HIGH SPEED SERVO CONTROL OF MULTI-AXIS MACHINE TOOLS

By

DOUGLAS D.A. RENTON, M.Eng. (Electrical and Computer Engineering)
McMaster University
Hamilton,Ontario, Canada

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Doctor of Philosophy

McMaster University

High Speed Servo Control of Multi-Axis Machine Tools

DOCTOR OF PHILOSOPHY (2000) McMaster University
(Mechanical Engineering) Hamilton, Ontario

TITLE: High Speed Servo Control of Multi-Axis Machine Tools

AUTHOR: Douglas D.A. Renton
M.Eng. (Electrical and Computer Engineering)
McMaster University
Hamilton,Ontario, Canada

SUPERVISOR: Dr. M.A.Elbestawi
Department of Mechanical Engineering
McMaster University

NUMBER OF PAGES: xv, 169

Abstract

High Speed Servo Control of Multi-Axis Machine Tools, is concerned with maintaining tight position control at high feed rates, in the presence of disturbances generated by the cutting process. This is an important goal due to general advances in machining technologies, as well as economic pressures in specific industries.

In this thesis a feed rate planning method, and servo loop control law are developed which explicitly account for the dominant machine tool servo characteristics, including actuator saturation. The feed rate planning method developed, 'Minimum-Time Path Optimization' (MTPO), enables coordinated motions to be planned with near maximum utilization of servo capabilities (a reserve must be left to handle modeling error, and disturbances). The result is a significant reduction in servoing time with out sacrificing path accuracy when compared to fixed acceleration and velocity limit path planning, or significantly reduced path error, when compared to fixed feed rate planning (assuming equal total time time). The servo loop level controller developed, Minimum-Time Tracking Control (MTTC), focuses on the primary sources of path error in high-speed servoing applications, specifically, limits on amplifiers, system inertia, and damping. This controller is relatively simple to compute, making it suitable for high speed servo control. MTTC's advantages are most clearly seen in

its consistently superior disturbance-rejection properties. MTTC is shown to regulate steps of various sizes better (faster/less over shoot) than Generalized Predictive Control (GPC), H^{∞}, or Proportional Derivative (PD) controllers. It is also shown to respond more linearly in magnitude, and phase (unity gain, zero lag) to sin waves at various frequencies.

A periodic observer is developed and implemented which significantly improves the rejection of cutting disturbances. Given that the major source of disturbances is the cutting process, an encoder on the spindle is used to synchronize compensation, with the disturbance. The instantaneous disturbance is estimated as the difference between the predicted acceleration, and the achieved acceleration. This estimate is then averaged as a function of spindle position, over successive spindle revolutions.

A linear servo motor is constructed to investigate commutation strategies which allows each winding of a permanent magnet linear motor to be driven separately. This enables commutation strategies to be implemented which reduce the heat generated in the coils of the linear motor, increasing the continuous holding force of the motor by approximately 18%.

A PC based Open Architecture Control (OAC) with soft motion control is developed which provides low level openness (All calculations including closing servo loops are performed on the PC, no special purpose motion control card is used.). This system enables the developed controllers, observers, and commutation strategies to be implemented in a cost efficient manner. It also eliminates many of the machine controller limitations which make path planning and computing difficult. Specifically, it enables the entire part path to be planned and computed in advance, stored, and executed, or played back in real-time. This eliminates the requirements of bounded execution times, and the difficulty of representing data in an intermediate form.

Acknowledgments

I would like to thank my research supervisor, Dr. Elbestawi for his support and guidance throughout the course of this research. I would also like to thank my friends at McMaster (especially Trevor, Stephen, and Rich) for their assistance, advice, and most importantly for making the past five years so enjoyable. Finally I would like to thank my parents, my sister Pam, and of course my wife Uta for their faith, and support.

Contents

List of Figures

Chapter 1

Introduction

High Speed Servo Control of Multi-Axis Machine Tools, is concerned with maintaining tight position control at high feed rates, in the presence of disturbances generated during cutting. This is important due to general advances in machining technologies, as well as economic pressures in several industries such as automotive, aerospace, and die and mold. General advances in high-speed machining technology, include higher spindle speeds and improved tool materials. These improvements are placing pressure on the servo control performance of machine tools. To alleviate this, new machine tools and improved controls must be developed. Improvements to machine tools under development include special-purpose machine tools, the use of advanced materials, the replacement of ball screws and ways with linear motors and roller guides, and the use of parallel link actuators. These advances are expected to result in higher feed rates, thus placing increased demands on the control system.

In the die and mould industry competitive pressures require reducing production cycle times, and achieving tighter tolerances, while maintaining or reducing costs. The solution is to reduce or eliminate hand finishing (which represents 25 to 38 percent of manufacturing time for a die), and rework. This is achieved through the use of "high speed machining", and tighter finish machining tolerances. These tolerances include cusp heights, and deviation from the target surface. This requires smaller step-over distances which results in significantly longer machining paths, with low cutter forces. To maintain machining times, faster feed rates are required [44] [36].

In chapter 2, the state of research in areas important in high speed servo control of machine tools is examined. Areas include 'Path and Feed-rate Planning', 'Servoing Algorithms', 'Open Architecture Machine Tool Controllers', 'High-Speed Machine Tool Design', 'Linear Motors in Machine Tools' and the actual process of 'High-Speed Machining'.

In chapter 3, a comprehensive time domain simulation is first examined. A time domain simulation is used, as opposed to a closed form transfer function due to the complexity of the system, and the desire to ensure that over simplifications are not made. Based on simulation results, initial conclusions on the important system parameters are made. A physical test plant is then fabricated, including a control system, on which these initial conclusions can be verified. This system is simply a rotary motor, and mass. It is verified that this system exhibits the dominant system properties, as indicated by the simulations. Initial control concepts are generated on this test stand. The control system is also developed to the point that other plants can safely be examined. As linear motor driven machine tools represent the state of the art, a linear motor test stand is then assembled. The examination of this system begins with developing accurate parametric methods to represent the characteristics which have been found to be dominant. The methods found include a simple small signal transfer function, and a representation of the large signal (saturation) limits, or axis performance envelope. The disturbance rejection properties of the plant alone are then examined, in an effort to determine what action will be required by a controller. The importance of feedback, as well as options are examined. Finally to test such things as increasing continuous holding force through alternate commutation methods, a linear motor is manufactured.

In chapter 4, a feed rate planning method is developed which takes advantage of the full performance envelope of each axis in an arbitrary path. This near-complete usage of the servo capabilities of a machine tool results in reduced cycle time or reduced path error (depending on the feed rate planning method it is being compared to). The issue of constant feed rate is addressed by replacing it with constant feed per tooth. This is achieved by controlling the spindle speed.

In chapter 5, servo loop control algorithms are examined. An algorithm is then developed that uses the small signal transfer function, the axis performance envelope as well as instantaneous position, velocity, and acceleration information of the target path and machine axis to improve servo performance in the presence of disturbances. Broad-band stiffness is then further improved with the addition of disturbance observers, including a standard DC observer, and a periodic disturbance observer. The standard observer is used in combination with a command sequence for automatic plant identification. Since the largest source of disturbances is the cutting process, an encoder on the spindle is used to synchronize compensation, with the disturbance. The instantaneous disturbance is estimated as the difference between the predicted acceleration, and the achieved acceleration. This estimate is then averaged as a function of spindle position, over successive spindle revolutions. This procedure provides the low frequency stiffness that the mass of the system cannot.

In chapter 6, the control algorithm developed is compared to industry standard controllers including Generalized Predictive Control (GPC), H^{∞}, and PD$\star$ (the PD$\star$ controller used the digital tachometer for velocity estimation, not the derivative of the position error). The comparison is performed on the linear motor test stand, using the PC based OAC developed. The tests show the improved disturbance rejection properties of the developed controller, as well as its more linear transfer function in terms of magnitude and phase. The further improvements in disturbance rejection made possible using the periodic observer are then examined.

In chapter 7 the conclusions of this work are summarized. The conclusions include that superior performance can be achieved by using a path planning method, and control scheme which focus on the dominant small and large signal characteristics of high performance machine tools; broad-band stiffness can be improved through the use of a periodic observer; and driving each winding of a permanent magnet linear motor separately can increase the continuous holding force of a motor by approximately 18%.

Chapter 2

Literature Review

2.1 Introduction

This research is motivated by the increasing demands which are being placed on machine tools in terms of final part accuracy, surface finish and machining time. Specifically, machine tool controls must advance to support new machine designs and the trend toward the use of higher spindle speeds which better utilize new developments in cutting tools. The use of low friction roller bearing slides, which exhibit lower damping and fast acting linear motors which lack the mechanical advantage associated with the ball screw, challenge existing controller algorithms. Furthermore, increases in tooth passing frequency and rapidly changing cutter/workpiece intersection geometries, increase the level of disturbance under which the control system must maintain its commanded tool path.

2.2 Path and Feed-rate Planning

Linear and Circular path and feed rate planners are common on CNC equipment today. In addition to interpolating the curve, there is generally a provision for constant feed rate during the body of the curve, and blending of path segments (ends). Less common is the provision for parabolas, splines, or free form curves [17] [62]. To cut these complicated shapes, the common practice is to use numerous straight line segments. This results in poor surface properties due to the variation in feed rate, part error and longer machining times than necessary [62].

The research in this area emphasizes the ability to interpolate free form curves, in multiple dimensions, minimize part error, provide constant feed rate, using minimal computational effort, and intermediate storage [17] [62] [33]. "..With this new interpolator, a constant feed is maintained along the cut.." [62] "The amount of geometric information transfered from the CAD to the CNC must be minimized but must still enable the machining of any general curve." [62] "varying feed rate in CNC machining causes fluctuant metal flow and machine vibration and consequently poorer surface quality." [63]

Through the use of PC based OAC with soft motion control, the limitations of the machine controller that make these goals difficult no longer exist. Specifically, there is no reason why the entire part path cannot be planned and computed in advance, stored, and executed, or played back (without the need to compress the intermediate data) in real-time. This eliminates the requirements of bounded execution times, and the difficulty of representing the data in an intermediate form. For example, assuming a 5-axis machine, using 4 bytes of position data per axis per loop closure, and a 10 kHz update rate, the requirement would be 720MB per hour of machining time (assuming no compression). Even this, 'worst case' example is possible on a modern PC.

2.3 Servo Control Algorithms

The suitability of a control algorithm for a given task has two components: first, performance, and second, relative efficiency or cost. The task is the control of machine tool drives. As the computational power of computers continue to improve, the computational costs will diminish to near zero, and we will be left simply with the human time required to set the system up.

The factors involved that make machine tools difficult to control include:

- Amplifier / Motor Limitations
- Play / Backlash
- Flexibility of Lead Screw (position dependent)
- Several forms of friction
- Disturbances / Load Variations
- Manufacturing Imperfections such as 'tight spots' in ways or belts

There are three main control strategies used for machine tool control. They are Feed Forward Control, Cross Coupling Control, and Optimal Control. Optimal Control is further broken down into Predictive Control (Generalized Predictive Control GPC), Adaptive Control, and Learning Control [30].

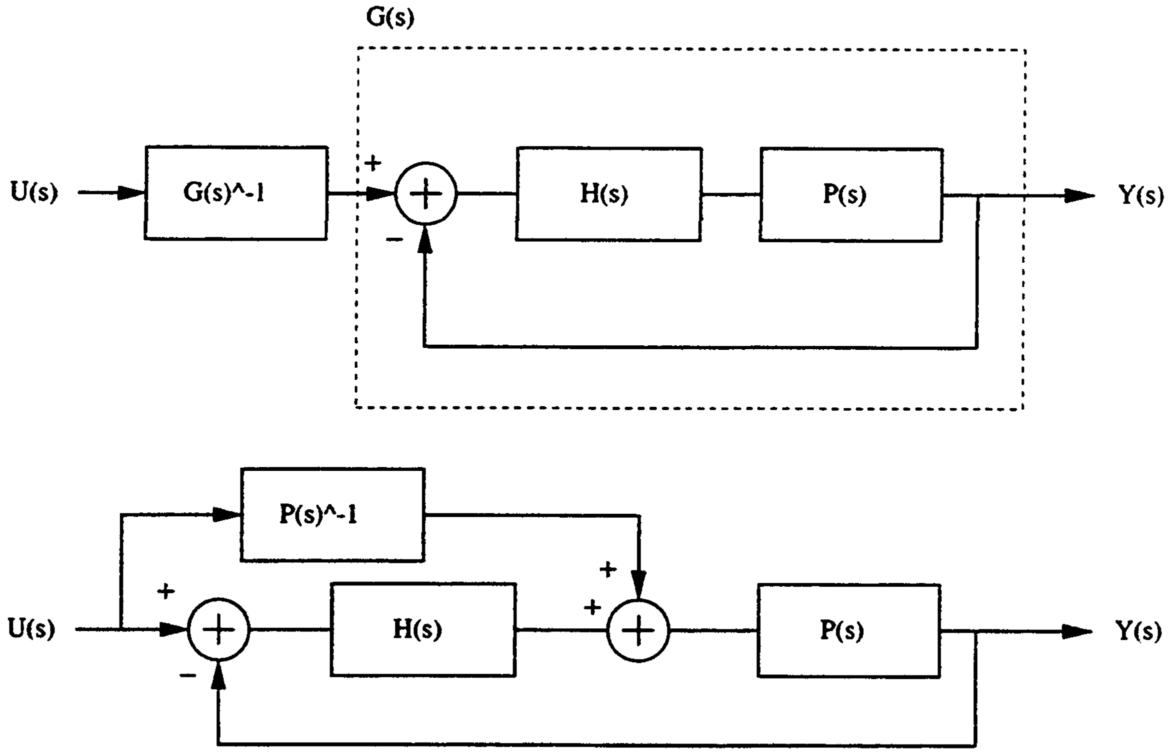


Figure 2.1: Typical Feed Forward Block Diagrams

2.3.1 Feed Forward

Feed forward controllers take advantage of knowledge of disturbances that can be measured before they affect a machine, by using a machine model to determine the command signal required to minimize error. Typically, the target path is the information known in advance, and this is used in combination with an inverted model of the servo axis.

Typical block diagrams are shown in figure 2.1.

Where:

$U(s)$	is the target trajectory
$H(s)$	is the feedback controller
$P(s)$	is the machine tool
$Y(s)$	is the actual trajectory

Tomizuka [52] introduced Zero Phase Error Tracking Control as a compromise when a machine model could not be inverted. While an improvement on standard pole-placement techniques, this strategy did not eliminate amplitude error or address the saturation problem of feed forward controllers.

Weck [57] developed an 'Inverse Compensation Method' to low pass filter the target path, thus reducing the saturation problem. This reduced the issue of loss of control due to command saturation, but it did not address requirements for coordinated motion.

Feed forward is open-loop compensation; it does not use past or current path error information in calculating command signals, nor does it address issues with coordinated motion. As a result, feed forward techniques respond poorly in the presence of disturbances or modeling inaccuracies.

2.3.2 Optimal Control

Optimal controllers create a command sequence which optimizes system performance over a specified horizon. For machine tools this is typically the minimization of the difference between target and predicted paths. The cost of the command signal is also often included. This can be important in machine tools as motors generally have thermal limits. Currently, GPC and its variations represent advanced optimal control.

Generalized Predictive Control

Generalized Predictive Control was introduced by Clarke [19] [20]. GPC uses a linear plant model to predict the output of a plant over a horizon, it then computes a command sequence which minimize a cost function over this horizon. The cost function includes error between the predicted plant output, and given future set-points, as well as control costs. The first command value of the computed command sequence is applied. The process is repeated iteratively at each sampling period.

GPC assumes a linear plant model. As such, it cannot address issues such as backlash and asymmetric performance. In its basic form, neither does GPC consider command saturation issues. This was addressed by Clarke [53] with 'GPC with Input Constraints'; however, by his own estimation this method was too computationally intense for servo control. A summary of the algorithm and terms follows, a complete description cam be found in Clarke's papers [19] [20].

GPC assumes a controlled auto-regressive integrated moving average (CARIMA) plant of the following form:

$$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t-1) + \frac{C(q^{-1}) \cdot \xi(t)}{\Delta} \tag{2.1}$$

Where:

q^{-1} is the backwards shift operator.

Δ is the differencing operator $1 - q^{-1}$.

$u(t)$ is the control input.

$y(t)$ is the measured output.

$\xi(t)$ is an uncorrelated random sequence.

And A, B and C are polynomials in the backwards shift operator q^{-1}. C however is often assumed to be 1 for simplicity

$A(q^{-1}) = 1 + a_1 \cdot q^{-1} + \ldots + a_{na} \cdot q^{-na}$

$B(q^{-1}) = 1 + b_1 \cdot q^{-1} + \ldots + b_{nb} \cdot q^{-nb}$

$C(q^{-1}) = 1 + c_1 \cdot q^{-1} + \ldots + c_{nc} \cdot q^{-nc}$

The GPC algorithm, derives it's control sequence by minimizing the cost function:

$$J(N_1, N_2) = E\left(\sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j)\,[\Delta u(t+j-1)]^2\right) \qquad (2.2)$$

Where:

N_1 is the minimum output horizon, generally the dead time of the plant.

N_2 is the maximum output horizon, generally set to the rise time of the plant.

$\lambda(j)$ is the control-weighting sequence.

$w(t)$ is the sequence of future set-points.

The predicted outputs can be represented as follows:

$$\hat{y} = G \cdot \bar{u} + f \tag{2.3}$$

Where:

$$\begin{aligned}\hat{y} &= [\hat{y}(t+1), \hat{y}(t+2), \ldots, \hat{y}(t+N)]^T \\ \bar{u} &= [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N-1)]^T \\ f &= [f(t+1), f(t+2), \ldots, f(t+N)]^T\end{aligned}$$

$$G = \begin{bmatrix} g_0 & 0 & \ldots & 0 \\ g_1 & g_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix} \tag{2.4}$$

$g_0 \ldots g_N$ are the impulse response values of the plant

Given equations 2.2, and 2.3 $\bar{u}$ can be solved for as follows:

$$\bar{u} = \left[G^T G + \lambda I\right]^{-1} G^T (w - f) \tag{2.5}$$

An efficient recursive method for solving this in [19],[20].

Adaptive Control

Adaptive Control systems modify machine controller parameters on-line. This includes a range of methods from simple gain scheduling, to on-line machine model identification and compensator design [4].

Practical examples of gain scheduling include compensating for gravity in a z-axis, compensating for the amount of flexibility in a lead screw based on load position, or perhaps compensating for work piece mass as a machining process progresses. These are all effective and practical examples of adaptive control, or gain scheduling.

More advanced forms of adaptive control refer to on-line estimation of machine parameters. Examples would include *on-line model identification* or disturbance estimation. This form of adaptive control places requirements on the control signal, namely that it have sufficient variation to enable proper model identification or disturbance estimation. This can be overcome by only enabling adaptation when there is sufficient information content in the command signal. However, one must consider what one is estimating, and why. The machining process is an exact process requiring tight positional accuracy and the machine tool has been designed for repeatable and consistent behavior. There should not be any unpredictable or unmeasurable system component that is going to change significantly during the machining of a part. In fact, the presence of significant unpredictable variations likely indicates a problem with the machine. An exception to this would be cutting forces, or disturbance estimation. The success of this however, will depend on the overall efficiency of the servo system, the size of the disturbances and accuracy of the predictions. The most successful system will be where cutting forces are directly measured, which is simply another example of gain scheduling.

Learning Control

Learning control adds past experience, in the same or similar situations, to the information available. This enables compensation for repeatable disturbances, or machine modeling error. Assuming that the machine and disturbances are consistent, or at least slowly varying, learning control provides the system with optimal feed forward.

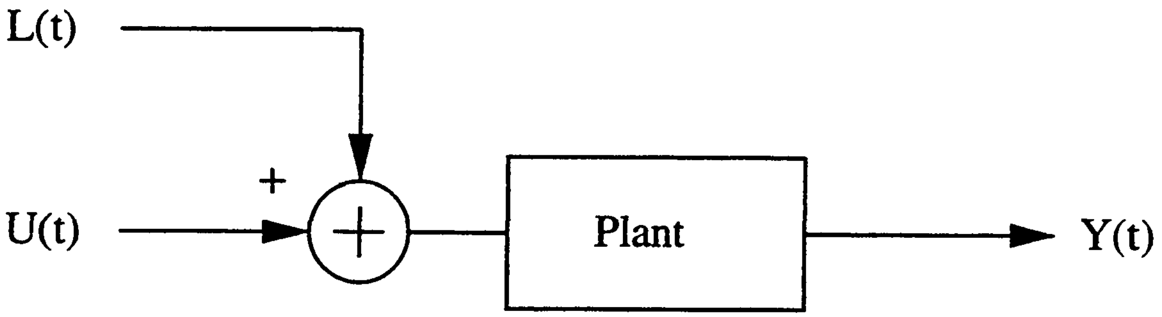


Figure 2.2: Learning Control Block Diagram

The following formula, combined with figure 2.2 summarize a typical learning controller.

$$L_N(t) = \sum_{i=0}^{N-1} F(U(t) - Y_i(t)) \tag{2.6}$$

Where:

$U(t)$	is the desired path
$Y_N(t)$	is the machine position at iteration N
$L_N(t)$	is the learned path error at iteration N
$F(e)$	is a function designed to provide stable convergence

Practical issues with learning control have hinged on the large amounts of data that was required to be saved [30]. At 2 kHz, 7.2 million sample points would have to be stored for each axis for an hour of operation. Assuming 5 axes, and 10 bytes per sample, this results in 360 million bytes of data. Considering that 25 G-byte disk drives sell for US$600 (June '99), the intermediate data problem appears past. Based on these numbers, that works out to $8.64 per hour of unique machining time for data storage (assuming no compression). If compression is used on stored data, this estimate could drop by a factor of ten. Clearly then, for a machine that is repeatedly making a handful of parts, learning control should be investigated.

It is important to note that, normally, learning control compensates only for issues related to servoing error. This is because the system encoders are used to calculate the actual system position. However, learning on final part geometry could also be performed. Other issues with learning control include that it usually only controls the command signal. This will be sufficient, as long as the system doesn't saturate. However, in the case of saturation of an axis, the ability to slow the system should be added. Additionally, the ability to speed the system up, to a limit, could also be added, as a method to optimize feed rate.

2.3.3 Cross-Coupled Control

Cross-Coupled Control (CCC) was introduced by Koren [28]. This method switched the focus of controllers from maintaining each axis at its target position to minimizing path error. CCC calculates (estimates) [13] the point on the target path that is nearest the machine's position, and uses this to determine the error for each axis (this can be seen in figure 2.3). This error signal is then used in combination with any one of a number of controllers to control the coordinated position (see figure 2.4 for a block diagram of the controller).

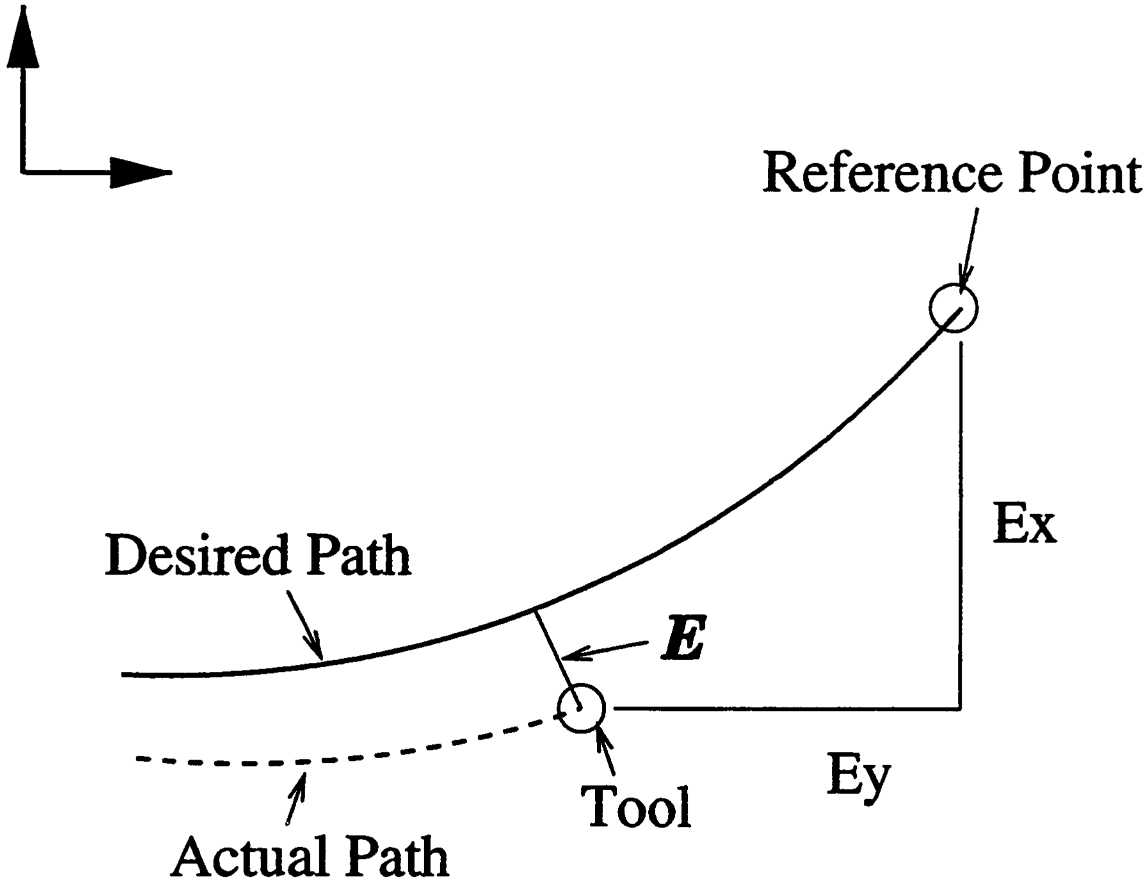


Figure 2.3: Contour Error used in CCC [30]

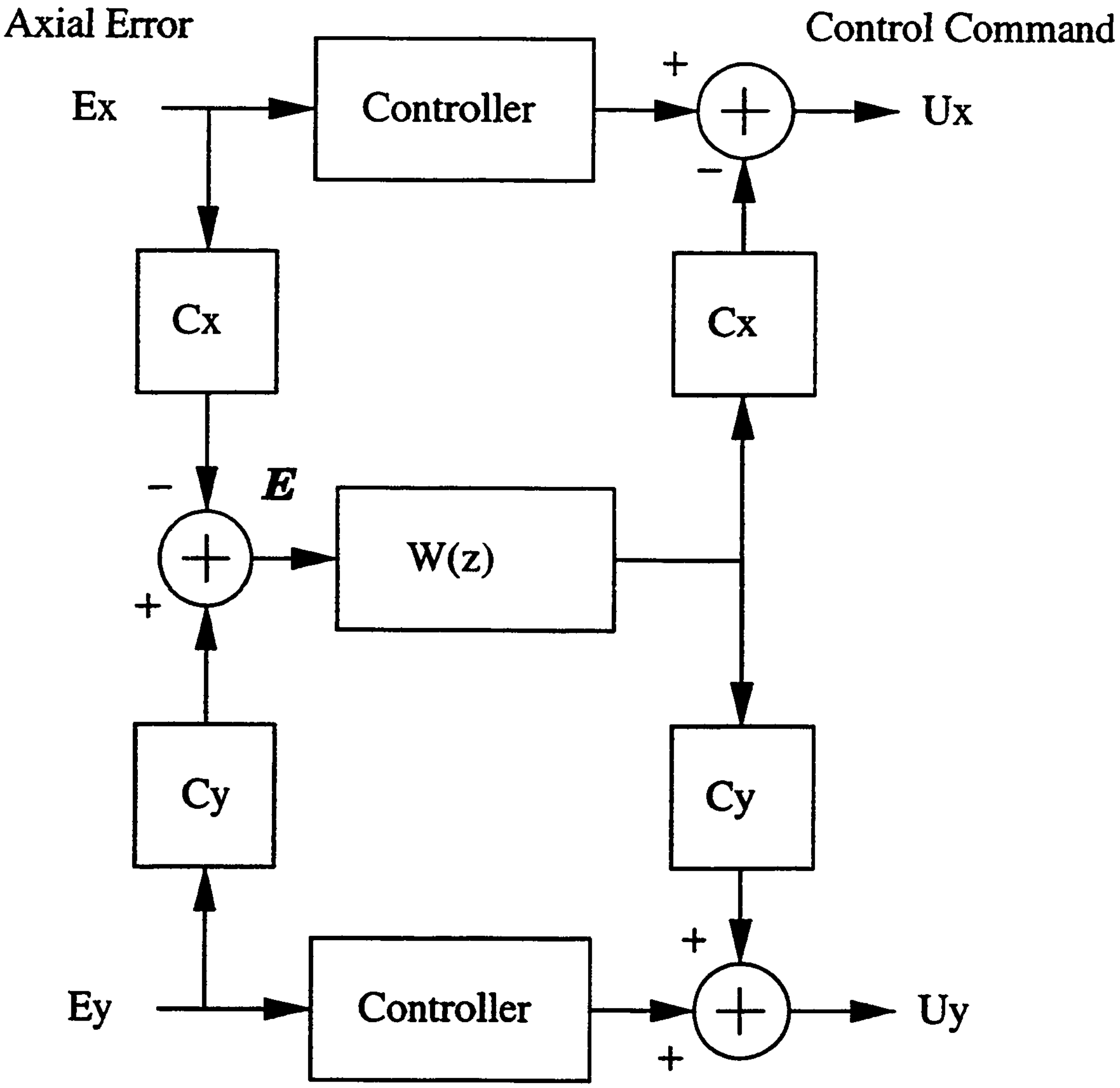


Figure 2.4: CCC Block Diagram [30]

Where:

E_x, E_y	are the X and Y Axis errors
C_x, C_y	are functions that determine X, and Y components of ϵ
ϵ	the shortest distance to the target path
$W(z)$	CCC Law

The difficulty with this method, as described, is in determining the point on the target path nearest the machine's position. This is accomplished by using various closed-form solutions, specific to the type of path being followed. This fails where two path segments meet, or for an arbitrary path. This seems to be an issue of developing appropriate formulae, or applying sufficient computational force. A form of CCC was developed by R.J. Seethaler and I. Yellowley [46] which slows the movement of the target, in real time, in response to excessive error on any individual axis.

The effectiveness of CCC is dependent on the method used to calculate the point on the target path nearest the machine's position, and on the control schemes used to control each axis once this point is computed. Clearly, CCC can address a number of the issues that other control schemes do not, like coordinating motion among several axes. One problem with CCC is that it lacks the ability to compensate for future path changes. Specifically, there is nothing to slow a CCC system down before encountering a corner or any other obstacle that could exceed the deceleration capability of one of the axes. This indicates that CCC should be used in combination with feed rate scheduling.

2.3.4 Other Control Schemes

Neural Networks

Neural Networks, depending on their size, require more computational power than Adaptive Control or Fuzzy Control, however, its requirements are usually still moderate, and its use is practical for servo control. The theoretical advantages of neural nets include self-training and the ability to compensate for non-linear elements.

Practical problems with neural nets include long training times and the requirement of a full spectrum of training data. The output of a neural net outside of its training data is unpredictable. Other problems are that system elements are not easily identifiable within a neural net, so nothing can be learned from the neural net, and elements like backlash or flexibility cannot be easily adjusted. Specifically, one could not train the net at the middle of table travel (ball screw system), even if position and some spring constant were fed to the net, and expect the net to handle operation at either end of the table optimally. Also, if the network is trained properly, along the entire length of travel, and a ball nut is tightened to reduce backlash, the network will require retraining, to optimally take advantage of the system improvements. The training is likely to take a significant amount of time. The same could be said about any other system variable.

Fuzzy Control

Fuzzy control, like gain scheduling, changes the control scheme, based on state information. Fuzzy control, however goes slightly further, in that any amount of logic, and look-up tables are used. This can be very effective in handling system non-linearities, such as backlash [29].

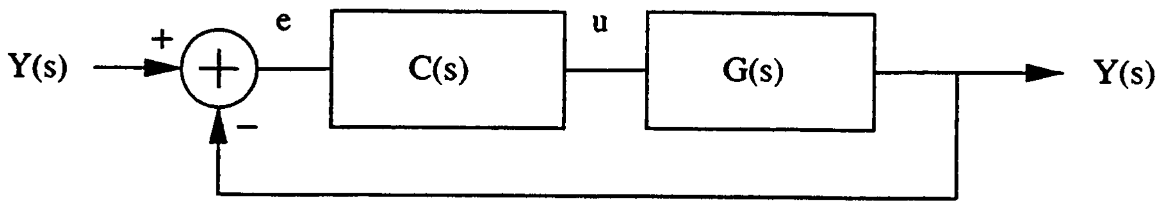


Figure 2.5: H^{∞} Block Diagram

In its simplest form, a look-up table, fuzzy control is impractical for machine tool control. There would simply be too many input-output relations to store, let alone generate, for smooth, continuous servo control. However, when combined with gain-scheduling, and other 'control' methods, fuzzy control can be used almost in a supervisory role to compensate for many system nonlinearities.

H^{∞} Control

An H^{∞} controller is a controller designed to minimize the H^{∞} norm of a set of design functions describing aspects of the desired closed loop performance. H^{∞} can be implemented as a simple single input, single output state-space controller, and can easily be computed at the frequencies required for high-speed servo control. The difficulty with H^{∞} control is two fold, first determining the functions to optimize, and second in finding a solution.

A block diagram of a H^{∞} is shown in figure 2.5. The weightings that are typically used are $W_p \cdot S$ to adjust the performance of the controller, $W_u \cdot R$ to limit saturation, and $W_r \cdot T$ to adjust the stability margin [11].

The requirement becomes to find the controller that satisfies the following:

$$\left\| \begin{array}{c} W_pS \\ W_uR \\ W_rT \end{array} \right\|_\infty \leq 1 \tag{2.7}$$

Where:

W_p is the performance weighting

W_u is the saturation weighting

W_r is the stability weighting

and, where:

$$\begin{aligned} T &= \tfrac{GC}{1+GC} \\ R &= \tfrac{C}{1+GC} \\ S &= \tfrac{1}{1+GC} \end{aligned}$$

In order to test H^∞ control, matlab's robust control tool-box was used for finding the solution. This result was then used as the H^∞ controller for the linear motor. There were a number of problems with this, however the procedure is clearly outlined in the matlab manual. The most significant issue is designing the weighting functions so that the H^∞ controller will have the desired properties. For machine tools desired properties include minimal steady state error, minimal following error to a ramp, and critically damped responses to steps. These do not relate clearly to the design weighting functions used in H^∞. It is also easy to specify desired properties for which a solution fails to be found. In addition to these difficulties,H^∞ relies on a linear machine model, and does not address issues of actuator saturation.

2.4 Linear Motors in Machine Tools

The advantage of linear motor driven machine tools over conventional ball screw driven machine tools is the elimination of the flexible drive. As a result systems can be created who's first natural frequency is significantly higher than previously practical. This is an advantage since in practical systems it is difficult to accurately compensate for the resonance of the system, and accordingly it must be avoided. This is accomplished using a band stop, or low pass filter, or reduced controller gains. In any case, the performance of the system is much reduced past this frequency.

In addition to high dynamic applications, linear motor drives also find application in precision machining. The reason for this is that the high frequency noise generated by most conventional drives (ie the bearings from ball screws) can be eliminated. When combined with air or hydraulic bearings, such a system can machine surfaces with very low surface roughness.

2.4.1 Types of Linear Motors

There are three main categories for linear motors, these are induction, permanent magnet, and stepping motors.

As is the case with rotary motors, linear stepping motors, are not "high performance" rather they are low cost, and easy to control, in that they can be run in open loop. However they will not generate the high dynamic performance that permanent magnet motors will (as is the case for rotary motors), and they require position feedback or significant over design to achieve high stiffness and reliability in the presence of disturbances, eliminating any advantage in these situations.

Linear induction motors, themselves could be even less expensive than linear stepper motors. They are particularly advantageous in cases where there is a significant length of travel, and exact position is not critical. For use in precision machine tool control they require significant control electronics, typically a "variable frequency drive". The net result of this is that for use in machine tools, they are no-longer orders of magnitude less expensive than an equivalent permanent magnet motor. Never the less, with appropriate control electronics, they have been shown to have sufficiently high performance for use in machine tools. They also have some advantages, in that when turned off they have no magnetic fields, to attract chips, or to make assembly difficult or dangerous.

Permanent magnet linear motors operate as their rotary counter parts do. They are available in brush, and brushless configurations. Typically, though, they are three phase brushless. They offer the highest dynamic performance, and the highest cost. They also have strong magnetic fields that can not be "turned off", making cleaning, or removal of chips that find their way to the motor difficult. Assembling these motors due to these strong permanent magnetic fields can also be tricky, even dangerous, particularly in the single sided case.

Other distinguishing features, such as relative lengths of the primaries, and secondaries (unique issue to linear motors), number of sides, single or double, and the presence or absence of iron, apply basically equally to each type of motor.

The relative length of the primary and secondary, is usually not an issue. The primary (active) is short, and the secondary (passive) is long. In the case of stepping, and induction motors this choice is clear, in that this is less expensive. In the case of permanent magnet motors, the choice is less clear, however less power is wasted, and a smaller amplifier is required. Additionally, the primary can be made lighter than the secondary, and thus higher performance can be reached, which is the advantage of the permanent magnet motor.

The most significant variations, are the choice of single sided, or double sided motor, and the inclusion, or exclusion of iron in the primary. Double sided motors have no net force normal to the direction of motion. This is important because in the single sided case, in addition to the desired force in the direction of motion, force is also generated normal to the primary and secondary. This can result in numerous problems, including part error (due to the limited stiffness of the guides), and some sensor issues (reflective position sensors are sensitive to the gap between read head and scale, when these are mounted incorrectly, hight variation is incorrectly measured as position variation, and system stability is reduced). In the case of double sided motors, these forces cancel. Double sided motors however, suffer from reduced stiffness of the attachment of the primary to the system that is being moved, simply due to the difficulty of construction.

Iron is included in the primary, (most often in single sided motors) to contain magnetic fields, and to increase the motor efficiency. This leads to cogging, which is essentially a disturbance which must be rejected. In the case of permanent magnet motors, the magnets are often put at a slight angle to reduce this effect. In any case, this is detrimental to performance, and while effects can be reduced through accurate modeling, they can not be eliminated, and they complicate the control problem.

2.4.2 Literature on Linear Motors in Machine Tools

Alter and Tsao [1] examined the issue of stable turning using linear motor feed drives. They were motivated to use linear motors by the potential of improved performance by eliminating gear related mechanical problems. Theoretical, and experimental results showed that stable turning using linear motors is achievable. They concluded that stability is not a problematic issue for turning, and that attention should be focused on issues such as disturbance rejection, and trajectory tracking.

In references [2] and [3] Alter and Tsao examined H^∞ and l_1 optimal feedback and feed forward controllers as applied to linear motors. Again they cited their motivation, as potential performance improvements resulting from linear motor drives. In addition to position feedback, cutting force feedback was examined. It was found that in a practical system, H^∞ optimal position feedback could increase dynamic stiffness by 27% to 46%, and H^∞ optimal force feedback could further increase stiffness by 70% to 100%. H^∞ and l_1 optimal feed forward controllers were found to be limited particularly in bandwidth in practical systems, due to modeling uncertainty. Nonetheless, they succeeded in reducing RMS tracking error by more than 50%. They concluded that H^∞ and l_1 optimal feedback and feed forward controllers represent a viable control scheme for linear motors. They also concluded that careful planning of the mechanical system is still required to eliminate structural resonances and friction.

Pritschow and Philipp [41] investigated direct drives for machine tools. They cited as motivation increasing requirements on velocity, length of travel, and accuracy. They argued that conventional drive systems (ball screw, toothed belt and toothed rack) can not fully satisfy these requirements due to the characteristics of the mechanical drive elements. They argued that cost is a significant barrier to the wide spread use of linear motors. This cost results from two components, the magnets generally used, and the high resolution position feedback systems required to generate meaningful velocity information by simply differentiating the position feedback signal at the high servo loop update rates required by linear motors. They addressed these issues by first using an asynchronous linear motor, which does not require costly permanent magnets. They argued that these motors have developed to the point that they are equivalent in performance to permanent magnet linear motors. Accordingly, they are less costly, and easier to install and maintain. The second issue, the cost of high resolution scales for generating velocity feedback data at high servo loop rates, was addressed using observers. They also presented the transfer function of their test linear motor, and show that there are essentially no significant natural frequencies in their system. The motor has linear gain from command to acceleration up to 900 Hz, and is linear in phase up to 300 Hz. They concluded that linear motors indeed have advantages over conventional drives in acceleration, velocity and travel lengths, as well as having high stiffness, high position loop gain, and high position accuracy. They also stated that the drive system presented is cost competitive with conventional drives.

In [42] Pritschow and Philipp investigated feed forward controllers for direct drives. They highlighted the limitations of feed forward control, specifically the inability to react to parameter deviation, model errors, friction, and disturbances. They also acknowledged that feed forward control can not compensate for paths which can not physically be executed by the plant, and that such paths should not be attempted. They presented a plant model (motor, slide) that is simply a double integrator. They point out that non-causality is not an issue in the case of machine tool control, as the path is known in advance. Examples were then given showing up to a 50,000 times improvement in performance by using feed forward control. This was then critically examined, and shown to not be achievable in practice due to parameter uncertainty, which is argued to be as high as 25 %. Even at 5% parameter error it was shown that feed forward results in poor performance. It was argued that increasing K_v (the velocity loop gain) best reduces path error. They were also critical of predictive controllers, for offering no advantages over classical structures, and only having one parameter to adjust the control quality. They concluded that mechanically stiff, backlash free systems with low inertia are required for optimal path accuracy, and that linear motor drives best meet these requirements. They also concluded that K_v is more critical to path accuracy than feed forward control, and that in the case of stiff linear motor systems, this is only limited by the feedback system, and speed of the control computer.

In [40], Pritschow noted that after more than ten years of intensive development linear motors have not displaced conventional drives. He thus compared conventional and linear motor drives in order to help designers chose between the two systems. The advantages of linear motors were cited, particularly, better dynamics, higher acceleration, higher achievable velocities, and longer achievable lengths of travel. The disadvantages were cited as higher costs.

Schultz, Gao, and Stanik [22], examined dynamic contour accuracy of linear motor driven machine tools. They cited as their motivation, the ability of linear motor driven machine tools to exploit the manufacturing potential of high speed cutting in the mold and die industry. They attributed this ability, to the elimination of the elastic recirculating ball screw, which resulted in improved dynamics, and higher achievable

K_v factors. They distinguished between contour lag, and contour deviation, and argued that contour deviation is the main concern. The contour deviation was shown to be most dependent on jerk, and acceleration, and limiting these was argued, at the expense of machining time. Accurate feed forward control was argued to reduce contour deviation. They concluded that contour deviation optimization should be performed, rather than contour lag optimization .

Chen, and Tlusty in [12] investigated the use of lead-screws and low friction guide ways in high speed machines. They acknowledged the problems generated by the flexible lead-screw, and minimal system damping. The motivation is that these systems are very common. They noted that the 10kHz amplifier, with internal current control loop almost completely eliminates the electrical time constant of the motor (the jerk limit). They also noted that the 10kHz servo loop closure rate provides essentially continuous control. The best performance was realized when feed forward techniques are combined with acceleration feedback.

Otten et al. in [37] investigated the use of a learning feed forward controller on a linear motor driven system. They were motivated by the increasing popularity of linear motors, and the possibility of superior performance. It was argued that linear motors have a number of repeatable, but difficult to predict non-linearities, that can be learned, and compensated for, including torque ripple, and variable winding

2.5 Open Architecture Machine Tool Controllers

A machine tool controller was required to implement the advanced, computationally-intense algorithms selected for testing and comparison or developed in this thesis. OAC was looked to for the solution. According to [50] OAC is an enabling technology for the application of advanced controls in manufacturing. To date there has been a significant amount of work performed in this field. Some of the well known efforts include MOSAIC, NGC and OSACA.

2.5.1 Machine tool Open System Architecture for Intelligent Control (MOSAIC)

MOSAIC was one of the first attempts at developing an Open Architecture for machine Control (OAC). It was developed at New York University. It's original purpose was to enable research related to expert planning systems. Machine tool controllers at the time were unable to perform the data handling required to perform these tasks. MOSAIC was implemented using "C", real time UNIX, and the VME bus. MOASIC also made use of a low level motion control board, one example being the Creonics MCC [50] [59].

2.5.2 Next Generation Machine / Workstation Controller (NGC)

The NGC program was a US government initiated program, whose aim was to revitalize their machine tool industry through advancement in the areas of computer hardware, software, and process controls. The system was implemented using a workstation, which provided the user interface, and performed high level functions, as well as a real-time platform dedicated to machine control. The NGC program also described a "low end controller" implementation which did not implement high level functions such as process planning. The definition of the LEC also divided the original NGC architecture into non-real-time, real-time, and hard real-time tasks. This was in recognition of the fact that the time critical functions were almost exclusively implemented on dedicated custom hardware [59] [50].

