Con Proceedings, pp. 314-321, April, 1983.

[14] Perdikaris, G., and K. Van Patten, "Microcomputer Control Algorithms for DC Motor Drive Systems", Motor-Con Proceedings, pp. 261-269, April, 1983.

[15] Srinivasan, K., and C. Dogan, "Digital Control Algorithms for Electrohydraulic Machine Tool Feed Drive Servomechanisms", American Control Conference, pp. 449-457, 1984.

[16] Gadsby, C. G., A Microcomputer Based Machine Controller for Contouring Applications, M. Sc. Thesis, University of Manitoba, Winnipeg, Manitoba, December, 1986.

[17] SA-9000 Series PWM Amplifier Instruction Manual, Electro-Craft Corporation, Hopkins, Minnesota, November, 1985.

[18] Gennuso, M., and R. Gettys Hill, "Servo Trade-Offs - How to Determine Which Servo Drive System is Best Suited to Your Application", Motor-Con Proceedings, pp. 141-152, March, 1982.

[19] Weck, W., Handbook of Machine Tools, Volume 3 - Automation and Controls, John Wiley and Sons, Inc., Toronto, 1984.

[20] Tal, J., "Fundamentals of Motion Control By Microprocessors", Motor-Con Proceedings, pp. 98-106, April, 1983.

[21] Tal, J., Motion Control By Microprocessors, Galil Motion Control, Mountain View, California, 1984.

[22] Pressman, R. S., and J. E. Williams, Numerical Control and Computer-Aided Manufacturing, John Wiley and Sons, Inc., Toronto, 1977.

[23] Kaufman, H., and C. J. Uliana, "Model Reference and Optimal Control Applied to Computerized Numerical Control Systems", Conference on Applied Motion Control, pp. 31-41, 1986.

[24] Koren, Y., "Design of Computer Control for Manufacturing Systems", Transactions of the ASME, Journal of Engineering for Industry, Vol. 101, pp. 326-332, August, 1979.

[25] Schmitt, L. E., "PC Versus CNC - Which Do You Choose?", IEEE Transactions on Industry Applications, Vol IA-20, No. 5, pp. 1141-1145, September/October, 1984.

[26] Tal, J., "Motion Control Stability Compensation by Microprocessors", Incremental Motion Control Systems and Devices Conference, pp. 1-9, 1982.

[27] Linear Scale Digital Readout System Price List, Bulletin No. 9100, Mitutoyo Canada Ltd., Toronto, September, 1987.

[28] REX Series Encoder Price List, Sandtron Automation Limited, Oakville, Ontario, 1987.

[29] DC Motors, Speed Controls, Servo Systems, 5th Edition, Electro-Craft Corporation, Hopkins, Minnesota, 1980.

[30] Tamaki, K., Ohishi, K., Ohnishi, K., and K. Miyachi, "Microprocessor-Based Robust Control of a DC Servo Motor", IEEE Control Systems Magazine, pp. 30-36, October, 1986.

[31] Meshkat, S., "Adaptive Control Performance Improvement Using Digital Signal Processors", Motor-Con Proceedings, pp. 1-21, April, 1987.

[32] Stojic, M., "Design of the Microprocessor-Based Digital System for DC Motor Speed Control", IEEE Transactions on Industrial Electronics, Vol. IE-31, No. 3, August, 1984.

[33] Kuo, B. C., Automatic Control Systems, 5th Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[34] Dorf, R. C., Modern Control Systems, 4th Edition, Addison-Wesley Publishing Company, Don Mills, Ontario, 1986.

[35] Shinners, S. M., Modern Control Theory and Application, 2nd Edition, Addison-Wesley Publishing Company, Don Mills, Ontario, 1978.

[36] Polad, M., "Design of a High-Performance CNC Control Using Standard Microprocessor Hardware, Motor-Con Proceedings, pp. 425-430, April, 1983.

[37] Franklin, G., and J. Powell, Digital Control of Dynamic Systems, Addison-Wesley Publishing Company, Inc., Don Mills, Ontario, 1980.

[38] Masory, O., "Improving Contouring Accuracy of NC/CNC Systems with Additional Velocity Feed Forward Loop", Transactions of the ASME, Journal of Engineering for Industry, Vol. 108, pp. 227-230, August, 1986.

[39] Poo, A., Bollinger, J. G., and G. W. Younkin, "Dynamic Errors in Type 1 Contouring Systems", IEEE Transactions on Industry Applications", Vol. IA-8, No. 4, pp. 477-484, July/August, 1972.

[40] Kraft, S. D., and E. T. Wall, "Experimental Microprocessor-Based Adaptive Control System", IEEE Control Systems Magazine, pp. 35-39, February, 1986.

[41] Ohishi, K., Nakao, M., Kouhei, O., and K. Miyachi, "Microprocessor-Controlled DC Motor for Load-Insensitive Position Servo System", IEEE Transactions on Industrial Electronics, Vol. IE-34, No. 1, pp. 44-49, February, 1987.

[42] Ulsoy, G. A., Koren, Y., and F. Rasmussen, "Principal Developments in the Adaptive Control of Machine Tools", Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 105, pp. 107-112, June, 1983.

[43] Tal, J., and W. Baron, "Motion Control for Automation and Robotics", Robotics Age, pp. 8-11, December, 1984.

