

5-2007

# Design of a Controller for a Precision Positioning Machine

Joshua Tarbutton

Clemson University, trbtt@clermson.edu

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_theses](https://tigerprints.clemson.edu/all_theses)



Part of the [Engineering Mechanics Commons](#)

---

## Recommended Citation

Tarbutton, Joshua, "Design of a Controller for a Precision Positioning Machine" (2007). *All Theses*. 94.

[https://tigerprints.clemson.edu/all\\_theses/94](https://tigerprints.clemson.edu/all_theses/94)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

DESIGN OF A CONTROLLER FOR A PRECISION POSITIONING  
MACHINE

---

A Thesis  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Engineering  
Science

---

by  
Joshua Tarbutton  
May 2007

---

Accepted by:  
Dr. Thomas Kurfess, Committee Chair  
Dr. Laine Mears  
Dr. Ardalan Vahidi

## ABSTRACT

Control of high-precision machinery is necessary to understand manufacturing defects, maintain quality control, and obtain desired dimensional accuracy, surface roughness, and tolerances. When a controller is designed for high-precision applications, the effect of structural and parametric uncertainty, disturbances, and noise play a significantly more important role in the system performance. The level of modeling required to accurately represent the systems' structure, parameters, noise, disturbances, non-linearity, and etc. to design a high-precision controller will require expert knowledge and significant time investments. In practice, a significant amount of time is spent on tuning the controller even after the modeling and initial controller design has been accomplished.

An alternative to the above control design approach is to build a model via *system identification* and design a controller from the identified system. System identification can be used to build a model that minimizes the difference between the actual system response and the model response when acted on by the same input while incorporating the actual plant's disturbance and noise into the model. System identification has the potential to save valuable time and resources in industrial applications because it uses input-output data from the system to build a model thereby eliminating the difficulty of modeling by physical laws.

System identification was used to build an accurate model of a high-precision measurement system. The model built by system identification was compared to modeling by first laws and showed extremely similar results. Pole-placement control design based on the identified system was used to place the systems' dominant poles. The necessary gains to achieve the desired system response were determined by using the identified model and knowledge of the controller structure. The performance of the model-based controller was compared to actual data of the system and showed that control based on the identified model can be used to accurately control the precision measuring machine.

## DEDICATION

This thesis is dedicated to God who has given us a world of limitless exploration, my mother who now lives with Him, my wife Antonina who is the love of my life, and my daughters who fill my home with laughter.



## ACKNOWLEDGEMENTS

Dr. Kurfess, thank you for investing so much of your time and energy into my life and my success, for your high expectations and equivalent support, and for bringing me along on a great adventure. Antonina, thank you for your patience throughout school, your respect, support, grace, and most of all your love. Thank you mom and dad for teaching me to honor God, show up on time, work hard, and tell the truth. Lee Campbell, thank you for mentoring me; without your support and encouragement I would not be an engineer. Thanks to all my friends. Thanks to the Timken Company: Joe Pack for your support and for providing such an interesting project, Mike Thompson for your expert knowledge, advice, support, and encouragement, Dr. Kolarits for your help and guidance, Doug Sponseller, Joe Fortunato, Randy Clark, Dave Hensal, and Tom Callow for all of your help and support.





## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
DEDICATION .....	v
ACKNOWLEDGEMENTS .....	vii
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xiii
NOMENCLATURE .....	xvii
CHAPTER	
I. INTRODUCTION .....	1
Research objectives.....	5
II. LITERATURE REVIEW .....	7
System Identification .....	7
Non-Parametric Methods .....	9
Non-Parametric Estimation of Impulse Response: .....	10
Non-Parametric Estimation of Frequency Response: .....	12
Parametric Methods .....	16
Control Design .....	26
III. EQUIPMENT, INSTRUMENTATION, AND SOFTWARE .....	31
Universal Measuring Machine .....	31
Architecture .....	31
Controller .....	34
Tuning .....	36
System Simulation .....	36
Modeling of C-axis: .....	37
C-Axis Inertia Determination .....	41
C-Axis Simulated Vs. Actual Results .....	45
C-Axis Conclusion .....	45
Linear Motor Parameter Determination .....	47

Table of Contents (Continued)

	Page
Theoretical Eddy Current Damping.....	47
Experimental Eddy Current Damping .....	50
R-Axis Mass Determination .....	56
R-Axis Simulated Vs. Actual Results.....	63
Parameter Summary.....	66
IV. SYSTEM IDENTIFICATION.....	69
C-axis Identification of ARX models .....	71
Effect of Noise and Disturbance for C-axis Identification .....	79
V. CONTROL DESIGN.....	85
VI. CONCLUSIONS AND RECOMMENDATIONS .....	103
Conclusions.....	103
Recommendations.....	106
APPENDIX.....	109
REFERENCES .....	113

## LIST OF TABLES

Table		Page
1.	Black Box Models and Their Assumptions .....	19
2.	UMM Axis Motor and Bearing Type. ....	33
3.	Axis Encoder and Resolution.....	34
4.	PMAC Controller Variables .....	35
5.	C-Axis Catalogue Values.....	40
6.	Theoretical Eddy Current Determination Variables.....	49
7.	Regression Values for Different Frequencies .....	63
8.	Summary of Motor Parameters for the UMM .....	67
9.	Roots of C-axis model and identified ARX models .....	76



## LIST OF FIGURES

Figure	Page
1. General Linear Parametric Model.....	17
2. Universal Measuring Machine.....	31
3. Schematic of UMM architecture.....	32
4. DC motor model .....	37
5. Block Diagram of DC motor.....	39
6. DC motor model with amplifier dynamics added.....	40
7. 1,000 count step response .....	46
8. 10,000 count step response .....	46
9. R-Axis DAC voltage command and residuals vs. position at 100mm/s .....	53
10. Velocity for different servo commands .....	54
11. Alternate constant force eddy current damping determination.....	55
12. Both methods of eddy current damping.....	55
13. Stribeck Model of Force vs. Velocity.....	57
14. Force and position vs. time for 0.1mm Sine input at 1 Hz for 5 seconds.....	58
15. Force and position vs. time for 0.1mm Sine input at 5 Hz for 1 second.....	58
16. Force and velocity vs. time for 0.1mm Sine input at 5 Hz for 1 second.....	59
17. Force and acceleration vs. time for 0.1mm Sine input at 5 Hz for 1 second .....	59

List of Figures (Continued)

Figure	Page
18. Force and position vs. time for 0.1mm Sine input at 10 Hz for 0.5 seconds .....	60
19. Force vs. position for 0.1mm Sine Input at 5Hz for 1 second .....	60
20. Linear regression values for mass at different frequencies .....	61
21. Linear regression values for hysteresis at different frequencies.....	61
22. Linear regression values for Damping at different frequencies .....	62
23. Linear regression values for force offset at different frequencies.....	62
24. Block diagram for linear motor.....	64
25. Simulated/Actual data R-Axis 0.1 mm sine wave at 10 Hz for 0.5 sec.....	65
26. Simulated/Actual data R-Axis 0.1 mm sine wave at 10 Hz for 0.5 sec.....	66
27. 0.22 degree step input for C-Axis.....	72
28. Close up of step input showing time delay of approximately 1.3 ms. ....	72
29. Identification data for C-axis .....	73
30. Power spectrum of white noise input to PMAC .....	74
31. PSD of motor input from PMAC .....	74
32. Bode plot of system model and identified models.....	76
33. Comparison of models .....	77
34. Residuals from identified ARX models.....	78

List of Figures (Continued)

Figure	Page
35. Identification data with noisy input into C-axis motor .....	80
36. Identification data with added noisy output.....	81
37. Close up of output noise .....	81
38. Comparison of Actual, Box-Jenkins, and ARX models with output error .....	82
39. Step response using pole placement gains .....	93
40. Pole-zero map of system with pole placement approach .....	94
41. Close-up of pole-zero map.....	94
42. Pole locations as $K_P$ is varied from 1 to 1,000,000.....	96
43. Close up of pole locations as $K_P$ is varied from 1 to 1,000,000.....	97
44. Dominant poles affected by $K_P$ .....	97
45. Acceleration limit related to $K_P$ .....	98
46. Pole locations as $K_I$ is varied from 1 to 1,000,000 .....	98
47. Close up of pole locations as $K_I$ is varied from 1 to 1,000,000.....	99
48. Dominant poles affected by $K_I$ .....	99
49. Acceleration limit related to $K_I$ .....	100
50. Pole locations as $K_D$ is varied from 1 to 1,000,000 .....	100
51. Close up of pole locations as $K_D$ is varied from 1 to 1,000,000.....	101
52. Dominant poles affected by $K_D$ .....	101

List of Figures (Continued)

Figure	Page
53. Acceleration limit related to $K_D$ .....	102
54. Diagram of UMM controller and motor .....	109
55. Diagram of PMAC .....	110
56. Diagram of motor model.....	111



## NOMENCLATURE

(Symbols are listed in the order of appearance)

Symbol	Description
$C_f$	crest factor
$u(t)$	input at time $t$
$y(n)$	output at discrete time-step $n$
$h(k)$	impulse response at step $k$
$e(n)$	error at discrete time-step $n$
$R_{uy}$	cross-correlation between $u$ and $y$
$\sigma_u^2$	input signal variance
$Y(\omega)$	frequency domain input
$G(i\omega)$	frequency function of system
$U(\omega)$	frequency domain output
$y(t)$	input at time $t$
$u(t)$	output at time $t$
$U_\Omega(\omega)$	Fourier transform of input
$Y_\Omega(\omega)$	Fourier transform of output
$G(q)$	discrete transfer function of deterministic part of a system
$\hat{\Phi}_u(\omega)$	input spectrum
$v(t)$	disturbance

Nomenclature (continued)

(Symbols are listed in the order of appearance)

Symbol	Description
$U_N(\omega)$	discrete Fourier transform of input
$\hat{R}_{uy}$	estimate of cross-correlation
$E(\cdot)$	expectation
$\hat{\Phi}_{yu}(\omega)$	discrete Fourier transform of cross-correlation
$\hat{G}(e^{j\omega T})$	spectral estimate of transfer function
$A(q), B(q), C(q), D(q), F(q)$	discrete polynomial with shift operator $q$
$e(n)$	error at discrete time-step $n$
$H(q)$	discrete transfer function of stochastic part of a system
$\hat{y}(t t-1)$	model output prediction at time $t$ given output at $t-1$
$e(t)$	prediction error
$V_N(\theta)$	criterion function to be minimized
$\ell(\cdot)$	norm of prediction errors used in criterion
$\theta$	vector used to parameterize model
$\phi(t)$	past input/output vector
$x$	state vector
$A(\theta), B(\theta), C(\theta)$	state-space matrices

## Nomenclature (continued)

(Symbols are listed in the order of appearance)

Symbol	Description
$\hat{x}$	state estimate
$K$	Kalman gain
$CMDout(n)$	command output from PMAC at discrete time-step $n$
Ix30	PMAC proportional gain
Ix31	PMAC derivative gain
Ix32	PMAC feed-forward velocity gain
Ix33	PMAC integral gain
Ix35	PMAC feed-forward acceleration gain
Ix08	position scale factor
Ix09	velocity scale factor
CA(n)	command acceleration at discrete time-step $n$
CV(n)	command velocity at discrete time-step $n$
FE (n)	following error at discrete time-step $n$
IE(n)	integration error at discrete time-step $n$
AV(n)	actual velocity at discrete time-step $n$

## Nomenclature (continued)

(Symbols are listed in the order of appearance)

Symbol	Description
$e_0$	command voltage
$e_b$	back emf
$i_a$	armature current
$R_a$	armature resistance
$L_a$	armature inductance
$K_b$	back emf constant
$\theta, \omega$	angular position/speed
$T$	motor torque
$K_t$	motor torque constant
$J_L$	polar inertia
$b$	damping
$s$	Laplace variable
$PMAC_{COMMAND}$	PMAC command output
$K_{DAC}$	DAQ gain
$K_{TC}$	amplifier transconductance
$K_A$	amplifier gain
$I_C$	command current
$A_\theta$	servo command amplitude
$\Omega$	angular velocity
$\alpha$	angular acceleration

## Nomenclature (continued)

(Symbols are listed in the order of appearance)

Symbol	Description
$S$	servo command
$Res_C$	c-axis resolution
$f$	frequency in Hz
$emf$	electromotive force
$B$	magnetic flux
$L$	height of plate in magnetic field
$R$	resistance of conducting plate
$v$	velocity of conductor in mag. Field
$L_R$	effective height
$c$	plate thickness
$\sigma$	conductivity of aluminum
$I$	eddy current induced in plate
$F_{LZ}$	Lorenz force
$B_{eff}$	effective flux
$F_L$	linear motor (LM) load force
$F_C$	LM cogging force
$F_F$	LM friction force
$a$	LM acceleration
$m$	LM mass
$F$	LM thrust force

## Nomenclature (continued)

(Symbols are listed in the order of appearance)

Symbol	Description
$\phi$	LM phase offset
$x_1, x_2, x_3$	regression parameters
$F_{offset}$	equilibrium forces on LM
$F(s)$	Laplace transform of LM force
$X(s)$	Laplace transform of LM position
$ARX_{322}$	ARX model delay of 2 samples with third order denominator and second order numerator
$ARX_{323}$	ARX model delay of 3 samples with third order denominator and second order numerator
$BJ$	Box-Jenkins model
$CP(n)$	command position at discrete time-step $n$
$AP(n)$	actual position at discrete time-step $n$
$K_{vff}, K_{vffs}$	feed-forward velocity; simplified
$K_{aff}, K_{affs}$	feed-forward acceleration; simplified
$K_P, K_{Ps}$	proportional gain; simplified
$K_I, K_{Is}$	integral gain; simplified
$K_D, K_{Ds}$	derivative gain; simplified
$t_s$	settling time
$\zeta$	damping ratio

## Nomenclature (continued)

(Symbols are listed in the order of appearance)

Symbol	Description
$\omega_n$	natural frequency
$r$	z-domain radius
$\omega_d$	damped natural frequency
$\theta_p$	z-domain pole angle
$T_d$	discrete sampling rate





## CHAPTER 1

### INTRODUCTION

Since the introduction of the flyball governor in the 1780's, control has had an impact on nearly every area of society through its foundational role in advancing guidance systems, manufacturing processes, industrial processes, and communication systems (Murray, R. M. 2003). The design of a controller for a system requires *a priori* knowledge of the systems' dynamics (Ljung, L. and Glad, T. 1994). The performance of a controller is highly dependent on the underlying system model used in its design. Modeling for mechanical engineering applications is traditionally done by deriving the differential equations of motion using physical laws. The equations of motion give the structure of the system model and by combining the systems' parameter values such as mass, damping, and stiffness the resulting dynamic model of the system can be used for control design. Many classical and modern control techniques can effectively be used to appropriate control laws using this dynamic model.

In many cases, the dynamic model developed contains levels of uncertainty in its structure due to un-modeled effects. In addition, knowledge of the system parameters is not always available and the necessary estimation of values such as mass, damping, and stiffness leads to uncertainty in the parameters.

When the controller is deployed to control a real plant, environmental disturbances and noise must be compensated for by the controller. However, the

disturbances and noise are usually unknown prior to implementation and rarely enter the plant model. These uncertainty issues have led to the areas of adaptive and robust control which have addressed the stability and performance of the controller when there is large uncertainty in the structure and parameters. Although robust and/or adaptive control can guarantee stability in many cases, both are computationally expensive and require expert knowledge to implement. A limitation of the classical, modern, robust, and adaptive approaches is the fact that they begin with a model that is an idealization of the real system. Although modeling by first laws often leads to sufficient control design it is still limited to an idealized system. Because of this, tuning of the controller is necessary when it is implemented. Tuning is the process of varying the controller parameters, or gains, until the desired performance of the plant is obtained. This can be a time consuming step in control implementation; especially, when there exists high levels of model uncertainty and multiple degrees of freedom in the controller parameters.