2.5.3 Open Systems Architecture for Controls within Automated Systems (OSACA)

OSACA was a European effort to develop OAC for similar reasons as NGC, specifically, "to improve the competitiveness of the manufacturers of machine tools and control systems in the world market". (see www.osaca.org) It was the result of of two ESPRIT projects named OSACA: OSACA I (EP6379) and OSACA II (ESPRIT 9115). It was begun in November 1992 and was completed June 1998, the project cost 12.3 MECU, and 96 man years. The result of these projects was to specify and implement the OSACA architecture. Specifically this is a seven layer hierarchal model based on the ISO-OSI communications standard. OSACA deals substantially with high level aspects of OAC, in-fact four of the seven layers do not deal with the machine tool [50].

2.5.4 OAC Summary

While MOSAIC, NGC and OSACA are three of the largest efforts at developing OAC, they are by no means alone. There however, seems to be a very common component to all of these OAC implementations, and that is a low level motion control "card". One potential exception was the Evolutionary Test Bed Controller [60], which started out using a PC, and general purpose IO, but then migrated to a DSP multi-axis controller and VME controller with a real-time OS. It seemed the PC (i486 33Mhz) lacked a real-time, multi-tasking OS and the processing power to perform the required tasks.

Since no system was available which could implement the required demanding, arbitrary control schemes easily and in a cost-effective manner, one was developed. The system implemented all of the hard, real-time processing required to close servo loops on a PC motherboard, and used generic IO hardware (such as encoder counters and digital and analog IO) to interface with the machine. This provided the developer with the ability to modify the servo loop functions in software, using familiar development tools, as well as the power of modern computers. Significantly, no custom hardware is required, and both the economies of scale of the PC and the development work on UNIX are leveraged [48]. This was first implemented on a 200MHZ i686, running Free-BSD (UNIX), with the hard real-time tasks running cooperatively in a high priority ISR. Since this time Real-Time Linux (RT-LINUX) has been introduced, and the software has been ported to this platform. This is unique when compared to the large efforts, in that the Open Architecture Machine Tool Controller is actually implemented on an open and free operating system. This system is referred to as "PC based OAC with soft motion control". This work is currently being integrated with the OAC developed at the IMMRC by R. Teltz [50]. This uses a Delta-Tau motion card, which is being eliminated, and the project which was VME based is being ported to a PC running RT-LINUX.

2.6 High Speed Machining

High speed machining is difficult to define, and depends significantly on the material being cut, and the type of cutting operation (milling, turning, drilling). High speed machining however definitely uses higher spindle speeds and feed rates than has been common [51]. The central concern for this thesis are the reasons for this trend, and the resulting requirements on the servo system. The advantages of high speed machining are not simply higher metal removal rates. Other benefits include increased quality, reduced production time, and reduced costs.

Higher quality is achieved through a number of mechanisms. These include that at high speeds lower depths of cut are generally used which reduces cutting forces and therefor part and tool deflection is reduced. Also, heat remains in the chip which is removed preventing a build up of thermal energy in the part which can cause stresses to develop in the surface, and cause warping or thermal deformation. Additionally, parts tend to be less susceptible to forced vibration at higher frequencies [51], this is because the mass of the part acts like a second order low pass filter, which in effect increases the part stiffness at high frequencies. These differences, yield higher surface quality, and better part tolerances.

Production time is reduced as a direct result of the higher cutting speeds, which allow higher feed rates. Additionally, due to the higher quality and better tolerances, hand finishing particularly in dies and moulds is reduced.

The final advantage of high speed machining is reduced costs. This is achieved through reduced production times, and better utilization of equipment and tools. (A cutter at 20,000 rpm has ten times the utilization of a cutter at 2000 rpm) In addition to this high speed machining is performed dry, (or with very little lubricant) eliminating a manufacturing cost [44].

The benefits of high speed machining are not free. First there are increased demands on the machine tool, which require more advanced and costly machines.

Some of the extra demands include higher speed spindles, increased rigidity, and better contouring performance. In addition to this there are increased safety concerns due to the large amount of centrifugal energy stored in the spindle. This requires the machine tool to be fully enclosed, and capable of withstanding a tool breaking at high speeds [36] [44].

2.7 High Speed Machine Tool Design

Current areas of research for machine tool design center about providing the high feed rates, exceptional rigidity, and tight tolerances required by advances in high-speed spindles and cutting technology. The most promising method currently is the replacement of ball screws and ways with linear motors and roller guides.

Linear motors have two significant advantages over motor-ball screws, namely, no backlash and no flexibility. In addition, they can reach higher top speeds. These properties eliminate two of the most difficult to measure, and compensate for problems with current machine tools. Using roller guides in place of ways virtually eliminates a third source of nonlinearities, friction [12]. Therefore, the use of linear motors with roller guides creates a system which is essentially linear, with the exception of amplifier/motor saturation. This creates a straightforward, second-order system, with constraints on amplifier voltage and current. This is an ideal system, from a control perspective. Potential limitations are the increased throughput required of the controller, and the increased importance on the acceleration capabilities of the system [41] [7] [27].

Another promising area of machine tool development is parallel-link mechanisms. The advantage of parallel-link mechanisms over traditional orthogonal axis designs is the promise of lower moving masses and a stiffer structure. This should result in better dynamic performance and higher accuracy [21], [56].

The disadvantages of parallel-link mechanisms center around workspace limitations and variability. Specifically, the workspace of a parallel-link mechanism will be more limited, and its performance will vary widely within this envelope [21], [56] when compared to a traditional three-axis machine. This difference is smaller if a five-axis machine tool is considered. Parallel-link mechanisms also require very tight control to machine straight lines, or flat surfaces, common features of today's parts, while orthogonal axis designs are able to do so with ease (assuming that the surface lines up with an axis). This difference vanishes, however, for arbitrary surfaces, or surfaces not aligned with an axis. Additionally, each axis of a parallel-link mechanism may have a performance envelope which is dependent on the current orientation of the machine, thus complicating envelope mapping and path optimization procedures.

The common element in these developments in high speed machine tools is that it is certain that increased demands will be placed on controllers, in terms of throughput, flexibility, and servo control strategies.

2.8 Summary

An investigation of the issues central to high speed servo control of multi-axis machine tools revels that improvements are required in all areas. Path planners are focused on the difficulties of on-line path planning of elementary functions. This includes minimizing computational demands placed on machine controllers, and focuses unnecessarily on worst case computational times. Significant effort is also placed on methods to represent the cutter path in a compact intermediate form, due again to controller limitations, particularly in terms of memory, and data transfer rates. Path planners attempt to maintain constant surface feed, rather than directly addressing machine limitations, this results in poor machine utilization, and/or large part error. Servo control algorithms generally assume a complex linear plant model, and take no explicit account of saturation (in a computationally efficient manner). This results in under-performance. Linear motors appear to be the actuators of choice for new high performance machine tools. This is due to the excellent mechanical characteristics of machines based on this technology. Specifically, linear motor driven machines tend to have a significantly higher natural frequency than conventional machine tools. Open architecture controllers focus on high and medium level openness, little effort is directed at low level openness. As a result, no systems were found, which were suitable to conveniently implement new control methods. High speed machining advances and new high speed machine tool designs will certainly demand tighter tolerances, and higher feed rates, increasing the importance of control methods and controllers.

Chapter 3

Machine Model, Simulation, and Test Platforms

3.1 Introduction

There are many compromises to be made when researching a problem such as high speed machine tool control. The problems include safety issues, realism, dollar cost, and time cost. The safety issues include both personal safety and machine safety. In the development and testing of a new controller and algorithms, unpredicted and potentially damaging behavior is likely. The most obvious of which is an axis running away and colliding with an end stop. When linear motors are in use, this can happen in a fraction of a second. For this reason, initial testing was performed using a simulation. This ensured that technical difficulties could be worked out in as safe a manner as possible. The simulation however also provided some advantages over a real system, in that many different systems and controllers could be tested, simply by modifying a parameter. This provided a significant cost and time savings as compared to creating a physical test system which could be modified to model a spectrum of plants, or even finding actual plants for testing.

Simulation alone however was not believed to be sufficient to prove the validity of any conclusions. It was expected that there would be inaccuracies in the simulation, and unaccounted for factors. To address this, testing on physical systems was performed. The first test system was a rotary motor and mass. With the PC-based OAC functioning well on this platform, a linear motor was purchased, and testing was continued using the linear motor.

Based on the simulations and the testing of the motor and mass, the general concepts of this thesis were generated, namely, 'Minimum-Time Path Optimization', and 'Minimum-Time Tracking Control'. These methods are applicable to both conventional, and linear motor driven machine tools, however their advantages are most clearly exhibited in a high performance system. Additionally, it was found that sensor feedback was an important issue. It was found that analog velocity feedback was often noisy, and that differentiating position feedback provided unusable velocity feedback at low speeds. This led to an investigation into the use of advanced sensors, especially those that could be used on linear motors. It was found that the use of "digital tachometers", traditional accelerometers, and the "ferraris sensor" (an acceleration sensor based on the ferraris effect) could provide solutions to the aforementioned sensor feedback problems.

Through the use of simulation, PC-based OAC, various physical plants, as well as a number of types of sensors, an appropriate solution was found that satisfied the requirements of accurate results, safety, cost, and time.

3.2 Simulation

Initially a closed form solution, or model was investigated. A relatively simple block diagram of the mechanics of a servo system is presented in figure 3.11. The transfer function from force applied to position of the plant (M2) is given in equation 3.11. This block diagram does not include the effects of the amplifier, backlash, friction, or delay. A more general block diagram, including the amplifier effects is given in figure 3.10. This model still does not include effects of backlash, friction or delay, nor does it include a servo control loop. The major components in a servo loop are the sensors. To effectively model sensors, quantization and noise must be simulated. A final component would of course include disturbances. It was soon decided that the most practical method to model this plant would be to create a time domain computer simulation.

3.2.1 Model

In order to better understand the problems related to high-speed machining, a detailed time domain model was created. A block diagram is presented in figure 3.6. It was intended that this model should contain all significant effects that would be present in a typical plant used in high-speed machining. Elements that make this model different than typical Laplace or z-transform representations are the nonlinear effects. The sources of non-linearity include:

- Quantization of key signals and the addition of noise

- Amplifier current and voltage limits

- Stiction / Friction

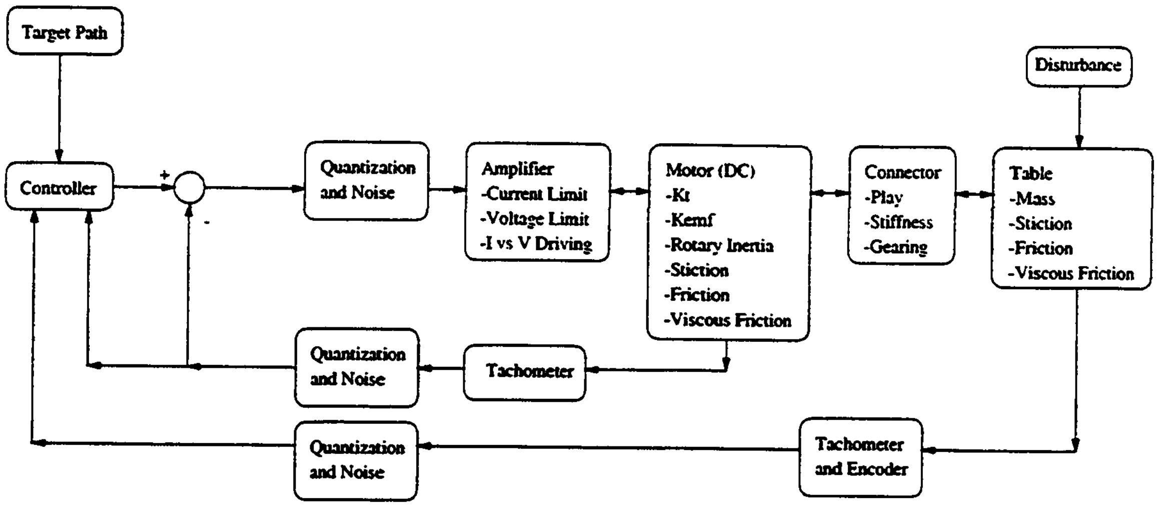


Figure 3.6: Simulation Block Diagram

- Backlash

By using this type of simulation instead of typical linear models, a more realistic representation of the issues associated with high-speed servo control was obtained.

3.2.2 Implementation

The model was implemented as a time domain computer simulation. Each box in the block diagram represents a subroutine. Each subroutine took a number of variables as inputs, performed the appropriate calculations, updated its state variables, and output results. These results were fed to the next block, and were used by the previous block on the next step. The time steps were made small enough that continuous functions were approximated.

The simulation was designed to reproduce the effects that a real machine tool would exhibit as closely as possible. The system was also designed to be modular, with clearly identifiable communication channels, and parameters linking the elements.

This aided in verifying the proper operation of the simulation, and made it easy to capture intermediate information.

Input - Output Files

The simulation is run from the command line, and takes four file names and an optional -t (train) as input parameters. The first file is the system file, it is a list of files. Each of these listed files contain the values for the important parameters for a simulated axis. A sample of some of these parameters is presented in table 3.1. To increase the number of axes in the simulation, simply increase the number of entries in the system file. The next file is the path file, on the first line it contains the number of samples along the path, and the number of axes of motion described. (This must match the number of axes in the system file, or an error is returned). On every line after this, there are numbers indicating where each axis is supposed to be at that instant in time. The third file is the disturbance file, it has the same format as the path file. The last file is the main output file, called the simulated path, this is a list of data that can be viewed directly with standard graphing utilities. It contains the path data for the target, the table, the motor, and the learned error. The last optional parameter is a -t (train) this option will enable the simulator to self tune simple controller parameters, such as PID, or Feed-Forward, in a recursive least squares sense. In addition to these specified files, there are a number of helper files that are output to the same names each time the simulation is run. These include a graph of the absolute error, the output of the amplifier (volts and amps) as well as a matlab ready file containing the control command, and the position of the table (particularly useful for diagnosing the system) this is normally used when the system is run open loop.

Quantization / Noise

This element of the simulation is capable of quantizing the output of the controller, as if it was being fed through an analogue to digital converter. It is also able to inject noise into the system, to simulate electrical interference that is often present in industrial settings. This element is helpful in determining the resolution of analogue to digital converter is appropriate, and what level of background noise is likely to be acceptable.

Amplifier

The amplifier can be current or voltage driving, and is current and voltage limited. It uses the command signal, the type of amplifier selected, the internal resistance and back-emf of the motor along with it's own current and voltage limits to determine the current applied to the motor at every system update. It is important to note that the internal amplifier motor-connector-table "loop" is updated more often than the external control loop. This provides for a more realistic simulation. This parameter is called system_N in the "Axis Definition File".

D.C. Motor

The motor is assumed to be a traditional D.C motor. It has a current - torque gain K_t, a back-emf per kilo- rpm K_{emf}, rotary inertia, stiction, friction, and viscous friction. The motor reacts within these limits to the current applied to it, as well as the torque applied to it by the connector. It calculates it's back-emf, and position, making this data available to the amplifier, and connector.

Connector

The connector encompasses everything used to connect the motor to the axis being moved. This can include gearing, belts, ball-screws and flexible couplings. The parameters used to represent this range of physical phenomenon are play, stiffness, and gearing. The connector takes the position of the table, and the motor, and calculates a force to apply to both.

Table

The table has mass, stiction, friction, and viscous friction. It reacts to the forces applied to it by the connector. and the disturbance (Load). It makes available it's position, for the connector, and the controller.

3.2.3 Load

The purpose of the load is to test the effectiveness of control schemes to disturbances, as well as to investigate the benefits of a learning controller. It is a force that is read in from a file, and applied to the table.

3.2.4 Simulation Results

The purpose of the simulation was to provide a facility for investigating as many system parameters as possible which influence servo performance, with the ultimate aim of isolating areas in which further effort could be placed to improve performance.

The effect of sensor noise was generally increased noise on the command signal sent to the amplifier. Most of the noise was filtered by the plant and did not translate to position variation, however this resulted in increased power consumption. In practice this would generate heat in the motor which would reduce the continuous torque limit of the system. There were also issues with the frequency content of the noise. High frequency noise was easiest to reject. Filtering the feedback was effective at reducing noise, however system performance was also reduced. Non-linearities in the velocity feedback most significantly reduced servoing performance and stability. Amplitude errors could be easily compensated for with a filter, or lookup table, however lag creates delay which could not be compensated for, and had the greatest impact. Quantization of sensor data resulted in reduced steady state performance. This included small oscillations, particularly in systems with little damping. This could be reduced if a dead zone was introduced, but this also reduced positioning accuracy. Quantization, and noise had much less of an impact in the forward loop, the only exception being for delay.

The effect of increased system mass was generally reduced acceleration capabilities, and reduced system bandwidth. There was also the issue of whether the system mass was concentrated in the table, or in the motor. The best results (assuming constant equivalent inertia) were achieved when the motor mass dominated, however this is clearly not practical. For a given table mass, it was best to reduce motor mass. Backlash, and flexibility in the transmission (connector / ball screw) reduced system performance. The effect of backlash can be reduced with a fuzzy control scheme, and a second encoder on the motor, however accurate positioning to a tolerance less than the amount of system backlash in the presence of disturbances did not seem practical, even in simulation. This fuzzy control scheme used the two encoders (one on the motor, and one on the table), and knowledge of the backlash to adjust the target position of the motor (or position error feed to the motor controller) by the known amount of backlash, depending on the direction the table was required to accelerate in. The effect of a flexible connector can be addressed by using the amount of flex in the lead screw as an indicator of the torque generated by the motor (note that the lead screw will have a position dependent spring constant). By adding an additional control loop around this, the effect of the flexible lead screw can be significantly reduced. The best results were achieved with minimal system backlash, and maximum transmission stiffness. (Figures 3.7 and 3.8 show simulation results for systems with excessive backlash, flexible connector, and stiction.)

Friction, stiction, and viscous friction all reduced system performance. Stiction most affected positioning accuracy. Integral control did not satisfactorily address stiction, due to 'windup', which tended to cause bursts of instability. This was very difficult to address generally, however again using a fuzzy control scheme this could be reduced. This was essentially a scheme that impacted the system, to start the system moving, and then used a regular control scheme. This was effective for motor stiction, and table stiction, if there was a stiff connector. When a flexible connector was used, the effectiveness of this method on compensating for table stiction was significantly reduced. Friction (constant, independent of velocity) could be dealt with by adding a constant to the command, depending on the direction of motion, and the sign of the command. Again this was most effective against friction in the motor and table, if there was a stiff connector. Velocity proportional friction was not as problematic, however it lead to increased following error at higher velocities. This effect could be reduced using velocity feed forward.

Limited amplifier current and limited amplifier voltage were found to limit system acceleration, and velocity. This created significant system non-linearities. Specifically, system response was no-longer independent of the amplitude of the input. This effect was most clearly visible in high dynamic systems. The evidence of this is shown in the responses to steps of varying amplitudes (see figure 5.36). The result of this was reduced system performance, specifically systems tended to overshoot in response to large steps, and responses were slower than necessary to small steps.

The final elements in the simulation examined were the target path, and the disturbance. Generally it was found that plants followed lower frequency signals (lower than servo loop closure rate, sensor bandwidth, and natural frequency of the system) more accurately than higher frequency signals. It was also found that the systems were best at rejecting low frequency disturbances, worst at medium frequency disturbances (system natural frequency), and improved again at higher frequencies, as the mass of the table started to dominate. It was also found that attempting to follow target paths that exceeded the capability (current limit) of one or more axes resulted in large path error. This can be seen in figure 4.26. This was also true of disturbances.

3.2.5 Simulation Conclusions

Sensor feedback was found to be important for servo control. The effects of high frequency noise, and repeatable amplitude errors could be addressed through signal conditioning. Sensor delay, low frequency noise, and non-repeatable errors could not be effectively compensated for, and tended to reduce system performance. Sensor

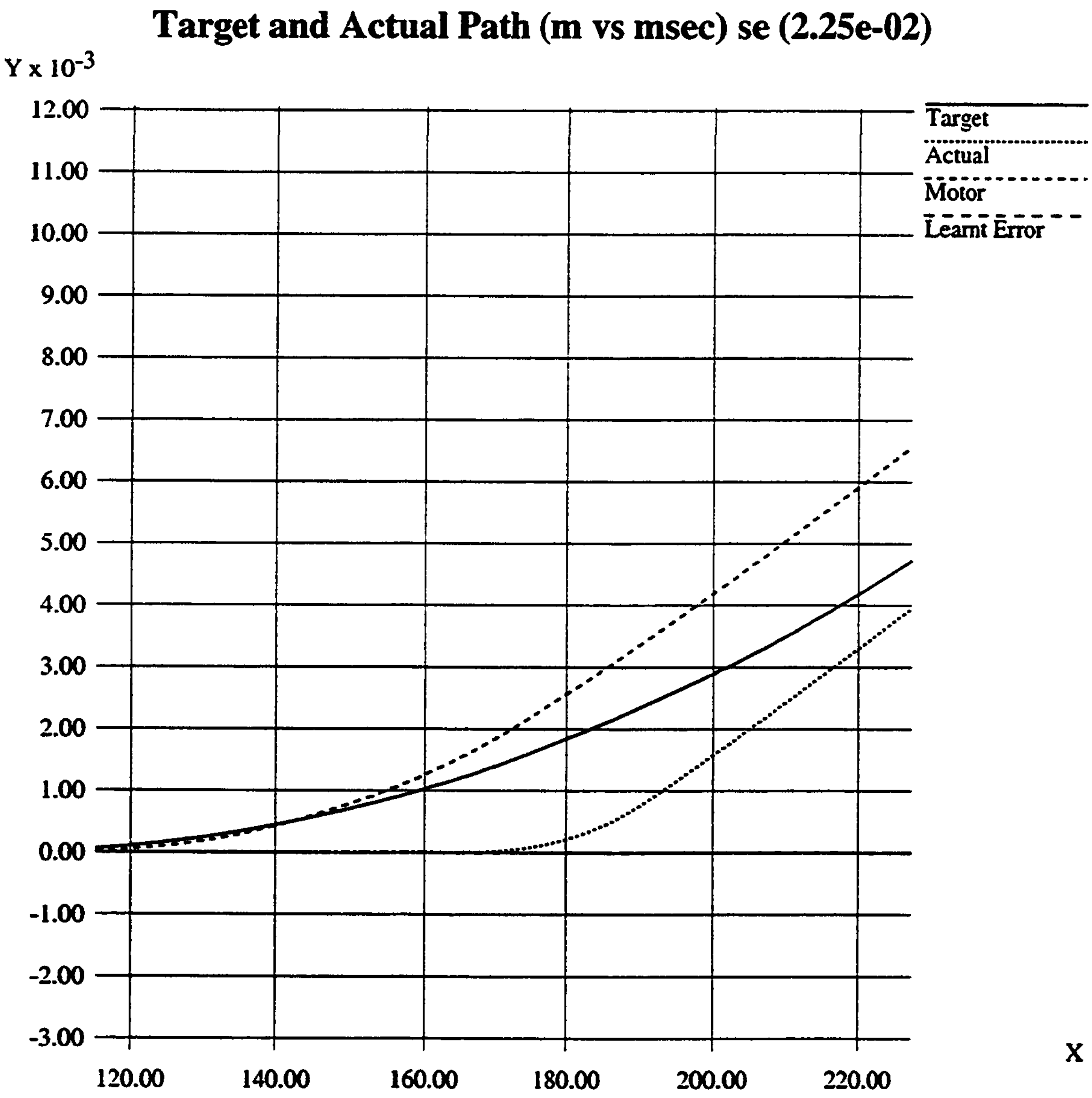


Figure 3.7: System with backlash, and flexible connector.

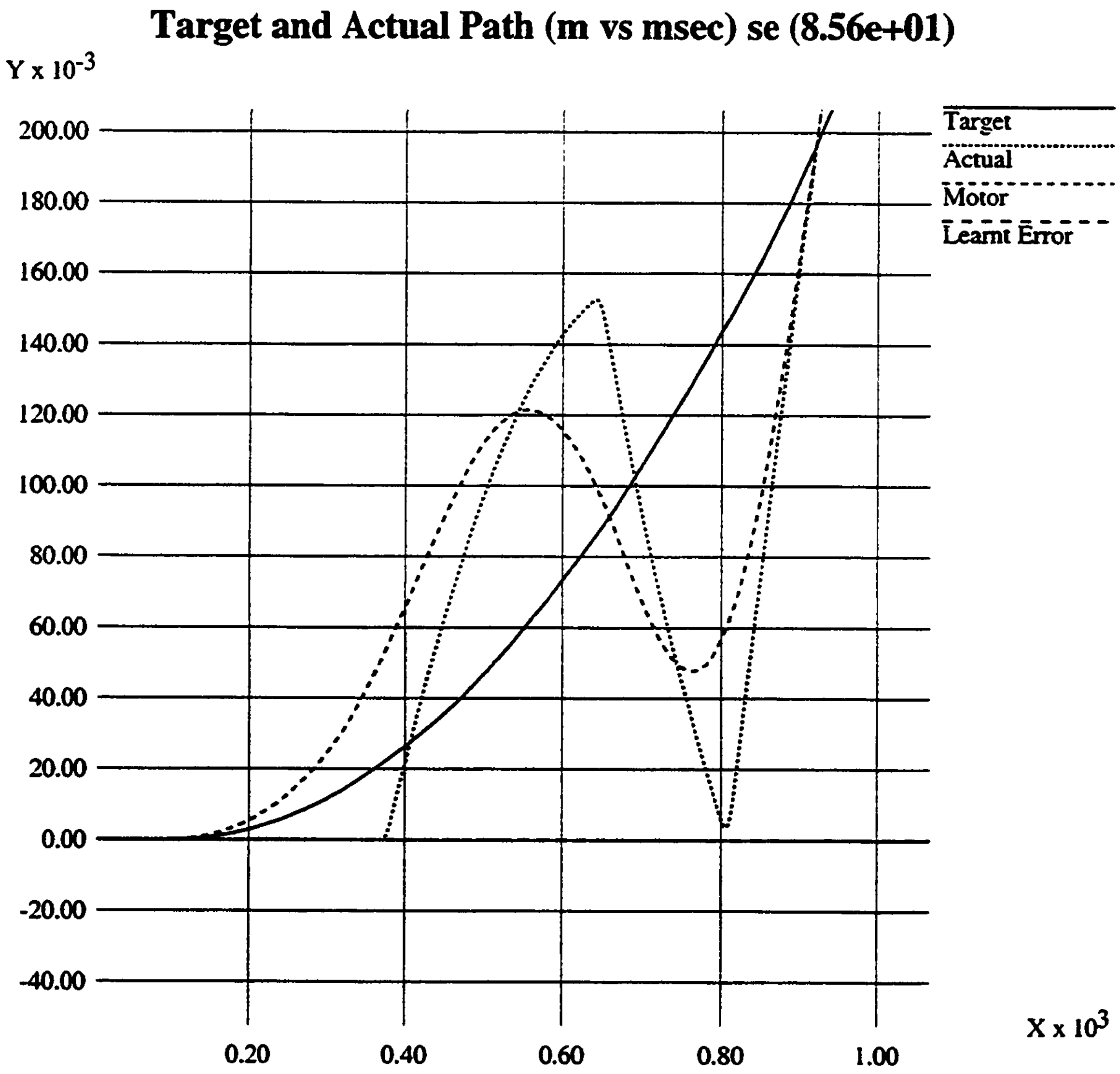


Figure 3.8: System with stiction, backlash, and flexible connector.

feedback from both the table, and the motor was found to improve servo control, and enabled the use of fuzzy control methods to address many system deficiencies. Increased mass reduced the system acceleration capabilities, and lowered system bandwidth. Table mass, was found to aid in rejecting high frequency disturbances. Backlash was found to reduce achievable positioning accuracy, and transmission flexibility reduced the bandwidth of the system. Both effects could be reduced, but not eliminated, through the use of a fuzzy control scheme, and an encoder on both the table, and motor. Velocity proportional friction was found to be largely compensated for using velocity feed forward. Friction could be compensated for using a similar scheme. Stiction however, was found to be much more difficult to address, particularly when the stiction acted on the table, and there was a flexible transmission. Amplifier saturation was found to introduce significant non-linearities. It's effects were found to be most visible in high performance systems. Target paths which exceeded the current capabilities of one or more axes of the system lead to large path error, as did disturbances, no control method was able to compensate for this.

These non-ideal features are found, to some extent, in virtually all servo systems for machine tools, however most can be reduced to levels where they do not limit system performance. This is particularly true of new high servo speed machine tools. Steps taken to reduce these features include improved feedback systems, including high resolution linear optical position sensors on the table, which directly measure table position, and a tachometer on the motor. Also, box-ways are being replaced with roller bearings, or hydrostatic bearings, significantly reducing stiction, as well as other forms of friction. Finally, linear motor based actuators are replacing ball screws, reducing the motor and transmission mass, and significantly increasing transmission stiffness. When all of these improvements are combined plant dynamics are significantly simplified, and performance capabilities are significantly increased. The servo amplifier/motor combination, however remains as a significant system limit, even when all of the best components, and methods are used.

The conclusions from the simulations then are that for a typical plant used in high-performance, high- speed machining, using typical paths, the dominant source of path error is unrealistic performance demands on individual axes, and that the dominant system nonlinearity, which can not be eliminated through careful design is actuator saturation. The unrealistic performance demands consist of excessive system acceleration requirements which exceed the current capabilities of the amplifier/motor combination. Actuator saturation was found to reduce usable loop gains, resulting in lowered system performance. It was found that these limitations were accurately reproduced by a real system consisting simply of a motor and mass.

3.3 Experimental Hardware

Simulation alone however was not believed to be sufficient to prove the validity of any conclusions. It was expected that there would be inaccuracies in the simulation, and unforeseen factors not taken into account. To address this, testing on physical systems was performed. The first test system was a rotary motor and mass which was used while developing the PC based OAC (Open Architecture Controller). With the PC-based OAC functioning well on this platform, a linear motor was purchased, and testing was continued using the linear motor. Finally, to investigate such issues as motor commutation a custom linear motor was constructed which allowed this.

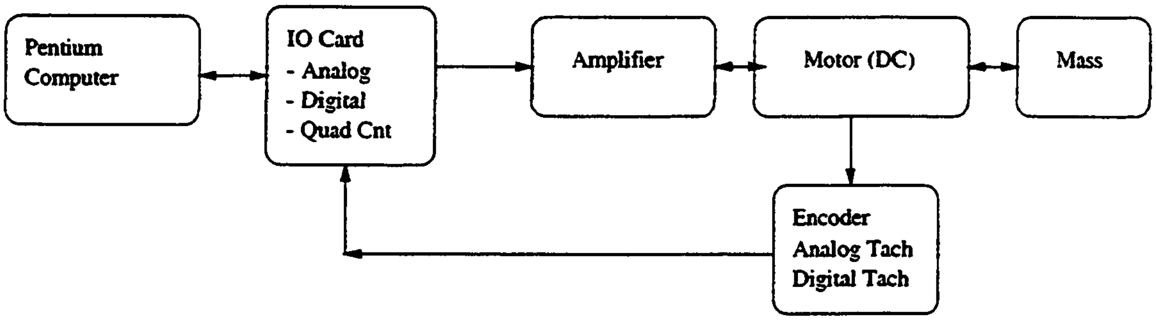


Figure 3.9: Test System Block Diagram

3.3.1 DC Servo Motor

The DC servo motor system was the first physical plant assembled. This consisted of a pancake motor, with integrated tachometer, and encoder, various masses could also be mounted on this motor. The purpose was to investigate control issues that the simulations had indicated were dominant. Specifically, system mass, damping, and amplifier saturation limits. This was a good choice for a first system, in that it clearly was subject to the factors which were found to be dominant, but it was also very resistant to poor control choices, and other errors which occur while developing a control system, and new control algorithms. A block diagram of this test system is shown in figure 3.9, and the performance envelope is shown in figure 4.30.

3.3.2 Linear Motor Test Facilities

The test facilities included two linear motor systems. The first was a commercially available system, on which most of the testing was performed. It was an ironness core, three phase, double sided long secondary, short primary type. It had approximately one meter travel, 1 micron resolution glass scale, and a sinusoidal three phase amplifier. Attached to this was a piezo electric acceleration sensor, and ferraris acceleration sensor.

The second system was constructed to investigate the details of linear motor design. It too was an ironness core, three phase, double sided long secondary, short primary type. Some of the less common features integrated into the system were a fiber composite frame and laminated iron cores to reduce eddy currents, and system inductance. Additionally, both leads of each of the three sets of coils were externally available, and were driven separately by brush DC servo amplifiers. This allowed alternative switching strategies to be investigated. The system had one meter travel and a 1 micron resolution glass scale.

3.3.3 Custom Linear Servo Motor

A custom-designed linear servo motor was constructed on which further, testing could be performed. A The details of the design are presented in appendix A. The construction of a linear motor provided a fist hand opportunity to investigate various design issues. Included in this is the motor frame, the motor core and the control of the coils.

The frame of the motor is composed of epoxy resin Glass Fiber Reinforced Composite. The GFRC has high mechanical strength while exhibiting excellent electrical properties and heat resistance. This material is non-conductive and therefore prevents losses associated with induced fields in the frame. Another design feature that minimizes unwanted field losses is in the construction of the two iron cores. Two arrays of magnets each sit upon a continuous laminated iron core The core was made with thin pieces of iron (approximately 0.015 inches thick) bonded together to form a block through which a magnetic field flows easily in one direction while current is blocked in the orthogonal direction by insulation, thus reducing the losses due to induced eddy currents. Energy lost in eddy currents in the core is of a concern for a number of reasons. One reason is the possibility of causing thermal deformation of the structure of the machine. The second perhaps more significant reason is that due to this inefficiency, the amplifier will have to drive the coils harder (resulting in more heat) to achieve a given level of transient performance (jerk).

Care must also be taken in selecting the gauge of wire used in the coils, due to the phenomenon known as the skin effect. The skin effect is the tendency for current to be restricted to the outer layers of a conductor at high frequencies. The result of this is that a conductor may exhibit a higher resistance to alternating currents than to direct currents. This is an issue that must be addressed in linear servo motors. The reason for this is to minimize the heat generated in the coil. The relationship between skin depth and frequency is expressed in formula 3.8. For copper K is 1. at 10 kHz, the skin depth is 0.0661 cm, and at 20 kHz it is 0.0467 cm. Given the current servo loop update rates of 10 kHz, the desire to increase this update frequency for high speed control, as well as the higher transients resulting from the switching of the amplifier, the coils should be composed of wire no thicker than 16 gauge, and preferable thinner than 20 gauge wire. The coils used in the custom servo motor were 23 gauge.

$$depth_{incm} = (\frac{6.61}{f^{\frac{1}{2}}}) * K \tag{3.8}$$

$$K = (\frac{1}{\mu r} * \frac{\rho}{c})^{\frac{1}{2}} \tag{3.9}$$

where

μr = relative permeability of conductor

ρ =resistivity of conductor

c =resistivity of coper at $20 \deg C$ (1.724 micro-ohm-centimeter)

3.4 Validation of Simulation Conclusions

A general block diagram of a position servo system, including amplifier saturation (neglecting such factors as backlash, friction, sensor quantization, delays, hysteresis, mechanical imperfections etc...) is given in figure 3.10. Neglecting the amplifier saturation, and motor back emf (the small signal assumptions), the transfer function for this block diagram becomes as shown in equation 3.10. This transfer function is already complicated, and it does not account for amplifier saturation, a critical component in servo control. A simpler model, that captures the dominant plant characteristics is required.

The results of the simulation indicated that the model of the mechanical system for a high servo performance machine tool (figure 3.11 equation 3.11), could practically be replaced by a simpler lumped mass model as shown in figure 3.12 and equation 3.12. The reason being that the stiffness of the connection (K) between M1 and M2 (or motor and table) is high. It also indicated that the most significant source of path error was due to large signal effects.

$$\frac{x(s)}{I_c(s)} = \frac{K_i}{s \cdot I + R + K_i} \cdot \frac{\frac{1}{M_1+M_2}}{\frac{M_1 \cdot M_2}{K_s \cdot (M_1+M_2)} \cdot s^4 + \frac{M_1 \cdot D}{K_s \cdot (M_1+M_2)} \cdot s^3 + s \cdot (s + \frac{D}{M_1+M_2})} \tag{3.10}$$

3.4.1 Mechanical System Investigation

The benefit of a linear motor driven machine tool over conventional ball screw driven machine is that the flexible ball screw is replaced by a stiff direct drive linear motor. A general model is presented in figure 3.11. The transfer function of this system is given in equation(3.11). The assumption is that the stiffness of the connecting spring K is sufficiently large that the system can practically be represented as shown in figure 3.12. The transfer function then becomes as given in equation(3.12).

$$\frac{x(s)}{F(s)} = \frac{\frac{1}{M_1+M_2}}{\frac{M_1 \cdot M_2}{K_s \cdot (M_1+M_2)} \cdot s^4 + \frac{M_1 \cdot D}{K_s \cdot (M_1+M_2)} \cdot s^3 + s \cdot (s + \frac{D}{M_1+M_2})} \tag{3.11}$$

$$\frac{x(s)}{F(s)} = \frac{\frac{1}{M_1+M_2}}{s \cdot (s + \frac{D}{M_1+M_2})} \tag{3.12}$$

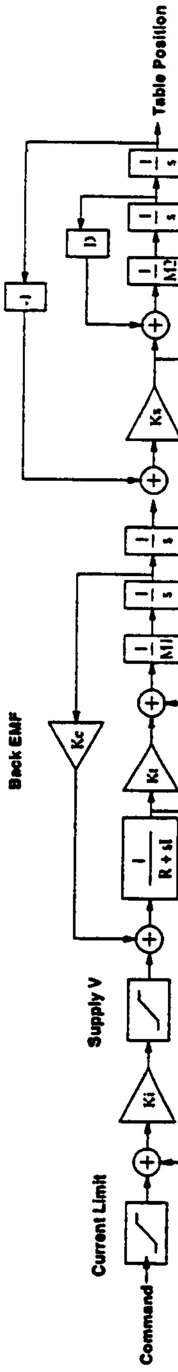


Figure 3.10: General block diagram of servo system.

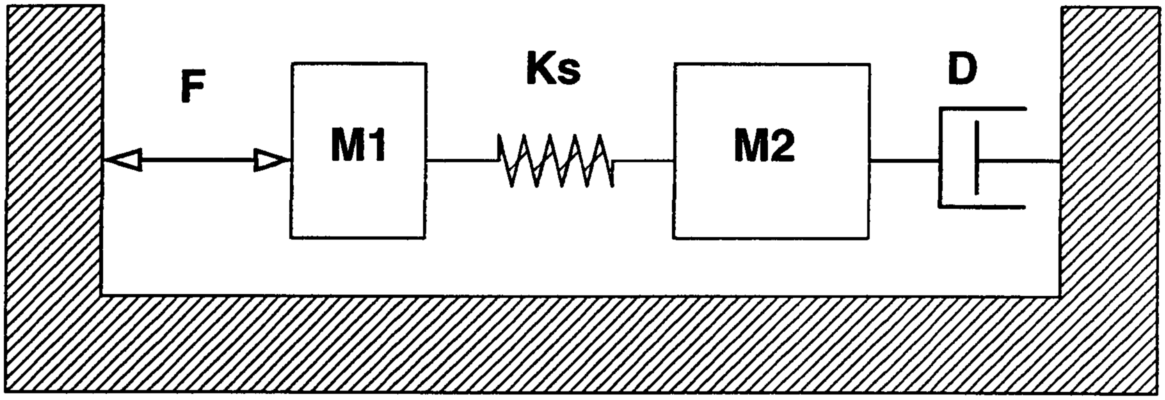


Figure 3.11: General diagram of mechanical axis.

Where:

D is the damping

F is the force generated by the motor

I is the inductance of the motor

I_c is the commanded current

K_e is the back emf gain

K_i is the voltage gain

K_s is the spring constant

K_t is the force gain

M_1 is the equivalent moving motor mass

M_2 is the table mass

R is the resistance of the motor windings

x is the position of the table

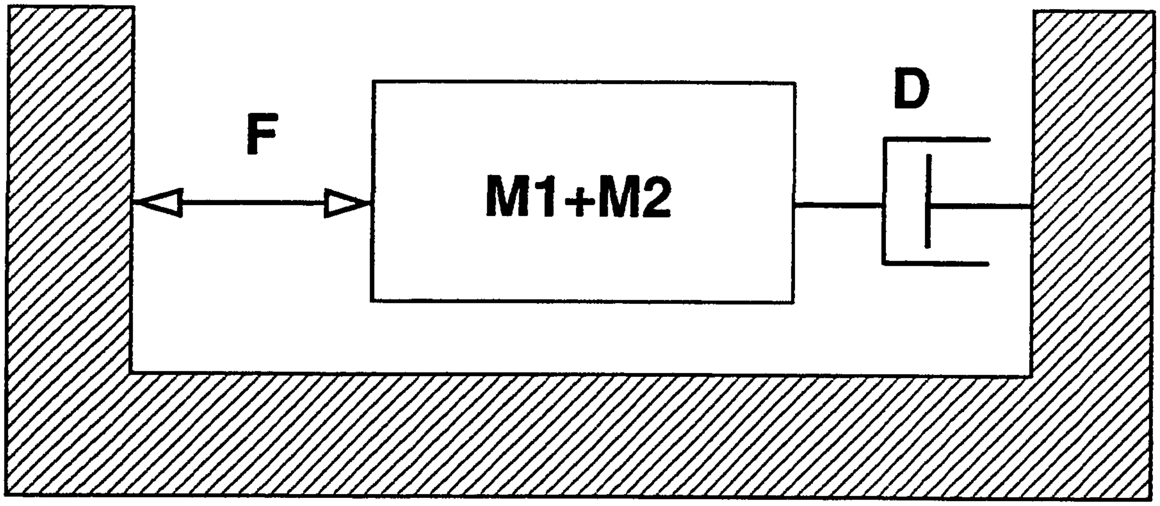


Figure 3.12: Diagram of ideal linear motors.

3.4.2 Small Signal, Current Driving Amplifier and Motor Transfer Function

The assumption made for the amplifier and motor is that the force exerted by the motor is directly proportional to the amplifier command. Ignoring supply voltage limits, current limits, and back emf (the small signal or linear control assumptions), the small signal transfer function from current command to force generated is given in equation(3.13). Assuming K_i is large, the force exerted by the motor is the current command multiplied by the torque, or force gain for the motor.

$$\frac{F(s)}{I_c(s)} = \frac{K_i}{s \cdot I + R + K_i} \cdot K_t \tag{3.13}$$

3.4.3 Validation of Mechanical, and Small Signal Models

In order to test the validity of these models, the transfer function from the controller command to the output of a commercial linear motor test stand was measured. This test was performed open loop, and a slow sinusoidal sweep was used as the set-point for the current command. The assumptions made by these plant models include that there are no significant plant dynamics. To measure the transfer function, the plant was subjected to a slow sweep of command current. The resulting gain, and phase lags were recorded. The first plant output measured was the position. The results were dominated by the double integrator, $\frac{1}{s^2}$. The velocity signal was then examined from the digital tachometer, the expected transfer function being $\frac{\dot{x}(s)}{I_c(s)} = \frac{C_1}{s+C_2}$. This also matched the expected plant very closely, up to the frequencies that could be measured. (While the digital tachometer used does not have a bandwidth limitation, it does have a resolution floor, or minimum required amount of position movement.) With the amount of energy used in the tests, the sensor was not moving sufficiently at high frequencies to accurately measure the velocity. The final measurement was the acceleration of the plant (figure 3.13). The results of this showed that up to 500 Hz the transfer function of system was reasonably constant. After this point there were resonances, and the gain decreased. This is mostly due to the finite stiffness of the attachment of the primary to the table, jerk limitations resulting from the inductance of the coils, and the limited supply voltage.