[44] Tal, J., "Performance Improvement in Bandlimited Position Control Systems", Incremental Motion Control Systems and Devices Conference, pp. 177-188, 1981.

[45] Baron, W., "Position Control of a Brushless Motor By Microprocessor", Motor-Con Proceedings, pp. 123-128, April, 1983.

[46] Phillips, C., and H. Nagle, Jr., Digital Control System Analysis and Design, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1984.

[47] KS-Series Operator's Manual, Parker-Compumotor Corporation, May, 1987.

[48] Stanley, B., and R. Barry, "Microprocessor Based Servomotor Control System", Incremental Motion Control Systems and Devices Conference, pp. 137-144, 1985.

[49] Wade, L., "Motion Control System Design Using Modular Components", Motor-Con Proceedings, pp. 333-350, October, 1986.

[50] Pemberton, T. J., "PID: The Logical Control Algorithm", Control Engineering, pp. 61-62, May, 1972.

[51] Pemberton, T. J., "PID: The Logical Control Algorithm - Part II", Control Engineering, pp. 61-63, July, 1972.

[52] Squires, D., Kmetz, J., Santos, J., and T. Isbell, "A Dedicated I.C. for D.C. Closed Loop Position Control", Motor-Con Proceedings, pp. 283-289, October, 1986.

[53] Juds, S., "Principles and Techniques of Digital Velocity and Position Servocontrol Using Microprocessor Technology", Motor-Con Proceedings, pp. 514-525, March, 1982.

[54] Krikelis, N., and S. Fassois, "Microprocessor Implementation of PID Controllers and Lead-Lag Compensators", IEEE Transactions on Industrial Electronics, Vol. IE-31, No. 1, pp. 79-85, February, 1984.

[55] Squires, D., Kmetz, J., and B. Cox, "A High Level DC Servo Controller Implemented as a New Generation of ASIC", Motor-Con Proceedings, pp. 175-182, April, 1987.

[56] Smith, C. L., Corripio, A. B., and J. Martin Jr., "Controller Tuning from Simple Process Models", Instrumentation Technology, pp. 39-44, December, 1975.

[57] Powell, B., "Direct Drive D.C. Servo Motors for Gearless Motion Control", Incremental Motion Control Systems and Devices Conference, pp. 109-120, 1985.

[58] Tomasek, J., Dictionary of Motion Control Calculations, Incremental Motion Control Systems and Devices Conference, pp. 1-12, 1981.

[59] Johnson, R., "Designing with Optical Shaft Encoders", Robotics Age, pp. 27-34, May/June, 1983.

[60] Wade, L., and L. Dadok, "Implementation of Quadrature Decoding Using a Universal Motion Control IC for Digital Closed Loop Systems with Encoder Feedback", Motor-Con Proceedings, pp. 290-296, April, 1984.

[61] Kostyniuk, T., Technical Reference Manual for the Microcomputer Based, Three-Axis Machine Tool Controller, Department of Mechanical Engineering, University of Manitoba, Winnipeg, Manitoba, 1988.

[62] Optoelectronics Designer's Catalog, Hewlett-Packard, Palo Alto, California, 1986.

[63] Linear and Conversion Products Data Book, Precision Monolithics, Inc., Santa Clara, California, 1984.

[64] IBM Technical Reference Manual, International Business Machines Corporation, Boca Raton, Florida, April, 1984.

[65] "Interchangeable Variable Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines", Standard EIA RS-274-D, Electronics Industries Association, Washington, D.C., March, 1979.

[66] Masory, O., and Y. Koren, "Reference-Word Circular Interpolators for CNC Systems", Transactions of the ASME, Journal of Engineering for Industry, Vol. 104, pp. 400-405, November, 1982.

[67] Hannan, H., "State-of-the-Art in Coordinated Axis Motion", Motor-Con Proceedings, pp. 416-424, April, 1983.

[68] SmartCAM CNC Programming System, Point Control Co., Eugene, Oregon, 1987.

[69] AUTO-CAM CAD Module Reference Manual, Publication No. D-XXDNC041R, ICAM Technologies Corporation, Pointe Claire, Quebec, 1988.

[70] National Aerospace Standard 979.

[71] Burdekin, M. S., "Cutting Tests for Accuracy Assessment", Technology of Machine Tools, Volume 5 : Machine Tool Accuracy, Lawrence Livermore National Laboratory, University of California, pp. 9.10-1 - 9.10-8, 1980.

[72] Weck, W., Handbook of Machine Tools, Volume 2 - Construction and Mathematical Analysis, John Wiley and Sons, Toronto, 1984.

[73] Kostyniuk, T., Wu, J., Balakrishnan, S., and N. Popplewell, "Vibration Analysis of a Milling Machine Retrofitted for CNC Operations", Proceedings of the 11th Canadian Conference of Applied Mechanics, pp. A-142 - A-143, May, 1987.

[74] IRD Mechanalysis Audio-Visual Customer Training Instruction Manual, IRD Mechanalysis Inc., Columbus, Ohio, 1975.

[75] Toutant, R., Report on the Feasibility of Using a Linear Encoder for the Digital Motion Control of a Milling Machine, Department of Mechanical Engineering, University of Manitoba, Winnipeg, Manitoba, 1988.