When a controller is designed for precision machines, the effect of structural and parametric uncertainty, disturbances, and noise play a significantly more important role in the controlled system performance. The level of modeling required to accurately represent the systems' structure, parameters, noise, disturbances, non-linearity, and etc. to design a precise controller will require expert knowledge and significant time investments. In practice, a considerable amount of time is spent on tuning such controllers even after the modeling and initial controller design has been accomplished. This is compounded when there

are multiple systems to tune. Even if detailed efforts are made to build an accurate model of the system, the actual system parameters may change over time and the system must be re-tuned. Changes in the system due to a crashes or component replacement will also necessitate re-tuning or result in sub-optimal control performance. An adaptive method of modeling that can be used to mitigate the uncertainty in parameters and capture these system changes is *system identification*.

System identification is the process of building a model based on an actual systems' input/output data. The model built by system identification can be used for model-based control. A model built by system identification minimizes the difference between the actual system response and the model response when acted on by the same input. System identification can be used to build a deterministic model of the plants dynamics as well as a stochastic model of the plant's disturbance/noise. The deterministic part of the model can directly be used for control design. The stochastic part of model can increase the accuracy of the parameters in the deterministic part of the model by properly filtering the plant disturbance and noise. For linear systems, the parameters of the identified model will approach those of the real system if input/output data is properly collected and the correct model structure is selected. Not only can system identification be used to create an initial model that incorporates the actual systems' dynamics but it can also be used intermittently or on-line to update the model parameters to account for time-varying parametric, structural, noise, and disturbance parameters. Therefore, a systems' model-based control performance, which is

related to the accuracy of the model, can be improved by proper system identification.

High precision systems, such as those used as tools to cut or measure parts with micron and sub-micron resolutions, are greatly affected by slight improvements in control performance. Control of high-precision machinery is necessary to understand manufacturing defects, maintain quality control, and obtain desired dimensional accuracy, surface roughness, and tolerances. Precision manufacturing and control is becoming increasingly more important in this society and around the world (Kurfess, 1996). The accuracy of precision machine motion is dependent on the ability of the controller to track a given trajectory which is dependent on the model that the controller was designed from. System identification can be used to build an accurate model of such a system and controller gains can be optimally determined by model-based control methods such as pole-placement.

The purpose of this research is to design an adaptive tuning and control method by system identification and pole-placement. A literature review of system identification is presented in Chapter II. The effect of parametric uncertainty as it relates to modeling is demonstrated by the level of effort required to identify these parameter by experiment in Chapter III. In Chapter IV it is shown that the modeling effort required to model and control a high-precision measurement machine using physical laws can be replaced by system identification. In Chapter V, a controller is designed based on the identified system using pole-placement. A region of feasible pole locations is explored. The

performance of the current machines' controller, based on a variant of Ziegler-Nichols, is compared to the performance of a controller designed by pole-placement using a model determined by system identification. Chapter VI summarizes how system identification can be automated and used to adaptively determine control gains that can virtually eliminate tuning. The use of system identification as an adaptive method of updating control gains has the potential to save valuable time and resources in industrial applications.

### **Research objectives**

In order to adaptively identify and tune a high-precision machine, the following objectives must be met:

- Review system identification methods and model based control design methods.
- Document the physics of a high precision measurement machine.
- Design a simulation of the plant and controller to verify that the physical modeling is in agreement with experimental data.
- Perform system identification to identify the model by using input-output data.
- Verify agreement of real system response and the identified response
- Develop a model-based controller from the identified system.
- Evaluate the performance of the controller designed from the identified model of the system vs. the current control design based on Ziegler-Nichols methods.



## CHAPTER II

### LITERATURE REVIEW

In this section the various methods of system identification and model-based control design are introduced. Within the practice of system identification there are numerous techniques to obtain system information. This information serves different purposes and yields different insights of the system. These techniques can be classified into parametric and non-parametric methods. Non-parametric methods have a rich history and were predominantly used prior to the 1960's. Parametric methods have dominated system identification ever since. Non-parametric methods still give physical insight and are still widely used but they do not directly result in a model for use in control design. Once system identification has been successfully completed, various model-based control design techniques can be used to design an optimal controller. These can also be broken down into two categories: classical and modern. This section concludes with a summary of the system identification and how they relate specifically to designing a controller for the system at hand.

#### **System Identification**

System identification was historically the work of non-engineering fields such as mathematics, time-series analysts, and econometricians (Gevers, M. 2005). System identification in these fields was referred to as estimation theory

and it contains a rich statistical history. Models were constructed by using first laws and Bode plots while control design was based on Bode, Nyquist, and Ziegler-Nichols plots. However, these approaches were limited to single-input-single-output or SISO systems (Gevers, M. 2006). With the combination of the introduction of state-space in the 1960's, the availability of affordable transistor-based computers, the minimum state realization by (Ho, B. L. and Kalman, R. E. 1965), and the introduction of the Maximum Likelihood for parametric models by (Åström, K. J. and Bohlin, T. 1965), system identification research began to attract great interest.

System identification can be broken up into parametric and non-parametric identification. Non-parametric identification methods are graphical in nature and result in qualitative information about the system (Eykhoff, P. 1974). System identification by control engineers was primarily done by non-parametric estimation until the 1960's. Non-parametric methods are used to estimate the impulse and frequency response of the system from a given set of data by frequency response analysis, correlation analysis, and spectral analysis. These methods are well known and were used by engineers to obtain qualitative information about system characteristics for modal analysis and graphical transfer function estimation (Åström, K. J., and Eykhoff, P. 1971), (Rake, H. 1980), (Wellstead, P. E. 1981), (Juang, J.-N. and Pappa, R. S. 1988). Non-parametric methods provide useful information about the system but they are limited to single-input single-output systems and do not immediately result in a model that can be used for control (Ljung, L. and Glad, T. 1994). Current system



identification for control is performed by parametric methods which directly result in a parametric model that can be used for control design.

Parametric methods are the determination of model parameters from time series data and have a rich statistical history that can be traced back to (Gauss, C. F. 1809). These methods were greatly developed by econometricians and time series analysts until the 1960's (Deistler, M. 2002). Parametric system identification by engineers became increasingly popular because of the need for model based control. The models resulting from parametric identification can directly be used for control design.

### **Non-Parametric Methods**

Physical insight and time-domain characteristics can be obtained by the transient response analysis of a system to an impulse or step response. Impulse and step responses can provide information such as the stability, the dominant time constants, time delays, and damping characteristics. Although, step and impulse responses can give such qualitative information they do not yield a model structure or model parameters. In addition, impulse responses can only be approximated and seldom can be used to excite real engineering systems. This is because there is either not enough available power to excite the frequency spectrum of interest or that an impulse input would damage the system. An alternative approach is the use of high energy signals as measured by a crest factor.

### Non-Parametric Estimation of Impulse Response:

Estimation of the impulse can be accomplished without an impulse input (Wellstead, P. E. 1981), (Rake, H. 1980). This is done by correlation analysis of the input and output when the input is a low crest factor signal. The crest factor is defined as the input of a signal divided by its root mean square value as shown in Equation(2.1).

$$C_f \equiv \frac{\max_t u(t)}{\sqrt{\frac{1}{N} \sum_{t=1}^N u^2(t)}} \quad (2.1)$$

Therefore, if the system is excited with filtered Gaussian white noise, random binary signals, or a pseudo random binary signals (PRBS), which are signals with low crest factors, then the system input can contain the frequency content of interest. Therefore, the system input can have almost as much content that theoretically is present in a true impulse. In practice PRBS are often used because they are easy to generate on a digital computer with the use of EXOR logic functions and shift registers (Wellstead, P. E. 1981). In addition to the use of these high crest factor inputs, other inputs may be used as long as the input and output are both filtered by a whitening filter. Use of such a filter ensures that the integrity of the correlation analysis is maintained.

Correlation analysis can be used to approximate the impulse response of the system. It is well know that if the impulse response of a system is known then

the system characteristics are completely described. The response of a discrete system can be given by Equation (2.2).

$$y(n) = \sum_{k=0}^{\infty} u(n-k)h(k) + e(n) \quad (2.2)$$

Where  $y$  is the output,  $u(n-k)$  is the input,  $h(k)$  is the impulse response, and  $e(n)$  is the error from the disturbance and noise. If the input to a system is zero-mean white noise and the disturbance and input are uncorrelated, then correlation analysis between the input and output can be used to estimate the impulse response as in Equation(2.3).

$$h(k) = \frac{R_{uy}(k)}{\sigma_u^2} \quad (2.3)$$

Where,  $R_{uy}$  is the cross correlation of the input and output and  $\sigma_u^2$  is the variance of the signal. The details of the calculation are given in (Ljung, L. 1999). The impulse response can be used to estimate the stability, dominant time constants, time delays, and damping characteristics but not the order or structure of the dynamic model. Because the impulse response relies on uncorrelated disturbances it is biased if used for closed loop identification (Ljung, L. and Glad, T. 1994). From a control design point of view the frequency response contains more useful information with respect to model construction; especially for first and second order systems.

### Non-Parametric Estimation of Frequency Response:

Non-parametric identification techniques are not limited to the time domain and there are three methods of particular importance in the frequency domain: frequency response analysis, Fourier analysis, and spectral analysis.

The frequency response of a system is simply the construction of the Bode plot from sinusoidal input-output data recorded at different frequency steps. The Bode plot can be used to approximate the DC gain, system type, time constant, damping, order, bandwidth, gain margin, and phase margin for a linear time-invariant system. However, this approach may not be well suited for some systems may be lengthy when sequentially traversing low frequencies (Kurfess, T. R. 1996). This simple yet powerful method was used with the sinusoidal transfer function analyzer up to the late 1960's when (Cooley, J. W. and Tukey, S. W. 1965) demonstrated that the computational difficulty of the DFT could be resolved by using and FFT and digital computers.

Another method to estimate a system model, or at least get a good qualitative estimate of the frequency response, is by Fourier analysis. Fourier analysis consists of taking the discrete Fourier transform of the input and output data and estimating the transfer function by their ratio. If the input output relationship is given by,

$$Y(\omega) = G(i\omega)U(\omega) \quad (2.4)$$

Then an estimate for  $G(\omega)$  can be computed by,

$$G(i\omega) = \frac{U(\omega)}{Y(\omega)} \quad (2.5)$$

Where the Fourier transform of the input and output yield,

$$U_{\Omega}(\omega) = \int_0^{\Omega} y(t) e^{-i\omega t} dt, \quad (2.6)$$

$$Y_{\Omega}(\omega) = \int_0^{\Omega} u(t) e^{-i\omega t} dt. \quad (2.7)$$

Then the estimate for  $G(\omega)$  can be expressed over the time  $0 < t < \Omega$  as,

$$\hat{G}_{\Omega}(i\omega) = \frac{U_{\Omega}(\omega)}{Y_{\Omega}(\omega)} \quad (2.8)$$

This is called the Empirical Transfer Function Estimate or ETFE. This can be used with good results if the input is periodic which causes the variance to decrease with larger data sets. However, if the input is not periodic then the variance does not decrease with larger data sets and equals the signal-to-noise ratio. A full theoretical analysis of the ETFE's is given in (Ljung, L. 1999).

One of the limitations to the above methods is that they can only be used when the input is a sinusoidal signal. Spectral Analysis is a powerful tool for frequency analysis where the input signal does not need to be periodic and can be directly applied to data. Spectral analysis is theoretically the Fourier transform of

a systems impulse response. However, because of the need for windowing and the nature of sampled data systems this is not done in practice and correlation analysis used instead. The spectrum of a signal is defined as the square of the absolute value of its Fourier transform at different frequencies. If the signal is the realization of a stationary stochastic process then the signals spectra can be defined in terms of its expectation and covariance (Wellstead, P. E. 1981). If a system sampled at time intervals,  $T$ , is described with input-output dynamics,

$$y(t) = G(q)u(t) + v(t) \quad (2.9)$$

then the signals spectral density at a certain frequency can be found by taking the Fourier transform of the correlation functions. Where  $G(q)$  is the discrete transfer function with the shift operator  $q$ ,  $y(t)$  is the output,  $u(t)$  is the input, and  $v(t)$  is the disturbance at time,  $t$ . The spectrum of the input signal is given by,

$$\hat{\Phi}_u(\omega) = \lim_{N \rightarrow \infty} \frac{1}{N} |U_N(\omega)|^2. \quad (2.10)$$

Where  $U_N(\omega)$  is the Fourier transform of the input data,

$$U_N(\omega) = \sum_{t=1}^N u(t) e^{i\omega t}. \quad (2.11)$$

If  $u$  and  $v$  are uncorrelated, then the cross-covariance of the input and output is given by,

$$\hat{R}_{yu}(kT) = Ey(t)u(t-kT) = \frac{1}{N} \sum_{t=1}^N y(t)u(t-kT) \quad (2.12)$$

And the DFT is,

$$\hat{\Phi}_{yu}(\omega) = T \sum_{k=-\infty}^{\infty} \hat{R}_{yu}(kT) e^{-j\omega kT}. \quad (2.13)$$

However, because the large time lags give high variances for the cross-covariance, windowing must be applied to weigh out larger time lag values. Windowing reduces the variations in the cross-covariance but also its resolution. If a system does not have resonances close to one another, then windowing can be applied to smooth out the frequency estimate without significant loss of information. There are a number of different windows that can be used but perhaps the most common are the Hamming and Blackman-Tukey windows. Full treatment of windowing is given by many authors such as (Oppenheim, A. V., Willsky, A. S. et al. 1997), (Oppenheim, A. V., Schaffer, R. W. et al. 1999). By taking the Fourier transform of the input and cross-covariance and applying the proper window, an estimate for the frequency response function can be obtained by Equation (2.14),

$$\hat{G}(e^{j\omega T}) = \frac{\hat{\Phi}_{yu}(\omega)}{\hat{\Phi}_u(\omega)}. \quad (2.14)$$

A limitation of spectral analysis for model identification is that it relies on the disturbance,  $v$ , being uncorrelated with the input,  $u$ . This is not the case for data generated in closed loop where the disturbances enter into the feedback. The non-parametric methods discussed above are excellent ways to obtain general information about a system that can compliment parametric identification. These methods are graphical and do not immediately result in a model which can be used for control design. However, the information obtained by using the parametric methods can help determine the order and delay for parametric methods. Parametric identification is much more powerful for control design because they directly result in a model as their output.

### **Parametric Methods**

Parametric system identification is the fitting of model parameters to a pre-selected model by using input-output data. The qualitative information by non-parametric identification can be used to select the proper model structure. Parametric identification can be seen as identifying the optimal parameters of a filter of pre-determined order. The parameters identified by system identification are the best approximation to the real model parameters with respect to a certain criteria such as the minimum of the norm between the estimate and residuals or the least-squares minimum.

Parametric models can be constructed for deterministic elements of the system that characterize intrinsic dynamics and for stochastic elements of the system addressing disturbances and noise. As such, the dynamics of the system



and the noise can be accurately represented in a single model that can be used for control design. Parametric methods are separated into Grey-Box estimation, where some of the parameters are known, and Black-Box estimation where none of the parameters are known. Most of the parametric methods can be described as variants of the general linear parametric model given in Figure 1.