3.4.4 Non-Linear and Large Signal Plant Models

Machine tools are primarily concerned with position error. The transfer function from current command to position is a second order low pass filter. As such, we

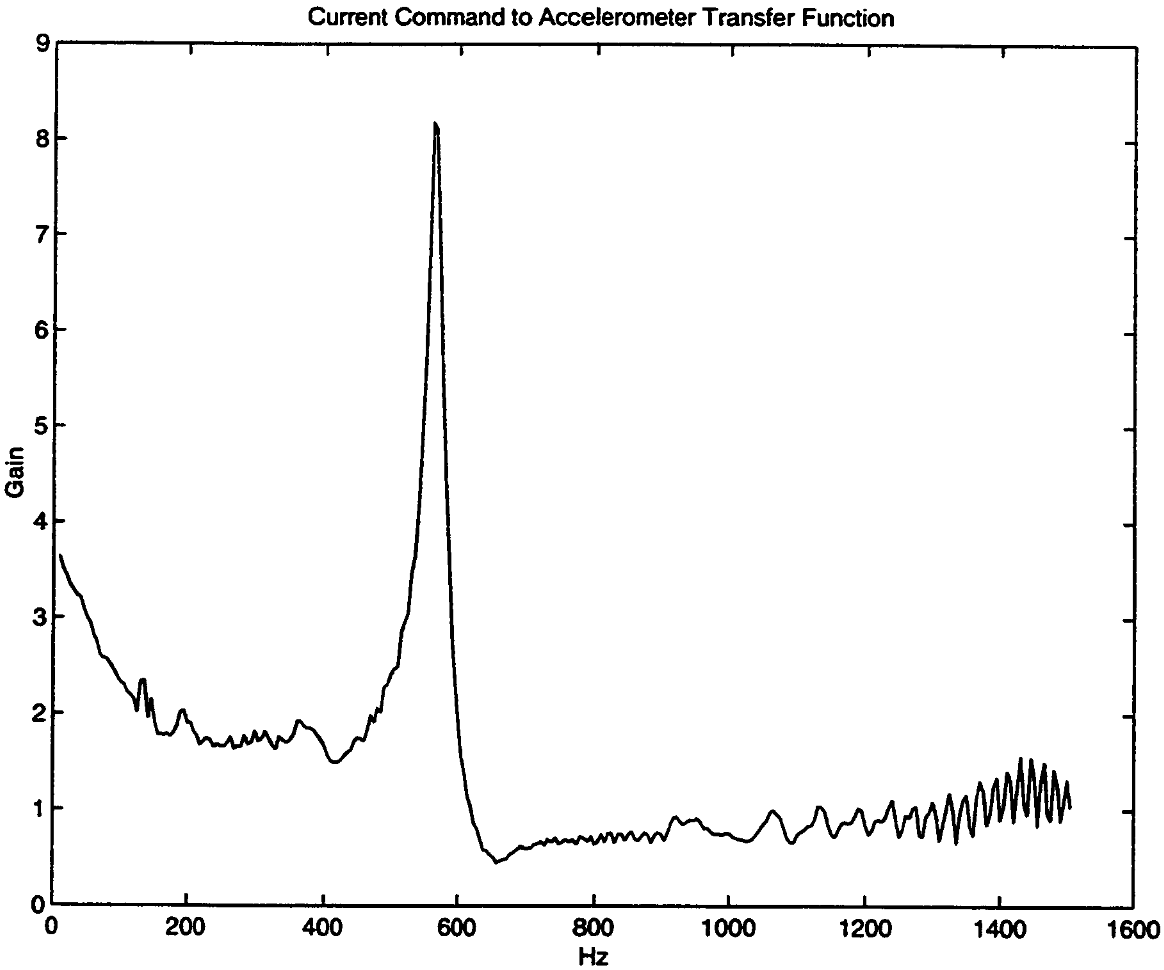


Figure 3.13: Linear Motor Current Command to Acceleration Transfer Function

are most concerned with modeling errors at low frequencies. Our assumptions are that the current limit and voltage limits are not exceeded. The voltage supply is usually chosen large enough that it mainly limits system jerk, and has little effect on low frequency modeling errors. The effect of the current limit on low frequency performance is difficult to describe using Laplace transforms, however the effects can easily be measured by applying full command forward, and backward and recording the resulting acceleration with respect to velocity (figure 3.14). This data describes the acceleration limits of the system with respect to velocity, or the performance envelope of the machine tool.

The conclusion from this investigation, is that $\frac{x(s)}{I_c(s)} = \frac{C_1}{s(s+C_2)}$ accurately represents the dynamics of the tested linear motor up to approximately 400 Hz. The low frequency, large signal plant limitations can accurately be characterized by a simple performance envelope test, as described. When combined, this information enables accurate predictions of plant performance, over the frequency and amplitude ranges of interest in machining.

3.4.5 Dynamic Stiffness

An important observation is that the plant is a second order low pass filter, in terms of force disturbances to position. If we ignore the resonance at 600 Hz, for a moment, we can determine the dynamic stiffness of the plant alone as a function of frequency. This is important, as it determines the frequencies that the controller will have to provide stiffness at, and at which frequencies the plant will be sufficiently stiff on it's own.

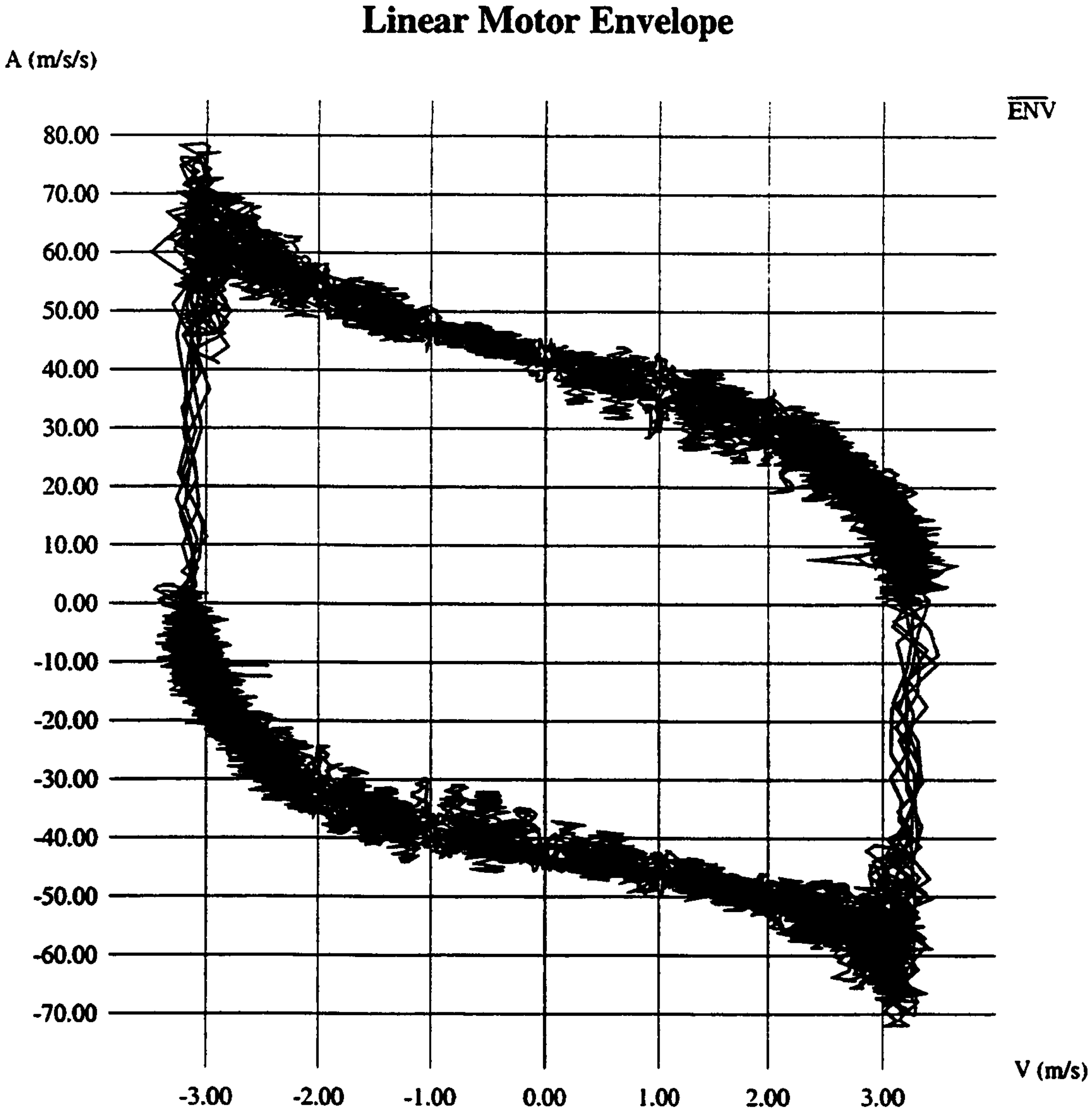


Figure 3.14: Performance Envelop of Linear Motor

$$x(t) = C \cdot sin(2\pi f \cdot t) \tag{3.14}$$

$$\dot{x}(t) = 2\pi f \cdot C \cdot cos(2\pi f \cdot t) \tag{3.15}$$

$$\ddot{x}(t) = -(2\pi f)^2 \cdot C \cdot sin(2\pi f \cdot t) \tag{3.16}$$

$$F(t) = m \cdot \ddot{x}(t) \tag{3.17}$$

Where:

f	is frequency in Hz
x	is position
$\dot{x}$	is velocity
$\ddot{x}$	is acceleration

For a given maximum position error C, the ratio F/m $(force/mass)$ required as a function of frequency can be calculated. A plot of various values of maximum position error C is shown in figure 3.15. This plot shows that for reasonable ratios of F/m the table alone will provide the required dynamic stiffness after a frequency of approximately 100 Hz.

3.5 PC-Based OAC

The motivation for the development of a PC-based OAC included flexibility, cost savings, and improved performance. Other benefits of the implementation include a clear, easy upgrade path. Specifically, the interest, for the purposes of this thesis, was to develop openness at the bottom level of OAC. This is in contrast to the bulk of the effort that has been exerted in such well known OAC projects as MOSAIC, NGC and OSACA which conceded this area to custom low level controllers. The second reason for focusing on this area is that it compliments the OAC work which has been completed to date in the lab, which addresses high and medium level OAC requirements.

While there has been some evolution of the implementation of the PC-based OAC with soft motion controller concept, the basics have remained consistent. The idea was to use as little custom hardware as necessary to interface to the machine, and perform all "control" on a PC. This is very different from what is traditionally done, which is to develop custom machine tool control hardware.

The largest advantage from a researchers perspective, or potentially someone producing custom machinery is the flexibility. As the system is currently implemented, running RT-Linux with loadable kernel modules, all development can be performed with out burning an eprom, or plugging in a down-load cable. While this may not seem like a significant advantage, a complete cycle of stopping the test machine, unloading the kernel test code, editing the offending source code, recompiling, reloading, and restarting the test machine can take well under 60 seconds (assuming that there was a small error in the source code). Also, C, and C++ can be used in the kernel code developed, and this is compiled with gcc or g++ (Gnu C, or C++). The use of

C, and C++ allows the developer to use other code, often without modification, again improving development time and confidence. Because the system is based on Unix, and has all of the standard facilities of Unix, there is no need to be near the controller to develop code, test, or perform diagnostics or maintenance. Prudence would advise against un-supervised remote development, or modifications in a real-life situation. However these abilities (given appropriate local supervision) could greatly reduce down times, service times, and the cost of modifications, especially for custom equipment. The fact that the facilities available on computers running Unix make up the backbone of todays networks provides the additional advantage that the machine controllers can now, on their own, participate in data collection, and presentation. Possible uses of this data include maintenance, process monitoring and accounting. When combined these factors alone provide an impressive argument for the PC Based OAC with soft motion controller concept. Figure 3.16 shows a sample screen developed early in this project which was created to allow remote modification of servo loop parameters, (from anywhere in the world) through Netscape.

The second area in which there are significant advantages are in the implementation. The PC-Based OAC with soft motion controller requires little hardware, and none if it has to be "custom" manufactured. Specifically, the hardware requirements are a computer, and an IO solution. The IO solution usually includes analog and digital IO, and quadrature encoder counters. There are currently several manufactures which provide the appropriate electronics. There are also other options available, including "smart drives" connected through serial links. The end result is that the user is now in control of the hardware, not the manufacturer of the controller. The user also benefits from the competition in the PC manufacturing market, as well as the spectrum of products available. Specifically, the user could buy a 486 for a very low performance cost critical application, or a brand new 600 MHz PIII, again adding options and flexibility. Also, these decisions don't have to be made at the beginning of the development cycle, they can be updated as the products ship, buying the appropriate computing platform, at the best price available. Perhaps more importantly, there is no need to make lifetime buys, once the software is developed, as there is a high degree of certainty that there will be a smooth upgrade path, and available product. Compare this to the effort and costs incurred by the developer that attempts to design manufacture and maintain his own controller hardware and development software.

3.5.1 Implementation

The system was implemented on a networked 200MHz Pentium Pro with 64 MB of memory, and a 4 GB hard-drive. The IO consisted of 8, 13 bit channels of analog input and output, and 24 configurable channels of digital IO. Additionally a single digital tachometer was constructed.

The software consists of the bulk of the implementation effort. It is currently running on RT-Linux. RT-Linux is an extension to Linux which provides facilities to run real-time tasks. It is implemented as a small hard-real-time kernel which runs standard Linux as a task. The result is the ability to run hard real-time tasks, with the standard benefits of a Unix Workstation, on a PC platform. While the real-time kernel, and the standard Linux kernel share memory space, and shared memory communications are possible between Linux processes and RT-Linux processes, the recommended, and standard method of communicating is through communication channels, or real time fifos. Using this as a basis for developing PC Based OAC with soft motion control provides a significant advantage, and time savings, when compared to trying to develop ones own operating environment. It also has the advantage of being free and open-source, which is a significant advantage when compared to many other proprietary real-time OS's, and agrees with the philosophy of OAC.

As the focus of this thesis is on servo control, most of the work and effort has been focused on the elements required to implement this. However the global plan can be seen in figure 3.17.

The real-time portion of the controller resides in kernel space, where there are few of the standard Unix operating safeguards. While this provides the developer maximum flexibility, it also requires ensuring the safety of man and machine. This is accomplished through software and hardware "watchdog" facilities. To date this combination has functioned exceptionally. In addition to this every effort has been made to provide the user with the ability to accomplish routine tasks in a "safe" environment where errors would be caught before any damage could occur. Examples of this would be entering invalid values for a PID control loop, the machine would be stopped as soon as position, or velocity error exceeded a specified threshold.

The real-time portion of the controller is divided up into several components, these include hardware abstraction, watch dog, servo loop, binary communication and instruction ques, ASCII or human readable communication and instruction ques as well as other special purpose functions which require real-time execution with minimal over head, called PLCs.

The purpose of the hardware abstraction layer is to provide a consistent interface to a spectrum of interface hardware in-order to simplify development and maintenance. This layer includes standard "abstracted" devices, such as analog IO, and digital IO. Position feed back is handled as high resolution analog feedback. These analog and digital values are stored in specific common variables which can be accessed by higher level routines, as well lower level routines which actually input or output the data. The low level routines are separated into two an input and output routine. The input routine runs before the main body of each "loop closure" or time step. It transfers the data either through DMA, reads, and writes from the IO hardware to shared variables. Other IO possibilities exist such as serial communications (which has been implemented), however this runs as a separate real time task, which updates data in a similar fashion, but at a much slower rate. Once the input routine is complete the body of the "loop" is called. Once these functions are complete, the output routine is called which similarly transfers the data out.

The body of the loop includes servo control, a binary command queue server and PLC's. The behavior of the servo loop can be changed by modifying variables, through the human readable ASCII interface, or new capabilities can be added off-line by altering the code. Currently there is provision for PID control with feed-forward, generalized predictive control (GPC), minimum-time tracking control (MTTC) and disturbance observers. The servo loop is currently updated at 10KHZ for PID and MTTC, and up to 5KHZ for GPC. Most of the GPC servo loop is implemented in C++ using a previously written array manipulation library. Substantially increasing the servo loop rates for the PID, and MTTC controllers will require migrating the IO hardware from the ISA bus to the PCI bus. The binary command queues are implemented as a concession to the high through put requirements of high speed, multi-axis machining. This interface implements essentially N symmetric interfaces. This means that two or more independent sets of actions could be proceeding simultaneously. One example is controlling two independent robot arms. The commands available are currently limited to those required for performing the required servo tests, however adding commands is straight forward. These implemented commands deal with drip feeding position, velocity and acceleration data, as well as recording data for analysis. Currently, data can be drip fed and recorded without issue at the 10kHz servo update rates. Other PLCs would include items which must occur quickly, with minimal overhead, or need to be isolated from users for safety reasons. Examples include the phase control of the coils of the custom manufactured linear motor. A PLC function is in charge of monitoring the encoder position and command signal generated by the servo loop, and determining the polarity and how much current should be applied to each coil. Another PLC is used to generate the capability envelope of an axis.

The watch dog function has been made as rigorous as possible to ensure the safety of the user, and equipment. It includes limits on absolute position, and velocity, limits on the differences between target and actual position and velocity, as well as command limiting.

The human readable ASCII interface is designed to provide maximum flexibility for the bulk of the OAC functions, which are not particularly time critical. The most significant feature of this interface is the ability to examine and modify (if allowed) essentially all of the variables used by the real time code. A scanner, and automatic data description program has been written which examines data definitions placed in a specific file “typedefs.h”, and creates a database of the internal variables, as well as other properties such as their names, and how to reference the data. Future work will include adding the ability to specify limits on the variable, and how the user should be able to access it.

The only limitations on these features are that the intermediate data can not exceed a specified size, currently 4000 characters. It is possible to eliminate this restriction however it has not proven necessary. Other improvements will be to perform the same type of operations on the "defines.h" file, so that the user need not remember the constants to set flags to in order to accomplish a desired task. An example of this would be switching between PID, and GPC servo control loops.

At present other logical operations and flow control are performed in user space, however the addition of other basic capabilities such as "if then else", "for loops", user defined data types, variables, and functions, are planned as future work.

3.6 Advanced Feedback

The purpose of a feedback control system is to compensate for unsatisfactory open loop performance. Machine tools depend on feedback systems to compensate for mechanical, and electrical variation, as well as external disturbances. High performance, precise control of machine tools depends on the feedback systems used.

Early motivations for developing this feedback theory included developing accurate repeaters for the telephone industry. It was found that the mass produced vacuum tubes had too much variation in amplifier gain. The only economical solution was to create a closed loop system. This had the effect of significantly reducing the overall system sensitivity to variation in the forward path (the expensive part), however the system was now just as sensitive to variation in the return path (see figure 3.18).

The relation is as defined as follows:

$$S_{\alpha}^{G} = \lim_{\alpha \to 0} \frac{\frac{\Delta G}{G}}{\frac{\Delta \alpha}{\alpha}} \tag{3.18}$$

G is the forward loop transfer function.

α is a parameter in G.

This is the percentage change in G resulting from a small change in α, or the open loop sensitivity to α.

The sensitivity of the closed loop transfer function T to variations in α, is given as:

$$S_{\alpha}^{T} = \frac{1}{1+GH} * S_{\alpha}^{G} \tag{3.19}$$

This shows that the feedback has reduced the close loop sensitivity T to variations in α by $\frac{1}{1+GH}$. The sensitivity of the closed loop transfer function to variations in parameters in the feedback path however is not reduced (assuming reasonable open loop gains).

$$S_{\alpha}^{T} = -S_{\alpha}^{H} * \frac{GH}{1+GH} \tag{3.20}$$

This shows that at reasonably large loop gains, S_{α}^{T} is not notably less than S_{α}^{H} [47].

3.6.1 Digital Tachometer

There are two ways to estimate velocity from an incremental encoder. The first is to count how many counts occur in a given period. This consept is outlined in figure

3.19. Ignore the line labeled "Edge Prior to Sample Time" in this figure, as this is used for the digital tachometer. Based on the two channels of the encoder (channels A, and B) the counting circuitry identifies when to increment the counter, this is identified by the line labeled "Count UP". If we assume that the servo loops are closed at periods labeled "Sample Time" then we would get estimates proportional to the line labeled "Counts per Sample Period" for the velocity. As shown in this example these estimates are imprecise, toggling between one and two. The concept of the digital tachometer is to add a high frequency clock to the system, to record the time between pulses, and to use this extra information to generate a more precise velocity estimate. As implemented for testing, the digital tachometer records the time of the last edge before the "Sample Time" and the number of counts that have occurred since the last "Sample Time". It is a simple matter then to estimate the velocity as $\frac{counts}{timerticks} * C$. The results of these two methods are shown in figures 3.20 and 3.21.

3.6.2 Acceleration Feedback - Piezo Electric and Ferraris Sensor

Acceleration feedback is an important source of feedback, particularly in plant modeling, and disturbance estimation. Two types of acceleration feedback were investigated, a piezo-electric accelerometer, and a ferraris sensor.

A piezo-electric accelerometer consists of a mass, attached to a piezo-electric material attached to an object to be measured. When the object being measured accelerates, the piezo electric material is either compressed between the measured object and the mass, or it is placed in tension, this produces a charge, which is amplified, and represents the acceleration of the object being measured, relative to the small mass. This has two problems for linear motors. The first is that the amplifier has a finite impedance, which drains the charge generated by the piezo-electric material, making it difficult to measure low frequencies, and impossible to measure DC acceleration. The second problem is that this measure of acceleration is absolute, not relative to the bed of the table. This presents problems for traditional orthogonal axis machine tools. If the machine is on a floating bed, the measurement will, in effect be filtered because of the movement of the machine. If the machine is firmly anchored, external disturbances will be measured and affect performance. The problem is likely to be worst for machines like a hexapod, as gravity, and the changing orientation of the motor will conspire to generate incorrect readings. A practical solution to this would be to use a second accelerometer to measure the movement of the bed of the axis and estimate the acceleration of the axis as the difference of the two. This was not required for the test system.

A ferraris sensor measures the relative acceleration between the sensor and a bar of aluminum. The sensor is mounted to the table, and the bar runs the length of travel. The sensor consists of permanent magnets, and one or more coils. As the sensor moves, relative to the aluminum, the permanent magnets induce eddy currents in the aluminum, these eddy currents produce magnetic fields, the magnetic fields pass through the coils, and generate a potential proportional to the rate of acceleration between the sensor and the aluminum. Drawbacks include susceptibility to interference by magnetic fields and lower bandwidth than piezo electric accelerometers. However it is very well suited to measure the low frequencies that the piezo-electric accelerometer can not, or is ill suited for.

3.7 Switching Strategies for a Permanent Magnet Linear Motor

Typically, the current in the coils of a permanent magnet linear motor is characterized by a sinusoidal, 3-phase arrangement, where each phase of the current is 120 degrees out of phase with the next. The linear motor produced in the lab; however, allowed different switching strategies to be investigated as the current in each coil was under computer control. In order to investigate this, both leads of each set of coils, had to be accessible to the amplifiers. Each set of coils was then driven by a separate DC motor amplifier. This allowed the PC-Based OAC to optimally control each motor winding.

The main element in this optimization is the average magnetic field intensity over the coil. A major limitation of linear motors, is the ability to dissipate heat from the coils. By powering only the set of coils that is in the most intense magnetic field, the maximum continuous force of the motor can be increased. This can be seen in figure 3.23 and 3.24. Figure 3.23 shows the case where coils A, and C are each at the fringes of the magnetic field. In the 3-phase case, if coil B were driven with full current in the clockwise direction, coils A, and C would be driven with half current each. (The advantage of three phase circuits is that the current in the three lines sums to zero, eliminating the need for neutral wires. Additionally less power electronics are required.) Assuming the magnetic field intensity is approximately sinusoidal the same force could be achieved by increasing the current in coil B by 50% and eliminating the current in coils A, and C. This would reduce the heat generated by 25%. (The heat is generated due to the internal resistance of the windings.) The drawbacks of this strategy are the increased power electronics required (approximately double), and the number of power leads required to drive the motor. Figure 3.24 shows the case where coils A, and B would each be at 86% power, and coil C would now be off. No improvement can be made in this case, and it is now our heat limiting case. Given that the motor's continuous force was limited by the first case with three phase control, and is now limited by the second case with individual control of the coils, there would be an approximately 18% increase in continuous force. This example assumes sinusoidal magnetic field intensity, if a uniform magnetic field is assumed between the magnets, this increase could be as much as 33%. This commutation strategy also has the benefit of reducing the inductance seen by the amplifier, increasing the jerk capabilities of the system.

3.8 Summary

A detailed time domain simulation of a machine tool and control system was first performed in an effort to better understand the issues in machine tool control. A time domain simulation was performed instead of using closed form methods, or approximations, due to the complexity of the system, and the desire to not simplify elements of the system before they were investigated. Based on this simulation, the dominant system elements were determined. These were the mass of the system, the velocity proportional damping, and the current limits of the amplifier. A physical plant consisting of a rotary motor, and mass, and a control system was then constructed, on which initial simulation conclusions could be verified, further testing could be performed, the control system could be developed to a stable state, and initial control concepts could be generated. With initial conclusions verified, and a suitable control system developed, a linear motor test stand was assembled for further testing. A linear motor test stand was chosen because linear motor driven machine tools represent the state of the art in high performance machine tools. A parametric analysis of this system was performed in an effort to further validate the conclusions of the simulations, and of the simple test stand. This analysis also provided a method to represent the dominant characteristics of the system. Specifically, by using a simple parametric model, and a large signal performance envelope, the dominant characteristics of the system were represented. A further analysis of the system was then performed to determine the disturbance rejection properties of the plant alone. Specifically, the plant is a second order low pass filter. While the plant has no stiffness at DC, as the frequency of the disturbance increases, it soon provides the required system stiffness without the intervention of the controller. This observation is important as it provides a guide to the frequency range that the controller must provide stiffness at. While this frequency will vary depending on the magnitude of the disturbance, the mass of the table, and the tolerances, the analysis indicated that most systems should reject disturbances on their own by approximately 100Hz. The importance of accurate feedback was then examined, including options for improving feedback. Improved velocity feedback, as well as acceleration feedback were examined. It was found that by using a digital tachometer, velocity feedback could be considerably improved. It was also found that the ferraris sensor provided good low frequency feedback, however accelerometers provided better high frequency feedback. Ferraris sensors have the advantage that they provide feedback relative to the bed of the machine, while piezo actuators do not. Finally a linear motor was constructed in order to test such things as increasing holding force through alternate commutation strategies. It was found that indeed it was possible to increase the holding force by approximately 18% through these commutation strategies.

Parameter	Value	description
noise_t_type	1	White
noise_t_A	1.000000e-04	Amplitude
amplifier_t_type	1	Current
amplifier_t_max_voltage	1.700000e+02	Volts
amplifier_t_max_current	2.500000e+01	Amps
motor_t_resistance	2.000000e-01	ohms
motor_t_K_ti	9.800000e+00	N m / amp
motor_t_K_emf	1.000000e-02	V / rev / sec
motor_t_c (damping)	2.000000e-04	N m / rev / sec
motor_t_r (friction)	1.000000e-02	N m
motor_t_s (stiction)	2.000000e-04	N m
motor_t_sv (stiction decays to 0 by)	1.000000e+00	rev/sec
motor_t_m	3.000000e-03	Kg m$\hat{2}$
connection_t_play	5.000000e-06	m
connection_t_K	1.000000e+09	N/m
connection_t_gear	3.420000e+02	rev/m
table_t_c (damping)	2.000000e+00	N/m/sec
table_t_r (friction)	8.000000e+00	N
table_t_s (stiction)	2.000000e-01	N
table_t_sv (stiction decays to 0 by)	5.000000e-01	m/sec
table_t_m	3.000000e+02	Kg
feed_back_data_t_table_velocity_noise	0.000000e+00	
feed_back_data_t_table_acceleration_noise	0.000000e+00	
feed_back_data_t_table_velocity_quant	0.000000e+00	
feed_back_data_t_table_acceleration_quant	0.000000e+00	
feed_back_data_t_motor_velocity_noise	0.000000e+00	
feed_back_data_t_motor_acceleration_noise	0.000000e+00	
feed_back_data_t_motor_velocity_quant	0.000000e+00	
feed_back_data_t_motor_acceleration_quant	0.000000e+00	
system_dt	1.000000e-03	sec/control loop
system_N	20	basic cycles / loop
system_quant	1.000000e-01	granularity of D/A

Table 3.1: Sample Simulation Configuration Parameters

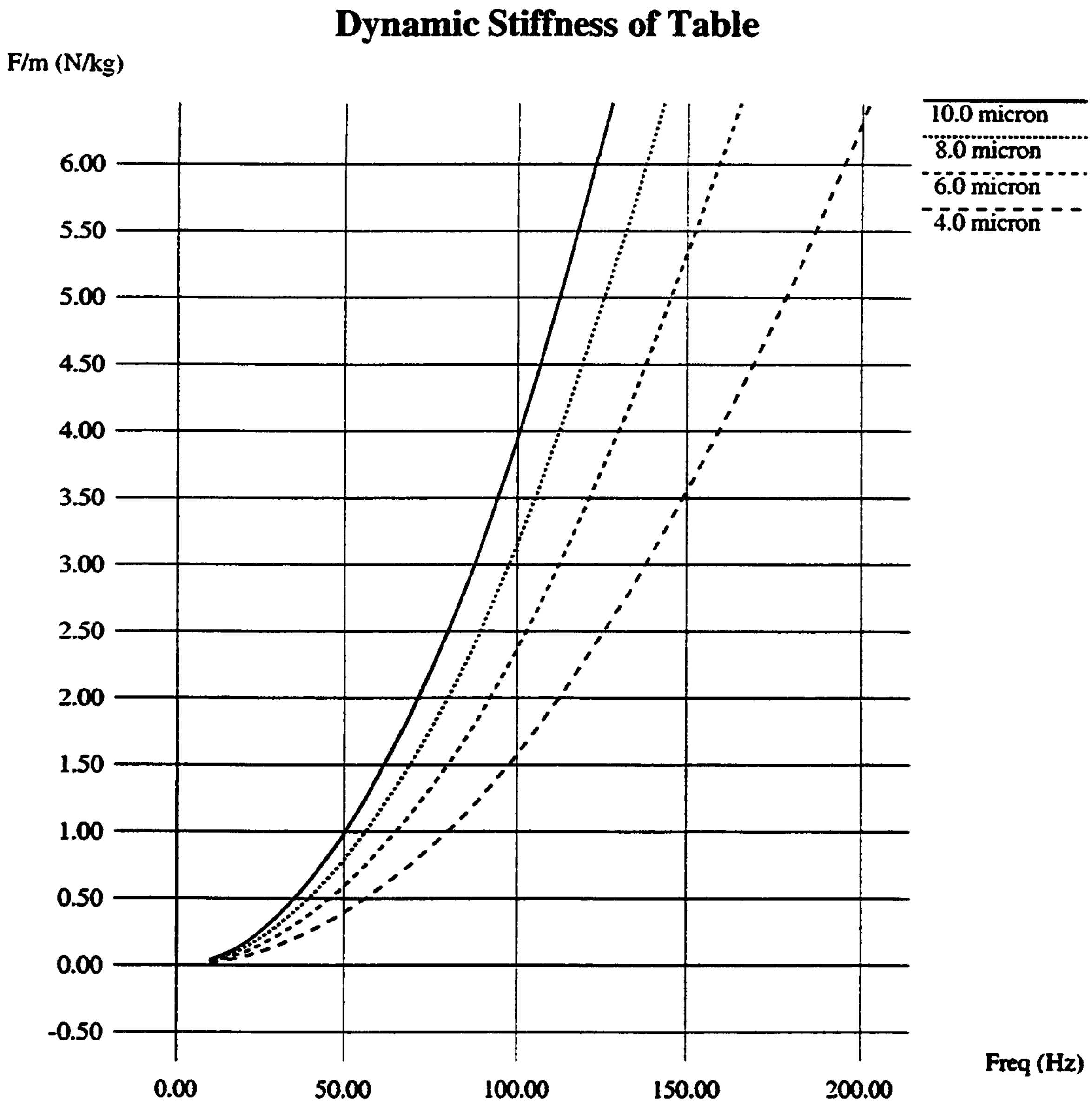
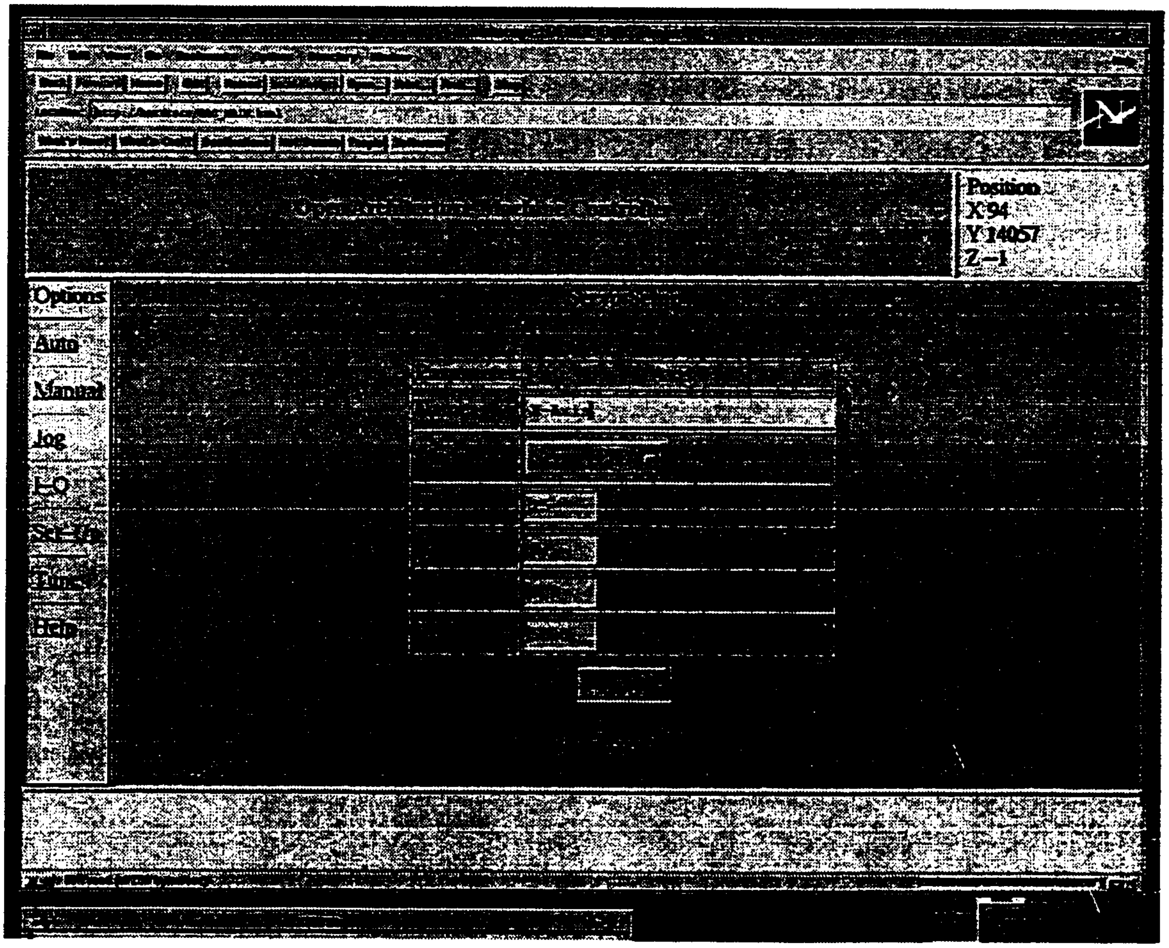


Figure 3.15: Ratio of disturbance force to table mass required to maintain given position tolerances, as a function of frequency



Position
X:94
Y:14057
Z:-1

Figure 3.16: Sample screen allowing access to system parameters through Netscape.

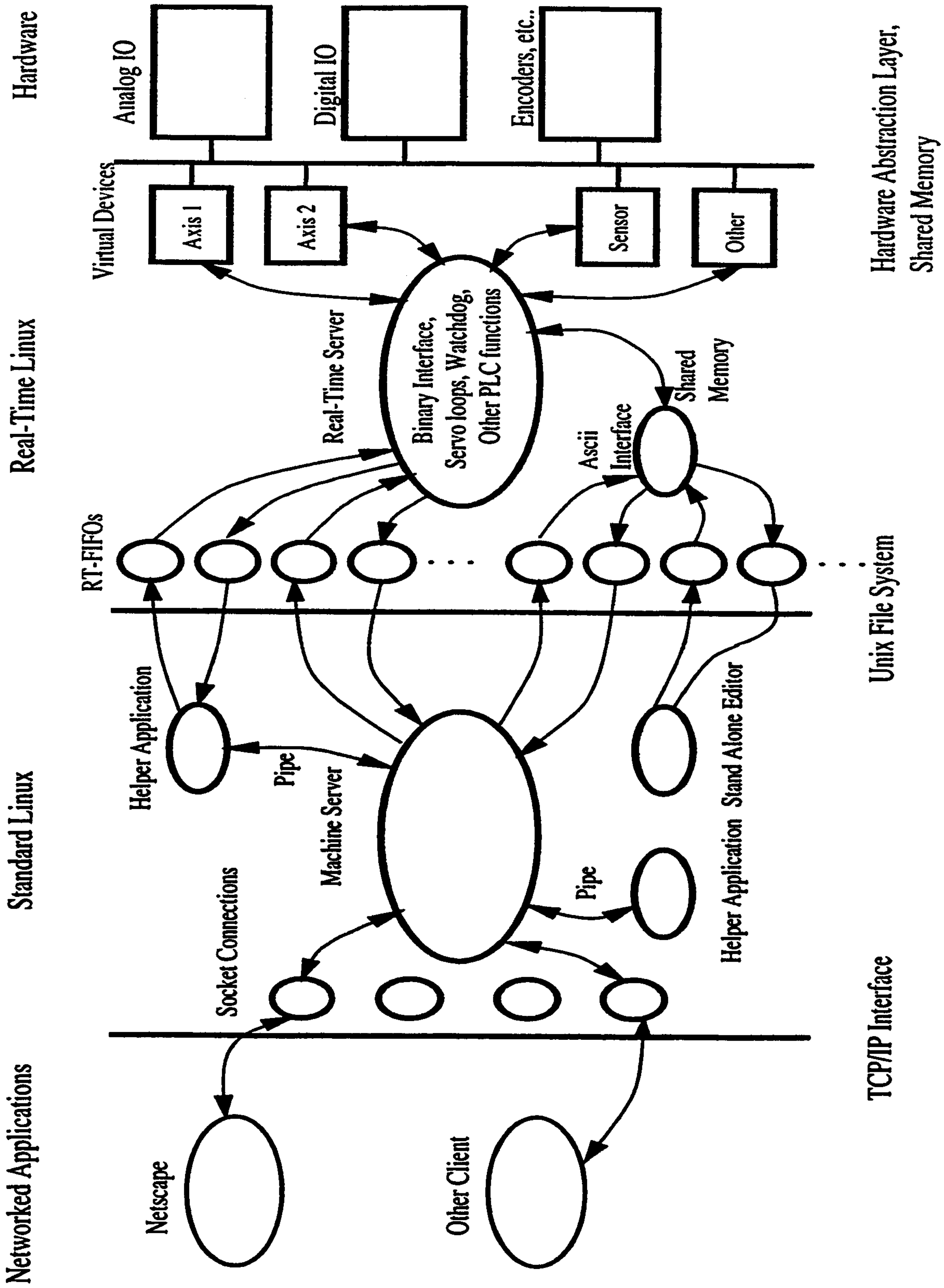


Figure 3.17: PC Based OAC Implementation Plan

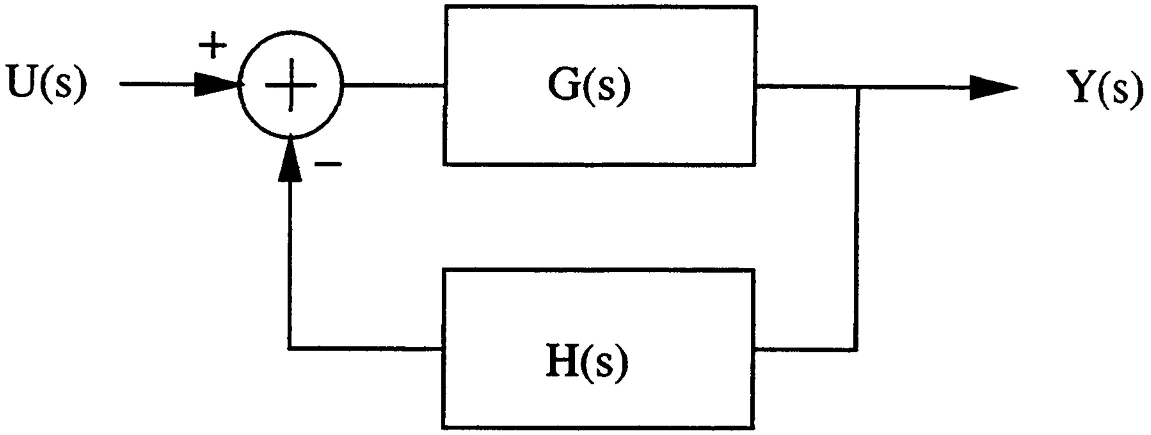


Figure 3.18: General Feedback Loop

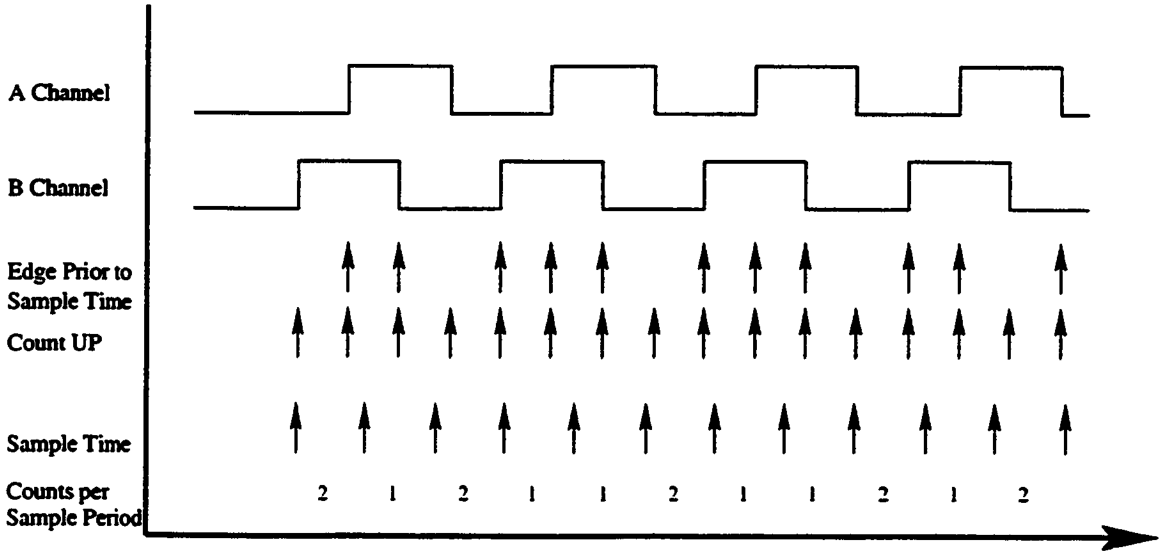


Figure 3.19: Digital Tachometer Consept

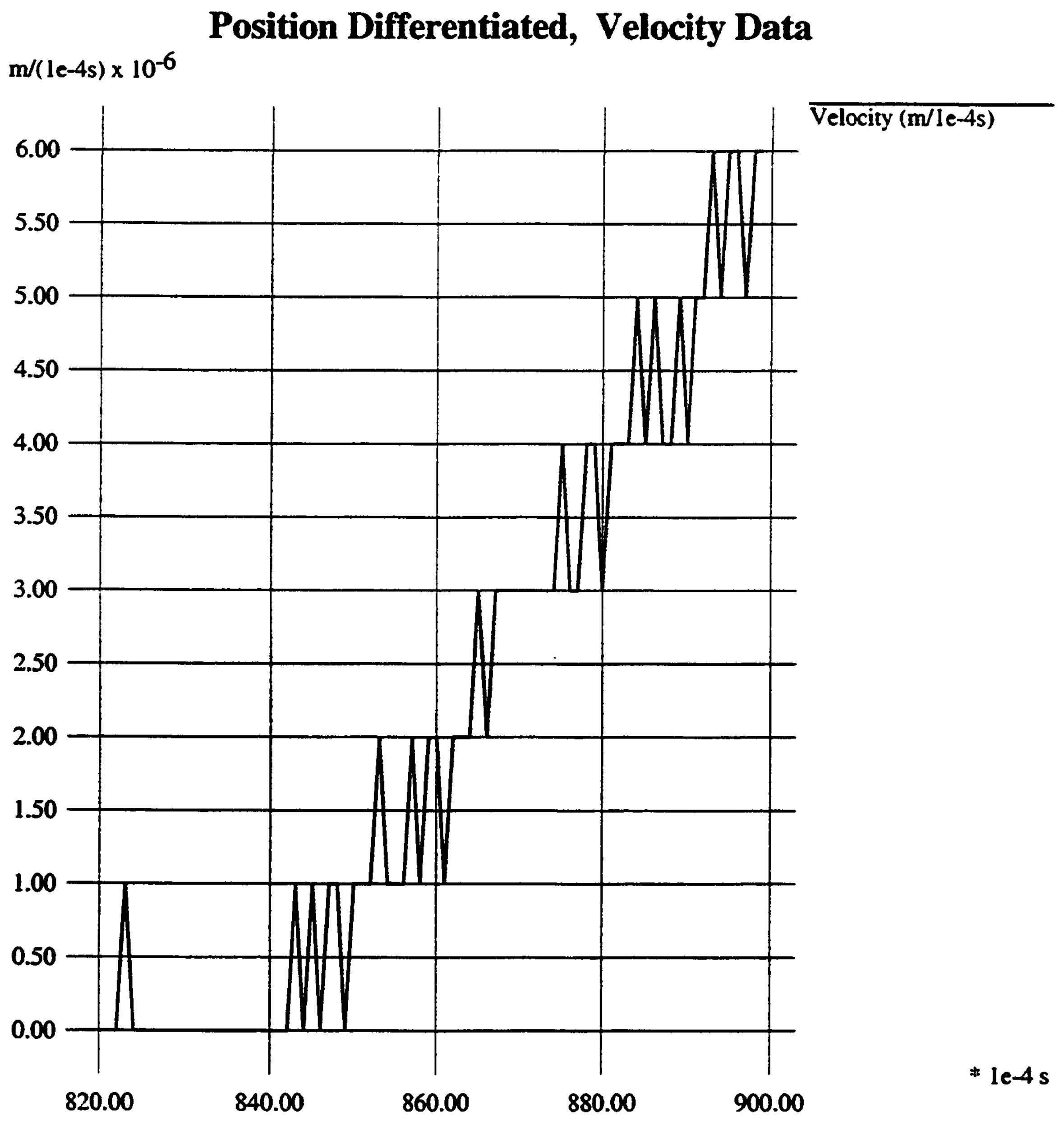


Figure 3.20: Velocity resulting from position differentiation

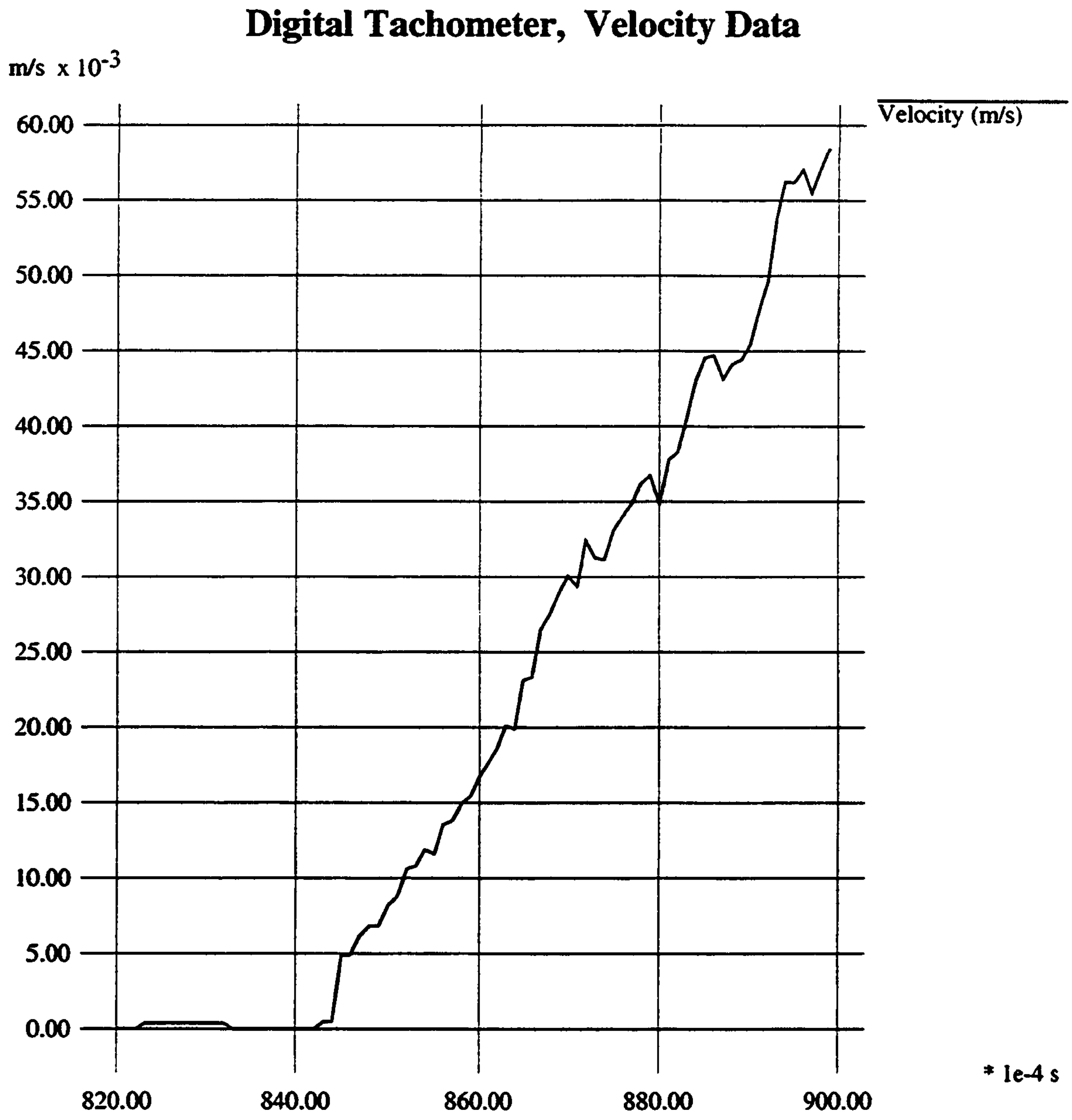


Figure 3.21: Velocity resulting from digital tachometer data

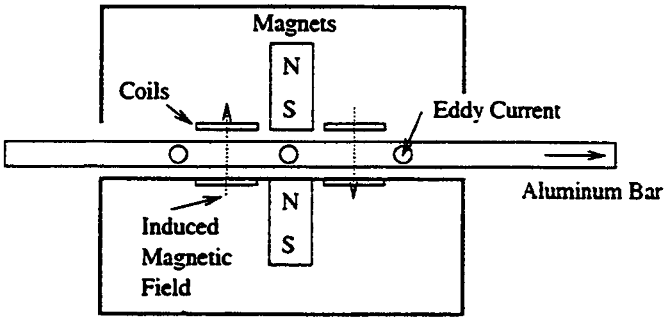


Figure 3.22: Diagram of ferraris sensor

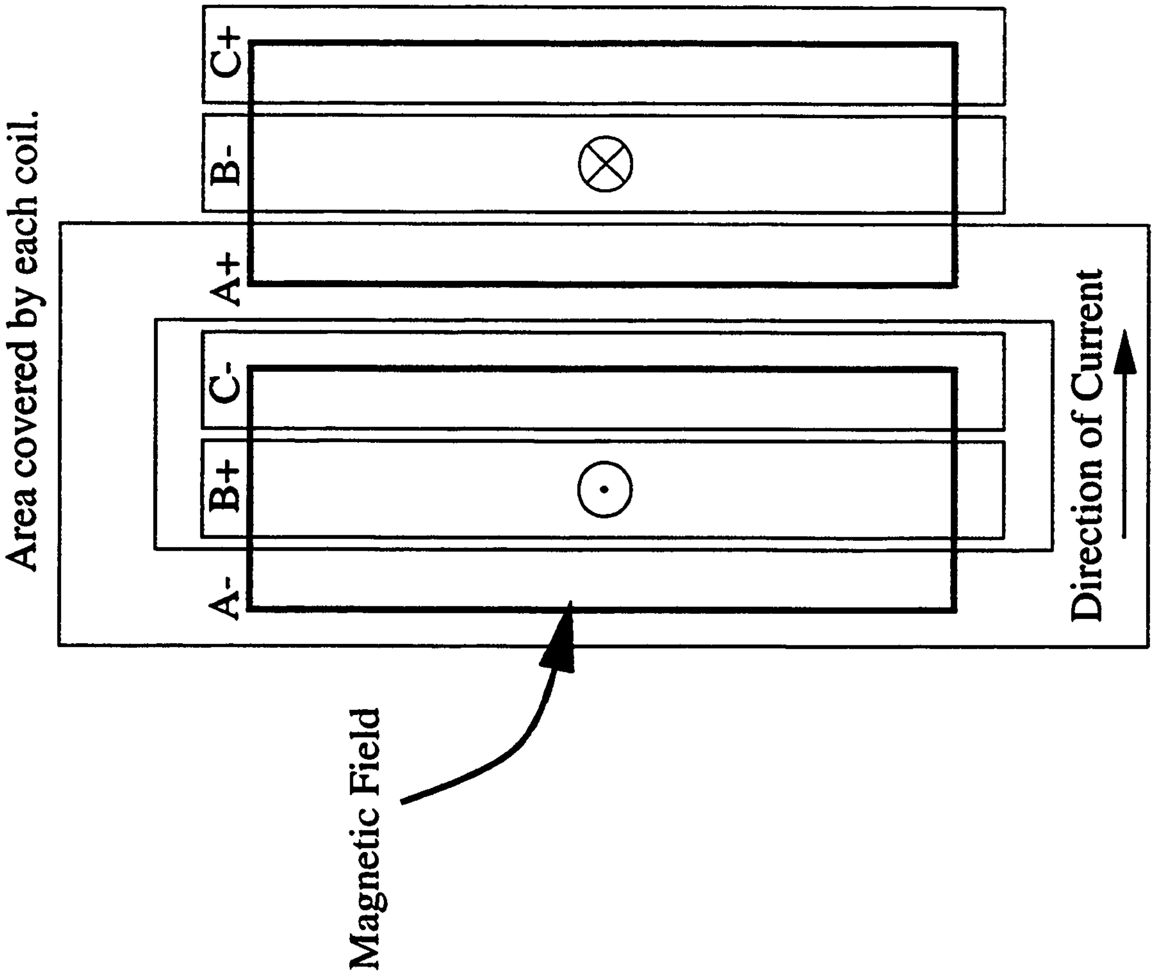


Figure 3.23: Representation of magnetic field experienced by Coils, position one

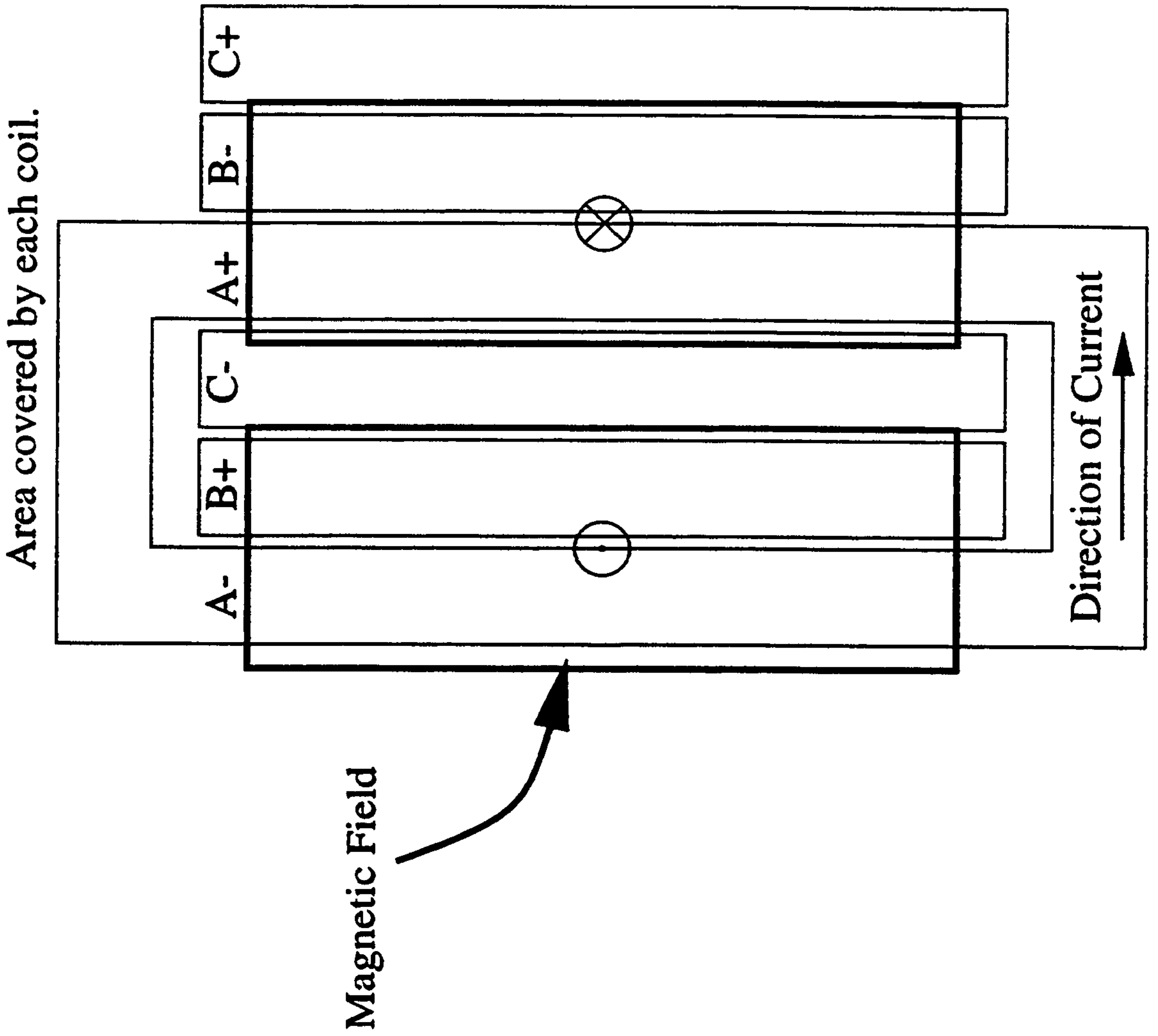


Figure 3.24: Representation of magnetic field experienced by Coils, position two

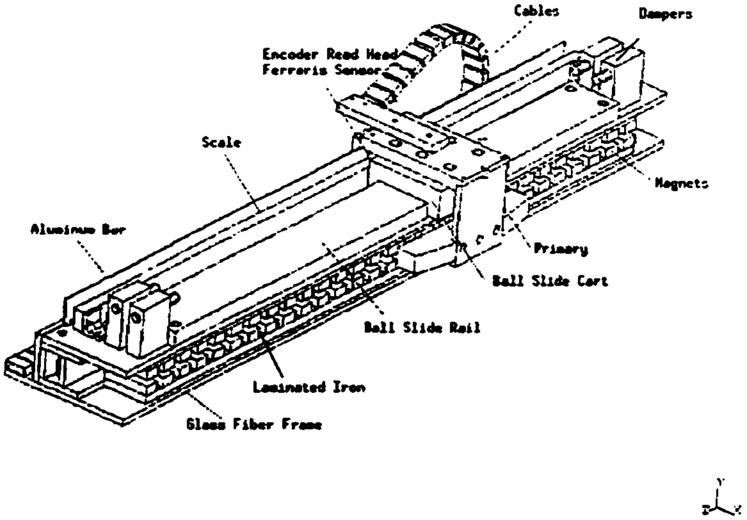


Figure 3.25: Custom-designed Linear Servo Motor

Chapter 4

Path Planning

4.1 Introduction

Control strategies for high performance machine tools (ball screw or linear motor driven) should take advantage of their simple plant dynamics (figure 3.13) and account for amplifier saturation (figure 3.14).

These issues must be addressed at the feed rate planning and servo loop levels. Two common feed rate scheduling techniques include fixed velocity and acceleration feed rate planning and constant feed rate planning. These techniques do not account for amplifier saturation and simple plant dynamics. The effect of constant feed rate planning on path error can be seen in figure 4.26. The excessive path error is a result of scheduling a path that exceeded the capabilities of the plant. When 'Fixed Velocity and Acceleration Limit Path Planing' is used the error is too small to be seen on this graph, and so the path error as a function of time is plotted in figure 4.27, however this technique schedules the feed rate sub-optimally, which results in longer machining time than necessary.

A technique which reduces machining time and/or path error by addressing the simple plant dynamics of high performance machine tools and saturation limits of amplifiers is developed in this chapter.

4.2 Minimum-Time Path Optimization

Fixed acceleration, and velocity feed rate planning when properly performed will ensure that the demands of the target path on the axes of the machine tool remain within their limits. However, this method does not take advantage of the full capabilities of a high performance machine tool and will result in longer machining times than necessary.

Minimum-time path optimization determines the maximum feed rate along an arbitrary multi-axis path that is possible without exceeding the capabilities of any axis. These capabilities are limited by the voltage and current limits of the amplifier/motor combination. Essentially the feed rate along the path is controlled such that each axis remains within it's capability envelop. The result is less path error, when compared to the fixed feed rate case, and less total time, when compared to fixed velocity, and acceleration case. This is important for movements during cutting, and rapid moves (30-50 percent of total machining time).

A traditional example of an unreasonable path is that of rounding a corner at a fixed feed rate. This demands infinite acceleration capabilities of the axis. The servo control loop cannot compensate for this sort of discontinuity. Much better results are obtained if modifications are made to accommodate the real limitations (see figures 4.26, 4.27). Figures 4.26 and 4.27 show the path error that results from traversing a corner with various controllers, given a fixed feed rate, and an 'optimized' feed rate, respectively. Figure 4.26 represents the fixed feed rate case, and the path error is large enough to be seen on a plot of the X-Y path. Notably, in this case, no servo control scheme was able to perform reasonably. Figure 4.27 represents the minimum-time path optimized case. The resulting path error is too small to be visible on a plot of the X-Y path as in the fixed feed rate case, so a plot of the path error is shown. The path and total path time are identical to those in figure 4.26, however the feed rate along the path has been 'optimized'. Figure 4.28 shows the actual feed rates.

The key then to high-speed machining is ensuring that the commanded path velocity and acceleration requirements are within the capability limits of the machine.

In machining, one is not normally able to modify the path to be traveled. Occasionally exceptions can be made, such as filleting corners. However, given a prescribed path, the only option is to control the speed (feed rate) at which the path is followed. While this option will immediately draw attention from persons concerned with the cutting process, we will argue that there are many cases where one has the freedom to do this. Also, the spindle can be controlled so that constant feed per tooth is maintained. Finally, there are only two options: path error, or velocity (feed rate) 'error'.

In order to optimize the path to the capabilities of a machine tool we need to know the machine's limitations. The limits are on the machine's velocity and acceleration capabilities. These limits are caused by the voltage and current limits that the amplifier and motor can tolerate. (We will ignore the case where the machine is limiting servoing speed. However, in such a case one simply uses the more restrictive limitation.)

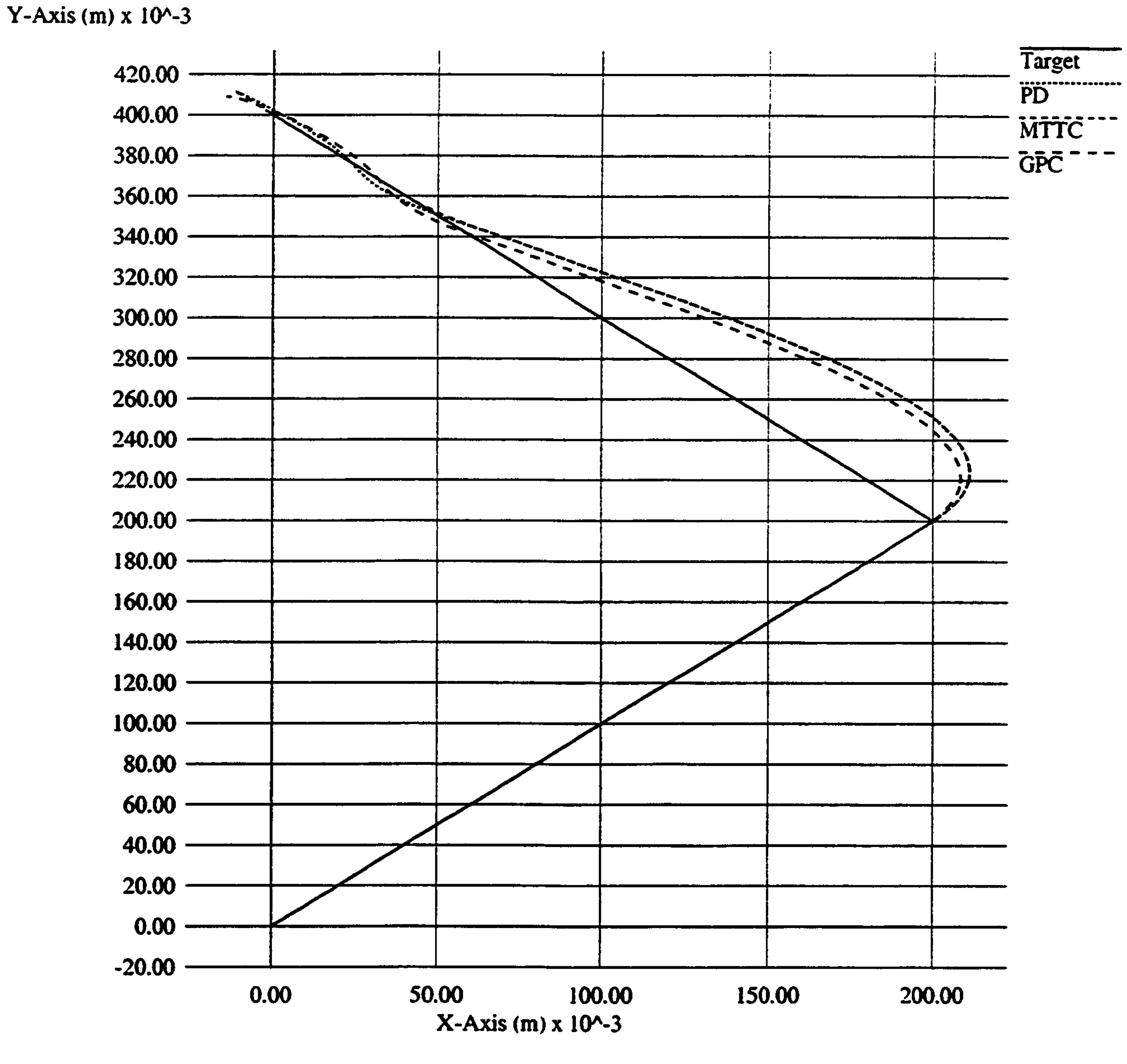


Figure 4.26: Resulting path from PD, MTTC, GPC controllers traversing a corner at constant feed rate, 'No Path Planning'

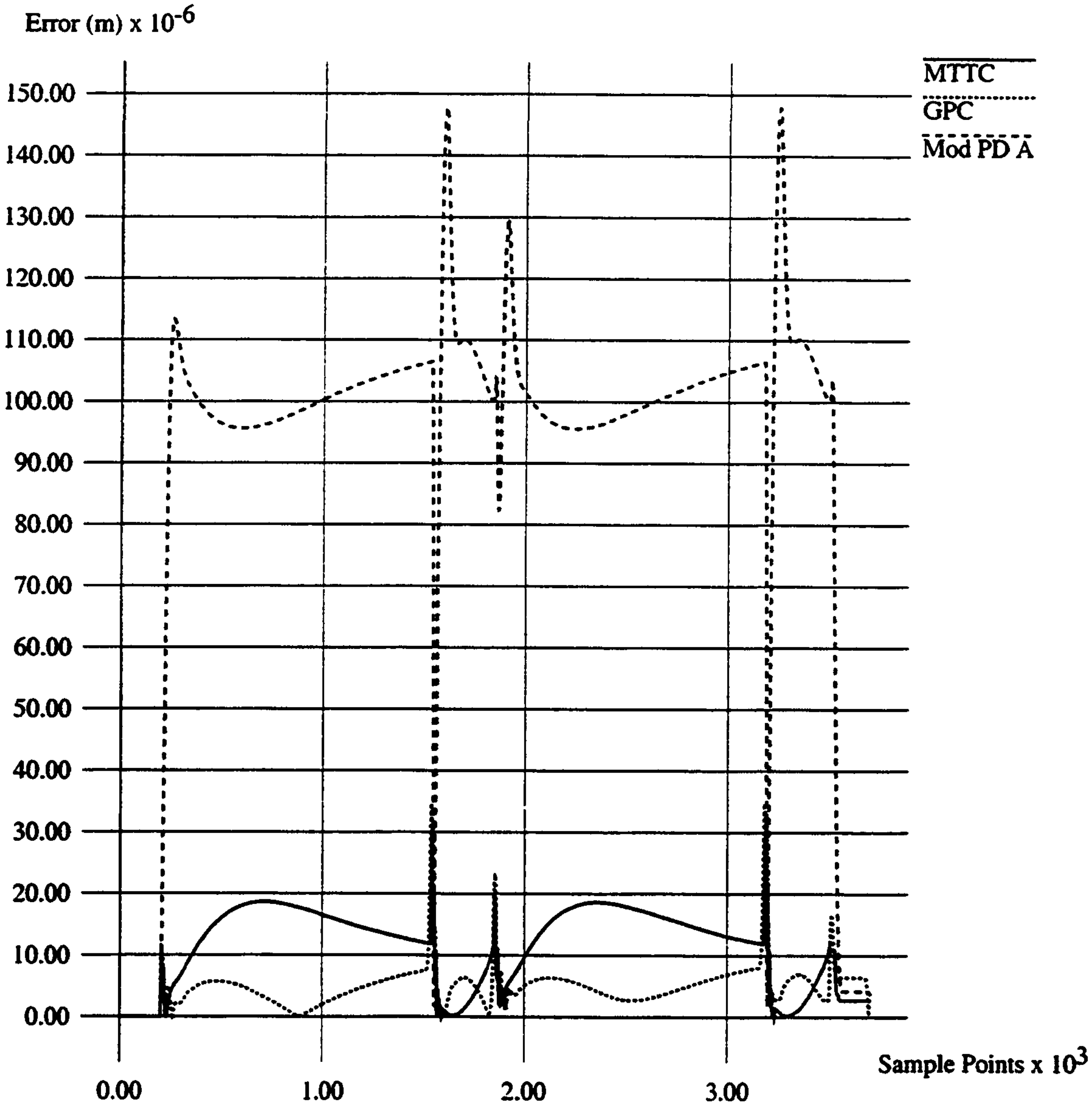


Figure 4.27: Resulting path error from PD, MTTC, GPC controllers traversing a corner with 'Optimal Path Planning'

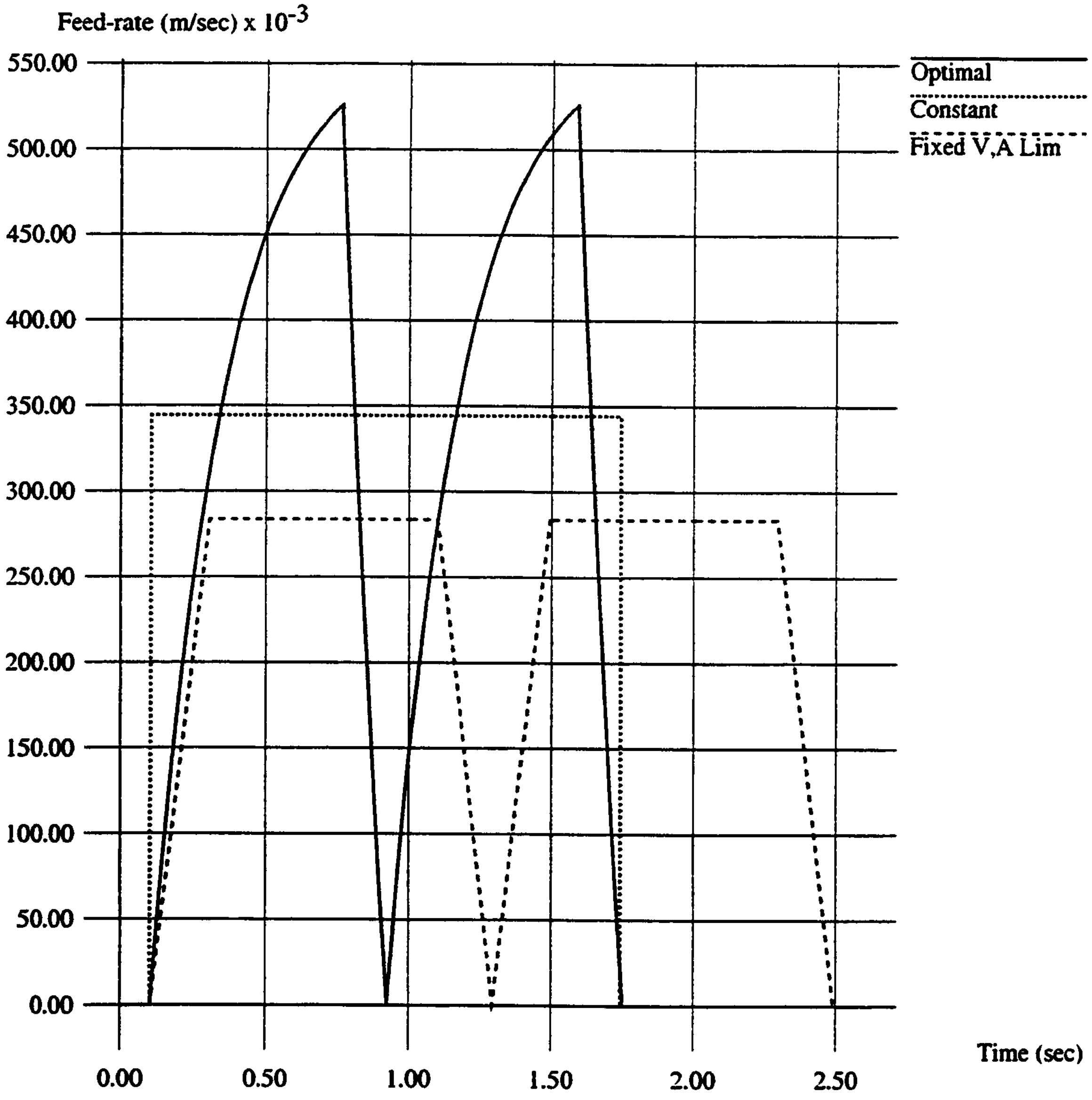


Figure 4.28: Feed rates for various path planners traversing a corner

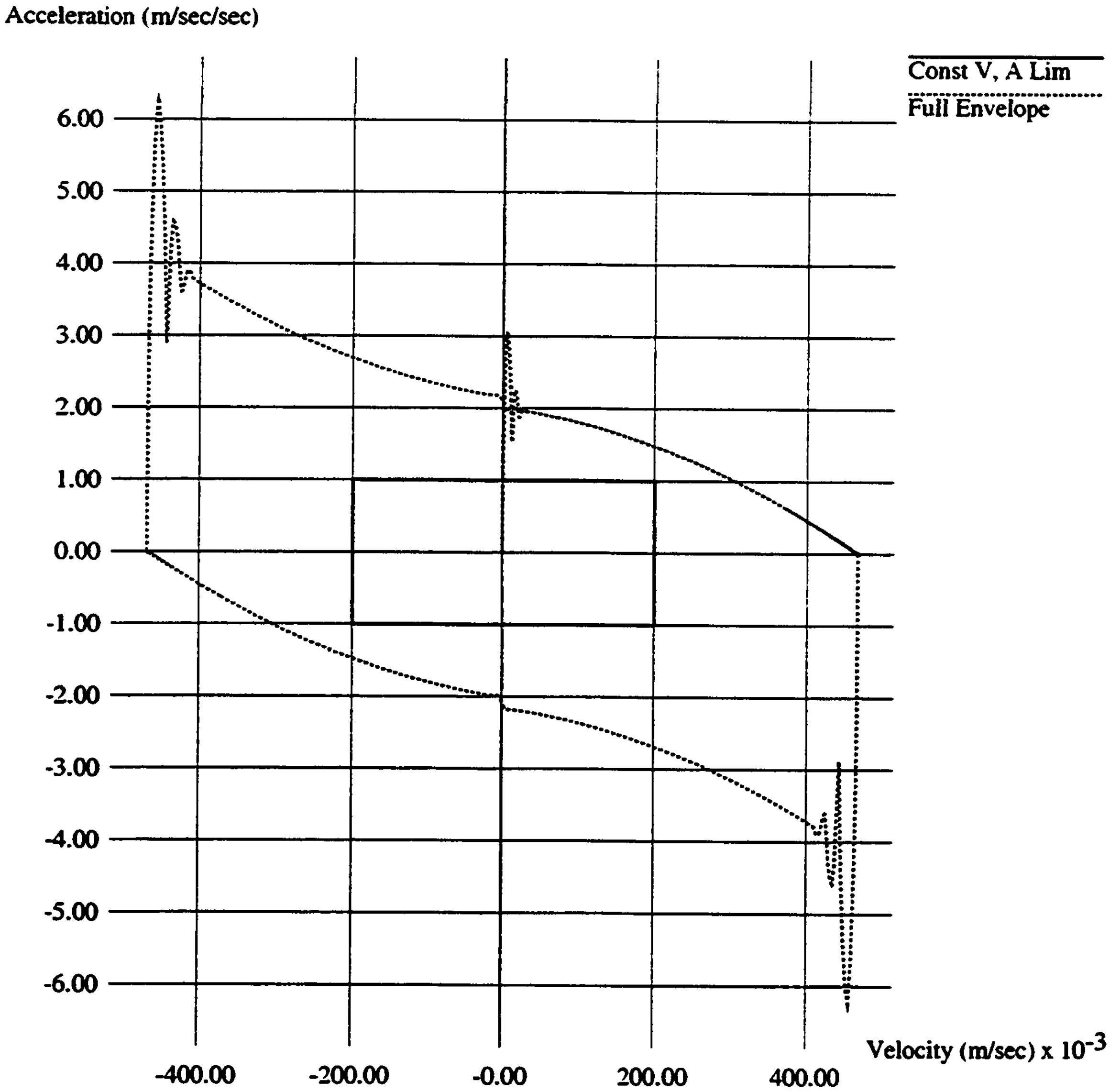


Figure 4.29: Simulated System Performance Envelope

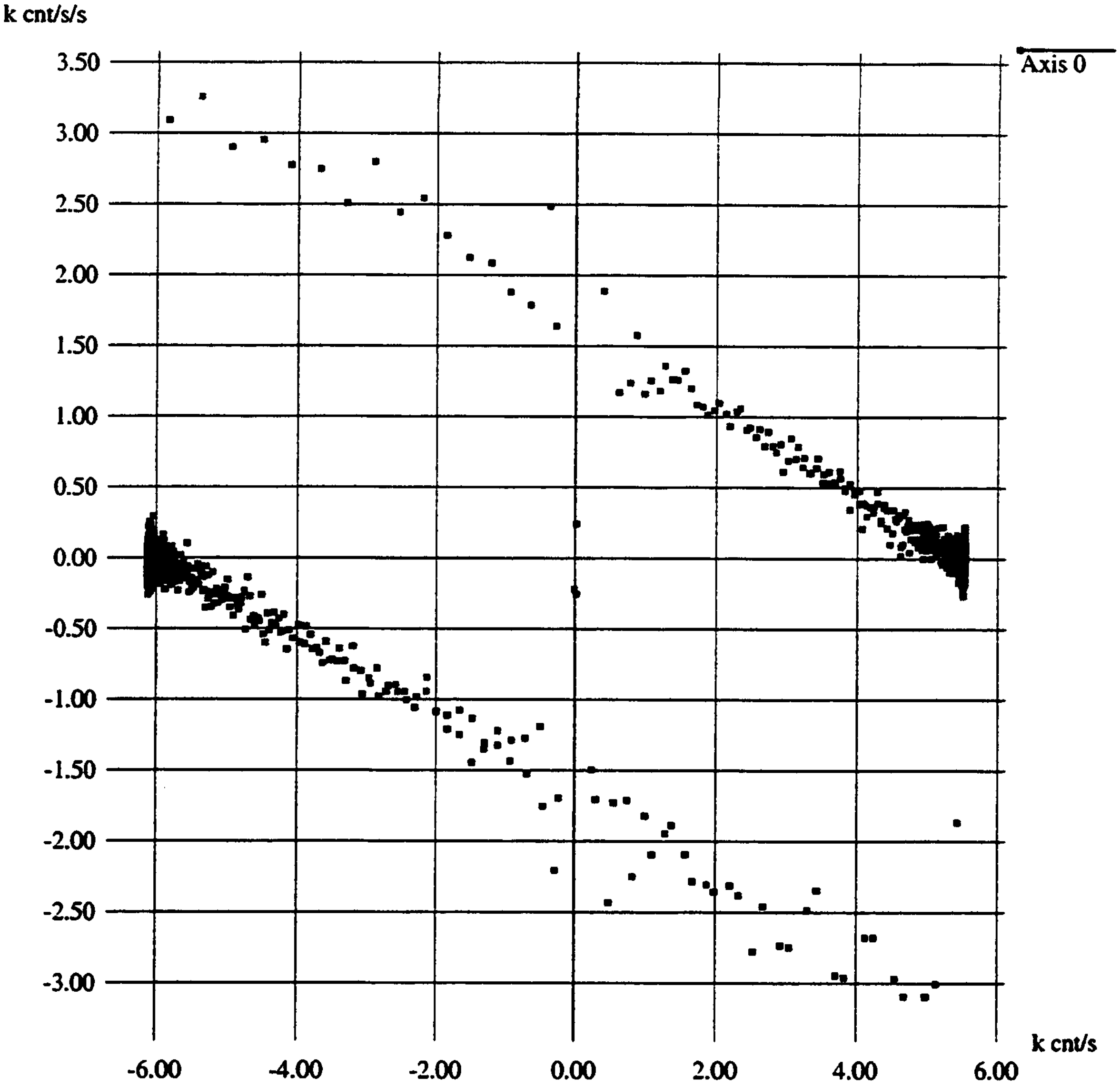


Figure 4.30: Test System Performance Envelope

Figures 4.29 4.30 and 3.14 show the velocity (feed rate) versus acceleration capabilities for a simulated system, a test system (rotary motor and mass) and a linear motor system respectively. These graphs were generated by applying full power forward, and then reverse, to the systems, and recording the resulting system velocity (feed rate) and acceleration. These graphs identify the working envelope of the machine tool. (The rectangle within represents the safe region of the total envelope that would be usable if we made the usual assumption that velocity and acceleration capabilities were independent. See figure 4.29.)

These limits on each coordinated axis must now be applied to the path in order to determine a coordinated motion that is within the capabilities of all the axes. The procedure used for this applies to any number of axes and is outlined as follows.

The performance envelope of each axis of the system is represented by an upper and lower bound. That is, for each axis, the acceleration boundaries can be described as functions of velocity.

$$A_{x_{max}}(\frac{dx}{dt}) = f(v) \tag{4.21}$$

$$A_{x_{min}}(\frac{dx}{dt}) = g(v) \tag{4.22}$$

The path to be followed is parameterized in an independent variable u :

$$\begin{pmatrix} x(u) & x'(u) & x''(u) \\ y(u) & y'(u) & y''(u) \\ z(u) & z'(u) & z''(u) \\ \vdots & \vdots & \vdots \end{pmatrix}, 0 < u < 1 \tag{4.23}$$

The functions must be continuous in their first derivatives.

We also know that at $u = 0$ and $u = 1$, all axes must be at rest, i.e., $\frac{dx}{dt} = 0$.

We estimate $\frac{dx}{dt}$ and $\frac{d^2x}{dt^2}$ using the following formulas:

$$\frac{dx}{dt} = \frac{dx}{du} \cdot \frac{du}{dt} \tag{4.24}$$

$$\frac{d^2x}{dt^2} = \frac{d^2x}{du^2} \cdot \frac{du}{dt} + \frac{dx}{du} \cdot \frac{d^2u}{dt^2} \tag{4.25}$$

An algorithm is used to determine the largest value of $\frac{d^2u}{dt^2}$ that will not violate the acceleration capabilities on any axis.

Algorithm 1 Calculate $u(t)$

```
$u = 0$, $\frac{du}{dt} = 0$
while $u < 1$ do
    Calculate the largest value of $\frac{d^2u}{dt^2}$ that will not violate the acceleration boundaries on any axis. This is accomplished as follows: Knowing $\frac{dx}{du}$, and using the current value of $\frac{du}{dt}$, calculate $\frac{dx}{dt}$ from equation 4.24. The corresponding acceleration limits are then determined from equations 4.21 and 4.22. These values are in turn used to find the corresponding maximum and minimum values of $\frac{d^2u}{dt^2}$ using equation 4.25. (Say, $(\frac{d^2u}{dt^2})_{max}$ and $(\frac{d^2u}{dt^2})_{min}$.) This is repeated for each axis, using equations of the same form as those referenced. The lowest value of $(\frac{d^2u}{dt^2})_{max}$ is thus the largest permissible value without exceeding the acceleration capabilities of any axis for a given feed rate. However, this value is not acceptable in the event that $(\frac{d^2u}{dt^2})_{min}$ exceeds it for any axis. If this occurs, decrease the current value of $\frac{du}{dt}$ iteratively to the point that an acceptable value is found.
    Advance $u(t)$ according to the estimate $u(t+\delta) = u(t) + \frac{du}{dt} \cdot \delta t + \frac{1}{2} \cdot \frac{d^2u}{dt^2} \cdot \delta t^2$
    Advance $\frac{du}{dt}$ according to the estimate $\frac{du(t+\delta)}{dt} = \frac{du}{dt} + \frac{d^2u}{dt^2} \cdot \delta t$
end while
```

Once algorithm 1 has been executed in the forward direction, the problem is reversed and repeated with the additional constraint that $\frac{du}{dt}$ must be less than or equal to $\frac{du}{dt}$ in the forward path.

The resulting path can now be followed using a proportional derivative controller. Figure 4.27 shows the path error that results as the system rounds a corner. The exact path is shown in figure 4.26. The feed rate is shown in figure 4.28.

The three cases used in figure 4.28 are 'Time Optimal Path Planning', 'Constant Feed Rate', and 'Fixed Velocity and Acceleration Limit Path Planing'. In the 'Time Optimal Path Planning' case and the 'Constant Feed Rate' case, equal total path times are used, while the path error is significantly different. The traditional 'Fixed Velocity and Acceleration Limit Path Planning' case is used to demonstrate the amount of cycle time that is currently being lost as a result of not using the total system performance envelope, while path error remains relatively unchanged.

Monitoring the amplifier can also validate the hypothesis that saturation effects are the cause of path error. Figures 4.31 and 4.32 show the amplifier signal for 'Time Optimal Path Planning' and for 'Constant Feed rate', respectively. With 'Time Optimal Path Planning', the amplifiers remain near a current limit (12 Amps) or a voltage limit, but do not saturate. By contrast, in the case of 'Constant Feed rate', the amplifiers have severe saturation problems. (The amplifiers saturate at 12 Amps.) With 'Fixed Velocity and Acceleration Limit Path Planing', the amplifiers do not saturate, but neither do they use their full capabilities.

Results indicate that in the case of high-speed machining, trajectory planning is required before advanced control algorithms can be expected to function. This is due to the limitations on amplifier current and voltage, which, if encountered, will lead to significant error. The other result is that there can be significant cycle time savings if feed rates are scheduled as proposed, instead of using fixed velocity and acceleration

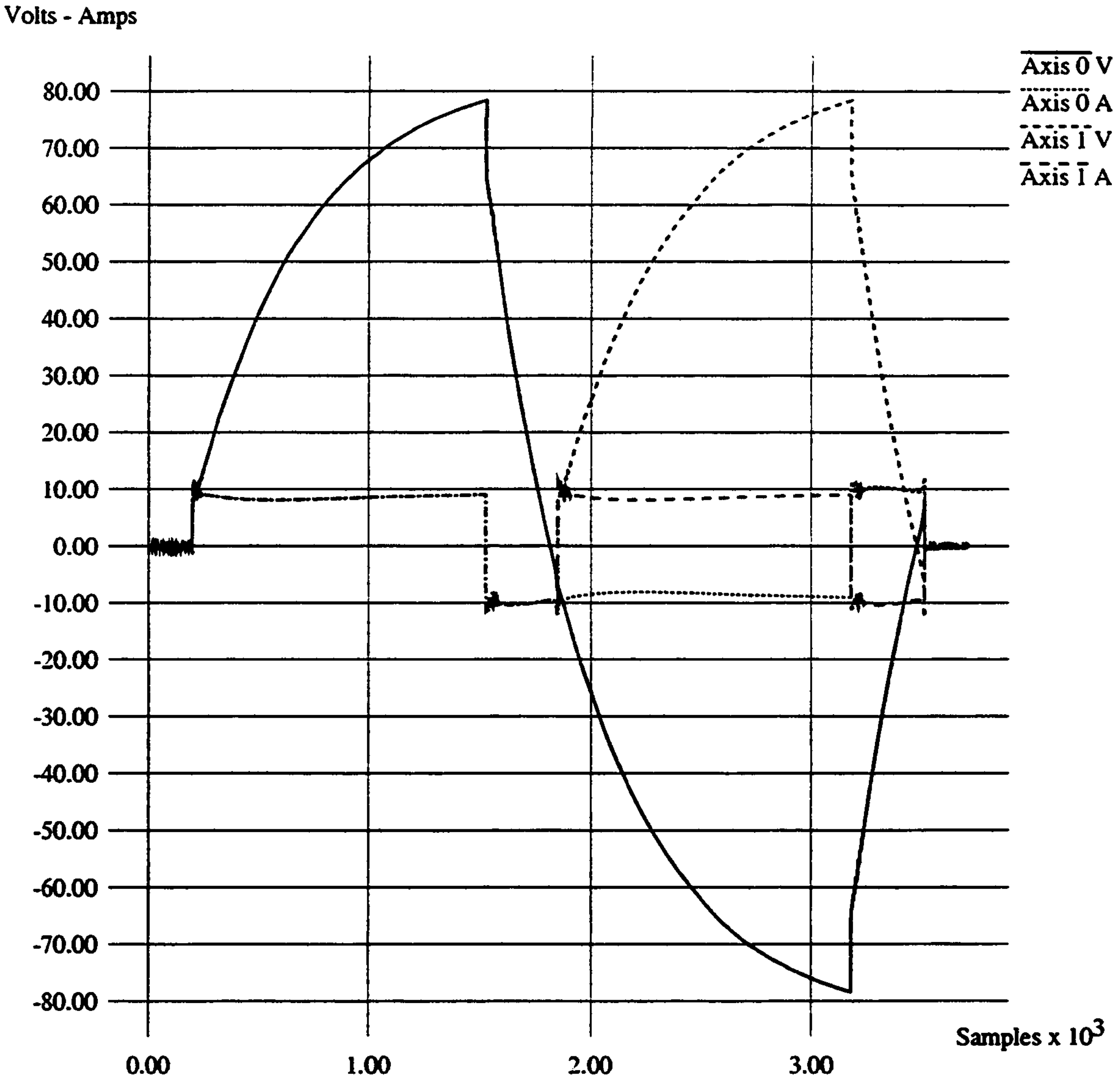


Figure 4.31: Amplifier signals ((A)mps, (V)oltage) resulting from traversing a corner, in the path optimized case

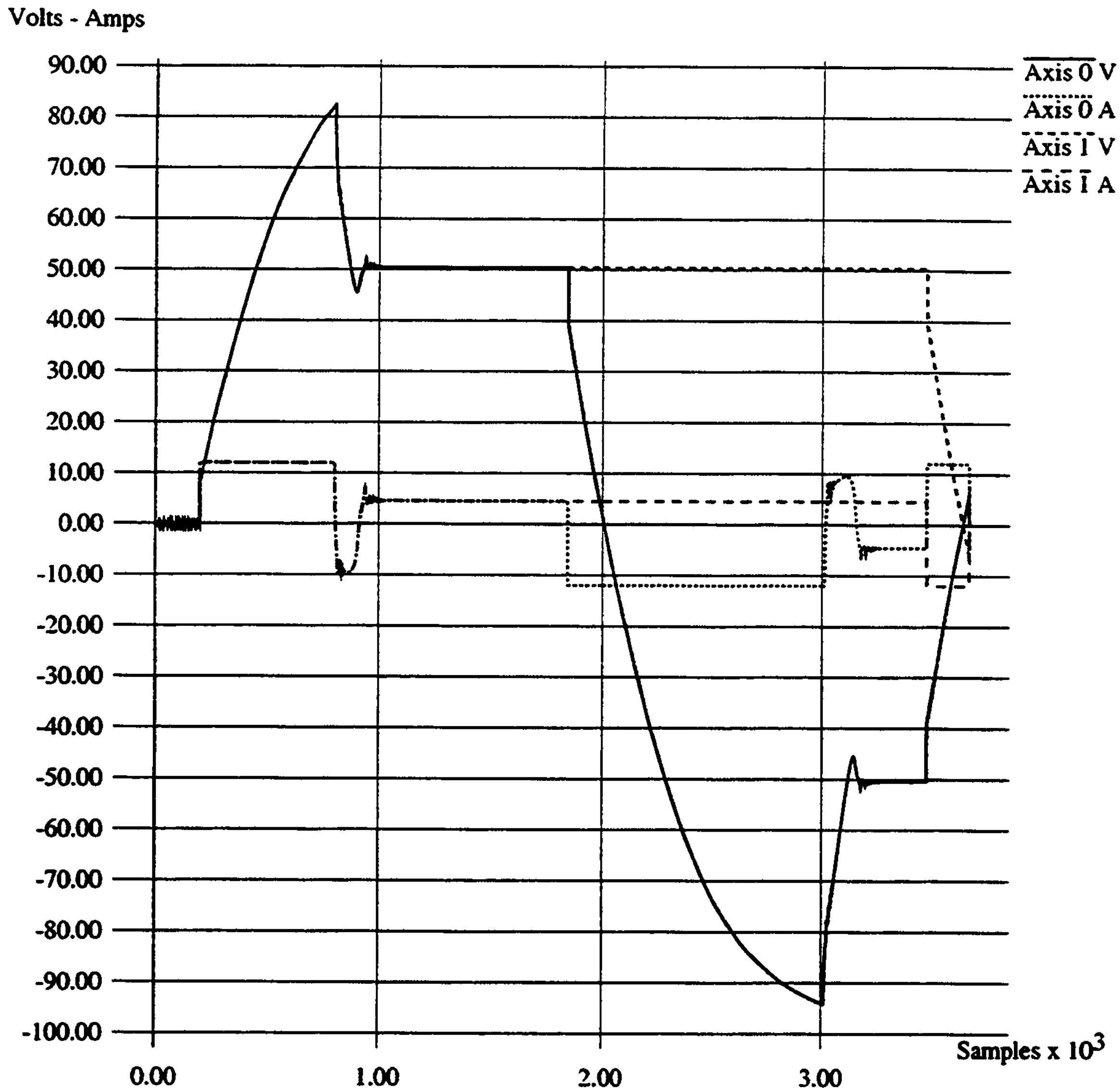


Figure 4.32: Amplifier signals ((A)mps, (V)oltage) resulting from traversing a corner, in the fixed feed rate case

limits.