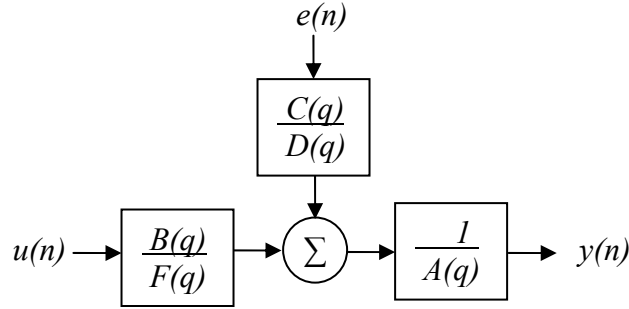


Figure 1: General Linear Parametric Model

Where the  $q$  is the discrete shift operator  $u(n)$  is the input,  $e(n)$  is the noise and disturbance,  $y(n)$  is the output and  $A(q)$ ,  $B(q)$ ,  $C(q)$ ,  $D(q)$ , and  $F(q)$  are finite difference equations. The output to input relationship of the general linear model can be described by discrete transfer functions,

$$y(n) A(q) = G(q)u(n) + H(q)e(n). \quad (2.15)$$

Where the deterministic part of the system model is given by,

$$G(q) = \frac{b_1 q^{-nk} + b_2 q^{-nk-1} + \dots + b_{nb} q^{-nk-nb-1}}{1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf}} \quad (2.16)$$

and the stochastic part of the system model based on the second order statistics of the error signal is given by,

$$H(q) = \frac{1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}}{1 + d_1 q^{-1} + \dots + d_{nd} q^{-nd}}. \quad (2.17)$$

Under varying assumptions and *a priori* knowledge about the system (which can be obtained by non-parametric methods as discussed above) the general linear model can be reduced to other forms. Selecting the order of the model and predicting how the error or disturbance enters a system is all that is need to select a specific model. The different assumptions leading to some of the different models are shown in Table 1. These models are described generally in (Ljung, L. and Glad, T. 1994) and in great detail by (Ljung, L. 1999).

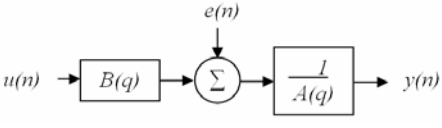
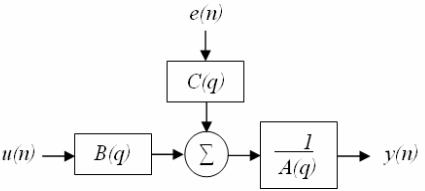
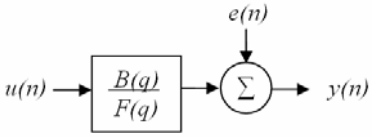
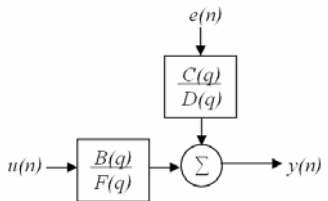
The general method to solve the model coefficients summarized in Table 1 is to minimize the prediction error of the selected model and the actual output, solve for the parameters using a correlation function, or by subspace identification. There are various techniques used in practice to minimize this prediction error. In general, when the system is in the form of (2.15) then the prediction is given by (2.18).

$$\hat{y}(t|t-1) = H^{-1}(q)G(q)u(t) + [1 - H^{-1}(q)]y(t) \quad (2.18)$$

And the prediction error is given by,

$$e(t) = y(t) - \hat{y}(t|t-1). \quad (2.19)$$

Table 1: Black Box Models and Their Assumptions

MODEL NAME	DISTURBANCE ASSUMPTION	OUTPUT DEPENDENCE
 <p style="text-align: center;">ARX</p>	Disturbances enter the process early and share the system dynamics.	Past outputs, current input, past inputs, and past dynamic disturbance.
 <p style="text-align: center;">ARMAX</p>	Disturbances enter the process early and share the system dynamics.	Past outputs, current input, past inputs, and past dynamic disturbance.
 <p style="text-align: center;">Output Error (OE)</p>	Disturbance properties are not modeled.	Past outputs, current input, and additive disturbance.
 <p style="text-align: center;">Box-Jenkins (BJ)</p>	Disturbance and system properties are independently dynamic.	Past outputs, current input, and dynamic disturbance.

In order to fit the parameters of a given model to the data criterion function based on the prediction error must be minimized (Ljung, L. 1999). The criterion function to be minimized is

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N e^2(t, \theta), \quad (2.20)$$

which measures how well the parameters fit the data. There are other criterion functions that can be used to measure the goodness of fit and the general case is

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \ell(e(t, \theta)). \quad (2.21)$$

If the function,  $\ell(\cdot)$  is selected to be the logarithm of the probability density function of the noise, then this general approach is called the *maximum likelihood* (ML) estimate of the model parameters (Ljung, L. 1999).

For high precision machines the sources of errors and noise are of a completely different nature than low precision machines. In high precision machines, noise and error sources from temperature variations, room acoustics, floor vibrations, machine resonances, quantization effects, pressure fluctuations in air bearings, misalignment of axes, electrical noise from surrounding electronics and power sources, and etc. have a much greater impact than in lower precision machines. Noise is present in all systems, but for precision machines noise is the greatest source of error and must be understood and eliminated as much as

possible. Precision machine philosophy is deterministic and the sources of noise and how they affect a system is usually or can be well understood.

If it is known that the noise enters the process early either an ARX or ARMAX model can be a candidate to describe the system and error dynamics. The assumption of both the ARX and ARMAX models is that the noise shares the same dynamics (or poles) as the system. Perhaps the most commonly used parametric model is the ARX model where the AR means autoregressive and corresponds to the  $A(q)y(t)$  in (2.22) and the X means extra input corresponding to the  $B(q)u(t)$  in (2.22) and the first row in Table 1.

$$A(q)y(t) = B(q)u(t) + e(n) \quad (2.22)$$

The system identification problem is then to compute the coefficients of  $A$  and  $B$  from the input-output data. The coefficients of  $A$  and  $B$  can be placed in a vector show in

$$\theta = [a_1, a_2, \dots, a_{na} \ b_1, b_2, \dots, b_n]. \quad (2.23)$$

If the past inputs and outputs are also collected in a vector,

$$\varphi(t) = [-y(t-1), -y(t-2), \dots, -y(t-na), u(t), u(t-1), \dots, u(t-nb+1)]^T \quad (2.24)$$

If the noise is Gaussian the maximum likelihood estimate of the parameters is estimated by the least squares solution (2.25).

$$\hat{\theta} = \left[ \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \sum_{t=1}^N \varphi(t) y(t) \quad (2.25)$$

Where  $\hat{\theta}$  is the minimization of the norm of the prediction error between the predicted output value from the ARX model and the measured value from the actual output (Ljung, L. 1999). If the noise is not Gaussian then the Instrument Variable (IV) method can be used to estimate the parameters. This method is described in regards to system identification by (Ljung, L. 1999) and (Stocia, P., Söderström, T. et al. 1985) and is a well known method in statistics for parameter estimation.

When the noise is a moving average and shares the same dynamics of the system, then an autoregressive moving average or ARMAX model can be used to describe the dynamics. The ARMAX model is an ARX model with an additional moving average or MA term applied to the error input. The ARMAX model in equation form is given in (2.26), where  $C(q)e(n)$  corresponds to the additional term accounting for the MA error dynamics

$$A(q)y(t) = B(q)u(t) + C(q)e(n) \quad (2.26)$$

In order to fit the parameters of the ARMAX model and minimize the prediction error a two-stage estimation approach is used. The maximum likelihood estimate for the ARMAX model was first introduced by (Åström, K. J. and Bohlin, T. 1965) for system identification. In practice, the first step to fitting the parameters is to obtain a rough estimate of the parameters by pseudo regression and then perform a minimization of the errors by a method such as Gauss-Newton. The ARMAX model allows for a higher degree of freedom in the error term than the ARX model and can be used to describe processes where the error is assumed to be a moving average of white noise.

The ARX and ARMAX models are both called equation error models because the error input to the model shares the same dynamics. If the error dynamics of a system do not share the system dynamics then an Output Error model should be used to describe this process. Output error models are used to describe systems where the errors due to noise and disturbances enter the system late in the process and do not share the deterministic poles. The two most common Output Error models are the Box-Jenkins (BJ) model and the Output-Error (OE) model which shares the same name as this class of model descriptions. Both of these models are shown in Table 1. The relationship between the input and output for the OE model is shown in (2.27).

$$y(t) = \frac{B(q)}{F(q)}u(t) + e(n) \quad (2.27)$$

In this equation, the error which is assumed to be white noise is decoupled from the system dynamics. Systems with randomly distributed errors due only to measurement of the output can be accurately described by such a model.

When the error is also dynamic but decoupled from the system dynamics then the Box-Jenkins model shown in (2.28) should be used.

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(n) \quad (2.28)$$

This model is named after statisticians G. P. Box and G. M. Jenkins (Box, G. E. P., Jenkins, G. N. et al. 1994). The parameters of both the OE and BJ are estimated by coarse estimation and then minimization. The instrument variable method is used to determine the parameters of  $B$  and  $F$  and minimization by Gauss-Newton is done to fit the entire model parameters (Ljung, L. 1999).

In addition to the above parametric methods there are also parametric state-space models. The parameters of the state-space models can be identified by using past data records in the same way as the parametric models above. The discrete state space model of the system consists of a number of first order difference equations in the following form,

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t) \quad (2.29)$$

$$y(t) = C(\theta)x(t) + v(t) \quad (2.30)$$



Where, the  $x$  is the system state,  $y$  is the system output, and  $A(\theta)$ ,  $B(\theta)$ , and  $C(\theta)$  are the state-space matrices which are parameterized by the unknown parameter vector,  $\theta$ . The state-space equations can be constructed by first law modeling in terms of the unknown parameters or by black box estimation (Nelles, O. 2001). When the noise in the system is more complex, a Kalman filter can be used to optimally filter the noise leading to the innovations model of the state space form in Equations (2.31)-(2.32), (Kalman, R. E. 1960a), (Kalman, R. E. 1960b), (Ljung, L. 1999). The innovations form of the state space model takes the error, or residuals, from the predicted output and actual output into account when determining the Kalman gain that minimizes the error in the state estimate.

$$\hat{x}(t+1, \theta) = A(\theta)\hat{x}(t, \theta) + B(\theta)u(t) + K(\theta)e(t) \quad (2.31)$$

$$y(t) = C(\theta)\hat{x}(t, \theta) + e(t) \quad (2.32)$$

Where,  $K$  is the Kalman gain applied to the process error and  $\hat{x}$  is the state estimate. Kalman filtering is a way of optimally estimating the state of the system based on the probability of the predicted state and its covariance. The error in the prediction of a state is used to update the probability of that state and correct its covariance. Therefore, the Kalman filter is a predictor-corrector filter that uses the probabilities of state estimates to weigh the estimates accordingly and determine the Kalman gain. The algorithm to compute the optimal parameters,  $\theta$ , is called subspace identification (Van Overschee, P. and De Moore, B. 1994), (Ljung, L. 1999). There are many books that deal with state-space modeling and control such

as (Åström, K. J. and Wittenmark, B. 1989), (Ogata, K. 1987), (Ogata, K. 2002). State-space models are advantageous in that they reflect the physics of the identified system better than any other parametric method (Ljung, L. 1999).

Parametric methods are powerful means for determining a system model. The flexible nature of parametric identification methods allows an accurate system model to be built. The model resulting from parametric system identification can be used directly for model-based control design.

### **Control Design**

The control of high precision machines requires an accurate model so the proper control structure and gains can be selected. The model parameters can be found using the system identification techniques discussed above. Without an accurate model, control design relies on methods that contain high uncertainty such as Ziegler-Nichol and variants of the same. System identification software is readily available for accurate model identification such as (LabVIEW), (MATLAB), (SOCIT), However, many industrial controllers still rely on control design from an uncertain model that results in long tuning times and requires expert knowledge. Parametric system identification can be used to construct a model or greatly improve its accuracy thereby removing the need for expert knowledge and long tuning times. In addition, models identified by data collected under closed loop, which is usually the case, are actually best for control design (Gevers, M. and Ljung, L. 1986).

Once the desired performance of a controller is specified, then there are a variety of control design techniques that can be implemented for precision control. All of the parametric models can be put into a transfer function or state-space. The most common linear time-invariant control design methods are discussed below.

Control design can be accomplished when the model is unknown. Ziegler and Nichols (Ziegler, J. G. and Nichols, N. B. 1942), (Ziegler, J. G. and Nichols, N. B. 1943) proposed methods by which PID control gains can be approximated from step response data. However, this approach leads to large overshoot and requires fine tuning of the system (Ogata, K. 2002). In order to use such a control approach for high precision systems this requires long tuning time and experience. If the system parameters change over time, then this time consuming process must be repeated. Model based control design is significantly more accurate and requires little or no tuning if modeled correctly.

If the model is known then control design can be done based on root locus analysis which was introduced by Evans (Evans, W. R. 1948), (Evans, W. R. 1950). By placing a controller with a certain control structure in the control loop, the dominant poles of the system can be placed where desired assuming that the system actuators and power supply is not limited. Then, the gains of the system required to place the poles at the desired locations can be determined by examining how the position of the dominant poles changes with respect to the change in gain (Ogata, K. 2002). Powerful design tools such as MATLAB's

control design toolkit offer graphical software to place the poles and observe the change in P, PI, PD, or PID gains.

If a state space system model identified by system identification is controllable, i.e. all the states can be driven to zero, and observable (Kalman, R. E., Ho, Y. C. et al. 1963), i.e. all the state can be estimated, then all of the states can be observed and controlled by pole placement of the controller. All of the system states cannot always be observed but if the system is observable, these can be estimated by state observers. A full treatment of discrete state space design is given by (Ogata, K. 1987) and continuous state space design by (Ogata, K. 2002).

In addition to root locus and state space controller design, frequency domain design based on the identified model can also be used to determine the compensators and gains to achieve the desired gain and phase margin of the closed loop system.

The above control design techniques apply to linear time invariant systems. Precision machines that are structurally designed for ultra-high performance are necessarily designed to be highly linear. The use of system identification is well suited for high precision machines where the use of granite structures, air bearings, tightly controlled temperature conditions, linear motors, laser interferometers, glass scales, and etc. eliminate common non-linearities. The use of a granite frame makes the system robust against temperature variations, the use of air bearings greatly mitigates the non-linear effects of friction, and the linear motors eliminate the presence of backlash from their ball-screw alternatives. Such components make the assumption of linearity easily

justified. Therefore, system identification of a highly linear precision machine can be used to obtain an accurate system model and design a high precision controller to achieve the desired performance.



## CHAPTER III

### EQUIPMENT, INSTRUMENTATION, AND SOFTWARE

#### Universal Measuring Machine

In this section the Universal Measuring Machine's (UMM) architecture, actuators, controllers, and mathematical axis models are presented. The UMM is shown in Figure 2.



Figure 2: Universal Measuring Machine

#### **Architecture**

The UMM is a continuous-contact or scanning probe measurement machine used to measure a parts geometry for quality control during

manufacturing. The UMM is basically a CMM with an R-axis, a Z-axis, and a fixed rotational C-axis. The R-axis moves the C-axis and part in the horizontal direction, the Z-axis carries a measurement probe in the vertical direction, and the C-axis, which is mounted on the R-axis, rotates the part. Two additional axes are used to center the part on the C-axis. The P-axis is used to horizontally center the part on the C-axis by pushing it into place. The W-axis is the vertical centering axis used to vertically position the pusher. A schematic of the UMM is shown in Figure 3. The R, C, and Z axes are all frictionless air-bearing axes driven by brushless servo-motors controlled by the Programmable Multi-Axis Controller (PMAC). The R- and Z-axes use Trilogy 310 series linear motors while the C axis uses an integral frameless rotary motor. The P-axis is a Trilogy 210 series linear motor and the W-axis is lead screw driven by a brushless servo motor. The above discussion is summarized in Table 2.