4.3 Summary

Simulation results and tests on physical plants clearly indicated that if not explicitly addressed path demands which exceed the capabilities of one or more axes will dominate part error. To avoid this source of error feed rates are typically over conservative. To address this issue a feed rate planning method, minimum time path optimization (MTPO), was developed which takes advantage of the full performance envelopes of the axes involved in a coordinated motion. The result of this improved usage of the capabilities of a machine tool is typically reduced cutting time, with out sacrificing part tolerances. This savings is beneficial both while cutting, and jogging.

Chapter 5

Control Algorithms

5.1 Introduction

Without trajectory planning, saturation effects are the dominant source of path error in high-speed machining. As such, it will be assumed that the path has been appropriately planned in advance, for all controllers. Responses to disturbances are used to demonstrate the robustness of a controller.

The main types of servo controllers include 'Error-Based Feedback Controllers', 'Feed Forward Controllers', 'Cross-Coupling Controllers', 'Optimal Controllers' (including GPC), 'Adaptive Controllers', and 'Repetitive Controllers' [30]. Path planning, as outlined, complements and improves the performance of all these controllers. Without disturbances, given an accurate model and a well-planned path, all of these control schemes perform acceptably. The simplest, with excellent performance for machine tool control, is 'Feed Forward, with Error-Based Feedback'. The largest variation amongst these servo controllers is found in their disturbance-rejection performance. The element common to these control schemes is a linear plant model (or the presumption of one in cases like PID). However, it has been shown that a machine tool has a number of non-linear system limitations, specifically, the current and voltage limits of the amplifiers. Another measure which is not commonly considered is minimum time. A controller which takes these factors explicitly into account in a computationally efficient manner is 'Minimum-Time Tracking Control' and is introduced at the end of this section.

5.2 Error Based, Feedback Controllers

Traditionally, error-based feedback controllers are PID. Feedback about the axis is provided through an encoder, and optionally through a tachometer. This scheme has a number of shortcomings:

- This controller requires following error to exert a command (or to change it with integration).
- It does not use a model of the system.
- It is unable to take advantage of future information.

In spite of these difficulties, PID can provide excellent control. In addition, it is very simple, enabling it to be computed extremely quickly.

It was found that the element that most adversely affected PID controllers for high-speed machining was, if present, the internal velocity loop (internal to the motor, amplifier). A small modification, which amounts to velocity feed forward, improves performance dramatically in high-speed machining. This, however, is different than

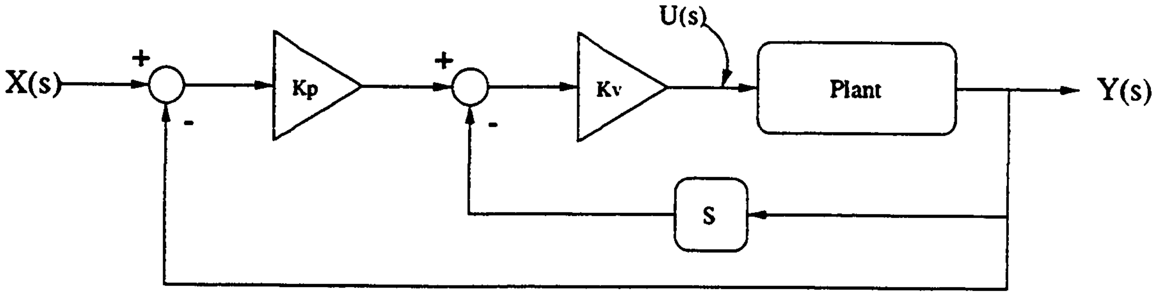


Figure 5.33: Position Control, with Velocity Feedback

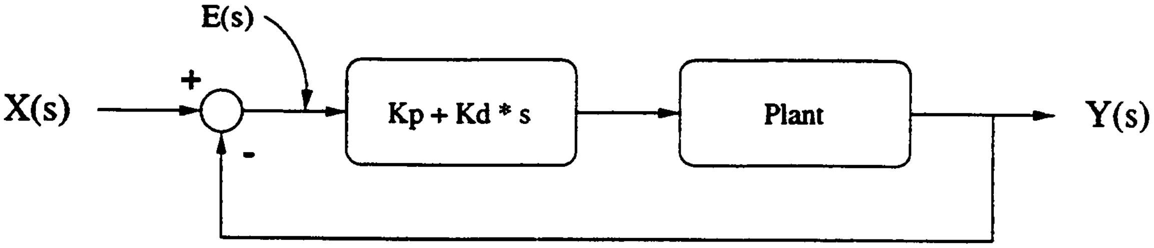


Figure 5.34: Position, Derivative Control

model-based feed forward, in that the feed forward is solely dependent on the path. This can be described as follows:

Upon inspection of the PD controller block diagram (figure 5.34), we note the following:

$$E(s) = X(s) - Y(s) \tag{5.26}$$

$$E(s) \cdot s \cdot K_d = (X(s) - Y(s)) \cdot s \cdot K_d \tag{5.27}$$

In the time domain:

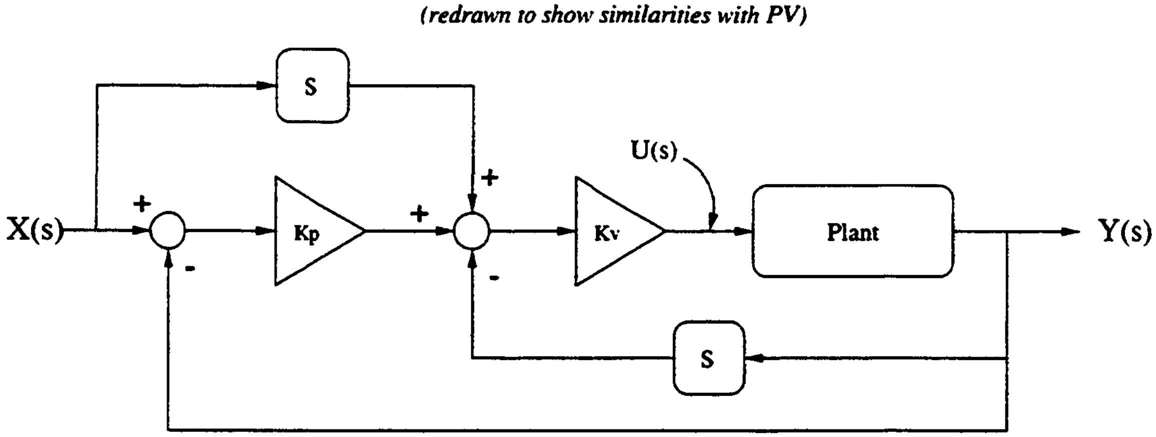


Figure 5.35: Re-arranged PD Controller

$$\frac{de}{dt}(t) \cdot K_d = (\frac{d(x(t) - y(t))}{dt}) \cdot K_d \tag{5.28}$$

$$\frac{de}{dt}(t) \cdot K_d = (\frac{dx}{dt}(t) - \frac{dy}{dt}(t)) \cdot K_d \tag{5.29}$$

As seen in equation 5.29, $\frac{de}{dt}(t)$ is actually the difference between the target velocity and current velocity. With this insight, we redraw the PD controller block diagram in figure 5.34 to more resemble the PV controller in figure 5.33. The re-arranged PD controller is shown in figure 5.35.

If we assume that the plant has an arbitrary transfer function P, then for the PD controller (figure 5.34) we have the following transfer function:

$$\frac{Y(s)}{X(s)} = \frac{K_p + K_d \cdot s}{\frac{1}{P} + K_d \cdot s + K_p} \tag{5.30}$$

The transfer function of the re-arranged PD controller (figure 5.35) is as follows:

$$\frac{Y(s)}{X(s)} = \frac{K_v \cdot K_p + K_v \cdot s}{\frac{1}{P} + K_v \cdot s + K_v \cdot K_p} \tag{5.31}$$

From these two transfer functions we can see that if we set $K_v \cdot K_p$ of the re-arranged PD controller (equation 5.31) equal to K_p of the PD controller (equation 5.30), and we set K_v to K_d, then the two transfer functions are identical. Therefore we will use the re-arranged PD controller (figure 5.35) for comparison purposes with the PV controller of figure 5.33.

The transfer function of the PV controller (figure 5.33) is as follows:

$$\frac{Y(s)}{X(s)} = \frac{K_v \cdot K_p}{\frac{1}{P} + K_v \cdot s + K_v \cdot K_p} \tag{5.32}$$

By comparing the denominators of the PV controller with the re-arranged PD controller (equations 5.32 and 5.31), we note that they are identical. We therefore expect no differences in stability.

We will now derive a measure for the steady-state error, e_{ss}, to a ramp input for each controller. If we assume that $U(s)$ (as indicated in each figure) is identical for both systems, then the steady state error for the PV controller can be derived as follows:

$$u(t) = K_v \cdot (-\frac{dy}{dt}(t) + K_p \cdot (x(t) - y(t))) \tag{5.33}$$

If, at steady state, we make the substitutions $(x(t)-y(t)) = e(t)$ and $(\frac{dy}{dt}(t)) = K_R$ (where K_R is the slope of the ramp) we have our measure of e_{ss}:

$$e_{ss} = \frac{u_{ss}}{K_p \cdot K_v} + \frac{K_R}{K_p} \tag{5.34}$$

For the PD controller we derive e_{ss} in a similar fashion, and it is as follows:

$$e_{ss} = \frac{u_{ss}}{K_p \cdot K_v} \tag{5.35}$$

Upon examination of the steady-state errors for the PV controller and the re-arranged PD controller (equations 5.34 and 5.35), we note that for an arbitrary plant, the steady-state error to a ramp input for the PV controller will always be greater than that of the re-arranged PD controller by a factor of $\frac{K_R}{K_p}$. This becomes a significant consideration in high-speed machining, when K_R is relatively high.

An example makes the point clear.

We require a number of assumptions and observations:

- the maximum input to the plant is 20 (Amps)
- $x(t)$, $y(t)$ are expressed in meters (m)
- at rest, with a position error of 0.25 (mm), it is found that the controller exerts full command
- at the target position, with a velocity error of 0.1 $(\frac{m}{s})$, it is found that the controller exerts full command
- at steady-state, with a velocity of 0.5 $(\frac{m}{s})$, the current is found to be 10 amps

From the assumptions and observations we can calculate K_v to be 200 $(\frac{V}{\frac{m}{s}})$, and K_p to be 400 $(\frac{\frac{m}{s}}{m})$.

The resulting following error for the PD system is:

$$e_{ss} = \frac{u_{ss}}{K_v \cdot K_p} = \frac{10}{200 \cdot 400} = 0.125mm \qquad (5.36)$$

For the PV system the following error is:

$$e_{ss} = \frac{u_{ss} + K_R \cdot K_v}{K_p \cdot K_v} = \frac{10 + 0.5 \cdot 200}{200 \cdot 400} = 1.375mm \qquad (5.37)$$

This is a ratio of 11 to 1.

Clearly then, for significant improvement at low cost, this modification to existing PID controllers with internal velocity feedback loops combined with 'Minimum-Time Path Optimization' should be considered.

5.3 Feed forward, with Error-Based Feedback

As mentioned, this is one of the simplest possible augmentations to PID control, whether it be velocity and acceleration feed forward, or Zero Phase Error Tracking (ZPETC). (A brief description can be found in section 2.3.1.) When used in conjunction with 'Minimum-Time Path Optimization', path error can be low. This controller performs no better, however, to a step input or to other disturbances than does PID. This is because in the case of a step input change, the command signal that is generated saturates the amplifier, resulting in lost control information, and in the case of a disturbance, the controller can not use the 'feed forward' element of the controller to react. An Inverse Compensation Filter Controller (IKF) will reduce the saturation problem. However, this is a linear operation, not able to take advantage of the performance envelope of the system.

This type of controller has the advantage over PID that it does not depend on path error to generate, nor to maintain a command signal. It also takes advantage of the target path velocity and acceleration information. However, if following error does accumulate, the feed forward information can actually hinder control. These controllers also ignore saturation limits.

5.4 Generalized Predictive Control (GPC)

Generalized Predictive Controller (GPC), performs exceptionally well, given a path which has undergone 'Minimum-Time Path Optimization'. This is because the controller, given appropriate values of lambda (λ) and other constants, never commands a value which would cause saturation, nor assumes that the system is capable of something which it isn't (again due to saturation limits).

However, because GPC has no knowledge of the system limitations, it will either perform sub-optimally to a disturbance (such as a step disturbance), or it will saturate. The other issue with GPC is the computational burden imposed by selecting N2 appropriately for a servo system. Specifically, the horizon for GPC should be 'equal to the settling time of the plant' [20] [10]. If we assume a loop closure rate of between 1kHz and 10 kHz, and a settling time, or time for the system to come to a complete stop, of approximately half a second, N2 should be between 500 and 5000. This type of horizon will impose a heavy computational burden. More typical horizons are on the order of 10 steps [10] [20] [18]. (A brief description of GPC can be found in section 2.3.2.) While not reproduced, GPC with input constraints exists. However, it is even more computationally intense to the point that '[The computational burden] is often too large to be useful for high-speed controllers using current microcomputers' [53].

If one uses a horizon of 30 steps and a loop closure rate of 2 kHz, these problems become visible. (See figures 5.37, 5.38, and 5.39.) As a result of the short preview horizon required for the practical implementation of GPC, and the fact that it does not consider saturation effects, GPC should be combined with 'Minimum-Time Path Optimization' and disturbances should be small.

5.5 Minimum-Time Tracking Control

Based on the information presented, a controller should take advantage of the following information:

- Known Current and Voltage Limits
- Plant Model
- Future Path Information

An ideal controller should bring the plant onto the known target path (position, velocity) as quickly as is allowed by the current and voltage limits of the amplifier. If we assume that our plant is essentially second order $\frac{C_1}{s(s+C_2)}$, then to optimally reject a disturbance a controller should apply full command towards the target for a period (t_0), then full command away for a period (t_1), resulting in a perfect match with the target path (position, velocity).

We can solve directly for the periods t_0 and t_1 by approximating the target path over a set of time intervals, where t_0 is the first control period, t_1 the second, T_0 the time instant before t_0, T_1 the instant after t_0 but before t_1, and T_2 the instant after t_1, but before t_2.

We let P_{T0}, P_{T1}, and P_{T2} be the target positions at times T_0, T_1, and T_2, respectively. Similarly, V_{T0}, V_{T1}, and V_{T2} are the target velocities, and A_{T0}, A_{T1}, and A_{T2} represent the target acceleration values. The actual position, velocity, and acceleration are denoted P_{A0}, V_{A0}, A_{A0}, etc.

Thus, at the instant T_1, we have:

$$V_{A1} = V_{A0} + A_{A0} \cdot t_0 \tag{5.38}$$

$$V_{T1} = V_{T0} + A_{T0} \cdot t_0 \tag{5.39}$$

$$P_{A1} = P_{A0} + V_{A0} \cdot t_0 + \frac{A_{A0} \cdot t_0^2}{2} \tag{5.40}$$

$$P_{T1} = P_{T0} + V_{T0} \cdot t_0 + \frac{A_{T0} \cdot t_0^2}{2} \tag{5.41}$$

At the instant T_2, then, substituting equation 5.38 for V_{A1}, and substituting equation 5.39 for V_{T1}, we have:

$$V_{A2} = V_{A1} + A_{A1} \cdot t_1 = V_{A0} + A_{A0} \cdot t_0 + A_{A1} \cdot t_1 \tag{5.42}$$

$$V_{T2} = V_{T1} + A_{T1} \cdot t_1 = V_{T0} + A_{T0} \cdot t_0 + A_{T1} \cdot t_1 \tag{5.43}$$

The actual position at the instant T_2 is

$$P_{A2} = P_{A1} + V_{A1} \cdot t_1 + \frac{A_{A1} \cdot t_1^2}{2} \tag{5.44}$$

Substituting equations 5.40 and 5.38 yields:

$$P_{A2} = P_{A0} + V_{A0} \cdot t_0 + \frac{A_{A0} \cdot t_0^2}{2} + V_{A0} \cdot t_1 + A_{A0} \cdot t_0 \cdot t_1 + \frac{A_{A1} \cdot t_1^2}{2} \qquad (5.45)$$

Similarly, the target position is found to be

$$P_{T2} = P_{T0} + V_{T0} \cdot t_0 + \frac{A_{T0} \cdot t_0^2}{2} + V_{T0} \cdot t_1 + A_{T0} \cdot t_0 \cdot t_1 + \frac{A_{T1} \cdot t_1^2}{2} \qquad (5.46)$$

Given $P_{A2} = P_{T2}$, $V_{A2} = V_{T2}$, $A_{T0} = A_{T1}$, $A_{A0} = C_0$,$A_{A1} = C_1$, we can solve for t_0, and t_1.

If we set equation 5.42 equal to 5.43 we can obtain a formula for t_1.

$$t_1 = \frac{(V_{A0} - V_{T0}) + (A_{A0} - A_{T0}) \cdot t_0}{-(A_{A1} - A_{T1})} \qquad (5.47)$$

Setting equation 5.45 equal to equation 5.46, substituting equation 5.47 for t_1, expanding and collecting, we obtain the following.

$$T_0^2 \left(\frac{1}{2} \cdot \delta A_0 - \frac{1}{2} \cdot \frac{\delta A_0^2}{\delta A_1} \right) + T_0 \left(\delta V_0 - \frac{\delta V_0 \cdot \delta A_0}{\delta A_1} \right) + \left(\delta P_0 - \frac{1}{2} \cdot \frac{\delta V_0^2}{\delta A_0} \right) = 0 \qquad (5.48)$$

Where:

t_0 is instant time 0 or now

t_1 is instant time 1 or the switch point

t_2 is instant time 2 or the intercept point

A_{p0} is plant acceleration at time 0

A_{t0} is target acceleration at time 0

δA_0 is plant acceleration at time 0 minus target acceleration at time 0

δA_1 is plant acceleration at time 1 minus target acceleration at time 1

δV_0 is plant velocity at time 0 minus target velocity at time 0

δP_0 is plant position at time 0 minus target position at time 0

T_0 is the time between t_0 and t_1

T_1 is the time between t_1 and t_2

This can be solved for t_0 using the quadratic formula. As t_0 approaches 0 it is time to start accelerating in the opposite direction.

$$\begin{aligned}((-P_{p0} - V_{p0}T_0 - V_{p0}T_1 - \frac{1}{2}(V_{t0} + A_{t0}T_0 + A_{t1}T_1 - V_{p0})T_1) + \\ (P_{t0} + V_{t0}T_0 + \frac{1}{2}A_{t0}T_0^2 + V_{t0}T_1 + A_{t0}T_0T_1 + \frac{1}{2}A_{t1}T_1^2))/ \\ (\frac{1}{2}T_0^2 + t_0T_1 - \frac{1}{2}T_0T_1) = A_{p0}\end{aligned} \tag{5.49}$$

This control algorithm rejects disturbances optimally. However, it only commands full forward or full reverse ($T_0 > 0$ or $T_0 < 0$), which we know will be problematic in a real system. This issue arises because our sensor feedback is inexact, and our real system is not exactly as modeled. This erratic switching can be prevented by creating a linear control region around the switching point. This can be accomplished using algorithm 2.

Algorithm 2 Calculate U, the command signal.

```
Pick $U$ to accelerate toward the target.
Set $A_{p0}$ to maximum possible acceleration given $U$ and current velocity.
Set $A_{p1}$ to minimum possible acceleration given $U$, current velocity, and prediction of $T_0$.
Solve for $T_0$ and $T_1$, using equation 5.48 and the quadratic formula.
if No Valid Values for $T_0$, or $T_1$ exist then {accelerating towards the target failed, try accelerating away}
    Pick $U$ to accelerate away from the target.
    Set $A_{p0}$ to maximum possible acceleration given $U$ and current velocity.
    Set $A_{p1}$ to minimum possible acceleration given $U$, current velocity, and prediction of $T_0$.
    Solve for $T_0$ and $T_1$, using equation 5.48 and the quadratic formula.
end if
Require: $T_0 >= C$ and $T_1 >= C$ {This detunes the algorithm, to prevent erratic switching.}
Calculate $A_{p0}$ based on equation 5.49
Use a system model to estimate $U$, the command signal, given the desired acceleration rate, and current velocity.
```

By applying this algorithm, the response of the controller is detuned to allow for unmodeled dynamics, modeling errors, and disturbances. When on path, and at the target's velocity, this controller method defaults to feed forward control. This control strategy takes advantage of the known current and voltage limits of the amplifiers, the plant model, as well as some future path knowledge. It also has the advantage of being easy to compute, making it applicable to servo motor control. Figures 5.36, 5.37, 5.38, and 5.39 show the responses to various disturbances of an identical system with three controllers. The PD controller (as described previously) is tuned to give an optimal response in one of the examples. This is done to demonstrate that as it does not consider saturation limits, it can not be optimal in all cases (see figure 5.36). Note that the proposed controller, minimum-time tracking control (MTTC), rejects the disturbance completely the quickest, and performs substantially better to the range of examined disturbances (figures 5.36, 5.37, 5.38, and 5.39).

5.6 Observers

Minimum-time tracking control, as described, does not take advantage of past errors. This is acceptable, assuming a static plant, and small, or random disturbances, as there is nothing to learn. This is the case for finish machining, which accounts for a large portion of high speed machining (high servo speed). The remaining work, presumably rough machining is unlikely to have the same tolerances on path error. However it is possible to take advantage of this final source of information to improve servo performance.

At this point the question arises, as to what to observe. In the case of MTTC, there are three system variables of interest. These are the mass of the system, the damping of the system, and any disturbance acting on the system. Typically, one would not expect the system damping, or mass to unexpectedly change during machining.

Possible causes for varying plant mass include a varying work piece mass, a gantry style machine with two parallel motors (the mass seen by each motor will vary depending on the position of the Y-axis carriage), or an unknown machine mass due to it being constructed with modular components [11]. For the purposes of high speed machining, an unknowable mass seems inapplicable. Attempting to observe the varying mass arising from a gantry style machine also seems unwise, as this could be predicted and compensated for in a straight forward manner. The third possibility is

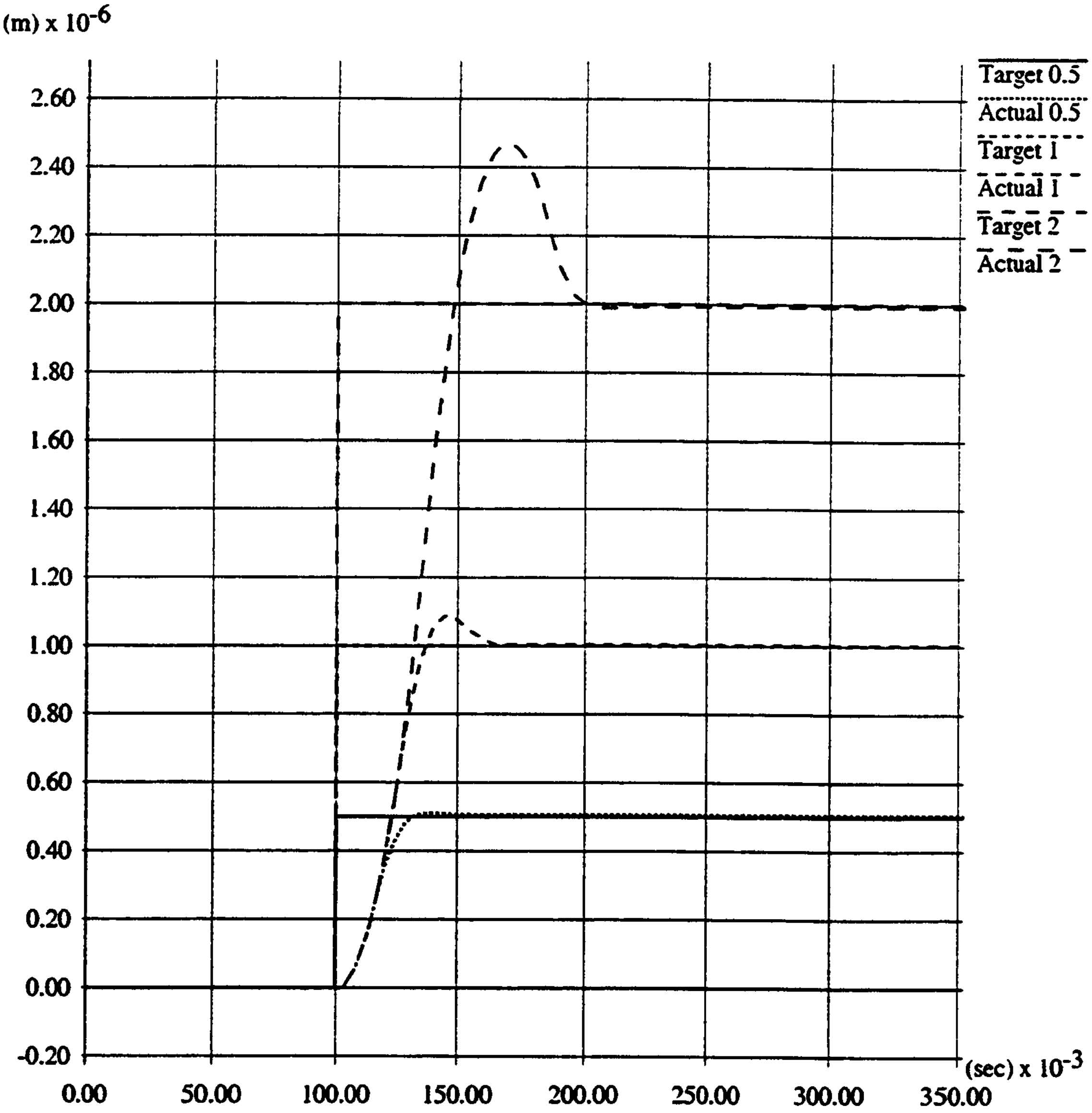


Figure 5.36: Responses of a PD controller to step inputs of various amplitudes

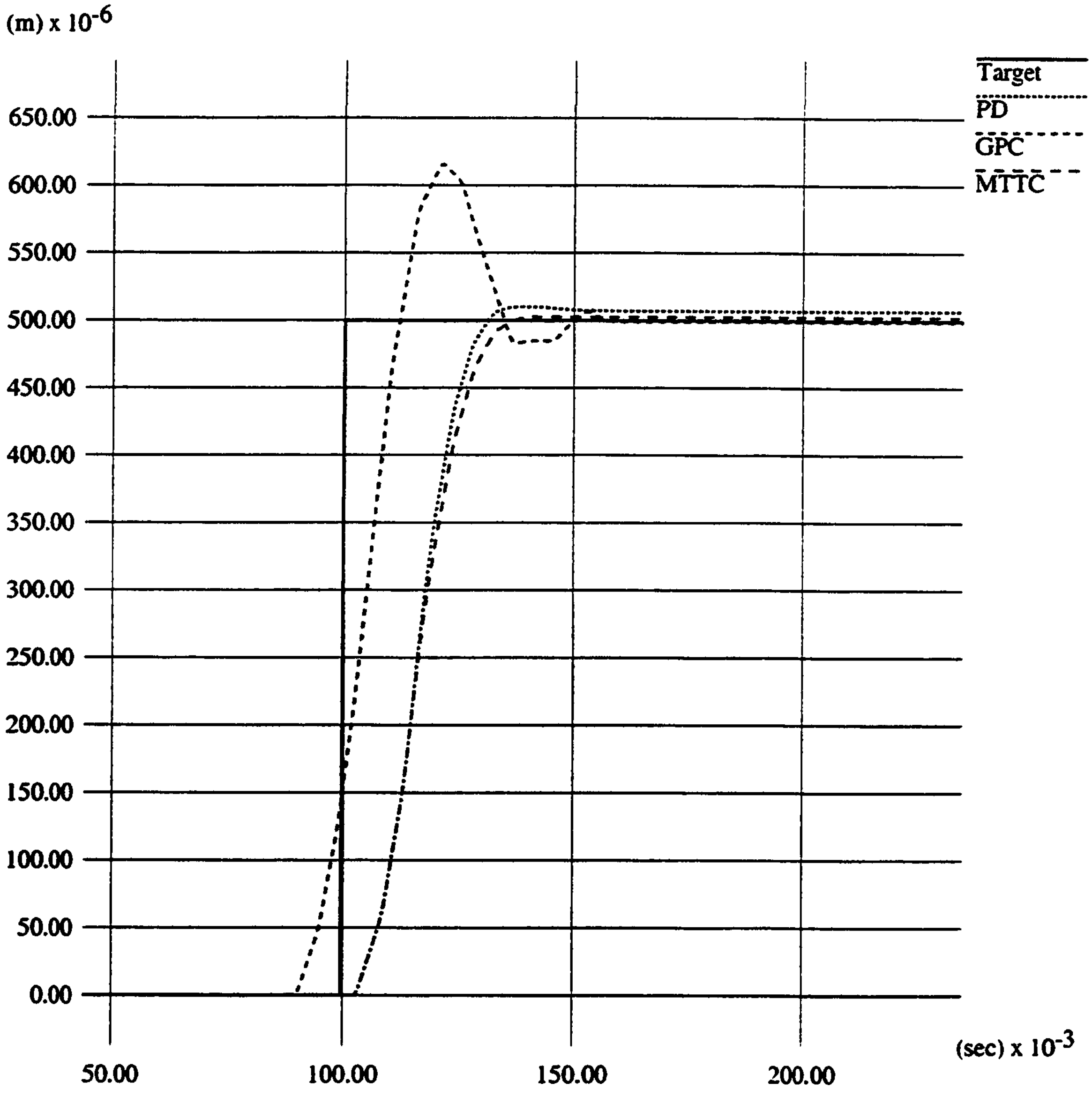


Figure 5.37: Responses of PD, MTTC and GPC controllers to a 0.0005 m step input

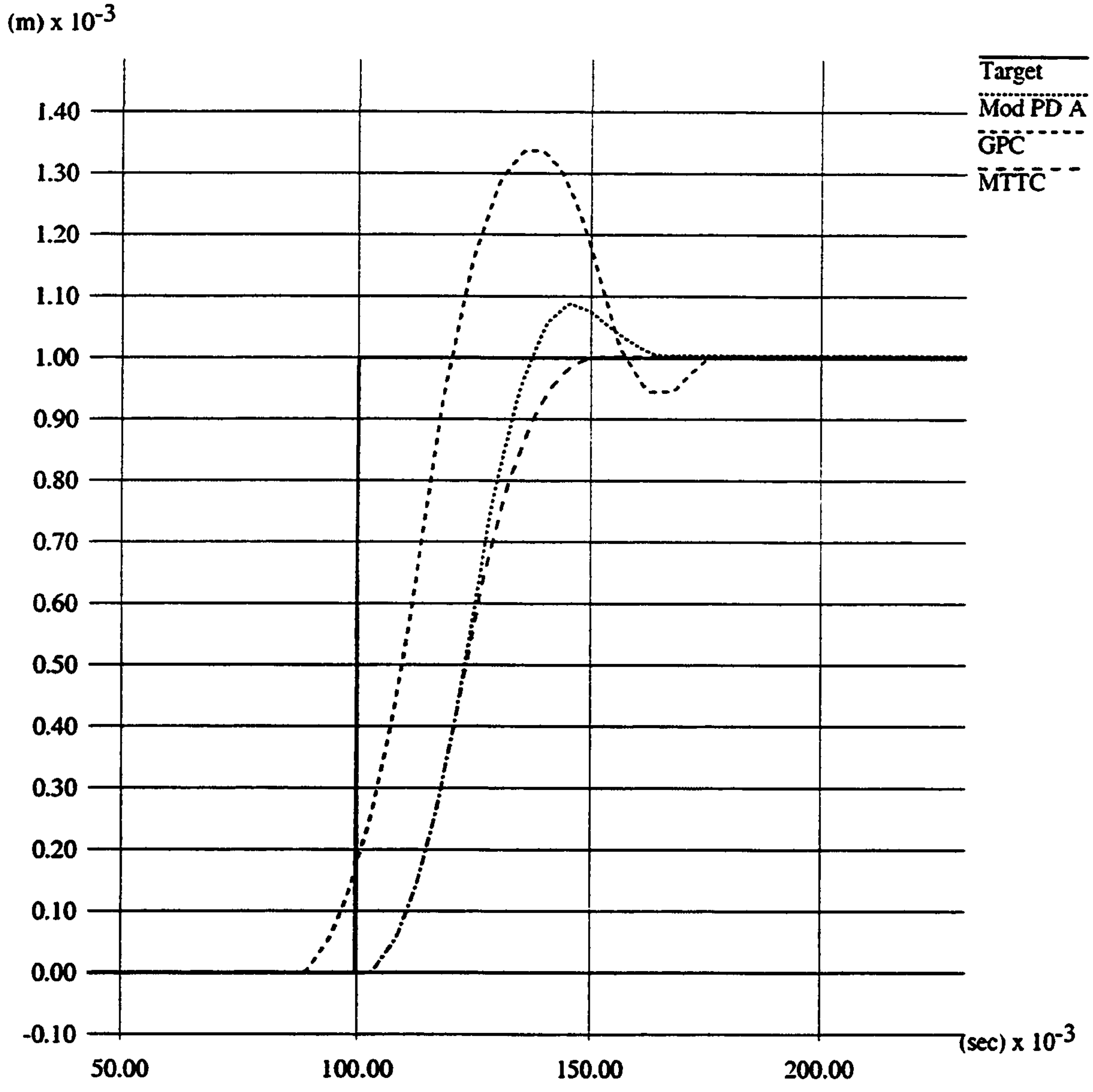


Figure 5.38: Responses of PD, MTTC and GPC controllers to a 0.001 m step input

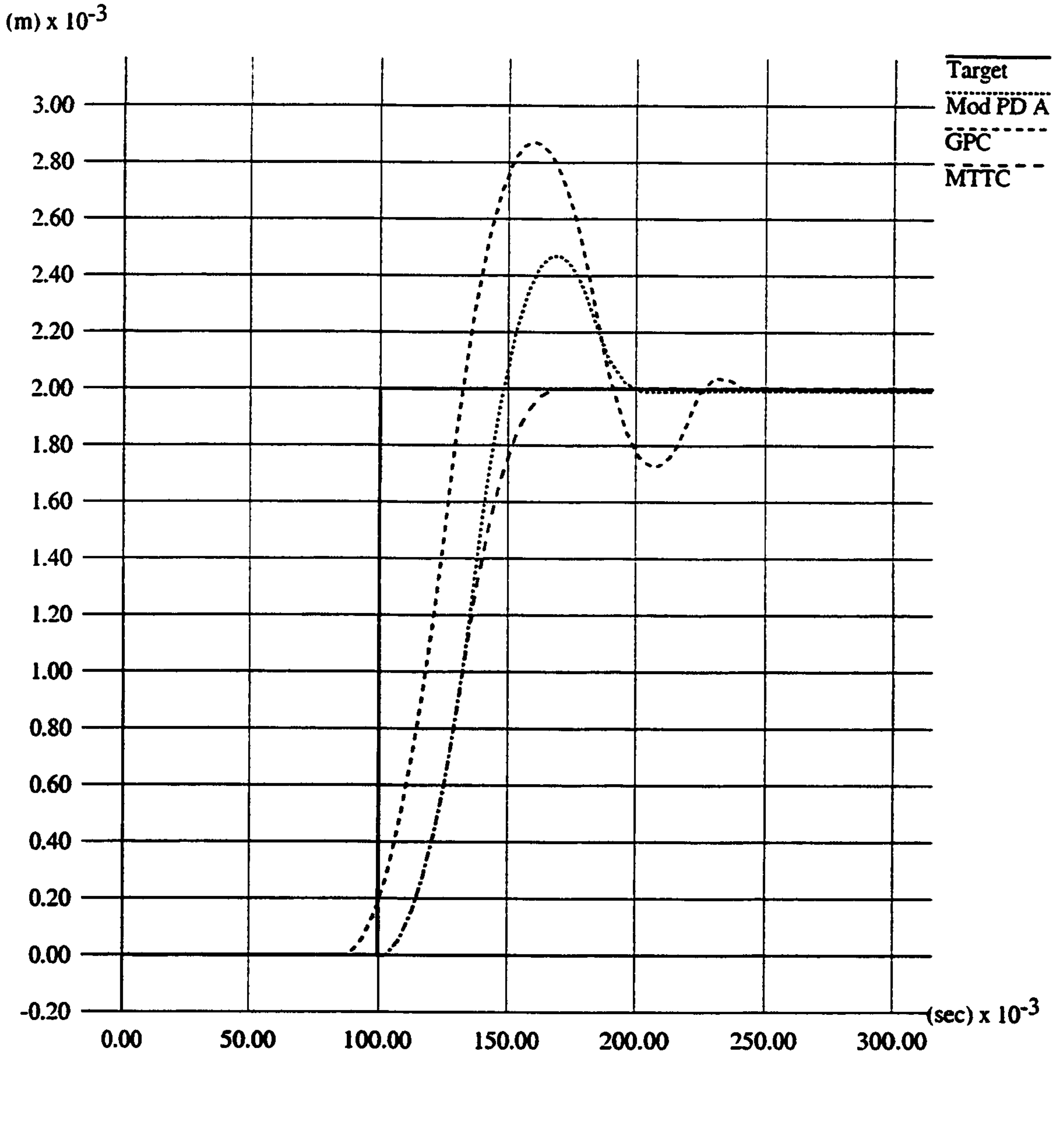


Figure 5.39: Responses of PD, MTTC and GPC controllers to a 0.002 m step input

the varying work piece mass, this should simply be measured at the beginning of a machine cycle. It is possible that the mass of the workpiece will change slowly during machining, and in this case an observer may be useful, however in the presence of numerous disturbances, it can be difficult to distinguish between a varying mass, and a disturbance. It will be argued that the estimate of the mass of the part should be updated during any rapid traverses, where there are no disturbances, and the effects of system mass are clearly measurable.

The final possible use for an observer is to compensate for disturbances, such as cutting forces. From a practical point of view, this is the most likely source of unpredicted plant variation, and the most likely cause of unexpected plant behavior.

Based on these assumptions, two additions were made to the test plant MTTC controller, the first was an automated testing sequence, which was designed specifically to produce the conditions where the plant mass, damping, and disturbance (perhaps gravity) could each be measured accurately. The second was an online disturbance observer. Each of these were based on least squares estimation, the methodology used is described in formulas 5.50 through 5.61. The results are impressive, and can clearly be seen in figures 6.56 through 6.58.

For the case of the automated testing sequence, the equation relating the command signal, mass, acceleration, velocity, damping, and disturbance is:

$$C = m \cdot a + v \cdot d + D + error \tag{5.50}$$

Where:

C	is the command to the motor.
m	is proportional to the table mass.
a	is the acceleration of the table in $\frac{m}{s^2}$.
v	is the velocity of the table in $\frac{m}{s}$.
d	is proportional to the velocity proportional damping.
D	is proportional to the disturbance force.

$$error = C - m \cdot a - v \cdot d - D \tag{5.51}$$

$$error^2 = (C - m \cdot a - v \cdot d - D)^2 \tag{5.52}$$

$$\frac{derror^2}{dD} = 2 \cdot (C - m \cdot a - v \cdot d - D) \cdot (-1) \tag{5.53}$$

$$\frac{derror^2}{da} = 2 \cdot (C - m \cdot a - v \cdot d - D) \cdot (-m) \tag{5.54}$$

$$\frac{derror^2}{dv} = 2 \cdot (C - m \cdot a - v \cdot d - D) \cdot (-v) \tag{5.55}$$

The squared error is minimized when $\frac{derror^2}{d}$ are zero. or:

$$\sum_{i=1}^{n} C_i = \sum_{i=1}^{n}(m \cdot a_i + v_i \cdot d + D) \tag{5.56}$$

$$\sum_{i=1}^{n}(C_i \cdot a_i) = \sum_{i=1}^{n}(m \cdot a_i + v_i \cdot d + D) \cdot a_i \tag{5.57}$$

$$\sum_{i=1}^{n}(C_i \cdot v_i) = \sum_{i=1}^{n}(m \cdot a_i + v_i \cdot d + D) \cdot v_i \tag{5.58}$$

In matrix notation $[Y] = [k] \cdot [X]$

$$Y = \begin{bmatrix} \sum_{i=1}^{n} C_i \\ \sum_{i=1}^{n}(C_i \cdot a_i) \\ \sum_{i=1}^{n}(C_i \cdot v_i) \end{bmatrix} \tag{5.59}$$

$$k = \begin{bmatrix} D & m & d \end{bmatrix} \tag{5.60}$$

$$X = \begin{bmatrix} \sum_{i=1}^{n} 1 & \sum_{i=1}^{n} a_i & \sum_{i=1}^{n} v_i \\ \sum_{i=1}^{n} a_i & \sum_{i=1}^{n} a_i^2 & \sum_{i=1}^{n}(a_i \cdot v_i) \\ \sum_{i=1}^{n} v_i & \sum_{i=1}^{n}(a_i \cdot v_i) & \sum_{i=1}^{n} v_i^2 \end{bmatrix} \tag{5.61}$$

This can then be solved by multiplying both sides by X^{-1}

In the case of the on-line disturbance observer formula 5.50 is rearranged as follows:

$$C - m \cdot a - v \cdot d = D + error \tag{5.62}$$

$$error = C - m \cdot a - v \cdot d - D \tag{5.63}$$

$$error^2 = (C - m \cdot a - v \cdot d - D)^2 \tag{5.64}$$

$$\frac{derror^2}{dD} = 2 \cdot (C - m \cdot a - v \cdot d - D) \cdot (-1) \tag{5.65}$$

The squared error is minimized when $\frac{derror^2}{dD}$ is zero. or:

$$\sum_{i=1}^{n}(C_i - m \cdot a_i - v_i \cdot d) = \sum_{i=1}^{n}(D) \tag{5.66}$$

In matrix notation $[Y] = [k] \cdot [X]$

$$Y = \left[\sum_{i=1}^{n}(C_i - m \cdot a_i - v_i \cdot d) \right] \tag{5.67}$$

$$k = \left[D \right] \tag{5.68}$$

$$X = \left[\sum_{i=1}^{n} 1 \right] \tag{5.69}$$

This can then be solved by multiplying both sides by X^{-1}. This shows that the least squares estimator in this case is simply the average estimate of D.

This scheme however does not address issues due to interrupted cutting. A sample of the forces resulting from cutting is shown in figure 5.40 Perhaps the solution is simply to use a high helix tool, and a sufficiently large enough depth of cut that the forces on the cutter are mostly DC. However this does not address issues associated with spindle run-out, and often this option is not possible. This gives rise to the notion of observing the periodic signature of cutting forces, and compensating for them. Since the source of these disturbances is the cutter, or spindle, the disturbances are correlated with spindle rotational position. Using an encoder on the spindle,

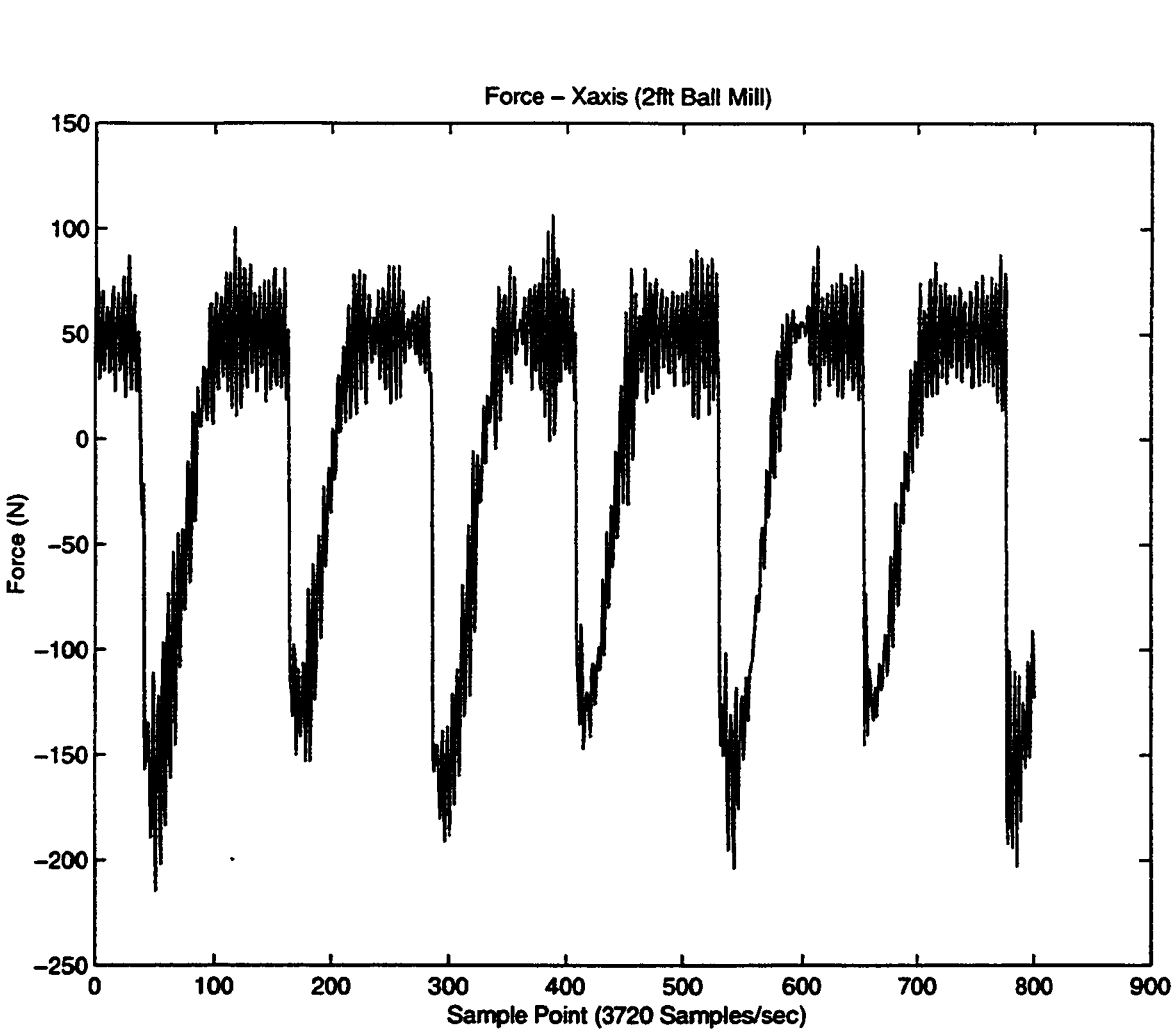


Figure 5.40: Sample cutter force

it is possible to create a straight forward record of the disturbances, in terms of spindle rotational position. Again, by using least squares analysis a prediction of the disturbance is generated, and can be compensated for. It is possible that this could be improved on somewhat, if the assumption that tool run-out was small was made, and that each tooth of the cutter was approximately identical. In this manner, the algorithm could correct it's prediction faster by a factor equal to the number of teeth on the tool. This decision will have be made on a system by system basis.

5.7 Summary

A review machine tool servo control laws reveled a number of common properties. These properties generally included an over parameterized model (especially given modeling uncertainty as high as 25%) and virtually no consideration of actuator saturation issues (GPC with input constraints being a notable exception). Most interestingly, no control algorithms were found which addressed saturation, and were (in terms of computational effort) appropriate for high speed machine tool control. Minimum-time tracking control (MTTC) was developed in an effort to satisfy these requirements. It explicitly addresses actuator saturation, and other significant plant parameters. The algorithm is also easy to compute, and easily incorporates feedback from a variety of sensors. Examples being position, velocity, and acceleration states can be fed directly to the algorithm. The result is a significantly improved ability (faster / less over shoot) to regulate disturbances, of any size. Broad-band stiffness has been improved through the use of a disturbance observers. These include a standard DC observer, and a periodic observer. The standard observer is used for automatic plant identification, in combination with a test command sequence. This identification could also be performed while jogging. The periodic observer uses the fact that the largest source of disturbances in a machine tool is the cutting process, and that these forces generated are periodic, and dependent on the spindle angular position. An encoder on the spindle is used to correlate instantaneous disturbances as a function of position. These measurements are then averaged over successive revolutions, and an accurate estimate of the disturbance force is generated. This enables the disturbance to be more completely rejected. Clearly this has limitations in terms of maximum frequency, and makes assumptions about the rate of change of cutting forces. However this disturbance estimation, and rejection method is most effective at lower frequencies, where the system is the least stiff on it's own, and therefore most in need of this extra capability. There is also an assumption that the cutting forces are slowly varying with respect to the tooth frequency, this is true in the majority of cases, and common practices, like lowering the feed rate on entrance, and exit of a cut make this assumption more true.

Chapter 6

Experimental Results

In order to validate the conclusions drawn from the simulations, and tests on simple physical plants, the tests were repeated on a commercial linear motor test stand. The results presented in this chapter were all performed on this system. Details of this system are described in section 3.3.2.

6.1 Controller Comparison

Minimum-time tracking control (MTTC) was compared to generalized predictive control (GPC) [19] [20], H^∞ [2] [49], and PD$\star$ (the PD$\star$ controller used the digital tachometer for velocity estimation, not the derivative of the position error) to determine it's relative effectiveness. Detailed descriptions of these controllers can be found in the cited papers. The H^∞ controller was generated using the "Robust Control Tool Box" from matlab, and was updated at 10 kHz (A brief description of H^∞ can be found in section 2.3.4.). The plant model used in the GPC controller was also generated using matlab, the control horizon was 30 steps, and was updated at 5 kHz. The data used to generate these controllers was that from an open loop pseudo random binary sequence (PRBS) (A brief description of GPC can be found in section 2.3.2.). The PD$\star$ controller was hand tuned, and the MTTC controller was generated using a least squares estimator as previously described.

To compare controller disturbance rejection capabilities and to emphasize the importance of including saturation information at the control loop level, the selected controllers were given small steps to regulate. A significant limiting factor for very small steps is the resolution of the position feedback mechanism, in this case one micron. The results show that only MTTC was able to consistently regulate the various steps (figures 6.41 - 6.43). Specifically, controllers which do not explicitly consider saturation limits under perform when rejecting small disturbances, and over shoot when rejecting larger disturbances. These controllers over shoot when presented with large disturbances because they exceed the current capabilities of the amplifier when decelerating. They also commanded acceleration rates which exceeded their current capabilities during the acceleration phase, however this has no effect. Evidence that these controllers are indeed saturating and causing over shoot can be seen in figure 6.47. Evidence that MTTC does not over shoot, because it does not issue commands which exceed the capabilities of the axis can be seen in figure 6.46. Systems which do not explicitly consider current limits under perform when regulating small steps because the system gain must be limited to maintain system stability, or to limit the system overshoot to more typical disturbances. This overshoot can be prevented in error based controllers "PID" by limiting the amplitude of the error signals that are fed to the controller. The result of this however is under performance when rejecting both small, and large steps. This is more difficult with GPC, or H^{∞}, however the results should be the same.

To demonstrate the dynamic performance of the controller, the transfer function of the closed loop position controller were generated (figures 6.49 - 6.55). MTTC control has the most linear (gain of one, zero phase lag) transfer function of the tested controllers, especially in the usable range of the controller. This is the most significant region for machine tool control, especially given a path which has undergone minimum-time path optimization. The small peak in the loop gain for the MTTC controller is due to the decision to accelerate with full command in the direction of the target when no intercept solution exists. This occurs when the target is accelerating away from the plant at a rate faster than the estimated limits of the machine tool. In such a situation, this is a reasonable decision, however this situation will not occur given a path which has undergone minimum-time path optimization. Given these results, the low computational over head of the controller, and the ease of tuning, MTTC is superior for the application.

6.2 Observers

In order to test the various observers, an actuator was used to push, and pull on the linear motor in a manner which simulates the forces generated in cutting. Tests were performed at a number of frequencies, and amplitudes, with no observer (figure 6.57), a traditional observer (estimate current disturbance force), and the periodic observer (estimate disturbance force as a function of spindle angular position) (figure 6.58). Clearly, the periodic observer is superior in rejecting the periodic forces generated by machining, at lower frequencies, where the effect of cutting forces on position error is greatest. This is important for linear motor driven machine tools, as the controller