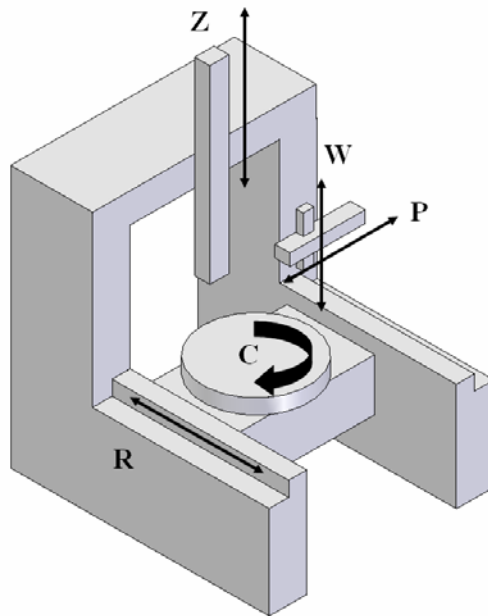


Figure 3: Schematic of UMM architecture



Table 2: UMM Axis Motor and Bearing Type.

<b>UMM</b>	<b>Motor</b>	<b>Manufacturer</b>	<b>Motor Type</b>	<b>Bearing Type</b>
<b>Z-Axis</b>	310-4S	Trilogy-Parker	Linear	Air
<b>R-Axis</b>	310-5P	Trilogy-Parker	Linear	Air
<b>C-Axis</b>	K254-100-H01-001	Bayside	Rotary	Air
<b>P-Axis</b>	210-2S	Trilogy-Parker	Linear	Rolling
<b>W-Axis</b>	CM231AE-00060	Compudor	Rotary	Element

Natural frictional damping of the air bearing motors is very small and damping must be done either by the controller or other means. Damping of the R- and Z-axis is accomplished by eddy current dampers. The eddy current dampers consist of aluminum blocks attached to the coil of the linear motors. As the R-axis moves through the magnet track the eddy currents induced in the aluminum block resist forward motion. Damping of the C-axis is done by the controller. Damping of the other W- and P-axes is done by both the controller and friction.

The position of the R-axis and Z-axis is determined by laser interferometers. The C-axis position is determined by a rotary encoder. The P- and W-axis position is determined by linear encoders. The axis and corresponding encoder and its resolution is summarized in Table 3.

The linear motors are commutated brushless DC motors. Two of the phases are commutated by the controller. The two commutated phases are fed into an amplifier and the amplifier commutates the third phase. The amplifier is a Glentek linear amplifier with a low-pass cutoff frequency of 523 Hz.

Table 3: Axis Encoder and Resolution

<b>UMM</b>	<b>Encoder Type</b>	<b>Resolution</b>
<b>Z-Axis</b>	Laser Interferometer	809,070 counts/mm
<b>R-Axis</b>	Laser Interferometer	809,073 counts/mm
<b>C-Axis</b>	Rotary Encoder	4,551 counts/deg
<b>P-Axis</b>	Linear Encoder	4,000 counts/mm
<b>W-Axis</b>	Linear Encoder	15,748 counts/mm

### **Controller**

The controller is a Turbo Programmable Multi-Axis Controller, or PMAC, made by Delta Tau. The controller is a multi-axis PID controller with feed-forward and feedback capabilities. The controller performs cascaded loop control to maintain desired position of the R-, Z-, and C-axes while maintaining the desired force in the probe along the surface of the part. In order to work within the available memory of the PMAC, and correctly deal with the different resolutions of the encoders, scale factors are used throughout the control loop. The PMAC command output is governed by Equation (3.1) where the  $n$  represents the time step. The output servo command is commutated and sent to a linear differential amplifier that is tuned for each axis. The PMAC takes approximately 0.443ms per servo cycle which is about 2257 samples per second which represents a loop closure rate of 2.257 kHz for control purposes. The command output of the PMAC shown in Equation (3.1) is essentially a PID filter with feed-forward

terms; the variables are listed in Table 4. The command output is given in encoder counts and limited to 32,767 encoder counts with a range of  $\pm 10V$  volts.

$$\text{CMDout}(n) = 2^{-19} \cdot \text{Ix30} \cdot \left[ \text{Ix08} \cdot \left[ \begin{array}{l} \text{FE}(n) + \frac{\text{Ix33} \cdot \text{IE}(n)}{2^{23}} \\ + \frac{\text{Ix32} \cdot \text{CV}(n) + \text{Ix35} \cdot \text{CA}(n)}{2^7} \end{array} \right] - \frac{\text{Ix31} \cdot \text{Ix09} \cdot \text{AV}(n)}{2^7} \right] \quad (3.1)$$

Table 4: PMAC Controller Variables

<b>PMACS Variable Name</b>	<b>Description</b>
Ix30	PMACS Proportional Gain
Ix08	Position Scale Factor
Ix09	Velocity Loop Position Scale Factor
Ix33	PMACS Integral Gain
Ix31	PMACS Derivative Gain
Ix32	PMACS Feed-Forward Velocity Gain
Ix35	PMACS Feed-Forward Acceleration Gain
CA(n)	Command Acceleration
CV(n)	Command Velocity
FE(n)	Following Error
IE(n)	Integration Error
AV(n)	Actual Velocity

## **Tuning**

In order to obtain the desired performance each axis of the UMM must be independently tuned. The UMM has been tuned by an experienced professional although the PMAC has tuning software that can determine the PID gains. The PMAC tuning algorithm is proprietary but likely a variant of Ziegler-Nichols and tuning done by the professional performs much better. All of the current tuning is done without a system model. Although the performance objectives are achieved by this approach, model based tuning is much more efficient. However, creating a model with accurate parameters is difficult and time consuming as discussed in the next chapter. Both the modeling and tuning can be performed by system identification. Algorithms to automatically tune each axis can be created to tune the system on start-up or at preset time intervals. This would allow the machine to correct the gains automatically for time-varying parameters or event changed parameters such as a collision.

## **System Simulation**

Determination of the dynamic models and their parameters for each UMM axis requires proper input-output data. Unfortunately, the input-output data were not available. Therefore, an accurate simulation created by physical modeling using first laws and manual system parameter determination was used to simulate identification data. The R- and C-axis were simulated because they are representative of all the other axes. To verify the models for the R- and C-axis the

PMAC controller was also simulated. In this section, the simulation and methods by which the R- and C-axis parameters were determined is presented. The simulation and actual data show extremely close agreement indicating that the modeling of the two axes and controller were done correctly.

### Modeling of C-axis:

The C-Axis is a commutated brushless DC motor as shown in Figure 4. The motor is modeled as a circuit with a resistor, inductor, and ideal motor that supplies torque proportional to the current less the back emf. By summing the voltages around the loop according to Kirchoff's Voltage Law yields equation (3.2).

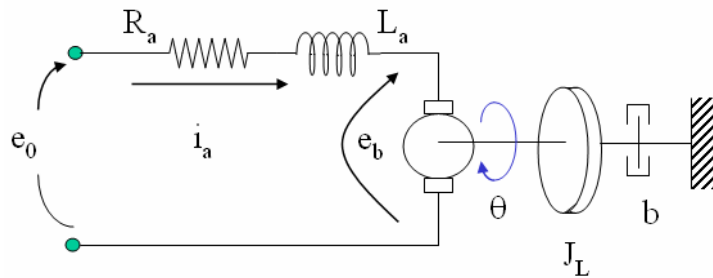


Figure 4: DC motor model

$$e_0 = R_a i_a + L_a \frac{di_a}{dt} + e_b \quad (3.2)$$

Where  $e_0$  is the applied voltage to the motor  $R_a$  is the resistance in the motor,  $i_a$  is the current through the motor,  $L_a$  is the inductance in the motor caused by the coils, and  $e_b$  is the back-electromotive force or back-emf. The back-emf is

proportional to the speed of the motor shaft as shown in Equation 4 where  $K_b$  is the back-emf constant and  $\omega$  is the angular speed.

$$e_b = K_b \frac{d\theta}{dt} = K_b \omega \quad (3.3)$$

The torque supplied by the motor is proportional to the current as seen in Equation 5 where  $K_t$  is the torque constant of the motor.

$$T = K_t i_a \quad (3.4)$$

The moment balance on the motor shaft is shown in equation (3.5). Where  $J_L$  is the rotational inertia and  $b$  is the viscous damping.

$$T = J_L \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = J_L \frac{d\omega}{dt} + b\omega \quad (3.5)$$

The Laplace transform of equations (3.2)-(3.5) assuming zero initial conditions yields equations (3.6)-(3.9).

$$e_0 = R_a i_a + L_a \frac{di_a}{dt} + e_b \implies E_0(s) = (R_a + L_a s) I_a(s) + E_b(s) \quad (3.6)$$

$$e_b = K_b \frac{d\theta}{dt} = K_b \omega \implies E_b(s) = K_b \Omega(s) \quad (3.7)$$

$$T = K_t i_a \Rightarrow T(s) = K_t I_a(s) \quad (3.8)$$

$$T = J_L \frac{d\omega}{dt} + b\omega \Rightarrow T(s) = (J_L s + b)\Omega(s) \quad (3.9)$$

The above equations can be combined algebraically to create the speed per voltage transfer function as in equation (3.10) or as a block diagram as shown in Figure 5.

$$\frac{\Omega(s)}{E_o(s)} = \frac{K_t}{L_a J_L s^2 + (L_a b + R_a J_L) s + (R_a b + K_t K_b)} \quad (3.10)$$

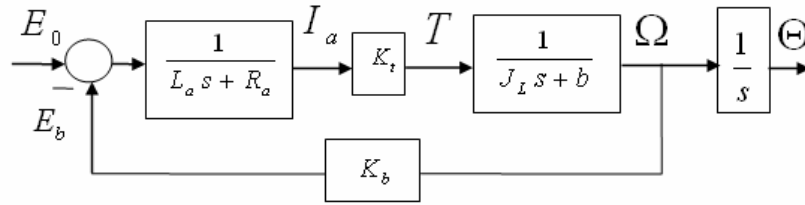


Figure 5: Block Diagram of DC motor

The proportionality constants,  $K_t$  and  $K_b$ , are given by Bayside and shown in Table 5. The integrator in Figure 5 can be used to get the position per input voltage transfer function which is the information obtained from the encoder.

The C-axis is actuated by an input command that comes from the PMAC in the form of  $\text{DAC}_{\text{counts}}$ . This value is then converted by a scale factor,  $K_{\text{DAC}}$  of  $20/65536 \text{ V/DAC}_{\text{counts}}$  and sent to the amplifier as a command voltage across the motor. The amplifier then turns this value into a command current proportional to

its transconductance,  $K_{TC}$  which is 0.5 Amps/Volt for the C-axis. The output position is converted from counts to degrees by a rotary encoder and is the feedback to the PMAC. The amplifier/motor block diagram is shown in Figure 6.

Table 5: C-Axis Catalogue Values

Winding Type	Motor Torque Constant [oz. in./Amp]	Motor Torque Constant [Nm./Amp]	Back Emf Constant [V-s/rad]	Resistance [Ohm]	Inductance [mH]
G	682.3	4.82	4.82	6.3	63.72
H	856.22	6.05	6.05	9.96	100.35

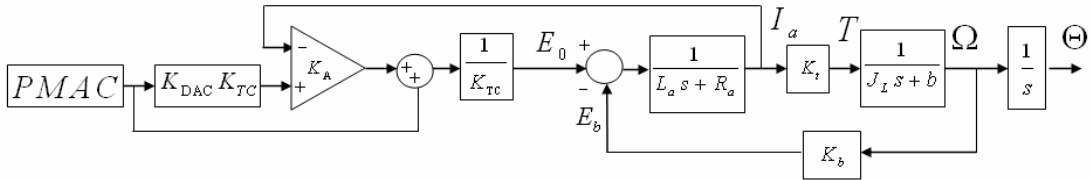


Figure 6: DC motor model with amplifier dynamics added

The amplifier dynamics are much faster than the motor dynamics. A reasonable assumption is that they are negligible. However, the amplifier is included in the model so that the variables associated with it can be adjusted to see their effect on the overall system. The amplifier also acts as a low-pass filter of 523Hz. When the PMAC command and amplifier dynamics are included in the model the input voltage is given by equation (3.11).



$$E_o(s) = PMAC_{\text{COMMAND}} \cdot K_{\text{DAC}} \cdot (1 + K_A \cdot (I_C - I_a)) \quad (3.11)$$

By combining equation (3.11) and the integral of the angular speed to get the angular position, (3.11) becomes,

$$\Theta(s) = \left( \frac{K_t \cdot PMAC_{\text{COMMAND}} \cdot K_{\text{DAC}} \cdot (1 + K_A \cdot (I_C - I_a))}{L_a J_L s^2 + (L_a b + R_a J_L) s + (R_a b + K_t K_b)} \right) \cdot \frac{1}{s} \quad (3.12)$$

In order to evaluate this model it was simulated in MATLAB's Simulink. The value of actual system's inertia was unknown and determined according to the following procedure.

### C-Axis Inertia Determination

The following discussion outlines an experimental determination of the UMM C-axis inertia. The actual position of the C-axis is in the form of Equation (3.13),

$$\theta = A_\theta \cdot \sin(\omega \cdot t) \quad (3.13)$$

Where  $\theta$  is the actual position in degrees,  $A_\theta$  is the amplitude of the servo command in degrees,  $\omega$  is the frequency in radians per second, and  $t$  is the time in seconds. Taking the derivative of position yields velocity and the taking the derivative again yields acceleration as shown in Equations 15 and 16 respectively.

$$\Omega = A_{\theta} \cdot \cos(\omega \cdot t) \cdot \omega \quad (3.14)$$

$$\alpha = -A_{\theta} \cdot \sin(\omega \cdot t) \cdot \omega^2 \quad (3.15)$$

From the moment balance on the C-axis shaft,

$$\sum T = J \cdot \alpha + b\Omega, \quad (3.16)$$

where,  $T$  is the torque in Nm,  $J$  is the rotational inertia in  $\text{Nms}^2$ ,  $\alpha$  is the angular acceleration in  $\text{rad/s}^2$ ,  $b$  is the damping coefficient in  $\text{Nms}$ , and  $\Omega$  is the angular speed of the motor in  $\text{rad/s}$ . The friction in the C-axis is assumed to be much less than the inertia times acceleration, therefore the damping term in (3.16) was neglected. This is reasonable because the C-axis air bearing is assumed to have minimal damping. Therefore, substituting equation (3.14) into equation (3.15) and neglecting the damping term yields Equation (3.17).

$$T = -J \cdot A_{\theta} \cdot \sin(\omega \cdot t) \cdot \omega^2 \cdot \frac{\pi \text{ rad}}{180 \text{ deg}} \quad (3.17)$$

Because the torque constants for the motor and the transconductance of the amplifier are known from the manufacturer data sheets, the servo command from the PMAC controller is translated into a command torque by Equation (3.18).

$$T = S \cdot Res_C \cdot K_{AMP} \cdot K_T \quad (3.18)$$

where,  $S$  is in DAC counts,  $Res_C$  is the range multiplier of 20 Volts/65536 DAC counts that converts the servo command into a voltage command,  $K_{TC}$  is the amplifier transconductance (0.5 Amps/Volt for the C-axis), and  $K_t$  is the torque constant for the C-axis motor of 6.05 Nm/Amp. The torque constant was determined from the Bayside catalogue. Combining equations (3.17) and (3.18) and substituting  $\omega = 2\pi f$  yields an expression for the inertia (3.19).

$$J = -\frac{K_{TC} \cdot K_t}{(2\pi f)^2} \cdot \frac{S \cdot Res_C \cdot 4551 \text{ enct / deg} \cdot 180 \text{ deg}}{A_\theta \cdot \sin(\omega \cdot t)} \cdot \frac{180 \text{ deg}}{\pi} \quad (3.19)$$

The  $S / A_\theta \sin(\omega t)$  term is simply the slope of the servo command position vs. actual position. The servo command is converted into voltage by the resolution of the DAC and the actual position is converted from encoder counts to degrees by the 4551 multiplier. This slope was determined experimentally. The experimental data was collected using Delta Tau's tuning software to perform a sine test by the following procedure:

1. Selecting a certain command frequency
2. Choosing the number of oscillations to get enough data that eliminates the transient effects
3. Record the servo command and actual position (This is a linear relationship if the axis is able to follow the command position)
4. Plot the servo command vs. actual position
5. The slope of this line is the variable  $S / A_\theta \sin(\omega t)$  in Equation 20.

6. Use this slope with the DAC resolution and encoder count scale factors to determine the rotational inertia of the stage.

The following is the result of a 0.4 Hz test and a 0.7 Hz test. Higher test frequencies became unstable. The technicians attribute this to a bug in the program. The slope changes for the different values but the frequency proportionally changes as well. Of the frequencies tested, the higher frequencies yielded steeper slopes and provide better resolution. The 0.7 Hz frequency is assumed to be more accurate than the 0.4 Hz frequency although for this reason.

*0.7 Hz Experimental Data:*

If the variables are defined as follows,

$$Res_C = 20V/65536 \text{ DAC counts}$$

$$K_{AMP} = 0.5 \text{ Amps/Volt}$$

$$K_T = 6.05 \text{ Nm/A}$$

$$S / A_\theta \sin(\omega t) = -0.26 \text{ DAC counts/ecnts}$$

$$f = 0.7 \text{ Hz}$$

$$J = -\frac{0.5A/V \cdot 6.05Nm/A}{4\pi^2 (0.7/s)^2} \cdot \frac{20V}{65536 \text{ DAC}} \cdot \frac{4551 \text{ ecnts}}{\text{deg}} \cdot \frac{-0.26 \text{ DAC}}{\text{ecnts}} \cdot \frac{180 \text{ deg}}{\pi}$$

$$= 3.23 \text{ Nm/s}^2 = 3.23 \text{ kgm}^2$$

*0.4 Hz Experimental Data:*

With the same variables as above except for a frequency of 0.4 Hz and slope of -0.0809 DAC the inertia becomes,

$J = 3.08 \text{ kgm}^2$  which differs from the above answer by only 5%.

The estimated inertia from measurement of the C-axis geometry is approximately  $3.1 \text{ kgm}^2$  which shows close agreement with the experimental results.

### C-Axis Simulated Vs. Actual Results

The C-Axis was simulated using MATLAB Simulink and the models are shown in Figure 54 through Figure 56. The actual control gains for the C-axis were used in the program as well as the actual scaling factors used in the real system. The parameters identified above are used in the simulation. A step test of 1,000 and 10,000 counts was performed using Delta Tau's tuning software. The actual and simulated results are shown in Figure 7 and Figure 8 for C-Axis step test of 1,000 and 10,000 counts respectively. This corresponds to a step input of 0.22 and 2.2 degrees respectively. The mean of the residuals of the simulated C-axis model compared to the actual data was 3.6682 counts and the standard deviation was 25.7364. The average error of the simulation was approximately 4% which showed that the simulation closely represents the actual data.

### C-Axis Conclusion

The rotational inertia for the C-axis was determined using the approach outlined above. The simulation of the C-axis closely represented the experimental results indicating that the parameters were estimated correctly. The approach

followed above can be replaced by a model identified by system identification if the proper input output data is available.

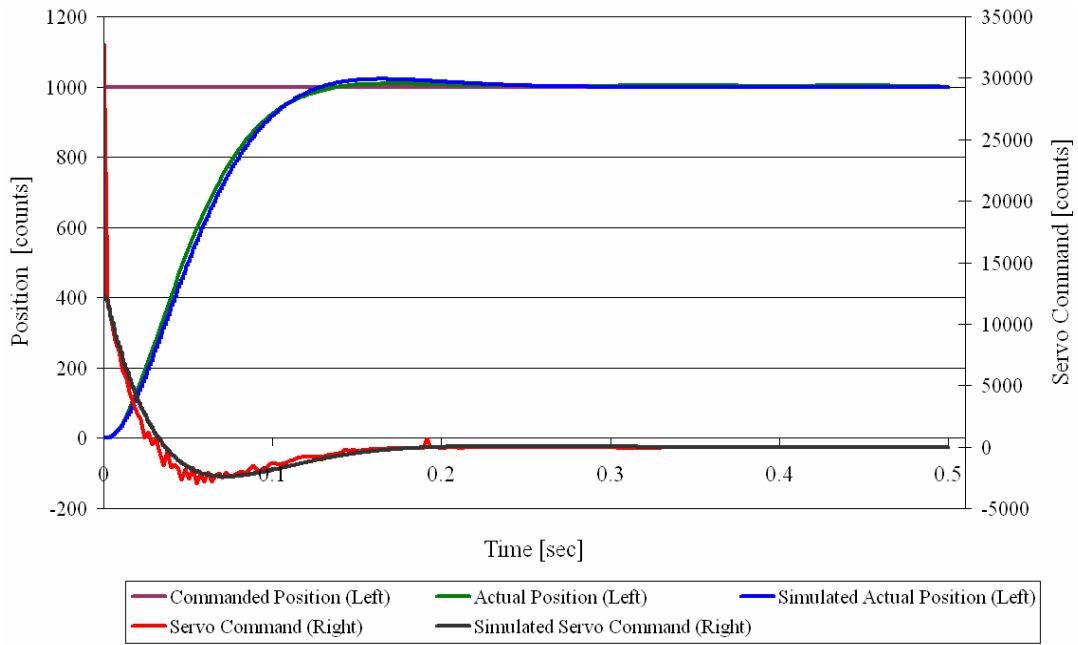


Figure 7: 1,000 count step response

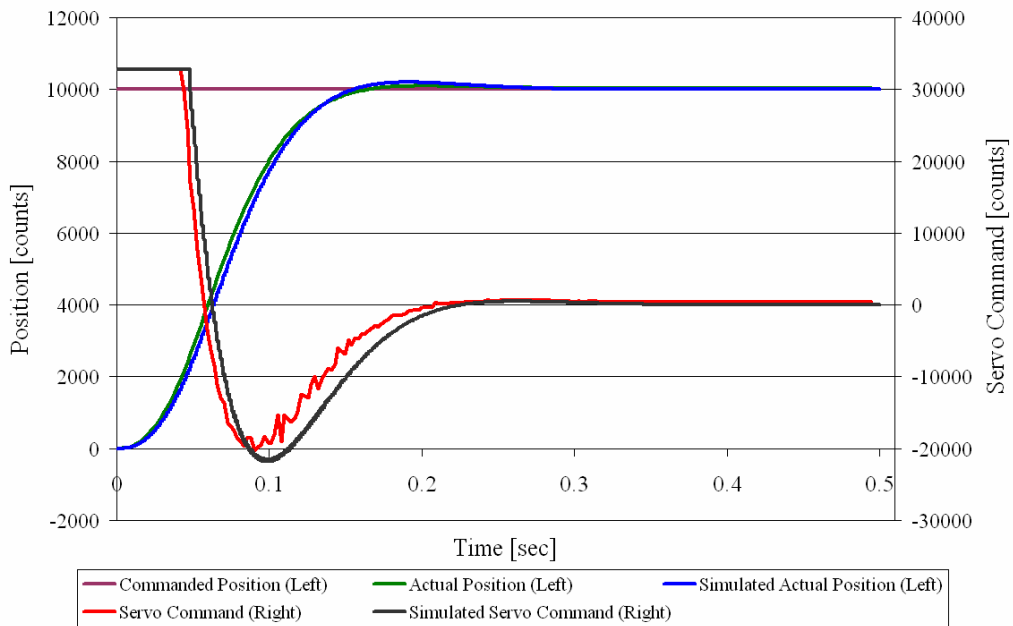


Figure 8: 10,000 count step response

## Linear Motor Parameter Determination

Simulation of the R-Axis required determination of the unknown mass and damping. This section discusses three methods by which the damping was determined and how the mass was determined with and without knowledge of this term.

### Theoretical Eddy Current Damping

Theoretical modeling of eddy current damping has been done by (Hughes, S. B. 2000) and (Sodano, H., Bae, J. et al. 2004). Damping on the R-axis is from the effect that eddy currents have on a conductor passing through a magnetic field. The R-axis magnetic field is perpendicular to the motion and induced by a rare earth magnet track. An aluminum plate is attached to the end of both linear motors on the R-axis. An electromotive force (voltage) is produced as this plate moves according to Faraday's law,

$$emf = BLv . \quad (3.20)$$

Where,  $emf$  is the electromotive force in Volts,  $B$  is the magnetic flux in the motor in Tesla, and  $L$  is the height of the plate in the field. The induced eddy currents are this voltage divided by the resistance of the plate. The resistance is given by (Caldwell, 1996) as,

$$R = \frac{L_R}{\sigma \cdot c \cdot x}. \quad (3.21)$$

Where  $R$  is the resistance in Ohms,  $L_R$  is the effective height of the plate in the magnetic field,  $\sigma$  is the conductivity of aluminum,  $c$  is the plate thickness, and  $x$  is the plate width. The effective height,  $L_R$  is less than the height of the plate in the magnetic field. This is an unknown parameter but using empirical results from (Hughes, S. B. 2000) the effective height is assumed to be 25% of the height of the plate in the magnetic field. The eddy currents are therefore a result of equations (3.20) and (3.21),

$$I = \frac{\sigma \cdot c \cdot x \cdot B \cdot L}{L_R} v. \quad (3.22)$$

The magnetic flux induced currents create a magnetic field that opposes the change in flux according to the Lenz's law. This flux then produces a force in the direction opposite to the velocity. This is described by Lorenz force law and can be used to calculate the force due to damping by equation (3.23).

$$F_{LZ} = IL \times B \quad (3.23)$$

By combining equations (3.20) thru (3.22) into equation (3.23) the Lorenz force law becomes,



$$F_{LZ} = \frac{\sigma \cdot c \cdot x \cdot B^2 \cdot L^2}{L_R} v. \quad (3.24)$$

However, according to (Cadwell, L. H. 1996), the effective magnetic flux is half the flux of the magnet and is approximated as,

$$B_{eff} = 0.5B. \quad (3.25)$$

Therefore, the Lorenz force equation becomes,

$$F_{LZ} = \frac{\sigma \cdot c \cdot x \cdot B_{eff}^2 \cdot L^2}{L_R} v. \quad (3.26)$$

The values for the variables in the equations above are summarized in Table 6. The theoretical damping is 687 Ns/m. To verify this experimental determination of the damping was also performed.

Table 6: Theoretical Eddy Current Determination Variables

Resistivity of Aluminum	2.82E-08 m <sup>3</sup> ·kg·s <sup>-3</sup> ·A <sup>-2</sup>
Conductivity of Aluminum	3.77E+07 m <sup>-3</sup> ·kg <sup>-1</sup> ·s <sup>3</sup> ·A <sup>2</sup>
Plate Thickness (c)	0.007 m
Plate Length (x)	0.2 m
Effective Magnetic Field Height (L)	0.05 m
Effective Length (LR)	0.013 m
Flux Density (B)	0.52 T
Damping (F/v)	687 N·s·m <sup>-1</sup>

## Experimental Eddy Current Damping

The linear motors on the R-Axis and Z-axis are damped by eddy current dampers. The servo command is broken up, or commutated, into three phases in order to pass current through the appropriate motor windings and produce the desired torque. Only the commutated servo command output was readily available for data collection. This problem can be overcome but will require time to code. The commutated servo command was logged using Delta Tau's tuning software. The amplitude of the command input was determined by fitting a sine wave via linear regression to the servo command. This command input was translated into a command current and subsequently commanded force by knowledge of the amplifier transconductance and torque constant for the motor. The damping is determined from this information as shown in the analysis that follows. The linear model is assumed to be in the following second order form as found in (Liaw, C. M., Shue, R. Y. et al. 2001).

$$F = ma + bv + F_L + F_C + F_F \text{signum}(v) \quad (3.27)$$

Where  $F$  is the thrust force of the motor in  $N$ ,  $m$  is the mass in  $kg$ ,  $a$  is the acceleration in  $m/s^2$ ,  $b$  is the damping coefficient in  $Ns/m$ ,  $v$  is the velocity in  $m/s$ ,  $F_L$  is the load on the motor opposing the thrust force in  $N$ ,  $F_C$  is the cogging force, and  $F_F$  is the Coulomb friction force. The cogging force is the force necessary to initially overcome the magnetic attraction between the linear motor coil and magnet track. The amplifier is powerful enough so the cogging force is negligible.

The friction force can be modeled by different methods as summarized by (Åström, K. 1998). However, the R-axis rests on air bearings and friction forces are assumed to be negligible. It is not known how much cogging force is present. However, the linear regression outlined below takes into account an offset force to account for this uncertainty. These assumptions reduce equation (3.27) to,

$$F = ma + bv, \quad (3.28)$$

Or, in terms of damping,

$$b = \frac{F - ma}{v}.$$

The damping term in the above equation was determined by constant velocity commands to the motor. The position and servo command (force) were recorded for each constant velocity command. By using this approach, the acceleration term in equation (3.28) becomes negligible and the damping is determined from the data for force and velocity. However, the force command is a commutated signal in which the amplitude is unknown. This can be determined by using a least squares regression on the commutated servo command as follows. The servo command is assumed to be in the following format,

$$S = A_{\theta} \cos(\theta - \phi) + b \quad (3.29)$$

Where  $\alpha$  is the amplitude of the servo command in volts, where  $\theta$  is the phase angle and  $\phi$  is the phase offset and  $z$  is the commutated servo command for the first phase. Using trigonometric identities equation (3.29) becomes,

$$S = A_\theta \cos(\theta) \cos(\phi) + A_\theta \sin(\theta) \sin(\phi) + b \quad (3.30)$$

Linear regression of equation (3.30) can be used to determine  $\alpha$  and  $\phi$ . The regression of  $z$  gives two linearly dependent terms  $i$  and  $j$ , as shown in equation (3.31).

$$\begin{cases} S = i \cos(\theta) + j \sin(\theta) + b \\ i = A_\theta \cos(\phi) \\ j = A_\theta \sin(\phi) \end{cases} \quad (3.31)$$

The terms  $i$  and  $j$  in the linear regression can be used to get the amplitude  $\alpha$  and phase offset  $\phi$  as shown in equations 34 and 35.

$$A_\theta = \sqrt{(A_\theta \cos(\phi))^2 + (A_\theta \sin(\phi))^2} \quad (3.32)$$

$$\phi = \arctan\left(\frac{A_\theta \sin(\phi)}{A_\theta \cos(\phi)}\right) \quad (3.33)$$

A plot of the actual servo command voltage, linear regression values, and residuals vs. distance along the R-axis is shown in Figure 9.