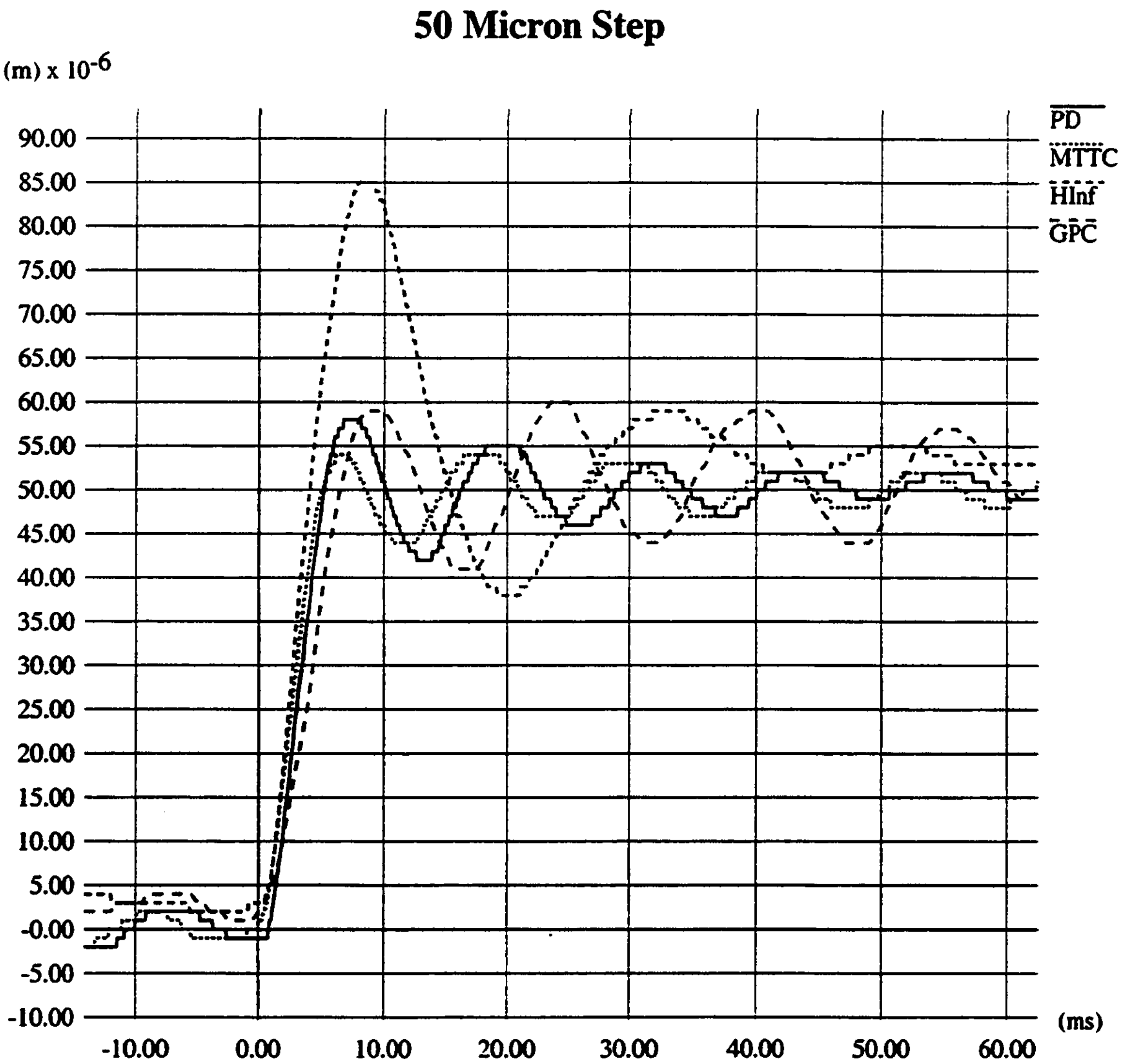


Figure 6.41: Various controllers respond to 50 micron step

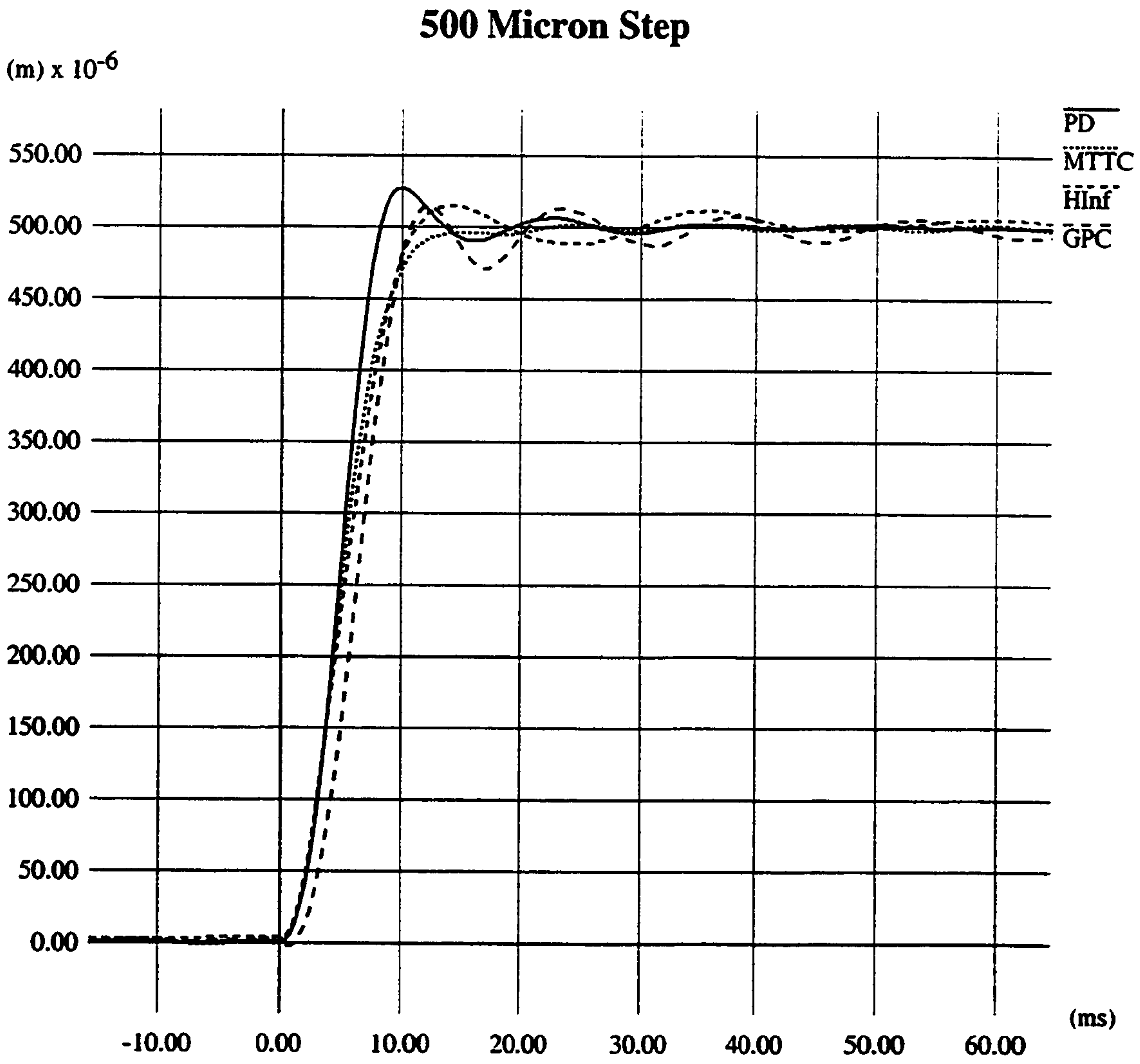


Figure 6.42: Various controllers respond to 500 micron step

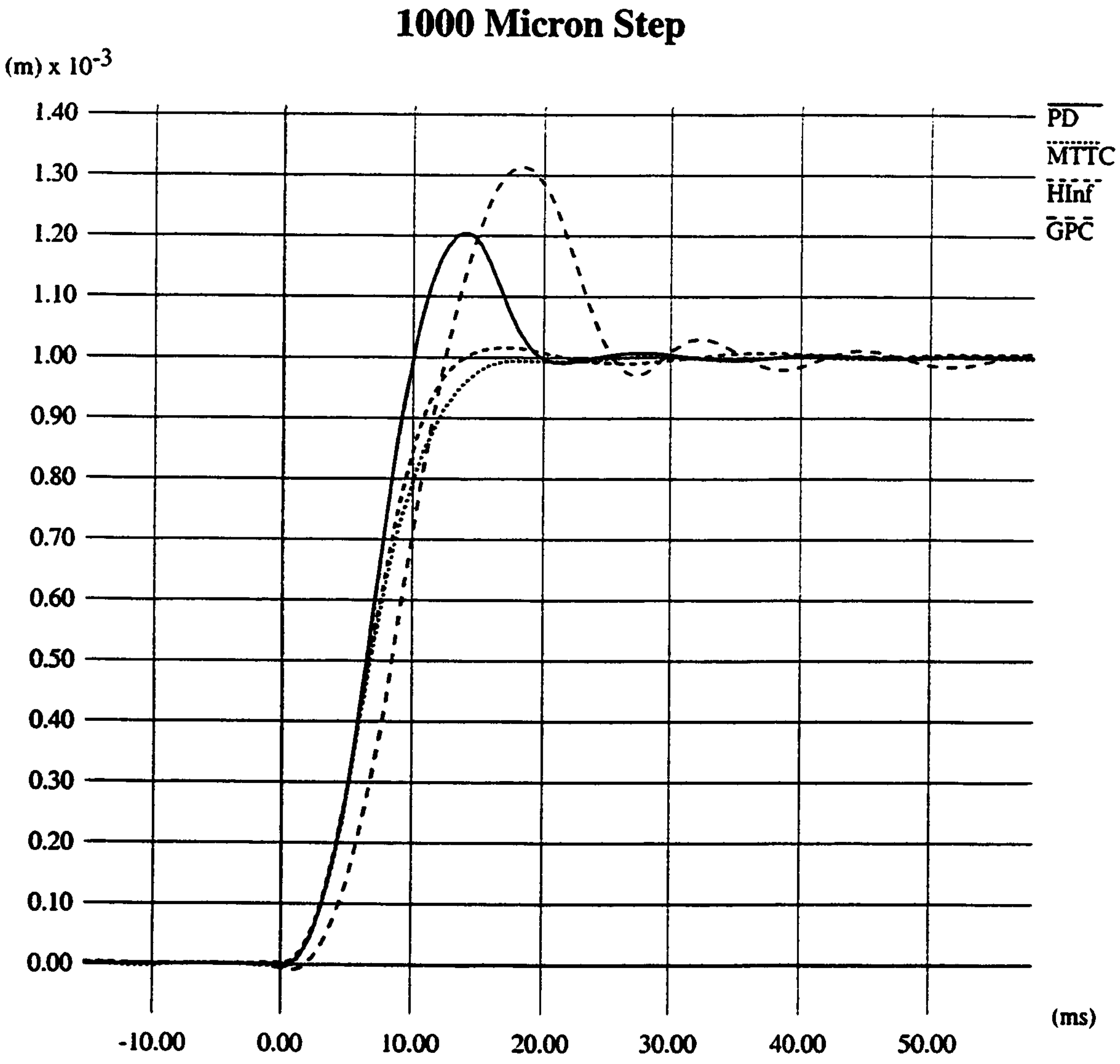


Figure 6.43: Various controllers respond to 1000 micron step

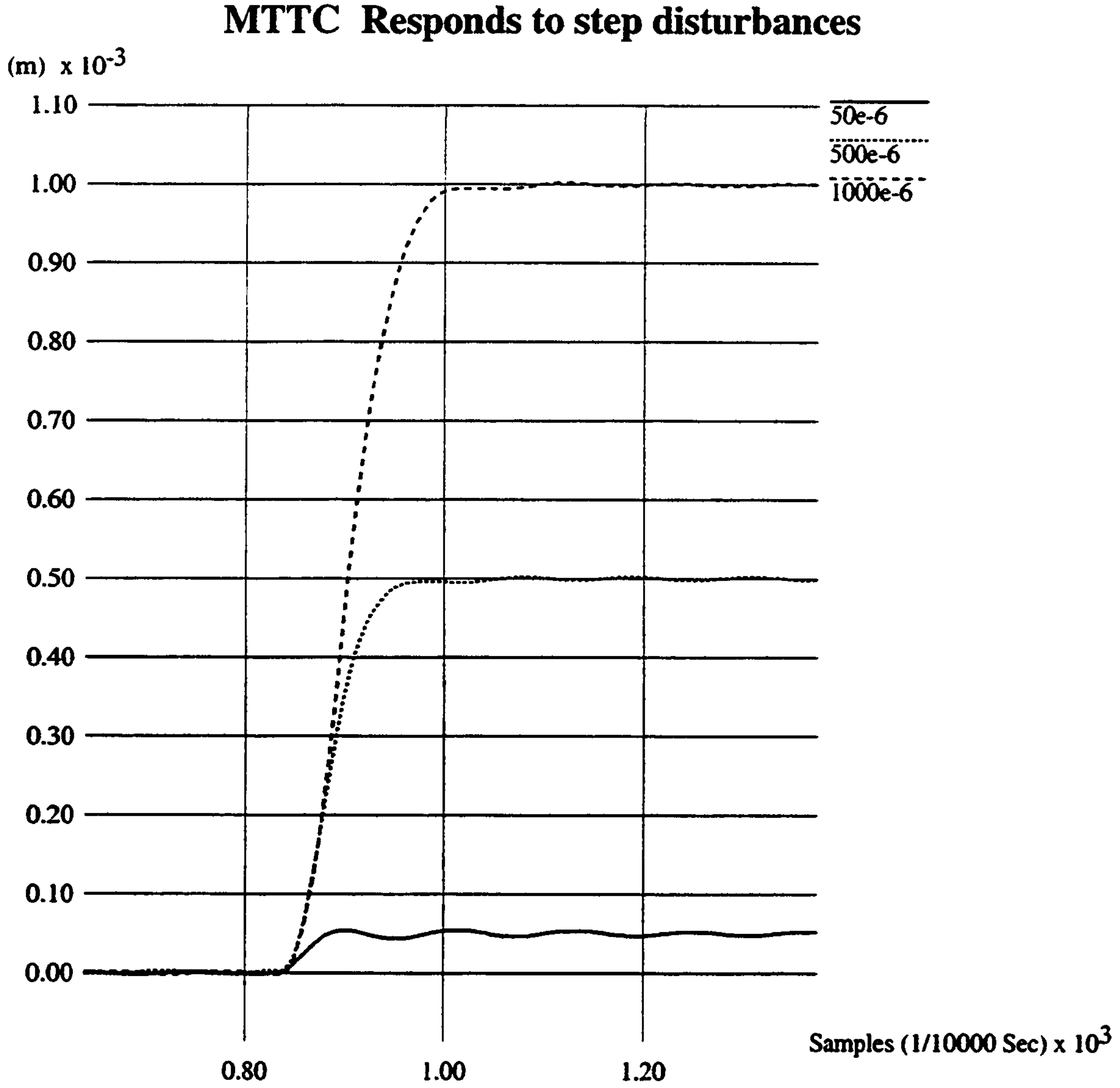


Figure 6.44: MTTC Responds to steps of various sizes

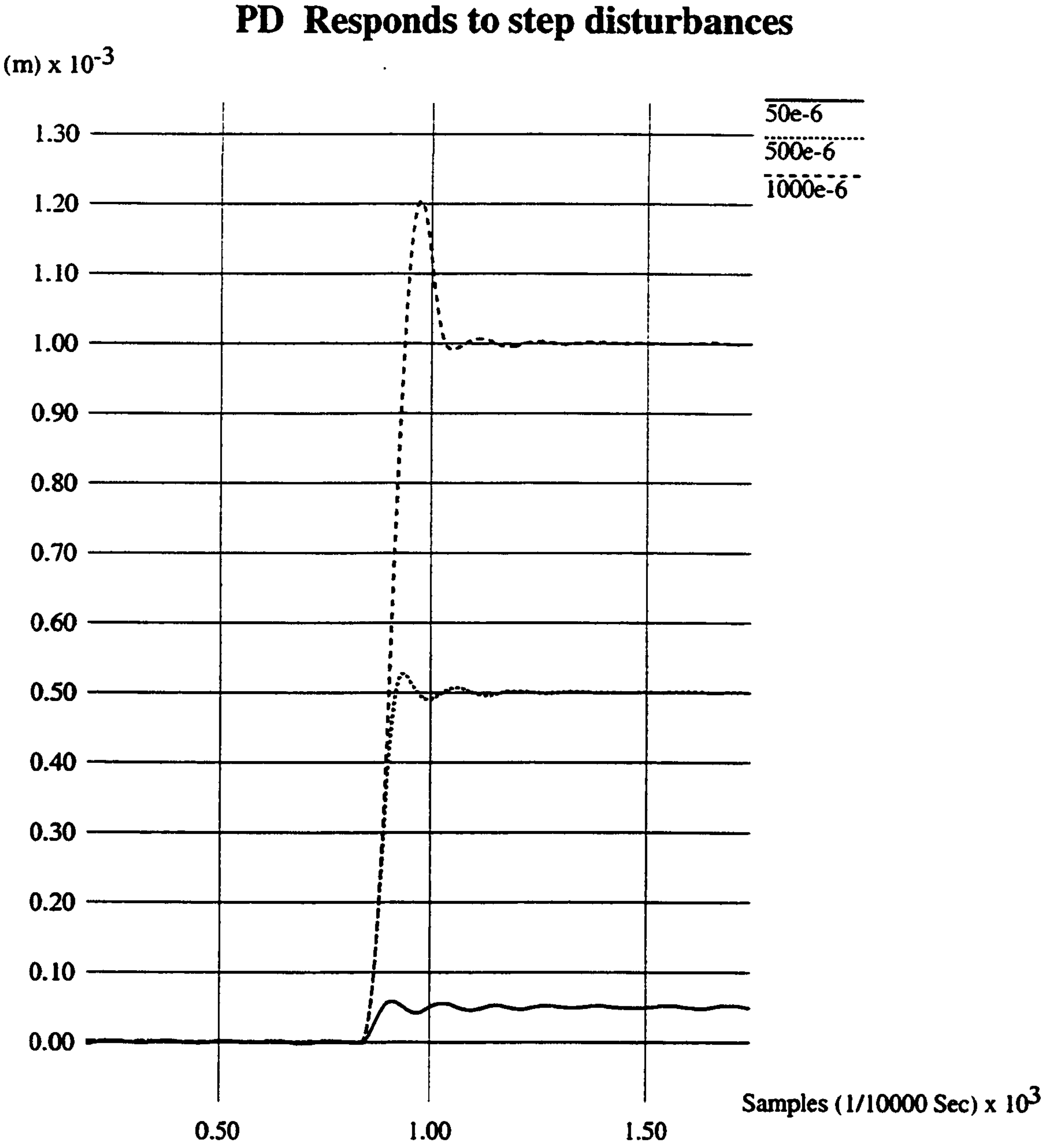


Figure 6.45: PD Responds to steps of various sizes

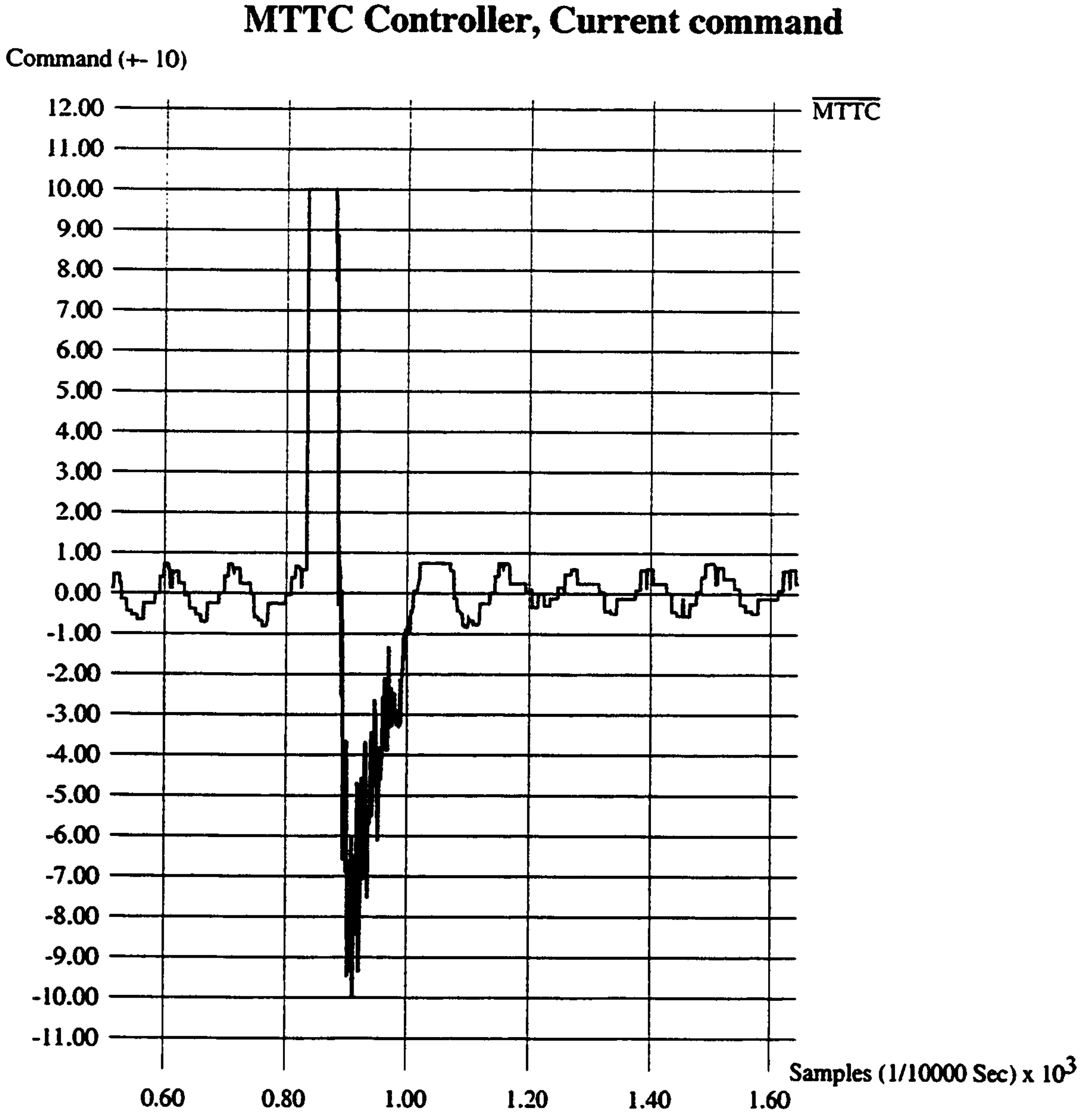


Figure 6.46: MTTC current command resulting from 1000 micron step

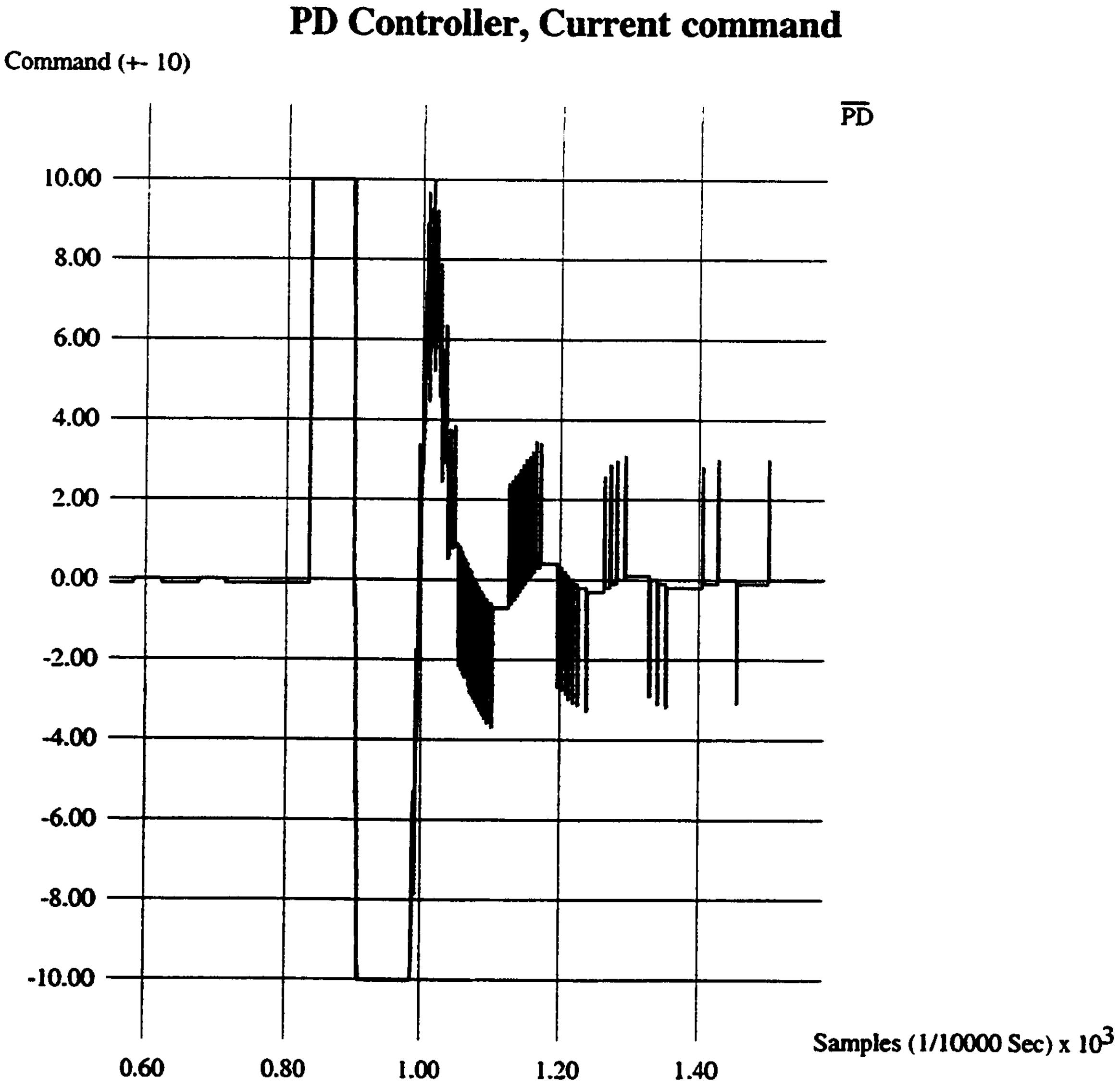


Figure 6.47: PD current command resulting from 1000 micron step

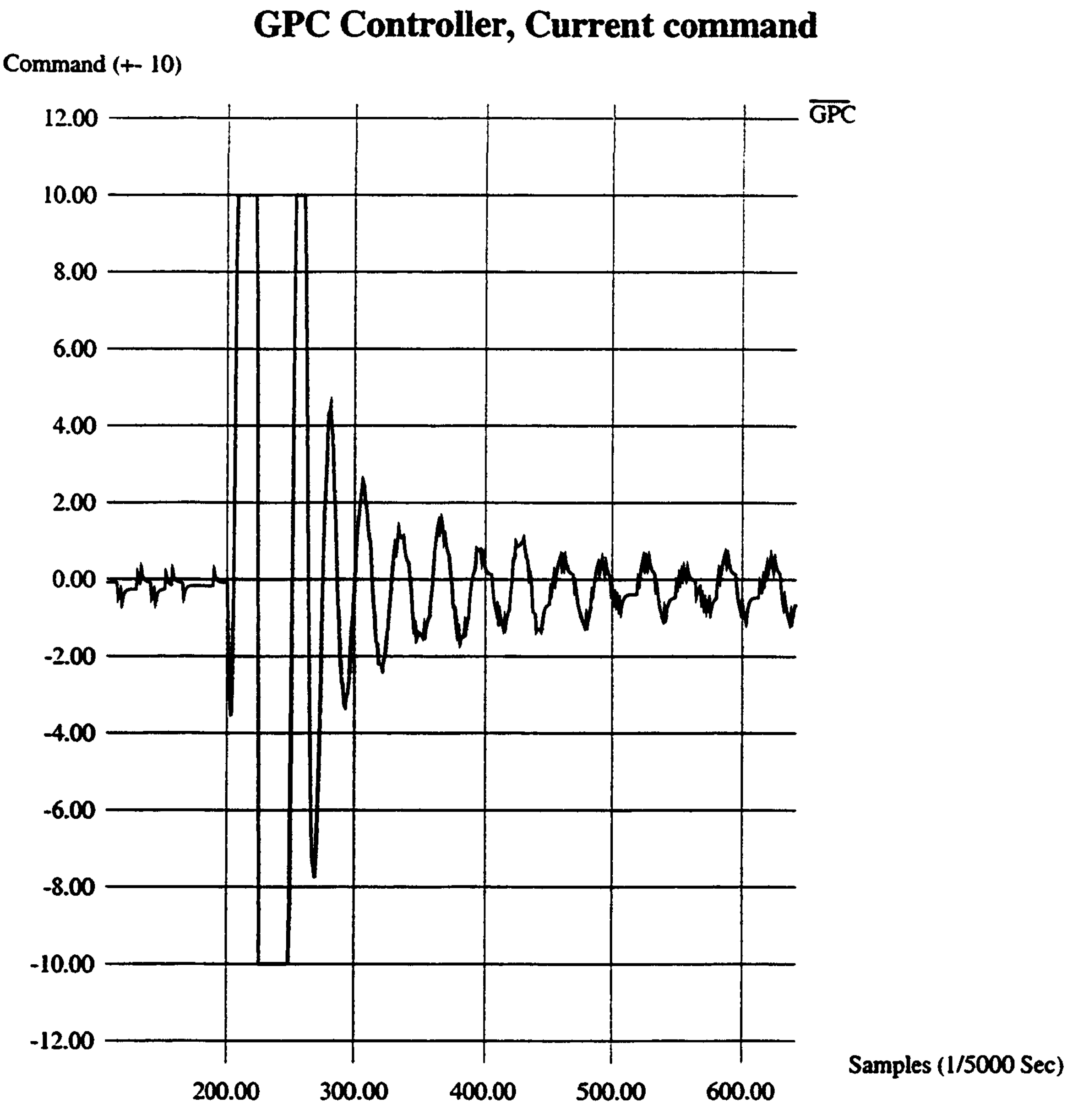


Figure 6.48: GPC current command resulting from 1000 micron step

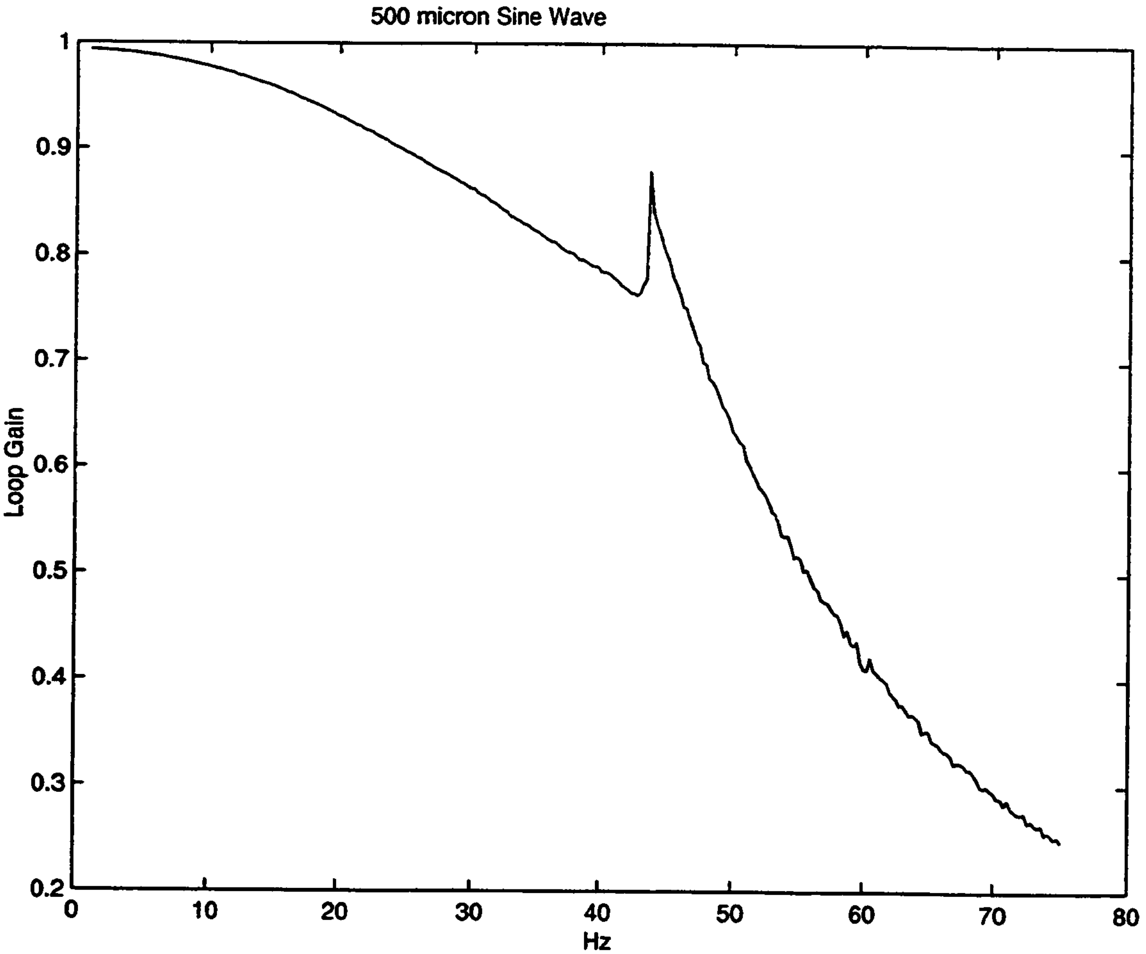


Figure 6.49: GPC Controller Magnitude of Transfer Function vs Frequency

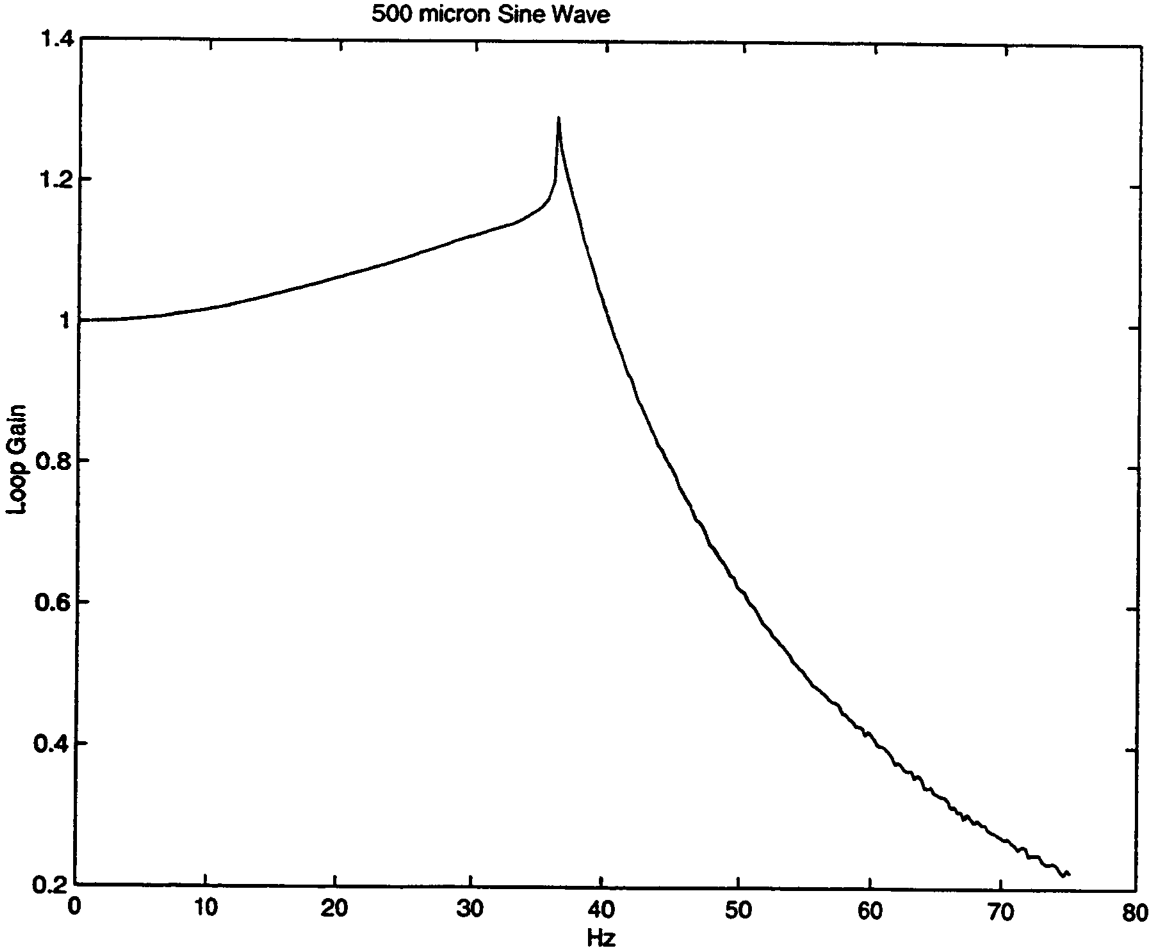


Figure 6.50: PD Controller Magnitude of Transfer Function vs Frequency

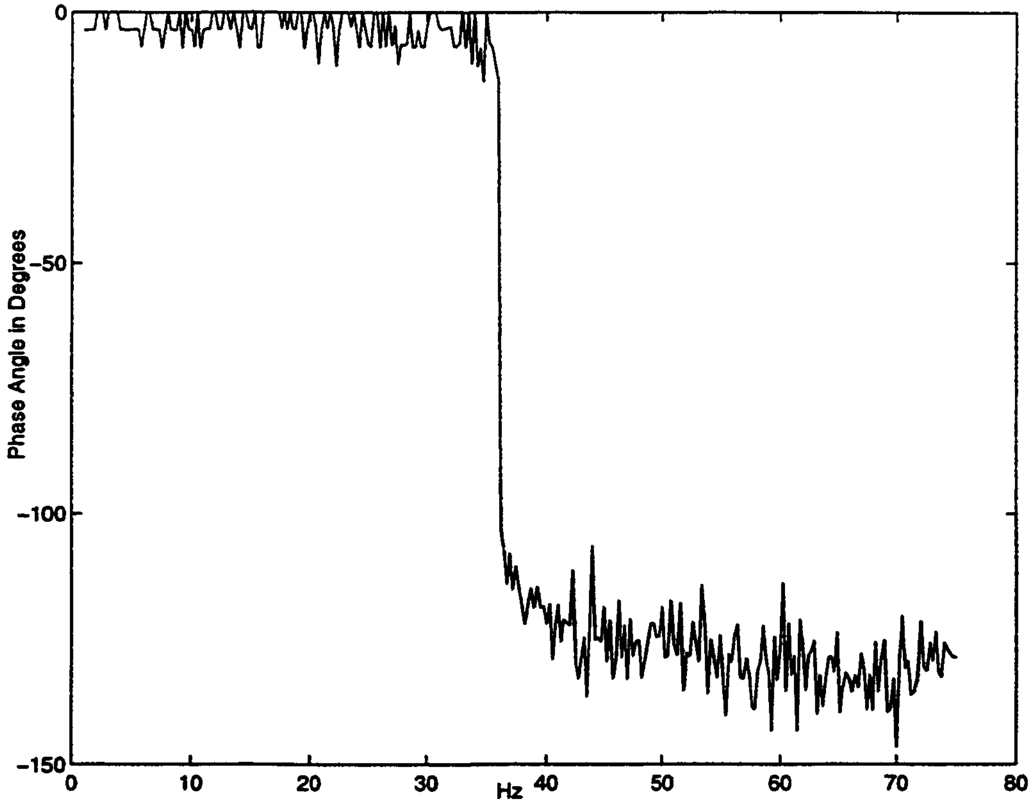


Figure 6.51: PD Controller Phase Lag of Transfer Function vs Frequency

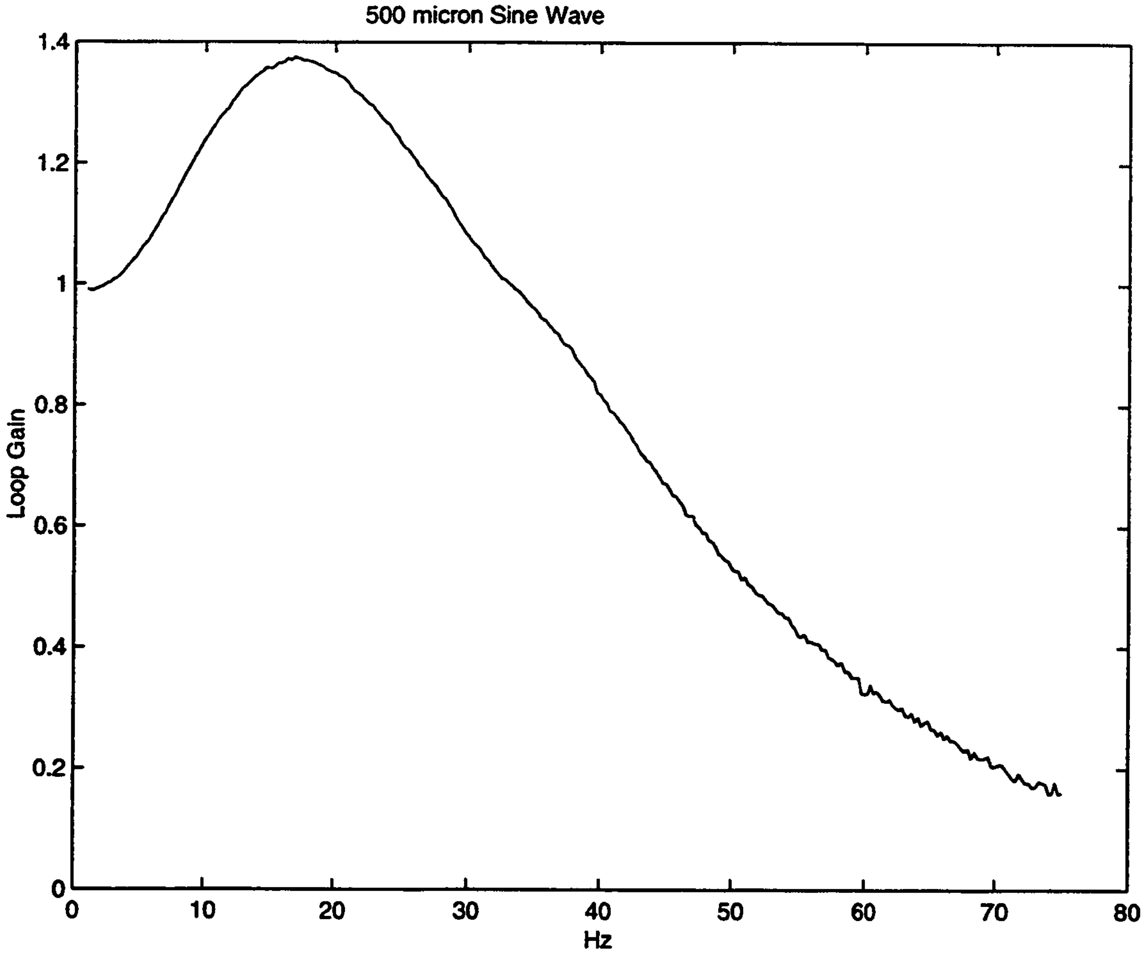


Figure 6.52: H-Infinity Controller Magnitude of Transfer Function vs Frequency

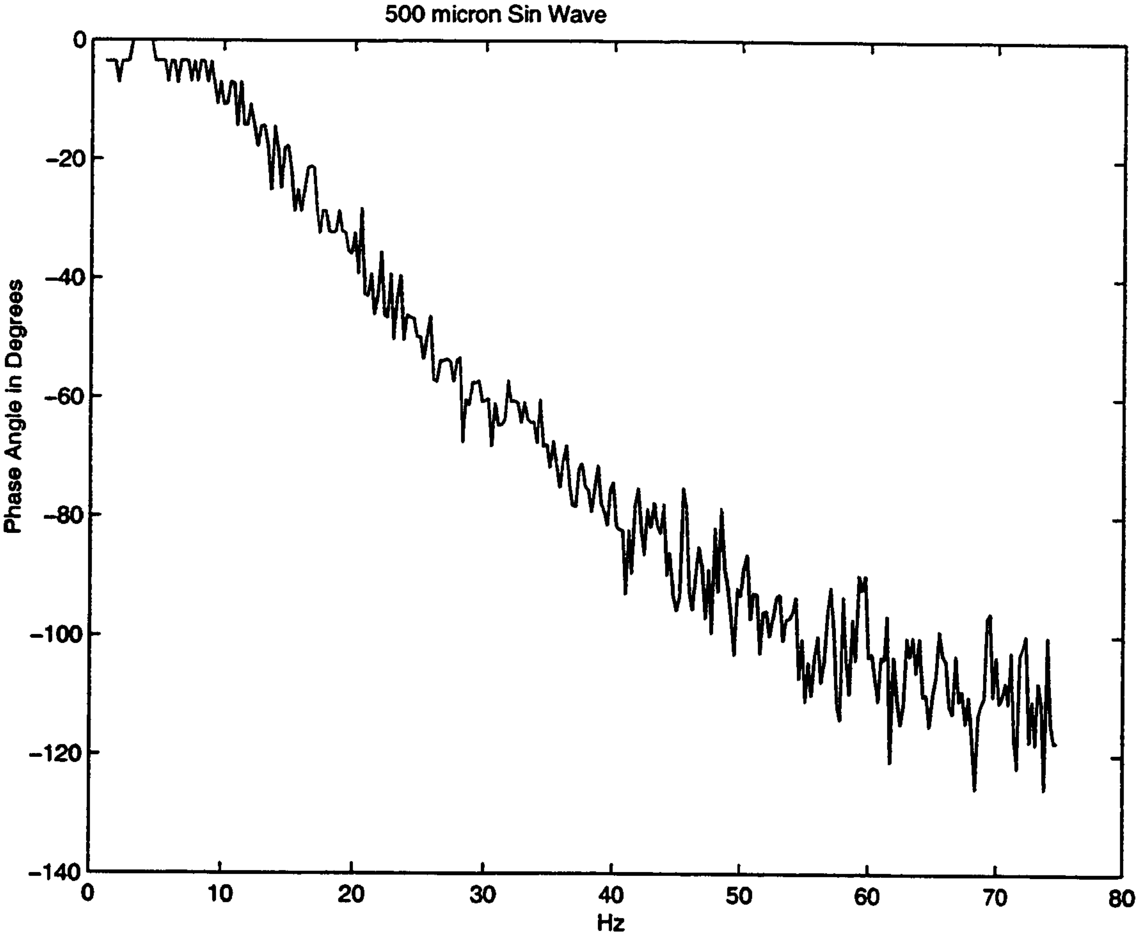


Figure 6.53: H-Infinity Controller Phase Lag of Transfer Function vs Frequency

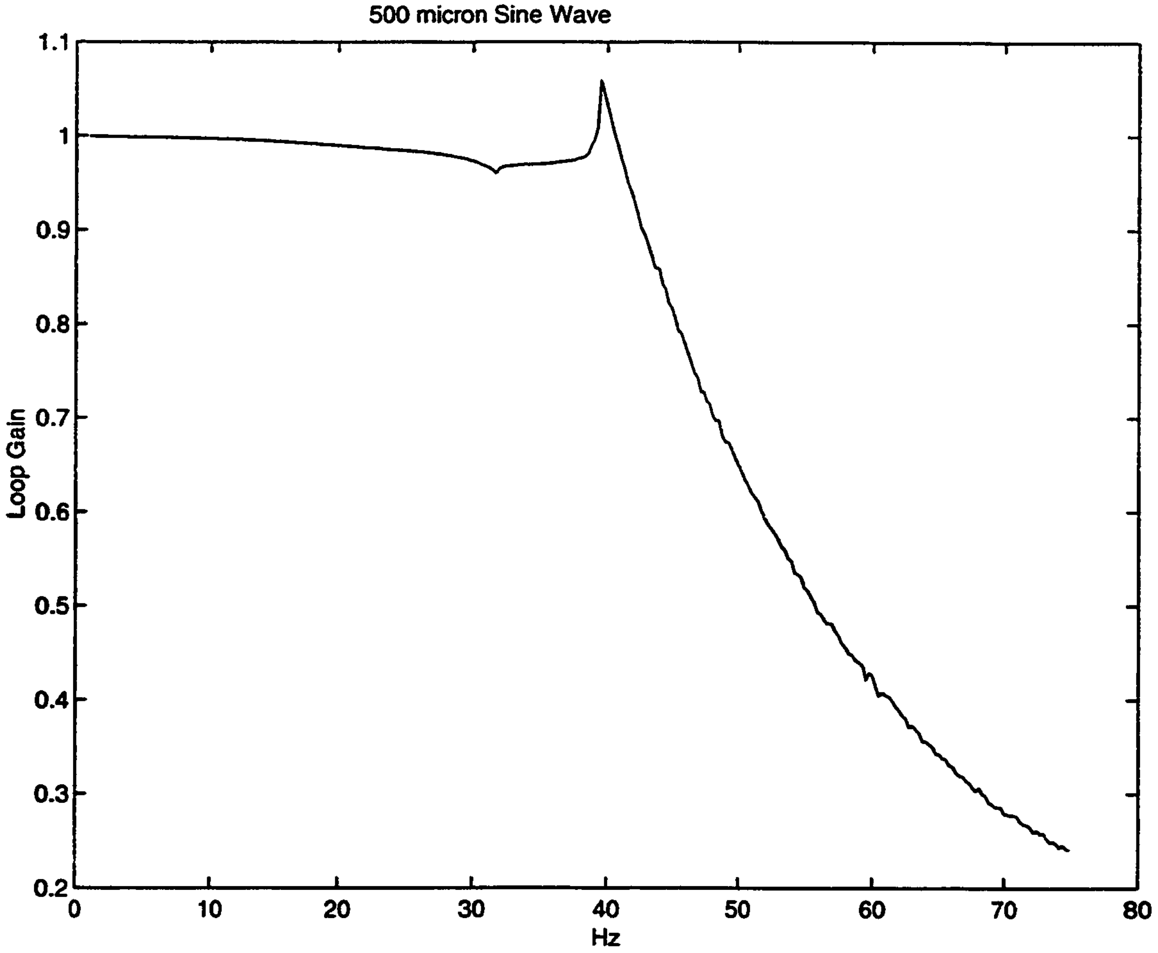


Figure 6.54: MTTC Magnitude of Transfer Function vs Frequency

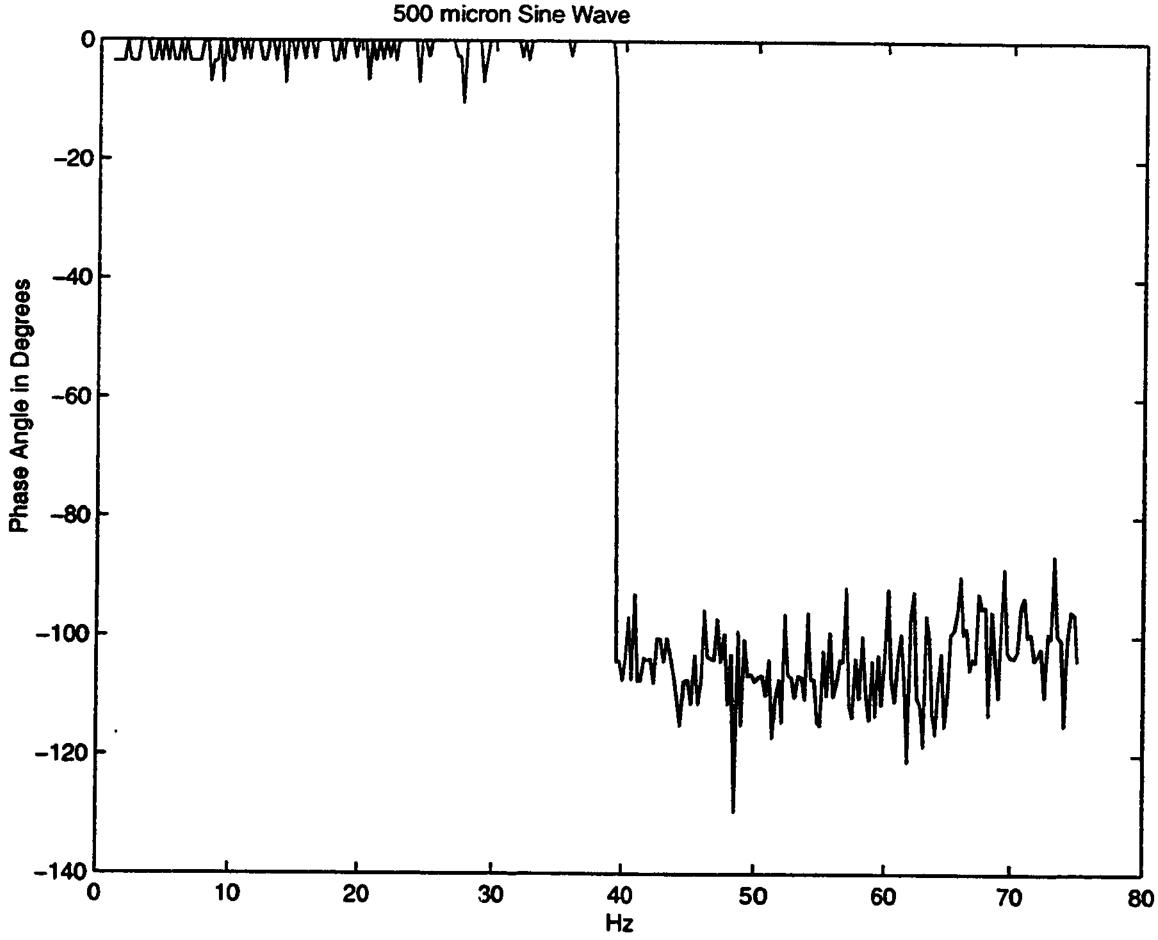


Figure 6.55: MTTC Controller Phase Lag of Transfer Function vs Frequency

must provide system stiffness at low frequencies. There will of course be limitations on the maximum disturbance frequency which can effectively be observed. However, given loop closure rates in excess of 10 kHz, high bandwidth servo amplifiers, and stiff machine designs, with natural frequencies over 500 Hz (particularly for single sided motors) the effective range for this scheme will exceed spindle speeds of 10-15000 RPM (assuming a two flute cutter). At these frequencies the machine alone will provide the required stiffness, and there will be no further need for performing the compensation. The other issue of concern is the rate at which cutting conditions vary. Rapidly changing cutter conditions will clearly reduce the effectiveness of predictions, however relative to the tooth frequency this rate is low and good prediction should be possible in the majority of cases. It is also common to reduce feed rates on entry, and exit to improve cutting results, this will reduce or eliminate the significants of transient cutting forces.

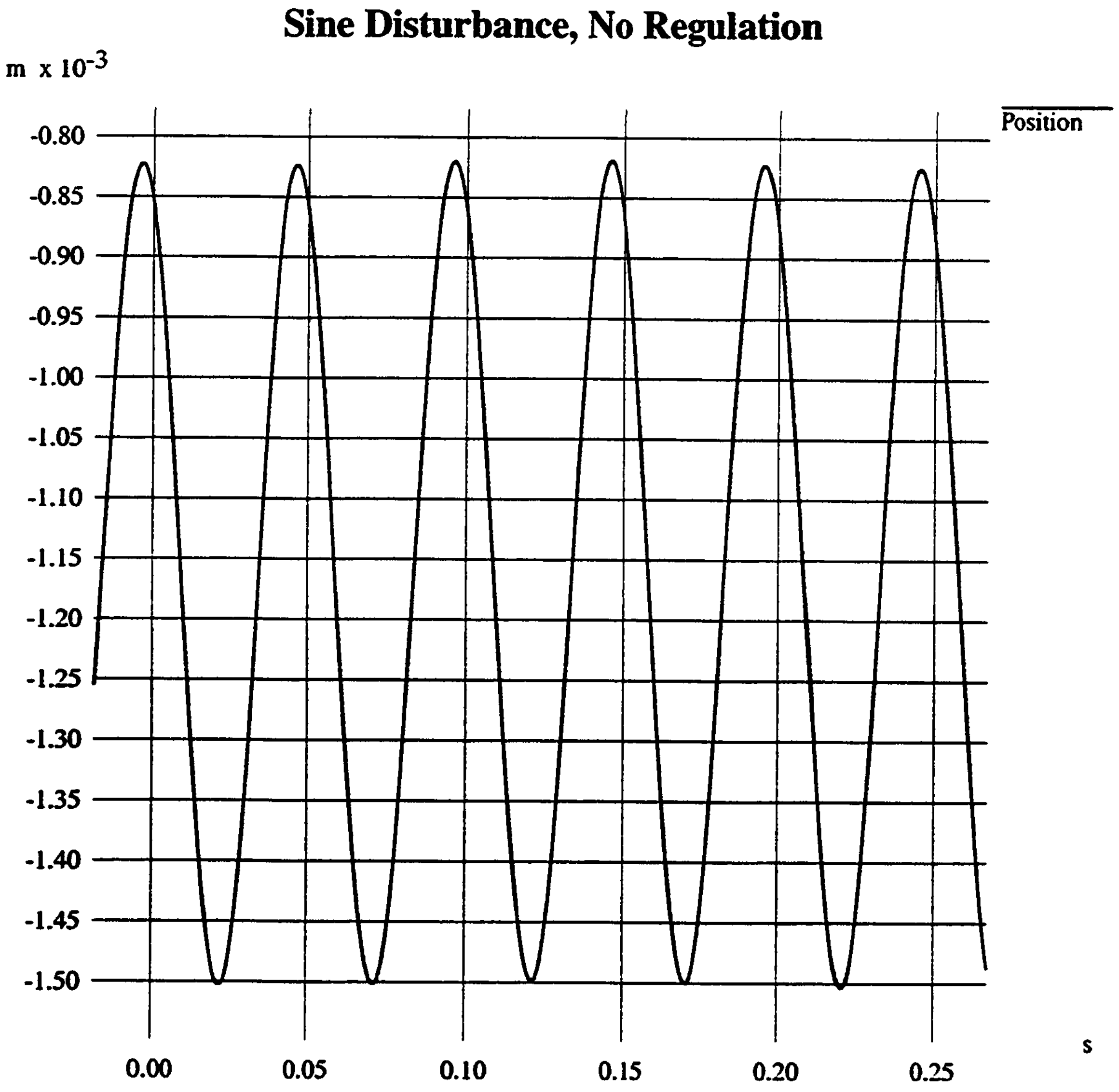


Figure 6.56: Effect of a 20Hz sinusoidal disturbance, no controller used.

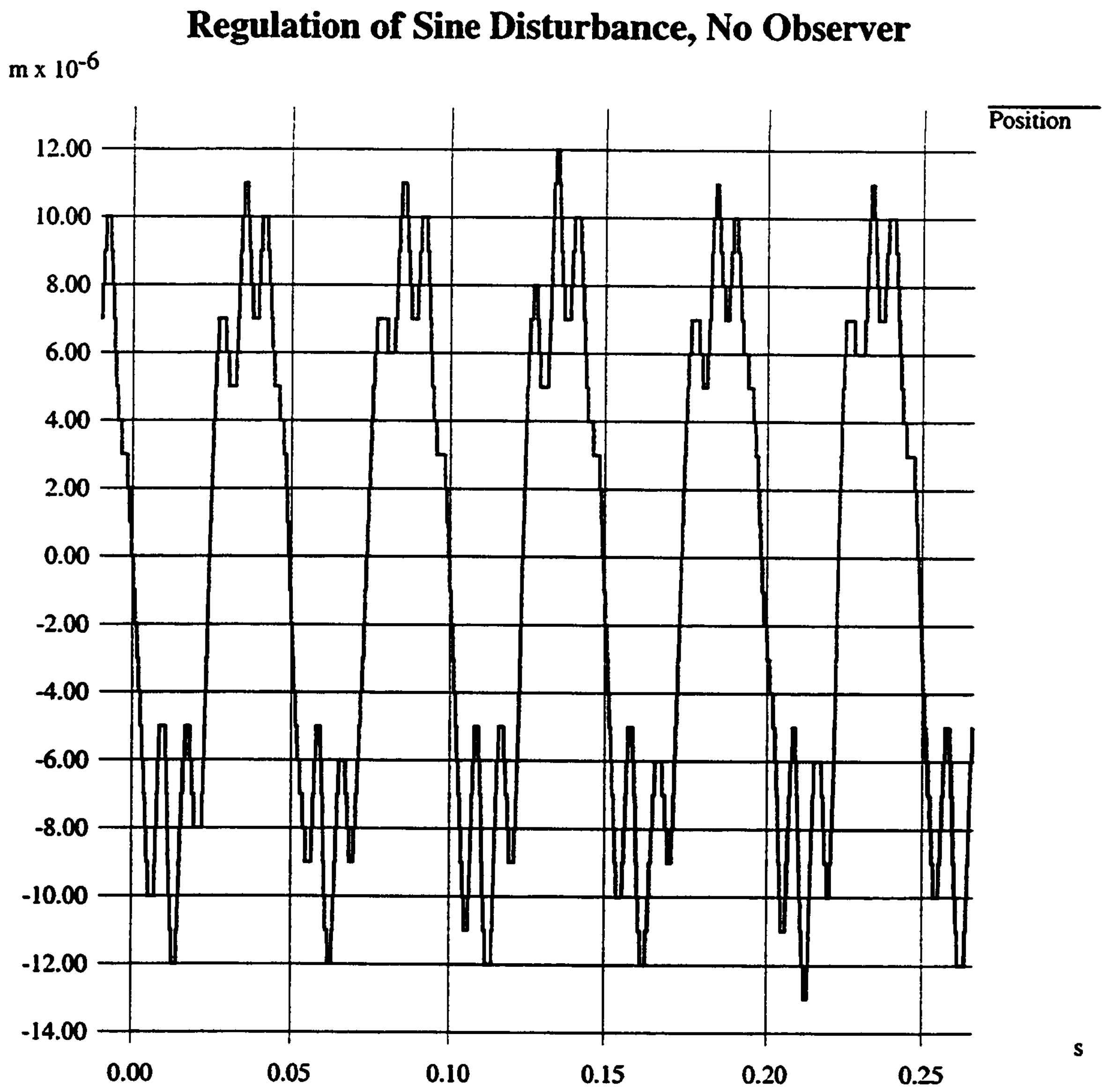


Figure 6.57: Regulation of a 20Hz sinusoidal disturbance, no observer used.

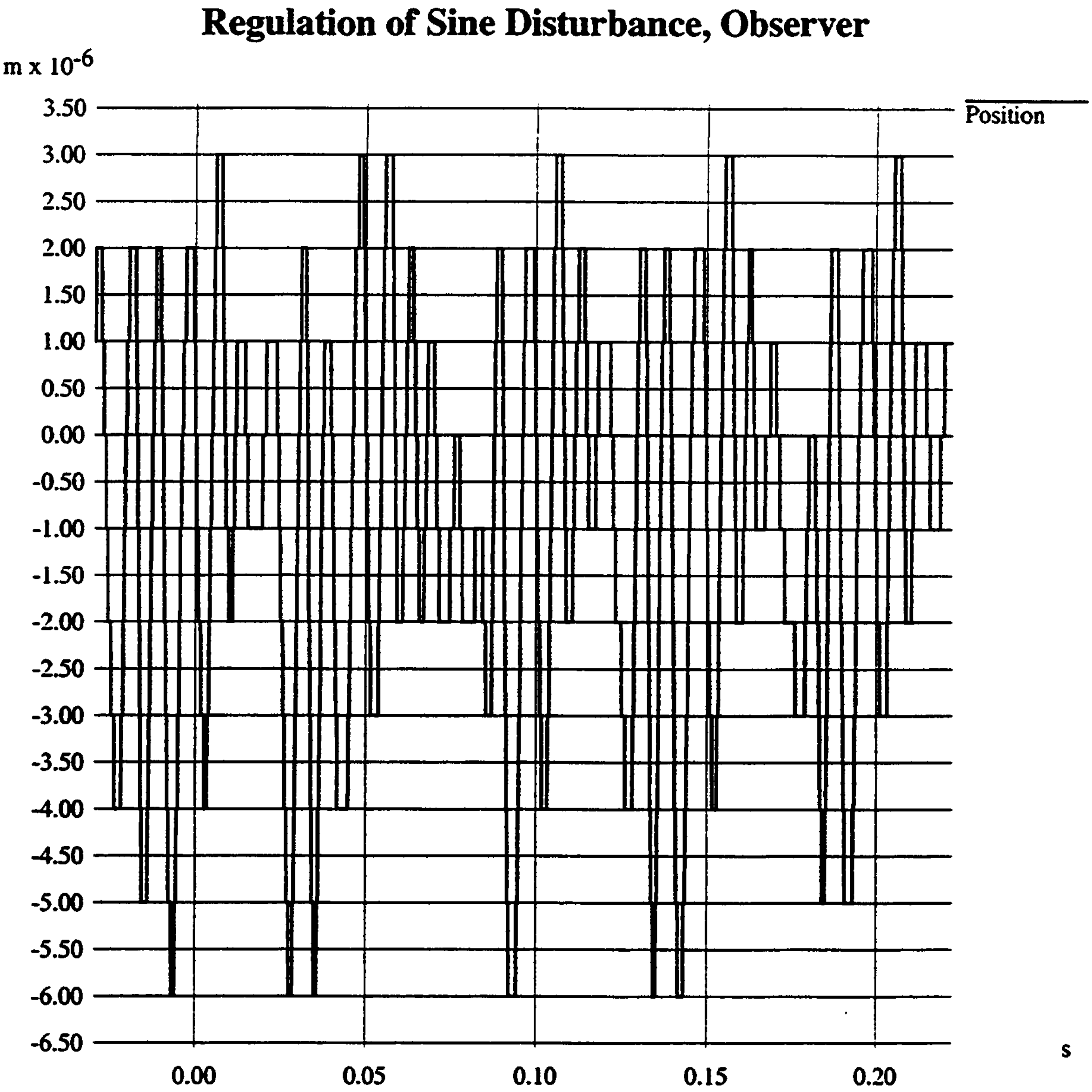


Figure 6.58: Regulation of a 20Hz sinusoidal disturbance, periodic observer used.

Chapter 7

Conclusions

Maximum utilization of advanced machine tools can only be realized if limits on amplifier current and voltage are explicitly addressed. Minimum-time path optimization (MTPO) addresses these issues at the feed rate scheduling level, and minimum time tracking control (MTTC) addresses them at the servo-loop level.

It was found that the largest source of path error, given a well-tuned system, was a commanded path that exceeded the velocity or acceleration capabilities of one or more of the axes. This source of error was minimized by determining the capabilities of the axes, and modifying the feed rate along the path such that these measured limits were not exceeded. Secondly, it was found that cutting time could be significantly reduced without compromising path accuracy by using the entire capability envelope of the system for feed rate scheduling, instead of fixed acceleration and velocity limits. All controllers tested were found to perform significantly better with MTPO.

A modified version of a proportional derivative control was presented which significantly improves the performance of proportional controllers with internal velocity loops at high speeds, without compromising stability, and retaining the improved feedback provided by the tachometer. This modification is simple, and could be implemented with minor controller modifications on existing systems. For these reasons it should be considered in combination with MTPO for existing machines.

A 'Minimum-Time Tracking Controller' (MTTC) was presented which focuses on the primary sources of path error in high-speed servoing applications, specifically, limits on amplifiers, system inertia, and damping. This controller is computationally efficient and robust. The advantages were most clearly seen in its consistently superior disturbance-rejection properties. MTTC was shown to regulate steps of various sizes better (faster/less over shoot) than GPC, $H,^\infty$ or $PD\star$ controllers. It was also shown to respond more linearly in magnitude, and phase to sinusoidal waves at various frequencies. This was particularly true up to the point where the commands exceeded the capabilities of the axis.

Broad-band stiffness can be improved through the use of a periodic observer. Since the source of disturbances is the cutting process, an encoder on the spindle is used to synchronize compensation, with the disturbance. The instantaneous disturbance is estimated as the difference between the predicted acceleration, and the achieved acceleration. This estimate is then averaged as a function of spindle position, over successive spindle revolutions. This procedure provides the low frequency stiffness that the mass of the system cannot.

Driving each winding of a permanent magnet linear motor separately enables commutation strategies to be implemented which reduce the heat generated in the coils of the linear motor, increasing the continuous holding force of the motor by approximately 18%.

PC based soft motion control has been instrumental in enabling this research. This system eliminated many of the limitations of the machine controller which make feed rate scheduling and path generation difficult. Specifically, the entire part path can be planned and computed in advance, stored, and played back in real-time. This eliminates the requirements of bounded execution times, and the difficulty of representing the data in an intermediate form.

With the exception of amplifier saturation, linear motors match the plant model $\frac{C_1}{s(s+C_2)}$ very closely. However they still resonate, simply at higher frequencies than traditional ball screws.

Appendix A

Linear Motor Design

1. List of Components

1.1 Purchased Components:

Linear Slide THK Type HRW 50 CR, 1000 mm rail length. Canadian Bearing.

High-Strength Magnets NFS33 ND-FE-B, Ni coating, 25 x 75 x 15 mm, magnetized 15 mm direction. Stanford Materials Co. , San Mateo, CA. (650) 348-3482.

Material for the motor's frame 3/8"-thick sheet of G11 epoxy resin Glass Fiber Reinforced Composite (GFRC), 36" x 48". P & A Plastics Inc., 1850 Burlington St. E, Hamilton. (905)547-1675. Material cost: $707.31. Machining (cutting) cost: ($200)

Linear Encoder LE18 Digital Linear Encoder. Gurley Precision Instruments, Troy, NY. (800)759-1844 www.gurley.com

Energy chain igus E-chain 05-5-038-0, 76 links (1.5 m) + mounting brackets 050-30-12. igus Inc., Rumford, RI. Toronto rep: Brian Miller, (800)965-2496. Cost: none. Value: US $5.00

Dampers Four small viscous dampers, model MC 150. ACE Controls, Inc.

1.2 In-House components (designed and manufactured at McMaster):

Coil assembly: Nine coils were made with 100 windings of 24 gauge wire, and set in epoxy in such a shape that they fit together as shown in (DWG-??) to form one solid piece.

Aluminum plate: A 3/4-in thick aluminum plate is screwed to the top of the sliding cart. Its purpose is to provide a mass to which all translating components can be attached. An aluminum shim was also manufactured in aid of attaching the encoder read head to the aluminum plate.

Connector plate: A 3/8-in thick vertical plate connects the 9-coil assembly to the top aluminum plate.

Extension bar: The end of the energy chain is affixed to a 1/2-in thick aluminum bar which in turn is screwed to the aluminum plate.

Mounting blocks for dampers (four): Machined aluminum blocks provide the housing for the four dampers (two at each end).

Mounting brackets for encoder (two): Small aluminum angles are used to mount the encoder to the frame.

Iron core (two): Each core (3 in x 0.5 in x 32.5 in) was constructed by bonding together approximately 200 pieces of very thin, insulated steel strips, each about 0.015 in x 0.5 in x 32.5 in.

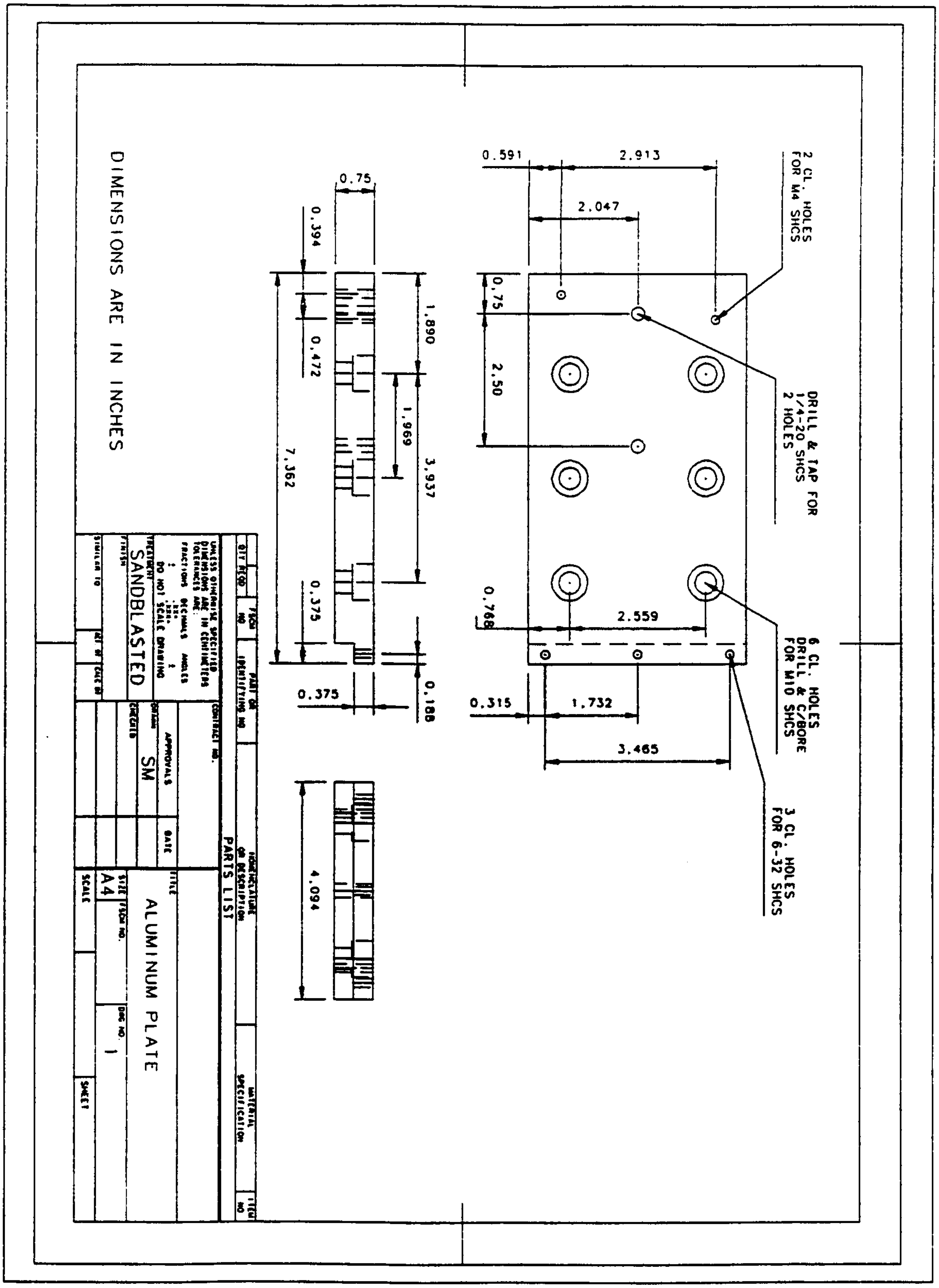


Figure A.59: Aluminum plate attached to the sliding cart allows for the attachment of other moving parts