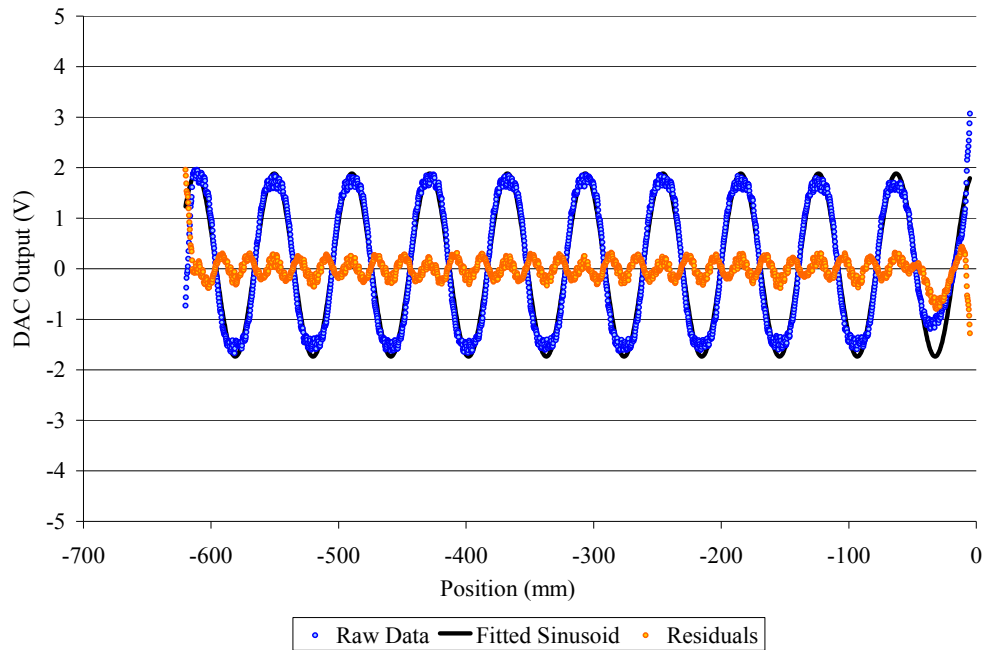


Figure 9: R-Axis DAC voltage command and residuals vs. position at 100mm/s

The linear regression of the commutated servo command signal showed close agreement with the actual servo command signal as shown in Figure 9. The above test is a constant velocity test which means that the acceleration in equation (3.28) is zero. Therefore, by repeating this test at different constant velocities the slope of the servo command vs. voltage  $F/v$  term is equal to the damping as shown in equation (3.34). This is graphically shown in Figure 10.

$$b = \frac{F}{v} \quad (3.34)$$

The slope of the line in Figure 10 is equal to the damping coefficient but the units are in Vs/mm. The damping coefficient is converted to Ns/mm by equation (3.35).

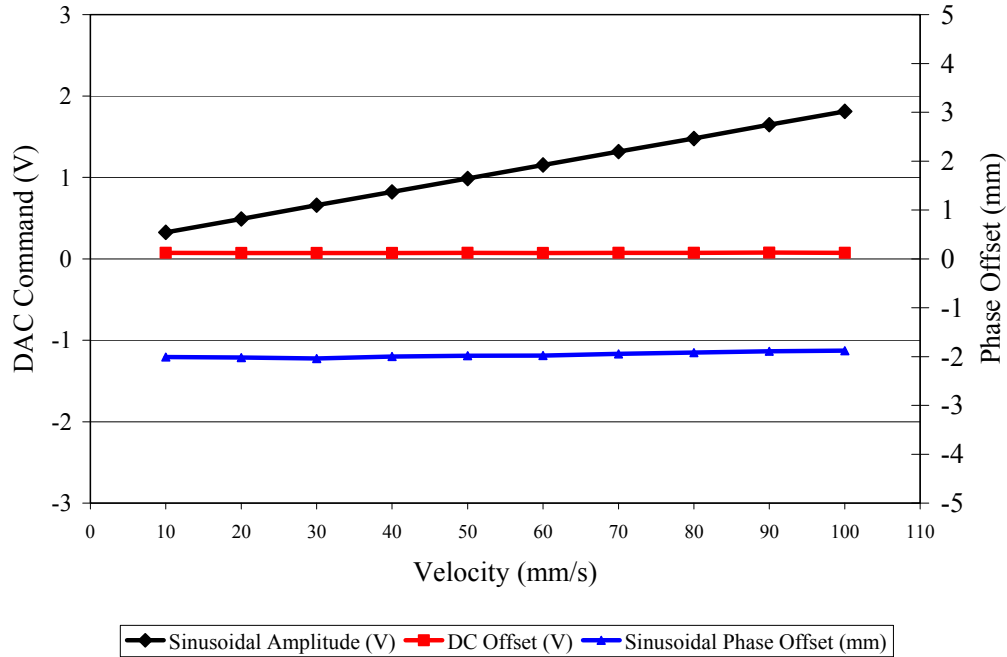


Figure 10: Velocity for different servo commands

$$b[\text{Ns/mm}] = b[\text{Vs/mm}] \cdot K_t \cdot K_{TC} \quad (3.35)$$

Where,  $K_t$  is equal to the torque constant of the motor (provided by the manufacturer) in Nm/Amp and  $K_{TC}$  is the amplifier transconductance in Amp/Volts and the result is scaled by 1m/1000mm. The result is that the damping coefficient of the R-axis is equal to 670 Ns/m.

An alternative method to determine the damping was also performed in which an approximately constant force was applied with a hand held force gage and the velocity measured for different forces. The ratio of the force over the velocity was the damping due to the eddy currents. This is shown in Figure 11 and

both methods are shown in Figure 12. The constant velocity test yielded 673 Ns/m and the constant force test yielded 709 Ns/m for  $b$ . The increased force is likely due to the fact that the velocity is not constant.

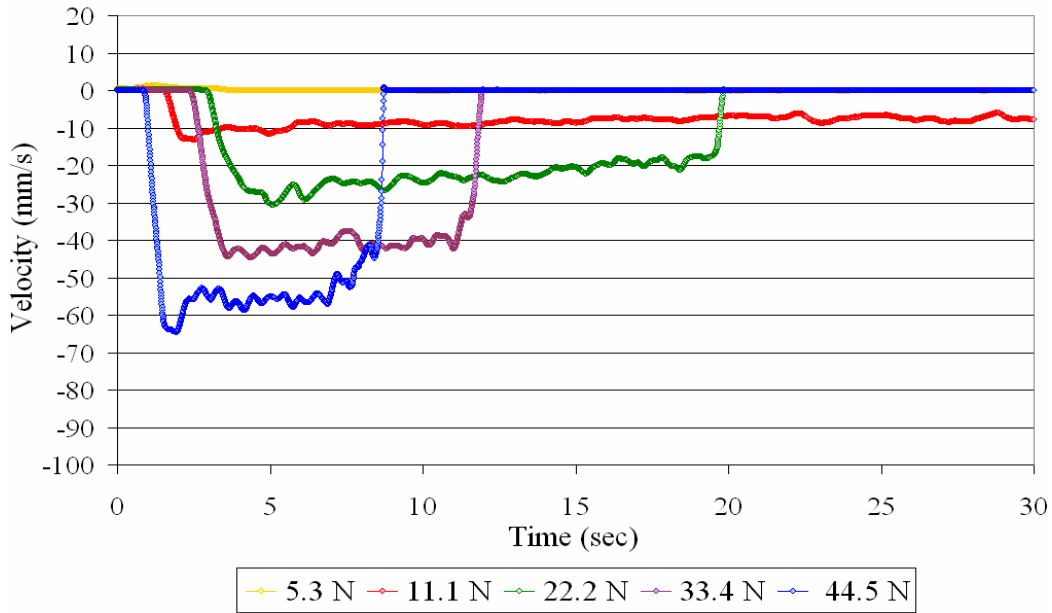


Figure 11: Alternate constant force eddy current damping determination

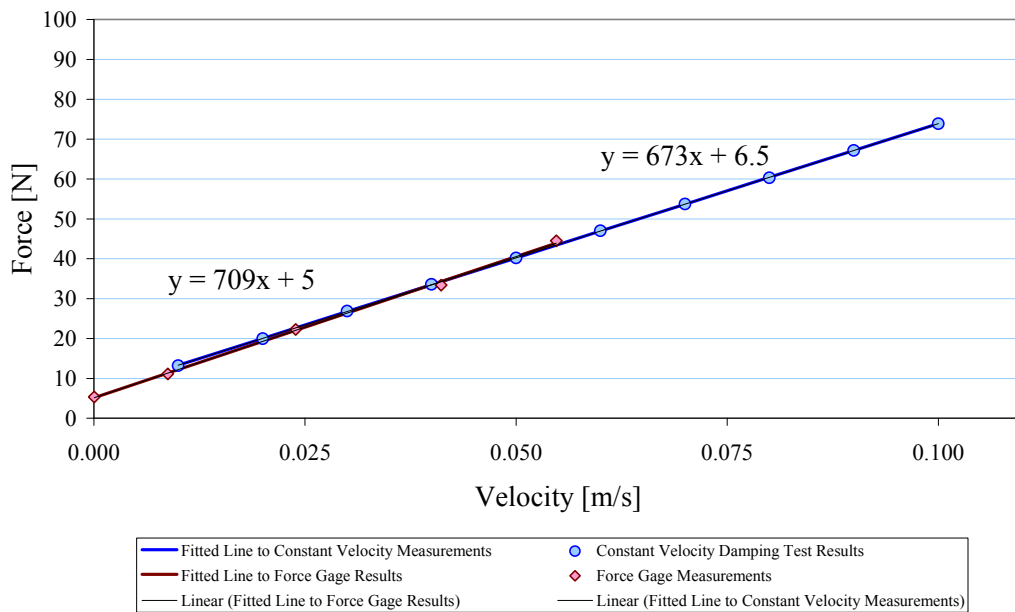


Figure 12: Both methods of eddy current damping

### R-Axis Mass Determination

The damping values above showed close agreement both theoretically and from two empirical tests. To completely model the R-Axis the mass must also be determined. Recall from equation (3.27) that the force on the linear motor is,

$$F = ma + bv + F_L + F_C + F_F \text{signum}(v).$$

The force,  $F$ , can be determined easily from the servo command output from the PMAC. To get the servo command in terms of force it must be multiplied by the DAC resolution to get volts, multiplied by the amplifier transconductance to get amps, and multiplied by the torque constant for the motor to get Newtons. A linear regression from knowledge of this force input and recorded data was performed according to equation (3.36).

$$F = x_1 a + x_2 v + x_3 \text{signum}(v) + F_{offset} \quad (3.36)$$

Where,  $F_{offset}$  is the sum of all external forces acting on the motor and includes cogging force and any external offset force,  $x_1$  corresponds to the fitted mass,  $x_2$  corresponds to the fitted damping,  $x_3$  corresponds to the fitted hysteresis. This hysteresis is due to the friction effect of stiction which is assumed to have a constant magnitude, changing only in sign with the direction of the velocity (Ellis, G. 2004). More accurate modeling of stiction has been presented by Stribeck (Åström, K. 1998). The actual position was obtained from the PMAC and the



numerical derivative was taken to get the velocity and likewise the acceleration. The values for velocity, acceleration,  $\text{signum}(v)$ , and  $F_{\text{offset}}$  were used in the linear regression.

Data for the linear regression was collected by a sinusoidal input to the R-axis linear motor at different frequencies and amplitudes. A linear regression was performed at each frequency to determine the coefficients  $x_1$ ,  $x_2$ , and  $x_3$ , as well as the  $F_{\text{offset}}$  term to determine the corresponding mass, damping, hysteresis, and force offset in the model. A summary of the regression coefficients is given in Table 7 and plot of each is shown in Figure 20 through Figure 23. The results of the low and high frequency linear regression are shown in Figure 14 through Figure 18. The hysteresis due to the small amount of stiction can be clearly seen in Figure 19. The solid lines in Figure 20 through Figure 23 represent the regression values obtained when all data sets from all frequencies were used and the damping term was *fixed* at 670 Ns/m as determined above using the constant velocity method. The linear regression values obtained were used in a simulation that is shown in Figure 25.

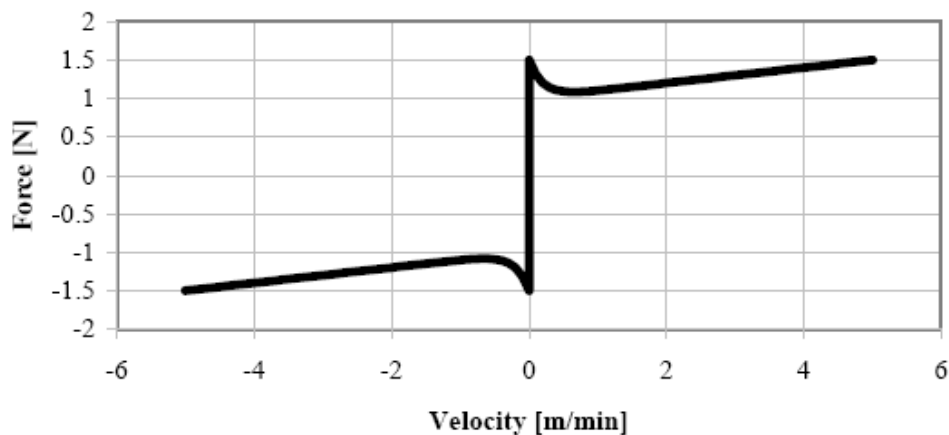


Figure 13: Stribeck Model of Force vs. Velocity

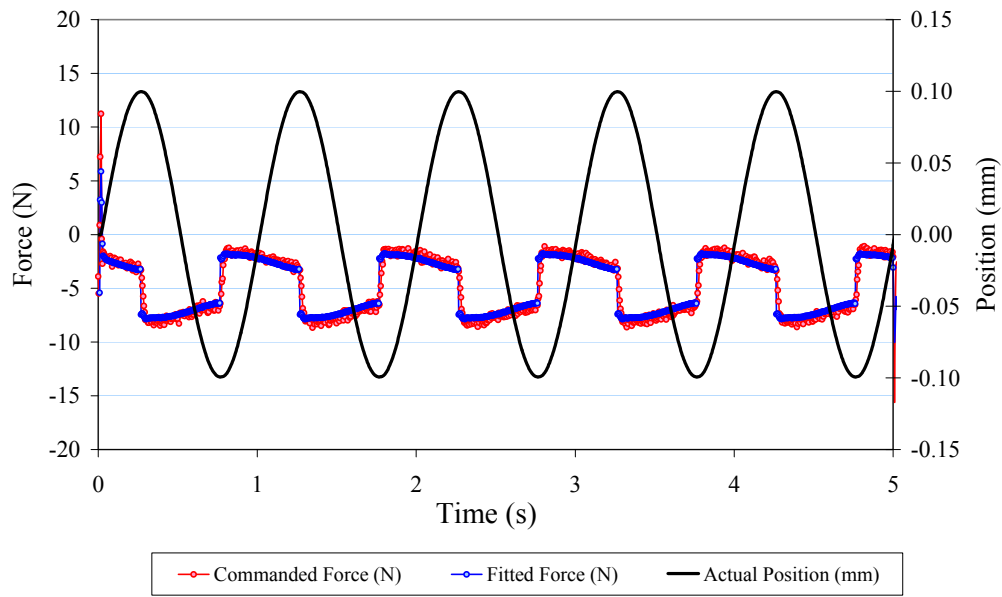


Figure 14: Force and position vs. time for 0.1mm Sine input at 1 Hz for 5 seconds

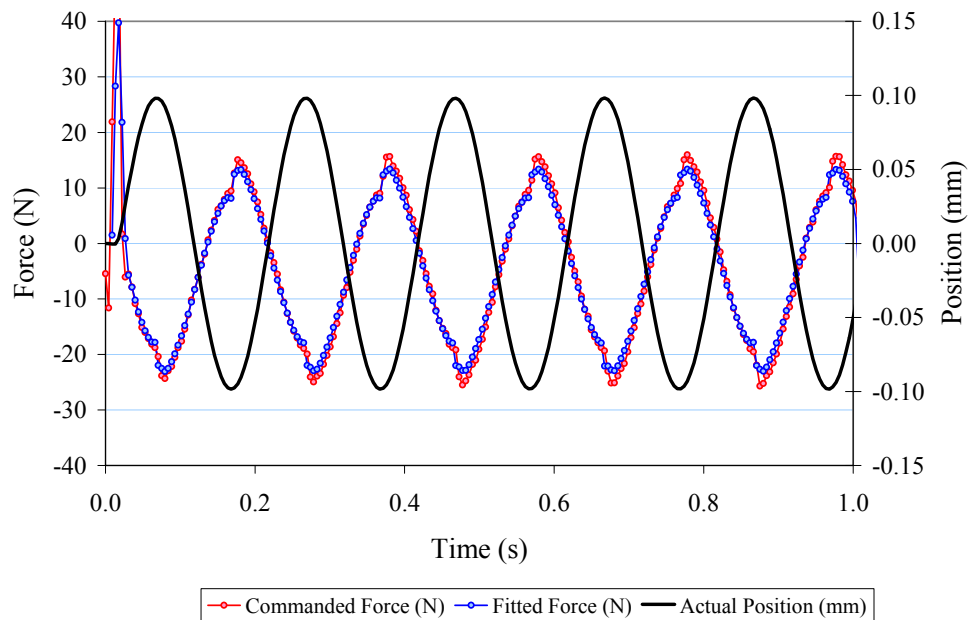


Figure 15: Force and position vs. time for 0.1mm Sine input at 5 Hz for 1 second

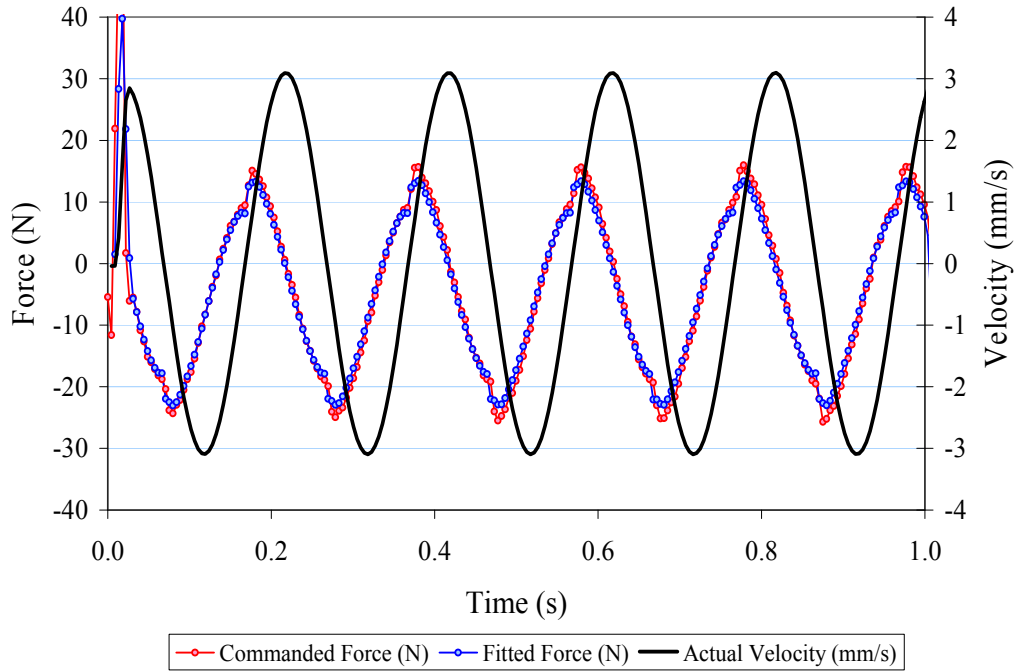


Figure 16: Force and velocity vs. time for 0.1mm Sine input at 5 Hz for 1 second

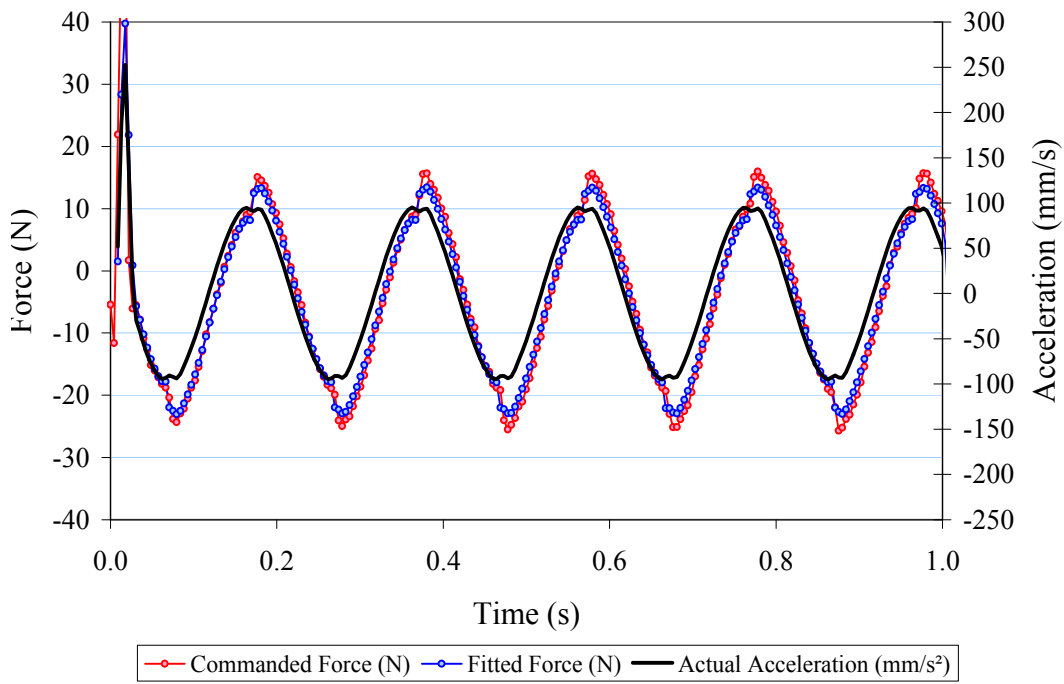


Figure 17: Force and acceleration vs. time for 0.1mm Sine input at 5 Hz for 1 second

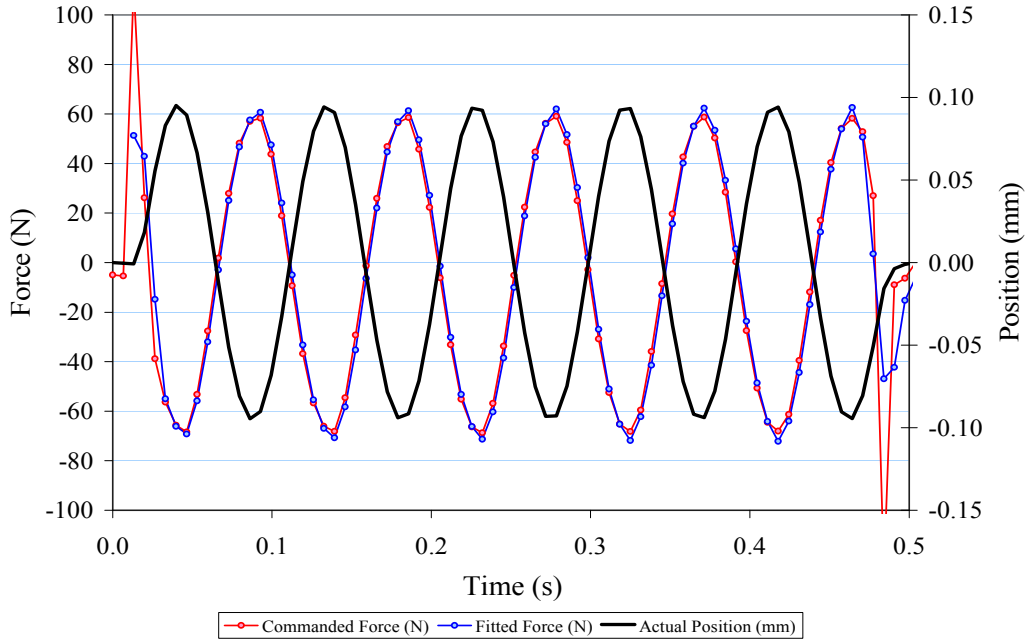


Figure 18: Force and position vs. time for 0.1mm Sine input at 10 Hz for 0.5 seconds

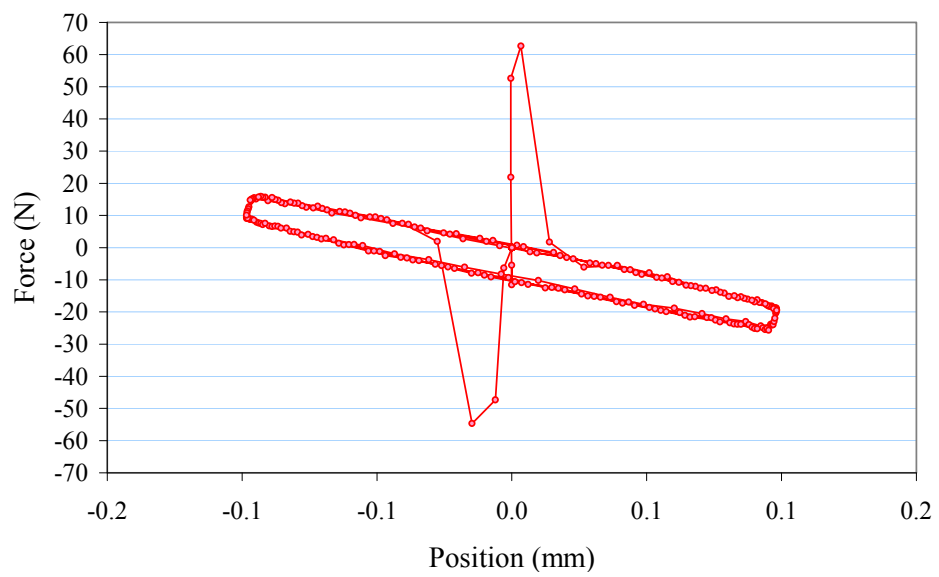


Figure 19: Force vs. position for 0.1mm Sine Input at 5Hz for 1 second

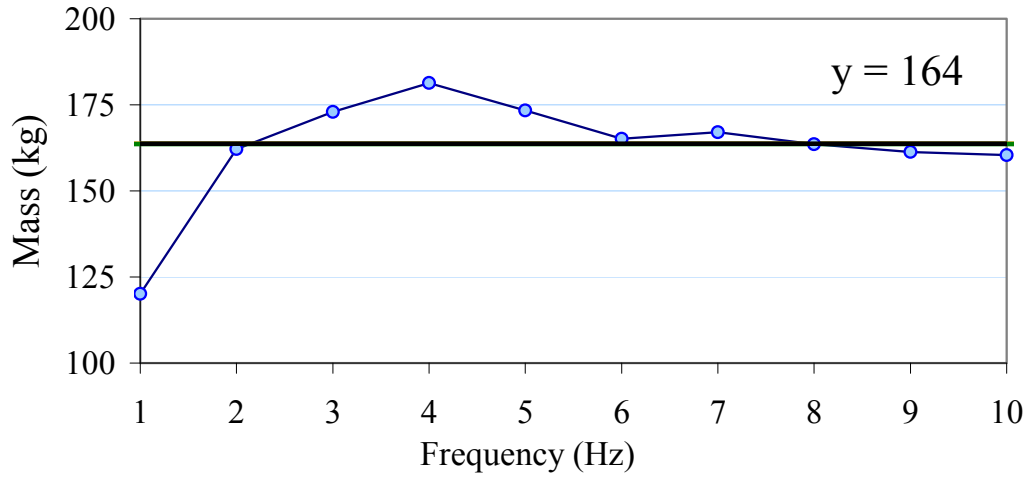


Figure 20: Linear regression values for mass at different frequencies

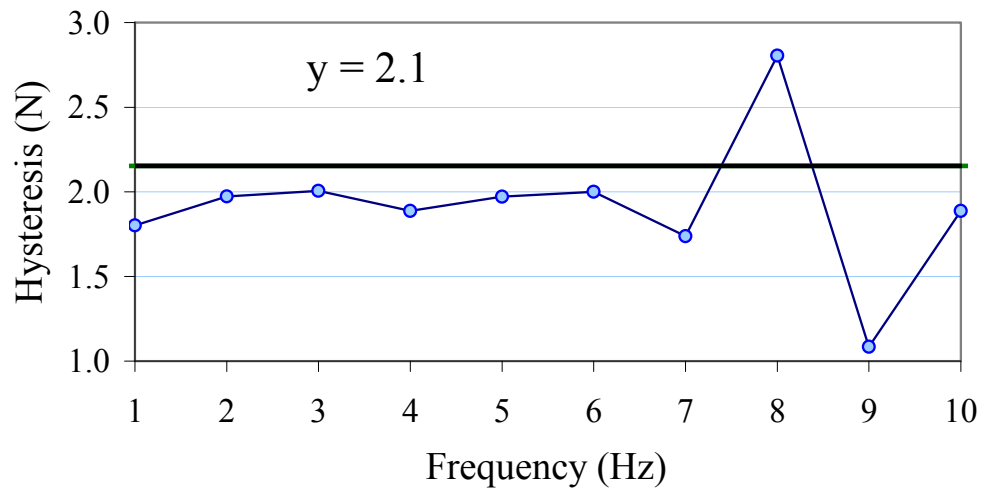


Figure 21: Linear regression values for hysteresis at different frequencies

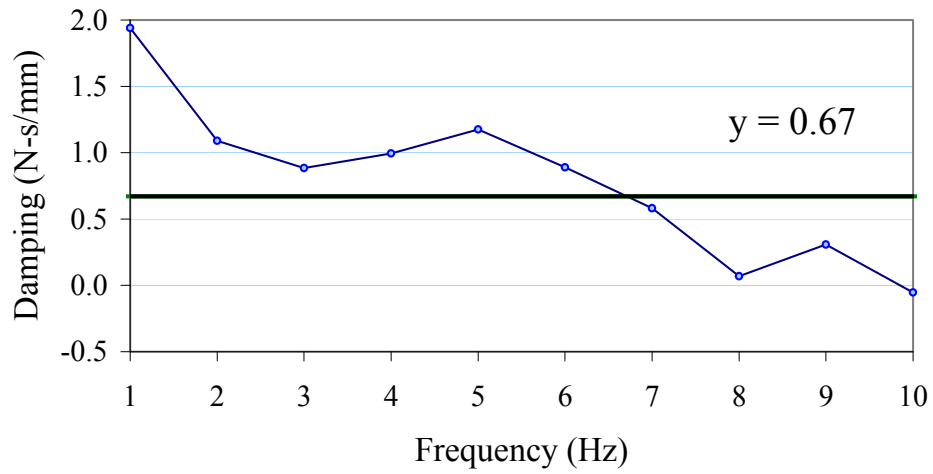


Figure 22: Linear regression values for Damping at different frequencies

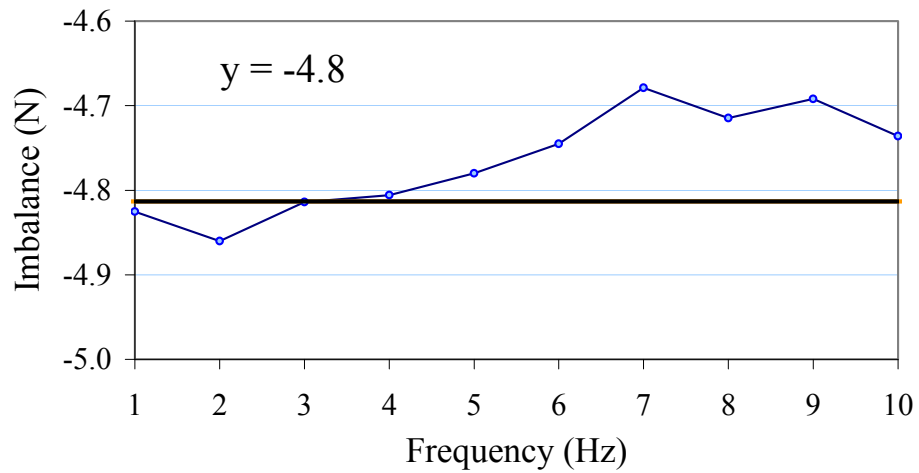


Figure 23: Linear regression values for force offset at different frequencies

Table 7: Regression Values for Different Frequencies

Frequency (Hz)	Mass (kg)	Hysteresis (N)	Damping N/(mm/s)	Imbalance (N)
1	120.121	1.801	1.940	-4.825
2	162.148	1.972	1.090	-4.860
3	172.937	2.006	0.885	-4.814
4	181.319	1.887	0.995	-4.806
5	173.315	1.970	1.175	-4.780
6	165.154	2.001	0.891	-4.745
7	167.014	1.738	0.581	-4.679
8	163.560	2.804	0.069	-4.715
9	161.256	1.083	0.307	-4.692
10	160.369	1.886	-0.054	-4.736

By including all the data at all frequencies and fixing the damping term at 670 Ns/m (as identified before) in the linear regression, the identified mass was 160 kg, the force imbalance -4.8 N, and the hysteresis/backlash accounted for 2.1 N of force. The force imbalance of 4.8 N would simply be the force necessary to maintain position of the R-Axis is there was an offset angle of 0.18°. The negative value of damping in Table 7 corresponding to the 10 Hz frequency indicates that damping is likely being done by the controller to maintain stability.