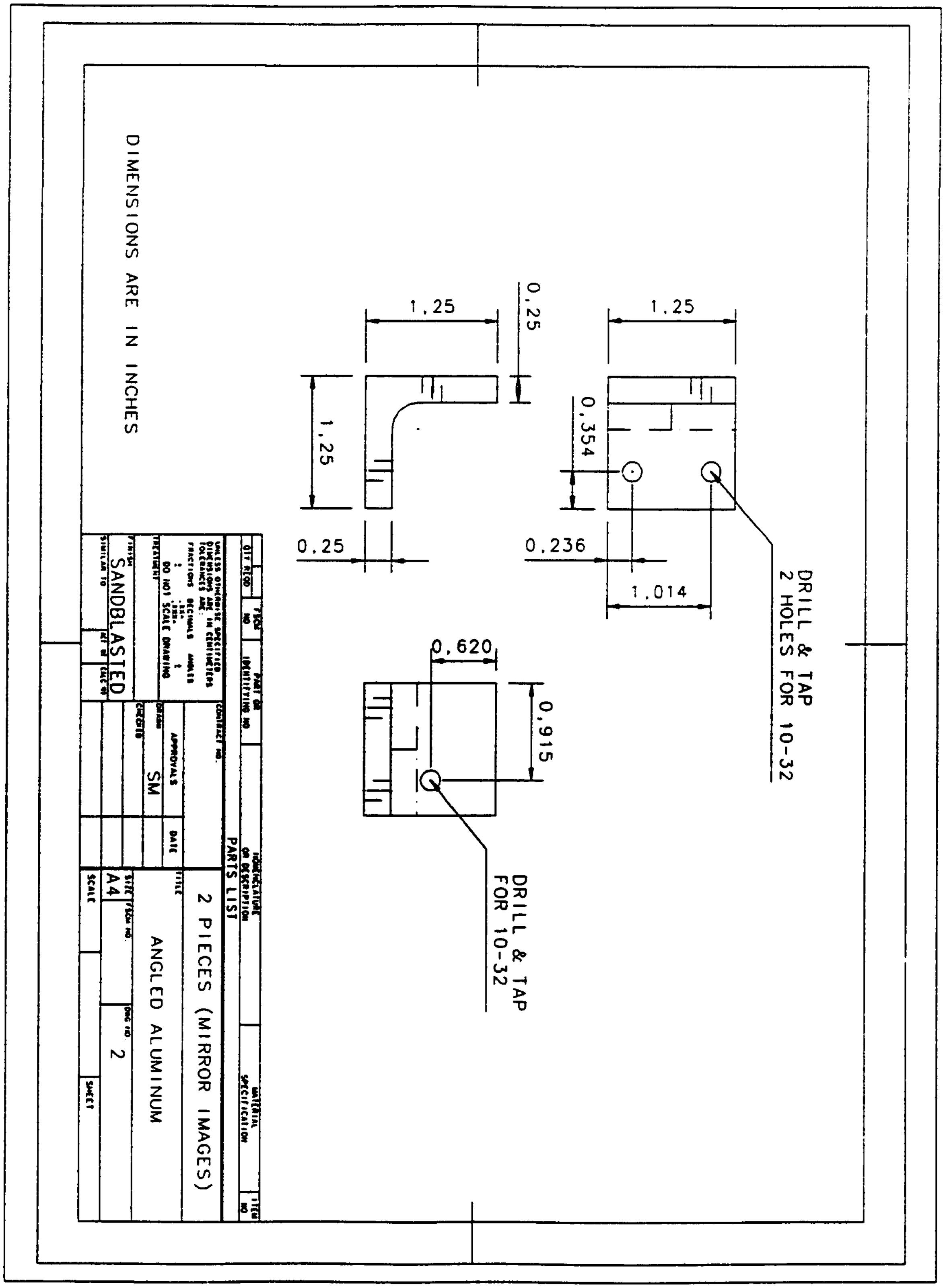


Figure A.60: Angle brackets (two) for mounting the linear scale

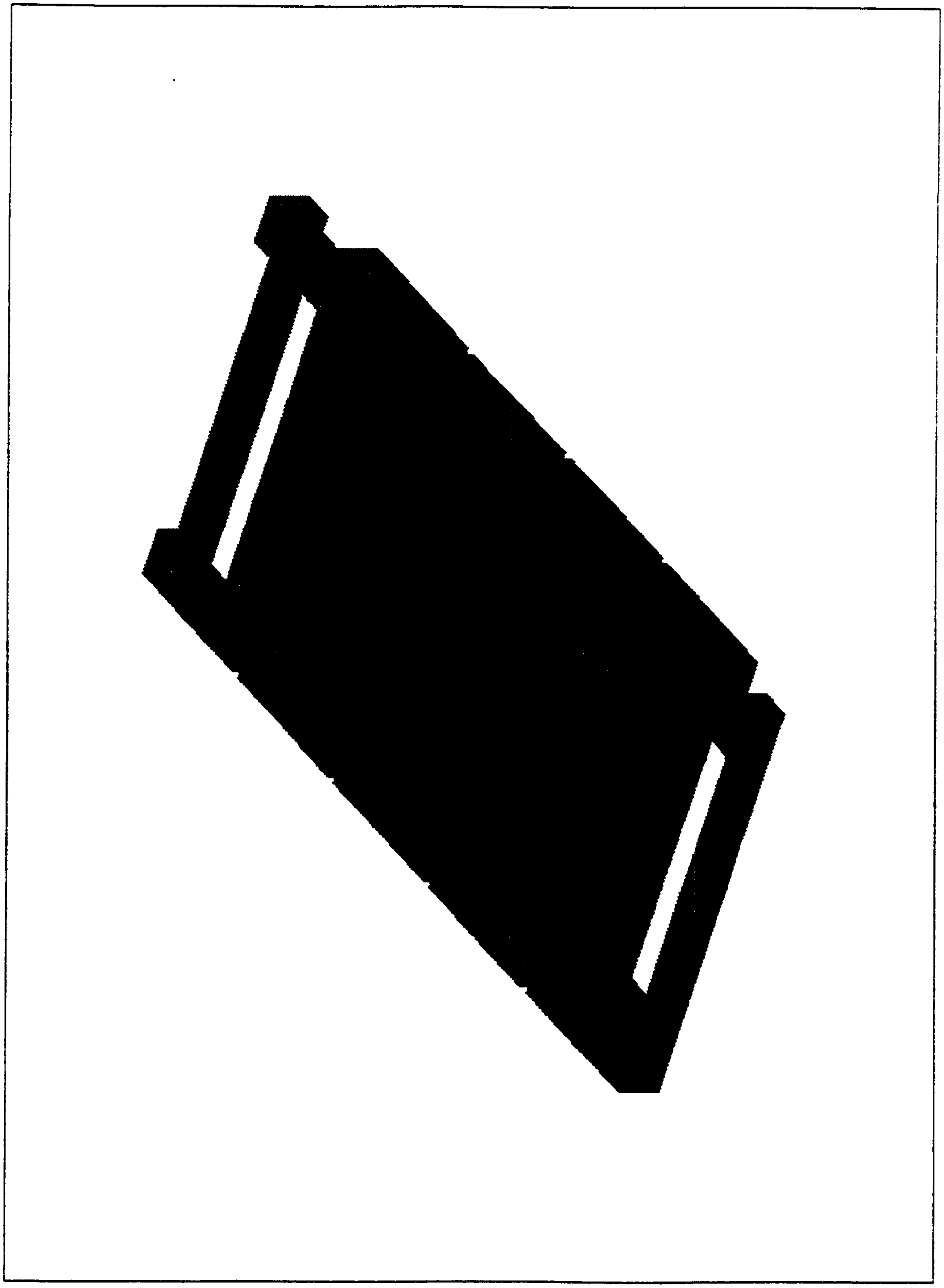


Figure A.61: Arrangement of nine coils, each encased in epoxy

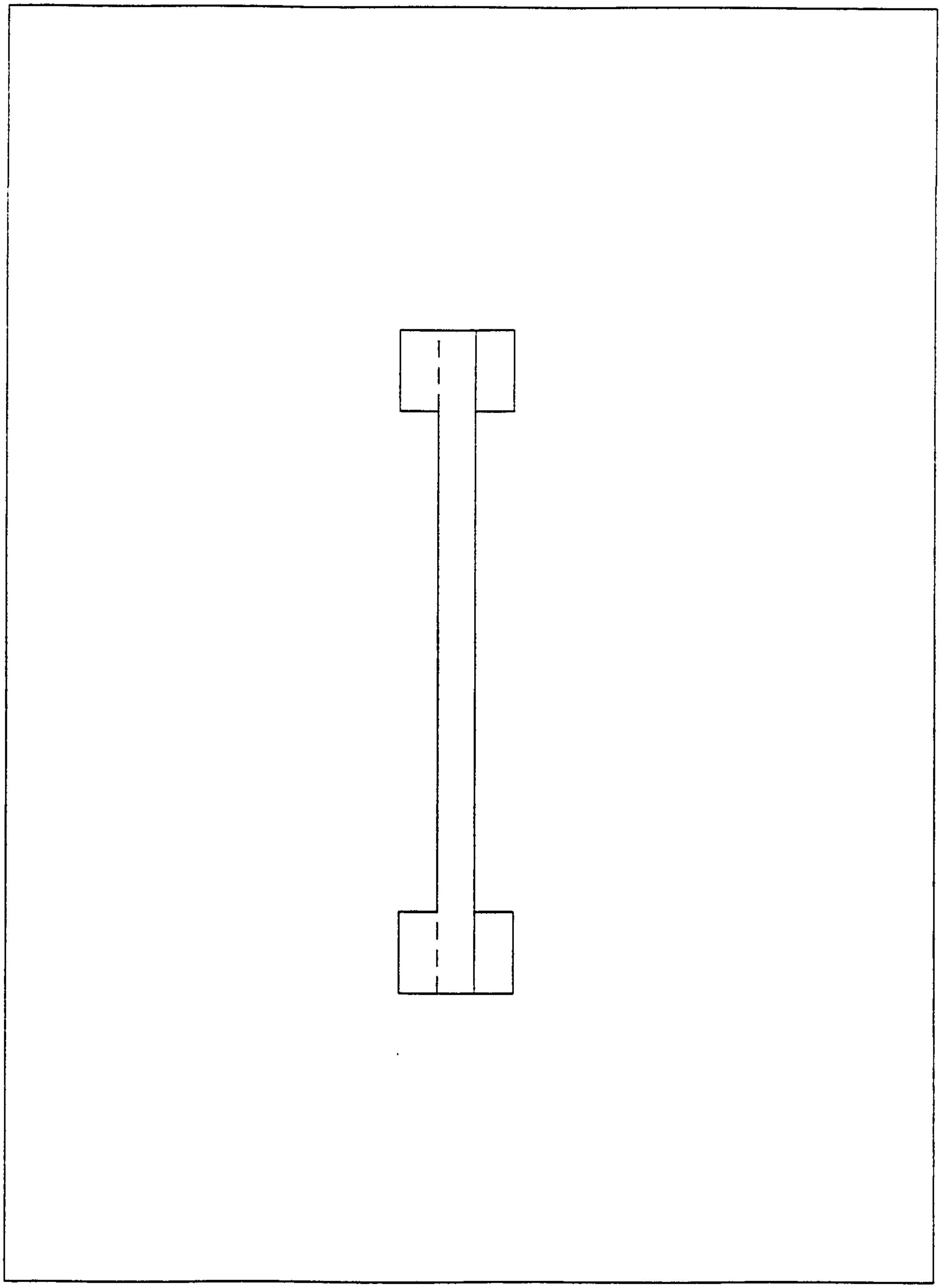


Figure A.62: Front view of nine-coil arrangement

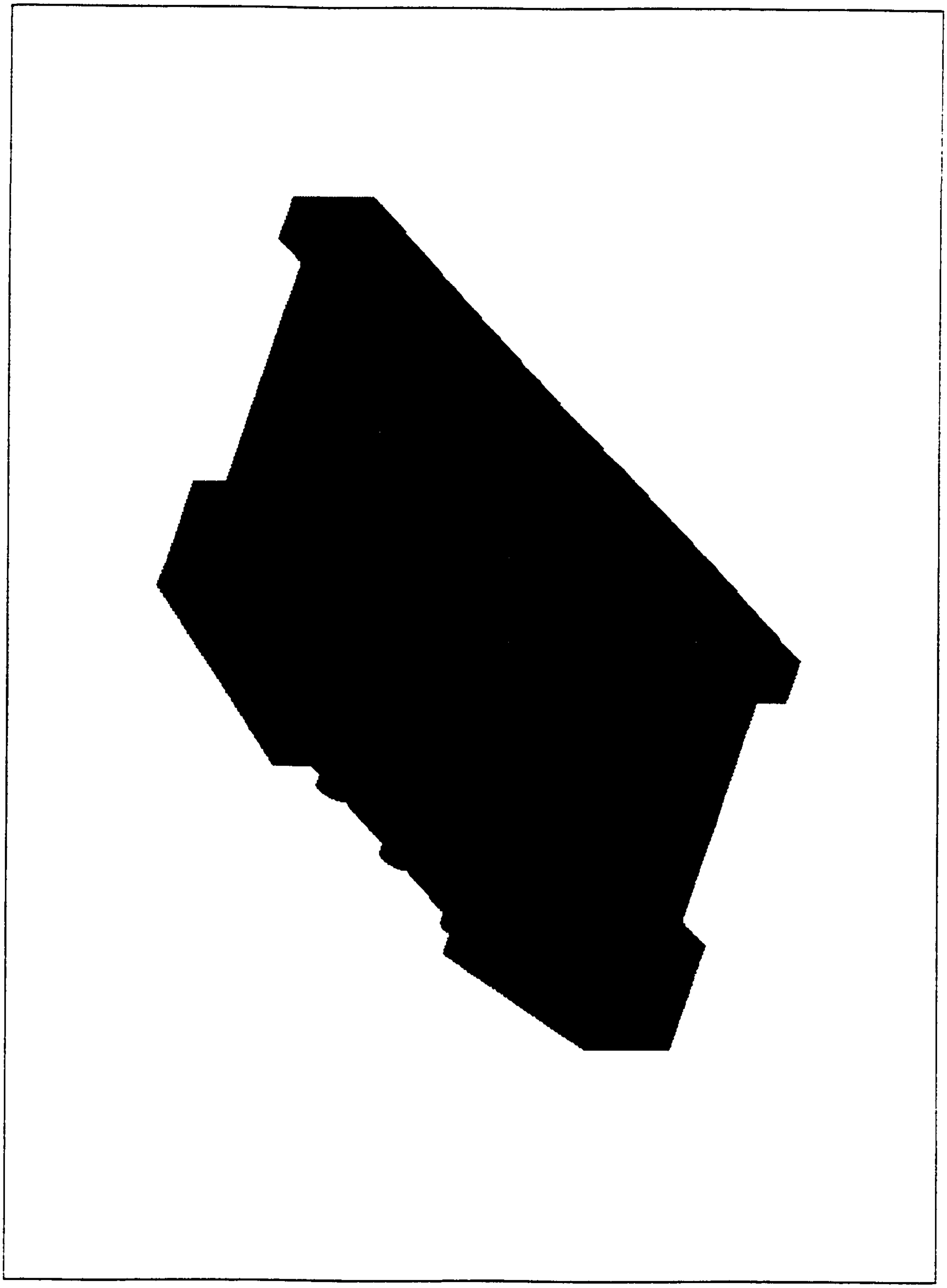


Figure A.63: Nine-coil assembly after being encased in epoxy

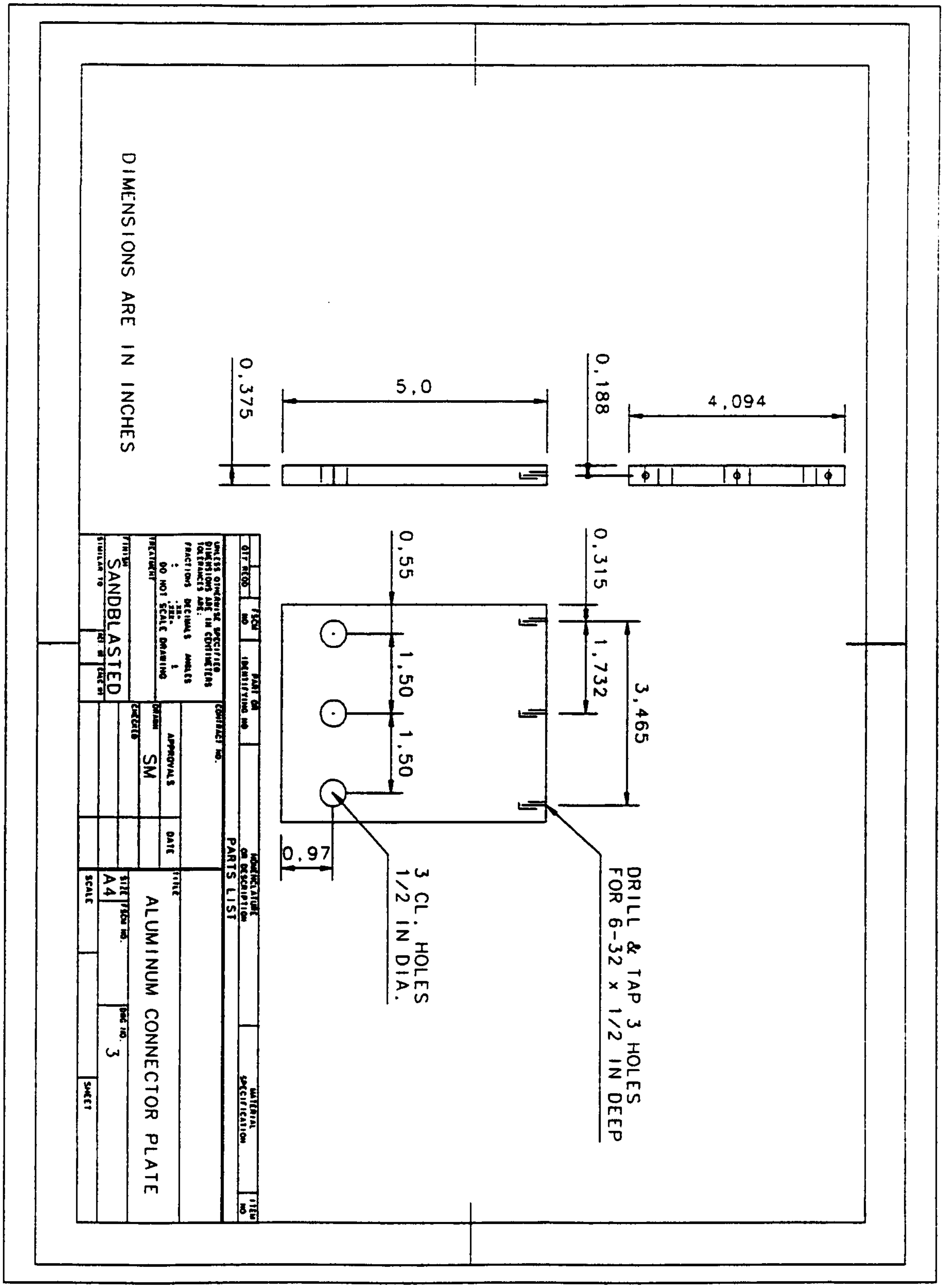


Figure A.64: Aluminum plate which connects the cart to the coil assembly

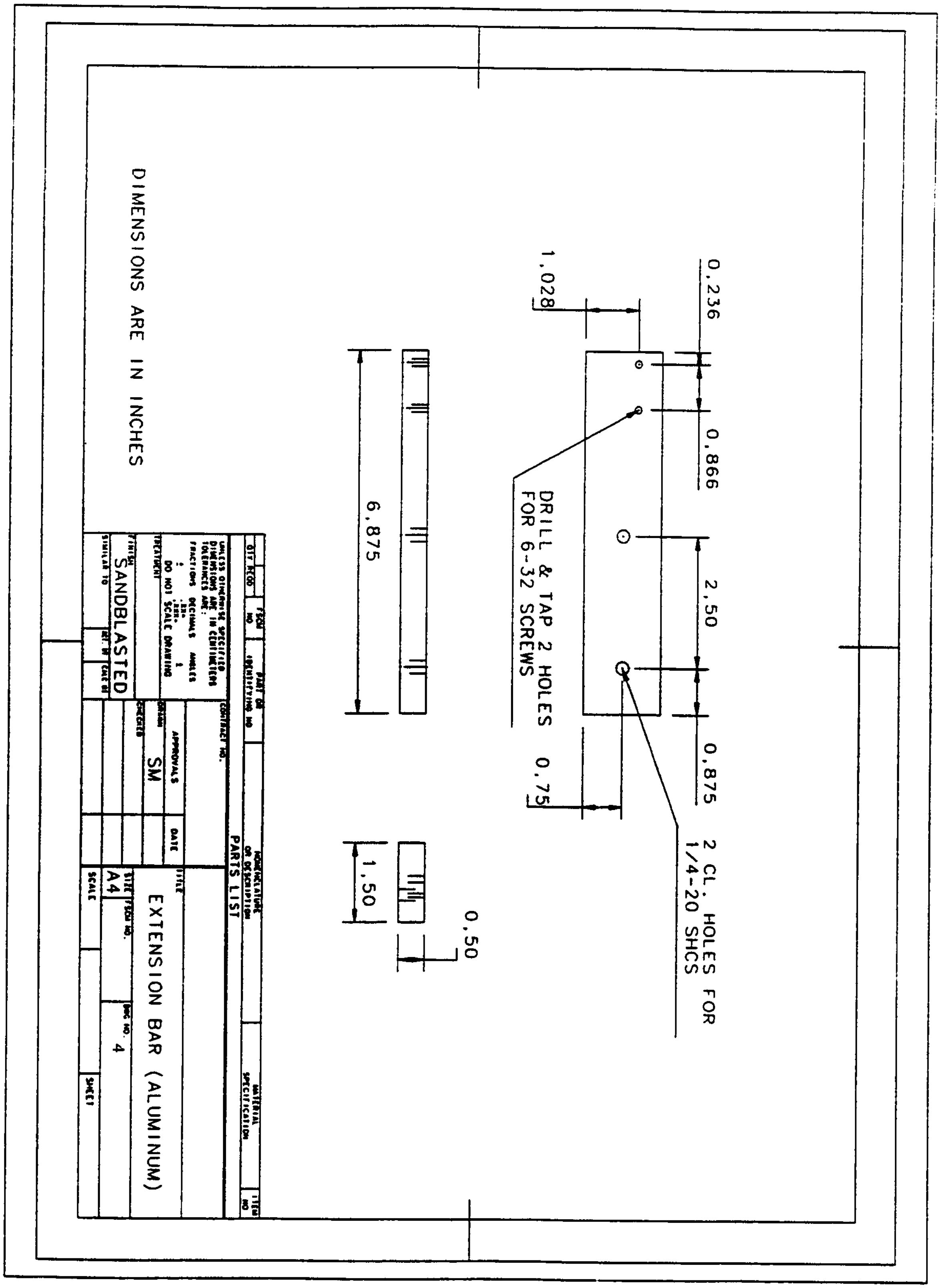


Figure A.65: Aluminum extension bar on which the energy chain is connected

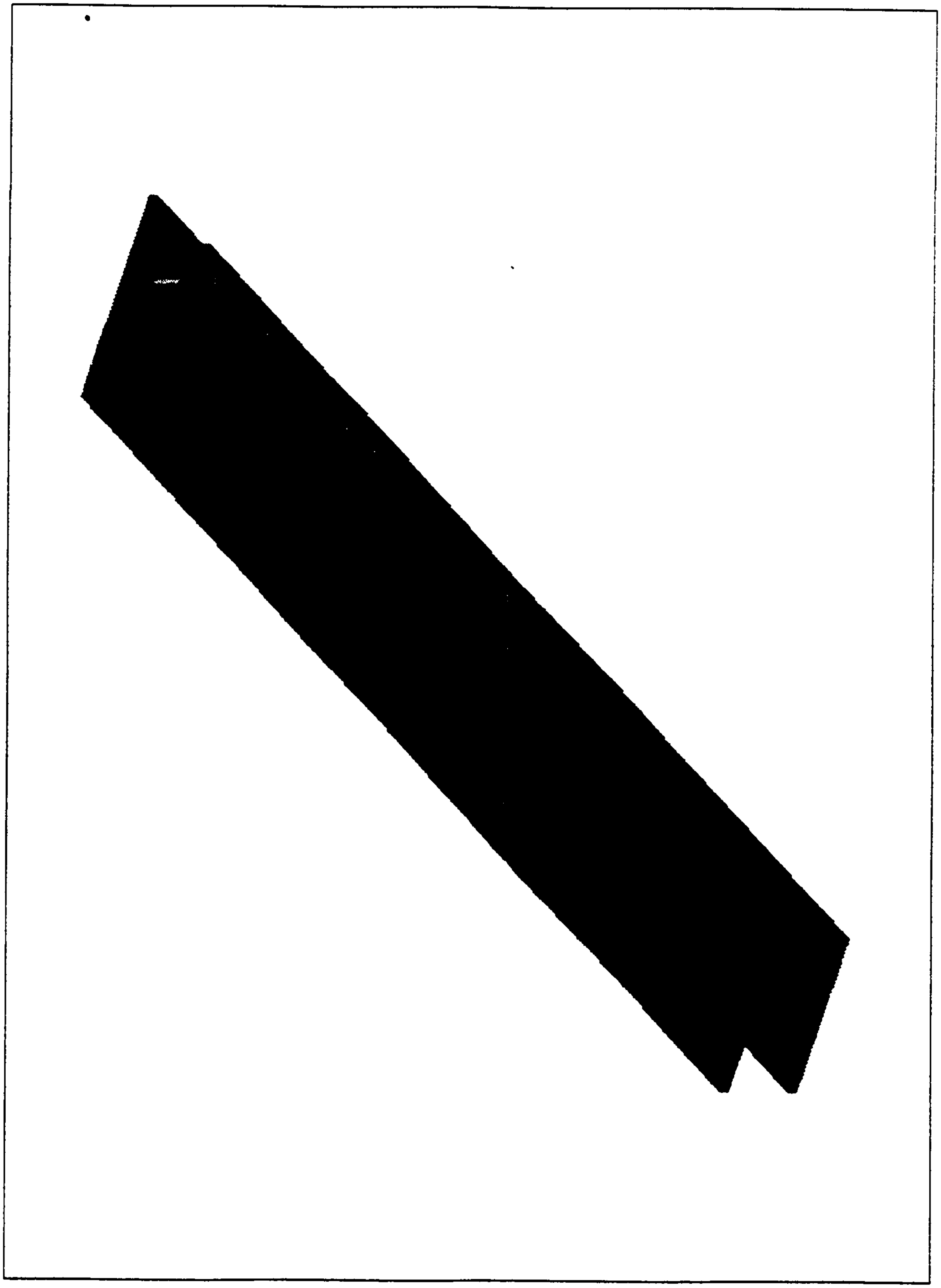


Figure A.66: Motor frame showing reinforcing 1/4-inch screws

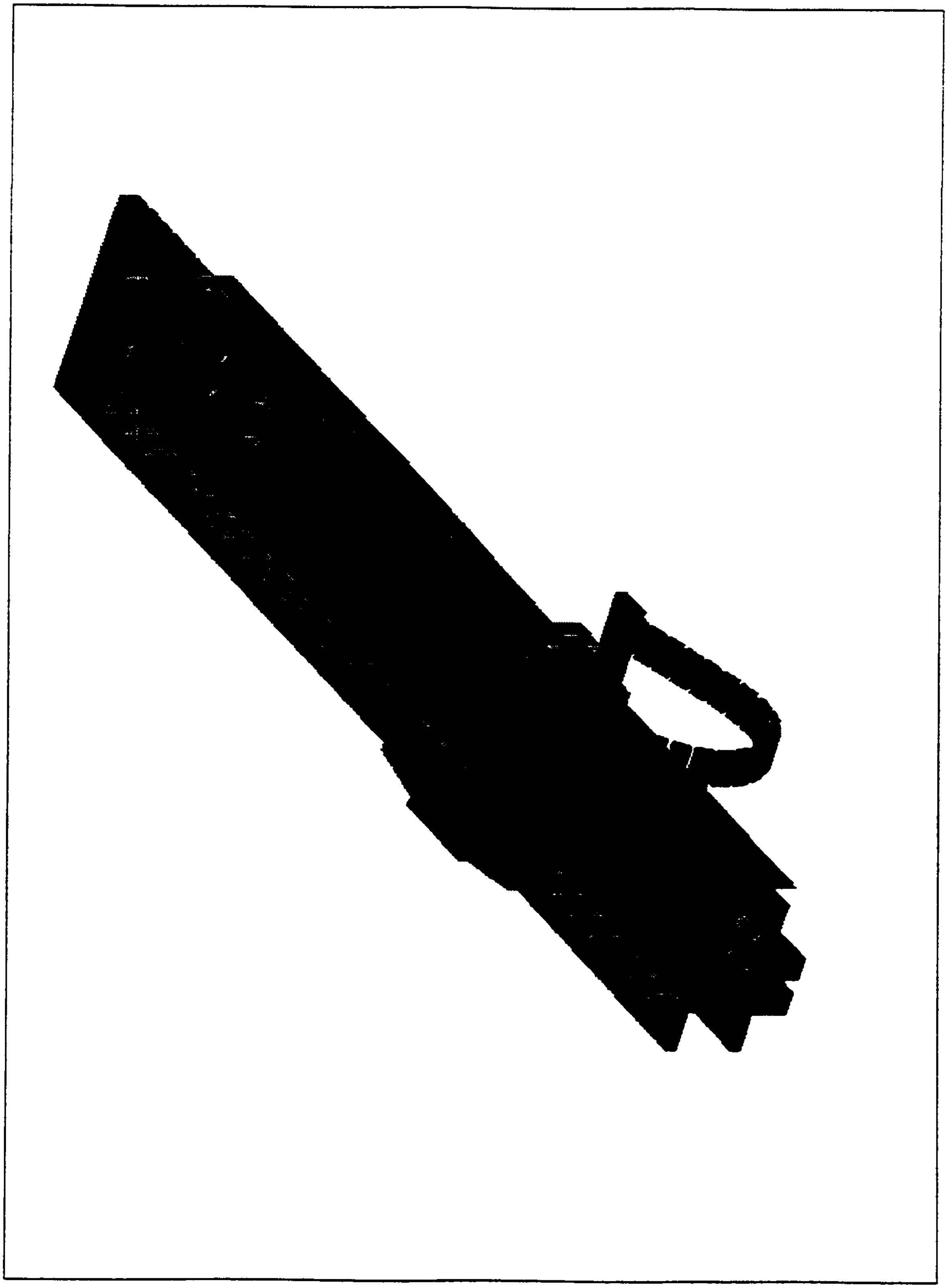


Figure A.67: Complete assembled custom linear motor

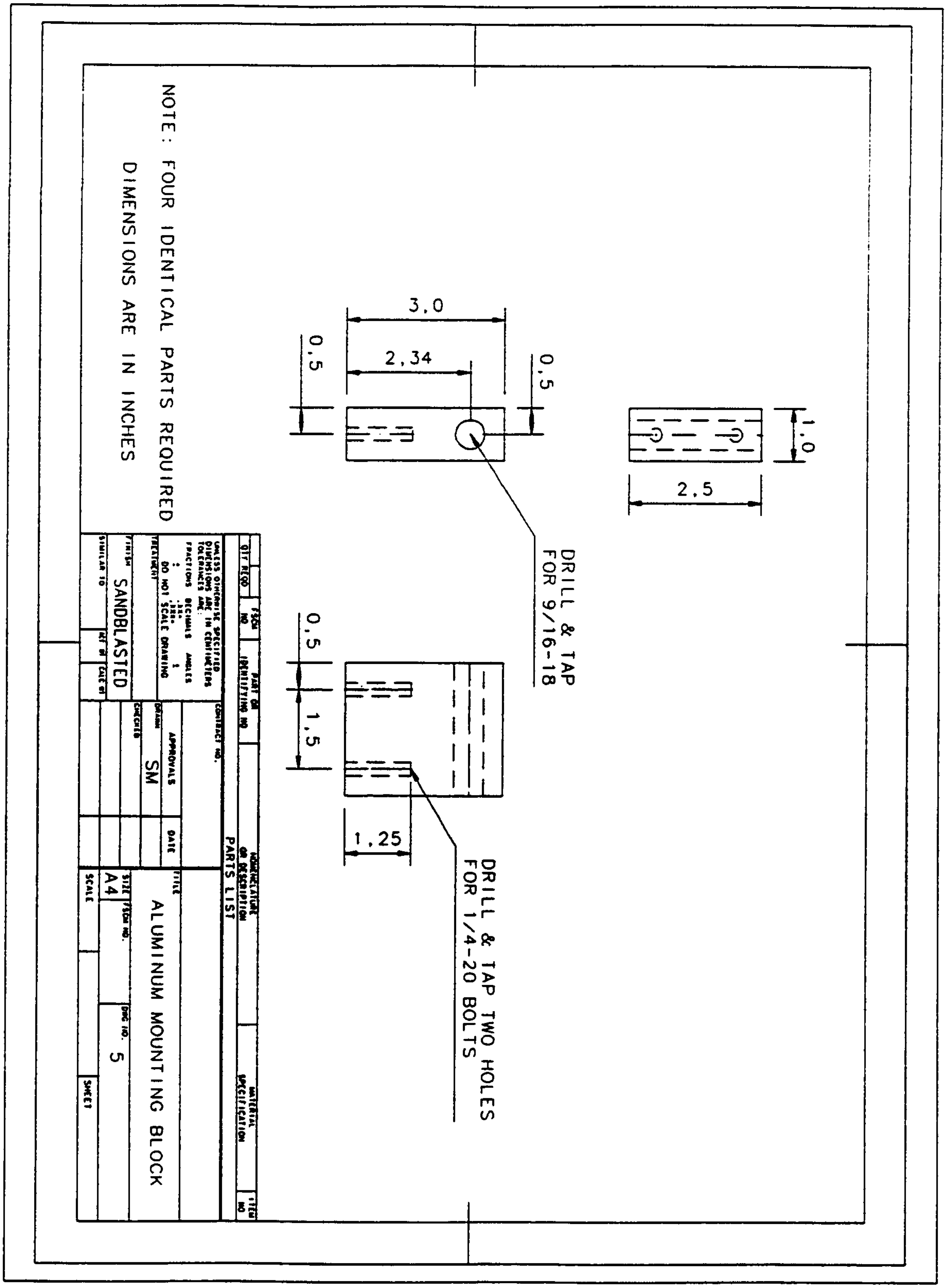


Figure A.68: Mounting blocks (four) that each provide the housing for a viscous damper

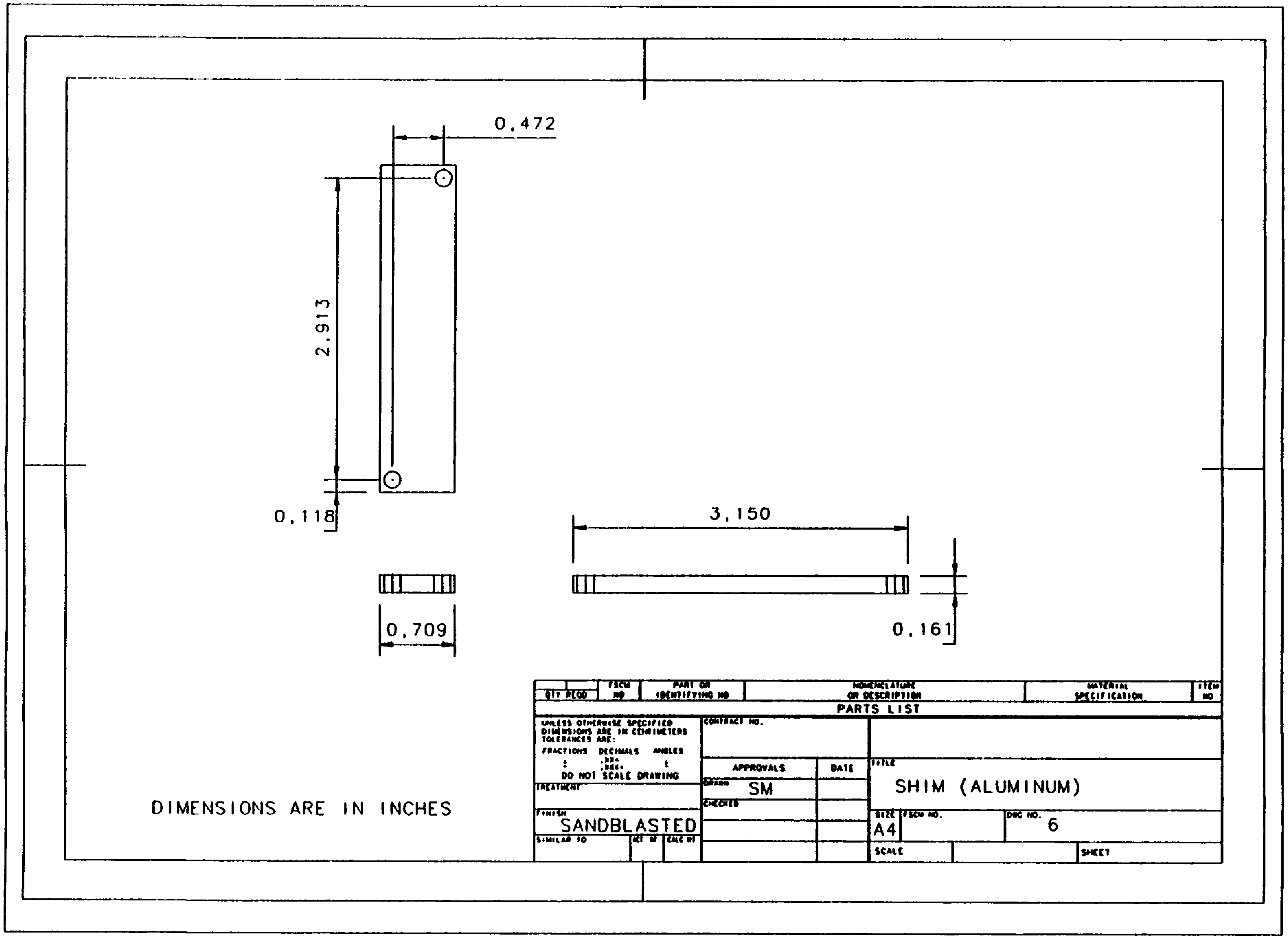


Figure A.69: Small shim custom machined to provide the interface between the aluminum plate and the read head of the linear scale

References

[1] D.M. Alter and T.C. Tsao, "Stability of turning processes with actively controlled linear motor feed drives", *ASME Journal of Eng. for Ind.*, vol. 116, pp 298–307, 1994.

[2] D.M. Alter and T.C. Tsao, "Control of linear motors for machine tool feed drives: Design and implementation of h^∞ optimal feedback control", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 118, pp 649–656, 1996.

[3] D.M. Alter and T.C. Tsao, "Control of linear motors for machine tool feed drives: Experimental investigation of optimal feedforward tracking control", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 120, pp 137–141, 1998.

[4] K. J. Astrom and B. Wittenmark, *"Computer Controlled Systems, Theory and Design"*, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1984.

[5] M. Barabanov, "A linux-based real-time operating system", Master's thesis, New Mexico Institute of Mining and Technology, June 1997.

[6] C.A. Barbargires and C.A. Karybakas, "Optimal control of first-order plants by dead-beat techniques", *Optimal Control Applications and Methods*, vol. 18 No. 5, pp 355–362, September-October 1987.

[7] Y.P. Li B.L. Zhang, P. Ma, C.X. Liao, and S.H. Xiao, "The high speed feed system with linear servo motor and its application in CNC machine tools", *IPMM'97: Australasia-Pacific Forum On Intelligent Processing And Manufacturing of Materials*, vol. 2, pp 1321–1327, 1997.

[8] G. M. Bone, "A novel iterative learning control formulation of generalized predictive control", *Automatica*, vol. 31 No. 10, pp 1483–1487, 1995.

[9] Richard Bonert, "Digital tachometer with fast dynamic response implemented by a microprocessor", *IEEE Transactions on Industry Applications*, vol. IA-19 No. 6, pp 1052–1056, November/December 1983.

[10] P. Boucher and D. Dumur, Generalized predictive cascade control (CPCC) for machine tools drives, *Annals of the CIRP*, vol. 39, pp 357–360, January 1990.

[11] H. Van Brussel and P.Van den Braembussche, "Robust control of feed drives with linear motors", *Annals of the CIRP*, pp 325–328, 1998.

[12] Y. Chen and J. Tlusty, "Effect of low-friction guideways and lead-screw flexibility on dynamics of high-speed machines", *Annals of the CIRP*, pp 353–356, 1995.

[13] H. Chuang and C. Liu, "Cross-coupled adaptive feedrate control for multiaxis machine tools", *ASME Transactions, Journal of Dynamic Systems, Measurement and Control*, vol. 113, pp 451–457, September 1991.

[14] David .W. Clarke, "Application of generalized predictive control to industrial processes", *IEEE Control Systems Magazine*, pp 49-55, April 1988.

[15] D.W. Clarke and C. Mohtadi, "Properties of generalized predictive control", *Automatica*, vol. 25/6, pp 859–875, 1989.

[16] G. Bastin D. Nesic, I. M. Y. Mareels and R. Mahony, "Output dead beat control for a class of planar polynomial systems", *SIAM Journal on Control and Optimization*, vol. 36 No. 1, pp 253–272, January 1998.

[17] D.C.H.Yang and Fu-Chung Wang, "A quintic spline interpolator for motion command generation of computer-controlled machines", *Journal of Mechanical Design, Transactions of the ASME*, vol. 116, pp 226–231, March 1994.

[18] D. Dumur, P. Boucher, and A.U. Ehrlinger, "Constrained predictive control for motor drives", *Annals of the CIRP*, vol. 45, pp 355–358, January 1996.

[19] C. Mohtadi D.W. Clarke and P.S. Tuffs, "Generalized predictive control – part I. the basic algorithm", *Automatica*, vol. 23/2, pp 137-148, 1987.

[20] C. Mohtadi D.W. Clarke and P.S. Tuffs, "Generalized predictive control – part II. extensions and interpretations", *Automatica*, vol. 23/2, pp 149–160, 1987.

[21] B. S. El-Khasawneh and P. M. Ferreira, "On using parallel link manipulators as machine tools', *Transactions of NAMRI/SME*, vol. XXV, pp 305–310, 1997.

[22] H. Gao H. Schulz and M. Stanik, "Analysis and optimization of the dynamic contour accuracy using the example of a linear motor machine tool", 2^{nd} *International Conference on High Speed Machining, Darmstadt, Germany*, pp 107–115, March 1999.

[23] C.H. Chew H. Van Brussel and J. Swevers, "Accurate motion controller design based on an extended pole placement method and a disturbance observer", *Annals of the CIRP*, vol. 43, pp 367–372, January 1994.

[24] R. H. Hosking, "Using high-speed data links in your data acquisition and DSP systems", *VMEbus Systems*, vol. 15 No. 2, pp 27–34, April 1998.

[25] F. Imamura and H. Kaufman, "Time optimal contour tracking for machine tool controllers", *IEEE Control Systems*, pp 11–17, April 1991.

[26] R. Jaenicke, "Using RISC processors to support floating-point DSP applications", *VMEbus Systems*, vol. 15 No. 2, pp 39–46, April 1998.

[27] Sinan J. Badrawy Jiri Tlusty, Scott Smith, David A. Smith, and Andrew P. Smith, "Design of a high speed milling machine for aluminum aircraft parts", *ASME - Manufacturing Science and Technology*, vol. 2, pp 323-332, 1997.

[28] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems", *ASME Transactions, Journal of Dynamic Systems, Measurement and Control*, vol. 102 No. 4, pp 265–272, December 1980.

[29] Y. Koren, "Control of machine tools", *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 119 No. 4(B), pp 749–755, November 1997.

[30] Y. Koren and C.C. Lo, "Advanced controllers for feed drives", *Annals of the CIRP*, vol. 41, pp 689–698, February 1992.

[31] P.K. Kulkarni and K. Srinivasan, "Optimal contouring control of multi-axial feed drive servomechanisms", *Journal of Engineering for Industry, Transactions of the ASME*, vol. 111, pp 140–148, May 1989.

[32] M. J. Karels M. K. McKusick, K. Bostic and J. S. Quarterman, *"The Design and Implementation of the 4.4BSD Operating System"*, Addison-Wesley Publishing Company, Inc., New York, 1996.

[33] Y. Koren M. Shpitalni and C.C.Lo, "Realtime curve interpolators", *Computer-Aided Design*, vol. 26, No. 11, pp 832–838, November 1994.

[34] A. D. McGettrick, editor, *"Real-Time Systems and Programming Languages (second edition)"*, Addison-Wesley Publishing Company, Inc., New York, 1996.

[35] C. W. T. McLyman, *"Transformer and Inductor Design Handbook"*, Marcel Dekker, Inc., 270 Madison Avenue, New York, New York 10016, 1978.

[36] n.a., "Die and mold finishing - how fast?", *Manufacturing Engineering*, pp 35–48, September 1995.

[37] G. Otten, T.J.A., De Vries, J.V. Amerongen, A.M. Rankers, and E.W. Gaal, "Linear motor motion control using a learning feedforward controller", *IEEE/ASME Trans. On Mechatronics*, vol. 2/3, pp 179–187, 1997.

[38] D. A. Patterson and J. L. Hennessy, *"Computer Architecture a Quantative Approach"*, Morgan Kaufmann Publishers, Inc., San Mateo, California, USA, 1990.

[39] G. Pritschow, "On the influence of the velocity gain factor on the path deviation", *Annals of the CIRP*, vol. 45, pp 367–371, January 1996.

[40] G. Pritschow, "A comparison of linear and conventional electromechanical drives", *Annals of the CIRP*, vol. 47/2, pp 541–548, 1998.

[41] G. Pritschow and W. Philipp, "Direct drives for high-dynamic machine tool axes", *Annals of the CIRP*, vol. 39/1, pp 413–416, 1990.

[42] G. Pritschow and W. Philipp, "Research on the efficiency of feedforward controllers in m direct drives", *Annals of the CIRP*, vol. 41/1, pp 411–415, 1992.

[43] D. Renton and M.A. Elbestawi, "High speed servo control of multi-axis machine tools", *Int. J. Mach. Tools & Manufact.*, vol. 40, pp 539–559, 2000.

[44] H. Schulz, "State of the art of high speed machining", *1st French and German Conference on High Speed Machining*, June 1997.

[45] H. Schulz and T. Moriwaki, "High speed machining", *CIRP Annals - Manufacturing Technology*, vol. 41/2, pp 637-643, 1992.

[46] R.J. Seethaler and I. Yellowley, "The regulation of position error in contouring systems", *International Journal of Machine Tools and Manufacturing*, vol. 36 No. 6, pp 713-728, June 1996.

[47] N. K. Sinha, *"Control Systems"*, Holt, RienHart and Winston, Inc., New York, 1986.

[48] M.A. Elbestawi T. Bailey, D. Renton, "PC based open architecture machine tool control", *ASME International Mechanical Engineering Congress and Exposition, Dallas, TX.*, 1997.

[49] K. Murata T. Mita, M. Hirata and Hui Zhang, "H^∞ control versus disturbance-observer-based control", *IEEE Transactions on Industrial Electronics*, vol. 45/3, pp 488–495, June 1998.

[50] Richard W. Teltz, *"Open Architecture Control for Intelligent Machining Systems"*, PhD thesis, McMaster University, 1998.

[51] J. Tlusty, "High-speed machining", *CIRP Annals - Manufacturing Technology*, vol. 42/2, pp 733–738, 1993.

[52] M. Tomizuka, "Zero phase error tracking algorithm for digital control", *ASME Transactions, Journal of Dynamic Systems, Measurement and Control*, vol. 109, pp 65–68, March 1987.

[53] T.T.C. Tsang and D.W. Clarke, "Generalized predictive control with input constraints", *IEE Proceedings*, vol. 135 Pt. D No. 6, pp 451–460, November 1988.

[54] M. Gringel U. Heisel, "Machine tool design requirements for high-speed machining", *Annals of the CIRP*, vol. 45 No. 1, pp 389–392, 1996.

[55] A. Galip Ulsoy and Y. Koren, "Control of machining processes", *ASME Transactions, Journal of Dynamic Systems, Measurement and Control*, vol. 115, pp 301-308, June 1993.

[56] M. Weck and M. Dammer, "Design, calculation and control of machine tools based on parallel kinematics", *Proceedings of the ASME - Manufacturing Science and Engineering Division*, vol. 8, pp 715–721, 1998.

[57] M. Weck and G. Ye, "Sharp corner tracking using the IKF control strategy", *Annals of the CIRP*, vol. 39 No. 1, pp 437–441, 1990.

[58] J. Wen and A. A. Desrochers, "An algorithm for obtaining bang-bang control laws", *ASME Transactions, Journal of Dynamic Systems, Measurement and Control*, vol. 109, pp 171–175, June 1987.

[59] P. K. Wright, "Principles of open-architecture manufacturing", *Journal of Manufacturing Systems*, vol. 14 No. 3, pp 187–202, 1995.

[60] A. Galip Ulosy Y. Koren, Zbigniew J. Pasek and Paul K. Wright, "Timing and performance of open architecture controllers", *Proceedings of the ASME Dynamics Systems and Control Division*, vol. 58, pp 283–290, 1996.

[61] A. Ulosy Y. Koren, Z. Pasek and U. Benchetrit, "Control of machine tools", *CIRP Annals - Manufacturing Technology*, vol. 45 No. 1, pp 377–380, November 1996.

[62] C.C. Lo Y. Koren and M. Shpitalni, "CNC interpolators: Algorithms and analysis", *Journal of Manufacturing Science and Engineering, Transactions o f the ASME*, vol. 64, pp 83–92, 1993.

[63] D.C.H. Yang and Jui-Jen Chou, "Automatic generation of piecewise constant speed motion for multi-axis machines", *Journal of Mechanical Design*, vol. 116, pp 581–586, June 1994.

[64] T. Yoon and D. W. Clarke, "Reformulation of receding-horizon predictive control", *International Journal of Systems Science*, vol. 26 No. 7, pp 1383–1400, July 1995.