#### R-Axis Simulated Vs. Actual Results

A linear motor model was created for simulation using the identified values above for mass, damping, and force offset. The simulation model (ignoring hysteresis) for the linear motor is,

$$F = ma + bv + F_{offset} \quad (3.37)$$

In terms of the Laplace domain,

$$F(s) - F_{offset}(s) = X(s)(ms^2 + bs) \quad (3.38)$$

In terms of the linear motors' transfer function of position to a force input,

$$G(s) = \frac{X(s)}{F(s) - F_{offset}(s)} = \frac{1}{(ms^2 + bs)} \quad (3.39)$$

The block diagram of the linear motor model is shown in Figure 24.

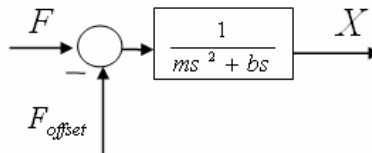


Figure 24: Block diagram for linear motor

The above model was coded in MATLAB using the parameters identified above. The actual and simulated response is shown below. There is about a 6% error in the simulated and actual position that is likely due to the fact that the hysteresis and amplifier dynamics are not included.



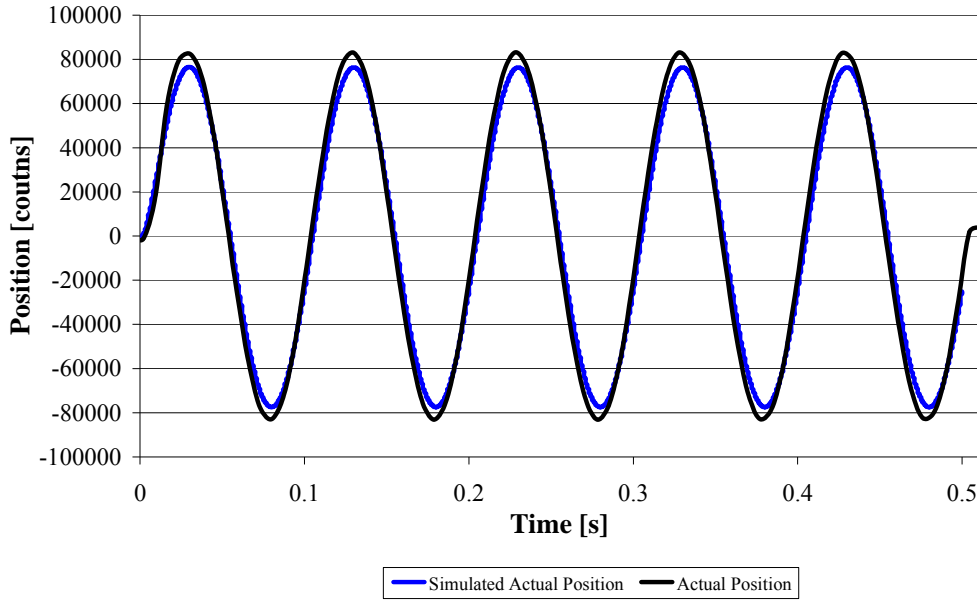


Figure 25: Simulated/Actual data R-Axis 0.1 mm sine wave at 10 Hz for 0.5 sec

Alternatively, the R-Axis may be modeled as a DC motor in the same way as the C-axis. For design and comparison purposes the resulting model is presented. The modeling of the R-Axis as a DC motor is the same as equations (3.6) through (3.10) where the torque is replaced by the applied force and the rotational inertia is replaced by the mass. This leads to a model for the R-axis given in equation (3.40) where the speed is integrated to get the position.

$$\frac{\Theta(s)}{E_o(s)} = \left( \frac{K_t}{L_a m s^2 + (L_a b + R_a m)s + (R_a b + K_t K_b)} \right) \frac{1}{s} \quad (3.40)$$

The amplifier dynamics can also be included in the same way as in equation (3.12). A comparison of the simulated and actual values for the same input is shown in Figure 26.

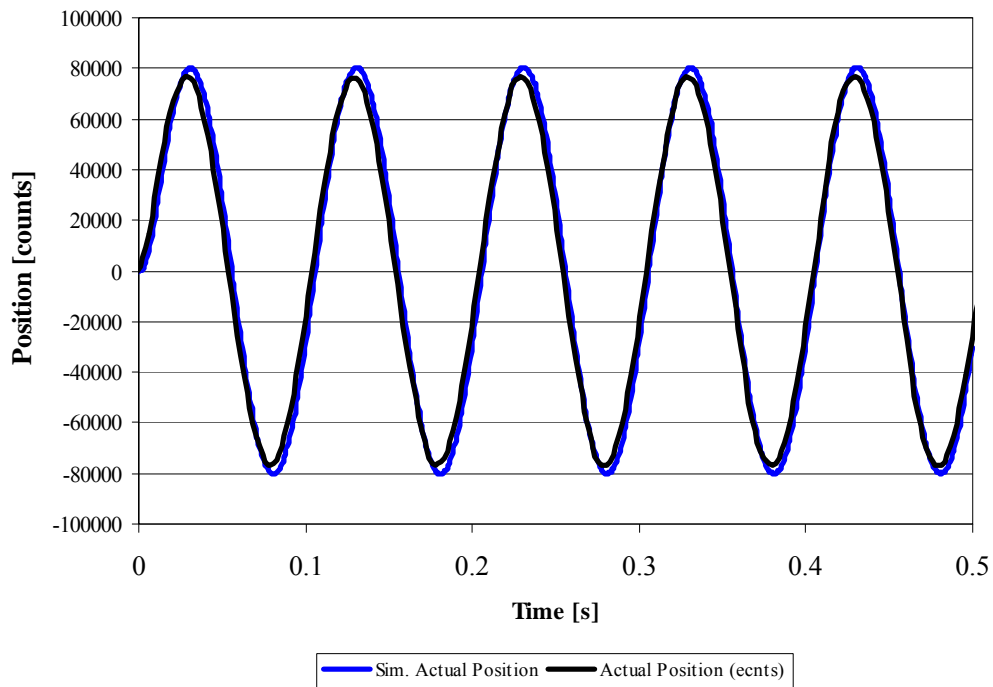


Figure 26: Simulated/Actual data R-Axis 0.1 mm sine wave at 10 Hz for 0.5 sec

Comparison of Figure 25 and Figure 26 shows that the simple mass-damper model and the more complex DC motor model are very close. The simple mass-damper model underestimates the necessary force to move the axis and the DC motor model overestimates the necessary force to move the axis. This uncertainty is present because neither model accurately captures the true parameter values.

### Parameter Summary

From the experiments discussed above, manufacturer's catalogue information, and UMM drawings, the parameters for the different motor axes were collected. It is evident that the parameter estimation by modeling and experimental identification is a time-consuming process. This process can be

replaced with system identification as discussed in the next section. A summary of the axis parameters is shown in Table 8.

Table 8: Summary of Motor Parameters for the UMM

UMM	Stage Mass [kg]	Damping [Ns/m]	Inertia [kgm <sup>2</sup> ]	Kt [N/A]	Ke [Vs/m or rad]	Resistance [Ohms]	Inductance [mH]
R-Axis	160	670	-	34.1	39.4 [Vs/m]	5.4	3.8
C-Axis	-	-	3.23	4.82	4.82 [Vs/rad]	6.3	63.72



## CHAPTER IV

### SYSTEM IDENTIFICATION

This chapter presents the results of system identification from input output data of the UMM described above. The models built from this data are compared to those developed by physical modeling as in the previous chapter. The level of effort required to model the system and determine the parameters by experimentation is significant. However, if proper input-output data is obtained then an accurate model can be determined by this information alone as demonstrated in this chapter.

The physical modeling and parameter identification approach detailed in Chapter III resulted in models and parameters for both the R- and C-axes. The model from the C-Axis DC motor is from equations (3.11) and (3.12) where the input is the input voltage to the motor and the output is the position read by the encoder.

$$\frac{\Theta(s)}{E_0(s)} = \left( \frac{K_t}{L_a J_L s^2 + (L_a b + R_a J_L) s + (R_a b + K_t K_b)} \right) \cdot \frac{1}{s}$$

Similar modeling of the R-axis resulted in equation (3.40),

$$\frac{\Theta(s)}{E_o(s)} = \left( \frac{K_t}{L_a m s^2 + (L_a b + R_a m)s + (R_a b + K_t K_b)} \right) \cdot \frac{1}{s}$$

The orders of the parametric model used for system identification can be determined by converting the continuous time models into discrete time models. Putting the parameter values into the continuous time models for the C-axis and R-axis and converting the models to discrete time models by a zero order hold yields equations (4.1) and (4.2) for the C- and R-axes respectively.

$$\frac{\Theta(z)_{C-axis}}{E_o(z)_{C-axis}} = \frac{2.646 \times 10^{-10} z^2 + 1.047 \times 10^{-9} z + 2.589 \times 10^{-10}}{z^3 - 2.957 z^2 + 2.915 z - 0.9573} \quad (4.1)$$

$$\frac{\Theta(z)_{R-axis}}{E_o(z)_{R-axis}} = \frac{1.37 \times 10^{-9} z^2 + 4.719 \times 10^{-9} z + 1.001 \times 10^{-9}}{z^3 - 2.531 z^2 + 2.064 z - 0.5332} \quad (4.2)$$

The order of the numerator and denominator in equations (4.1) and (4.2) is used to select the order of the numerator and denominator for the parametric model selected to represent the system. If the system, disturbance, and noise share the same dynamics, then an ARX model can describe the system. The ARX model for the C-axis and R-axis derived by this assumption and selection of the orders by the *a priori* knowledge of the system order yields an ARX model in the form,

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) = b_1 u(t-1) + b_2 u(t-2) + e(n) \quad (4.3)$$

In transfer function form the ARX model is,

$$\frac{y(z)}{u(z)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} + \frac{e(n)}{a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}. \quad (4.4)$$

Unfortunately, proper input/output data for the motor was unavailable and a simulated controller and motor were used to simulate the plant data. The simulation models are shown in the appendix.

### **C-axis Identification of ARX models**

In addition to the order of the system, the input to output delay is also needed to select the proper parametric model. The input-output delay of the C-Axis was determined by a step input command to the motor of 2.2 degrees. The step response of the motor is shown in Figure 27. The time it takes for the output to change from a given input determines the input-output delay. A close up of the initial response to a step input of 0.22 degrees to the C-axis reveals a time delay of 1.25ms. The sample time is 440 $\mu$ s so the 1.3 ms delay in output corresponds to a delay of 2.95 cycles. Therefore, the ARX model selected has a third order denominator, second order numerator, and a delay of two or three samples. In comparison, a step test on the actual system showed a delay of 2.25 samples indicating that the actual delay in the system is between two and three samples.

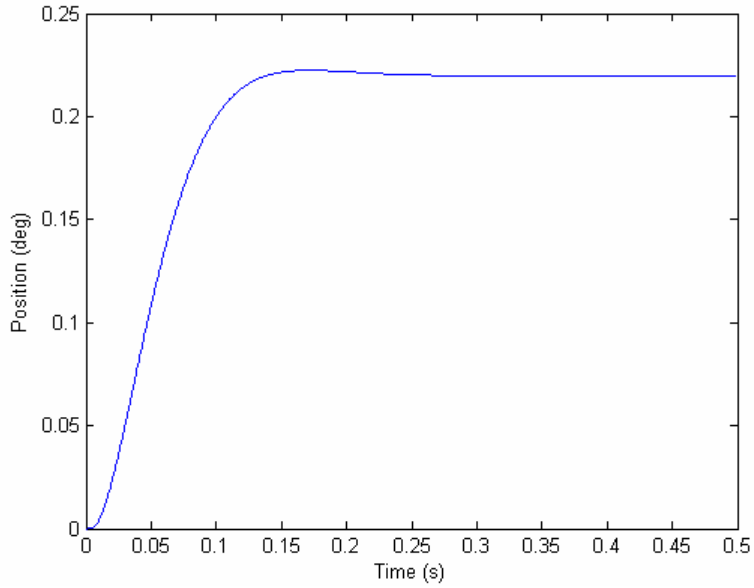


Figure 27: 0.22 degree step input for C-Axis

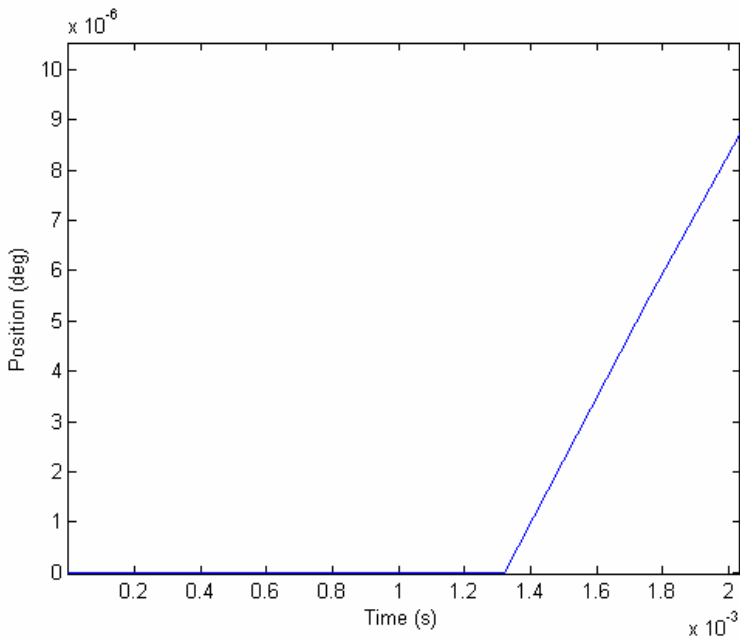


Figure 28: Close up of step input showing time delay of approximately 1.3 ms.

Band-limited white noise was input into the simulated controller to excite the frequencies of interest. The band is limited by the one half of the sampling period. The choice of input is not unique and any a variety of inputs may excite



the same system adequately. The power of the band-limited white noise was selected as 1000 for the C-axis in order to get measurable values from the encoder. The output of the white noise was updated every 0.88s which is slightly longer than the time constant of the C-axis. This allows the higher frequency content to be captured. The data was recorded for 100s and sampled at a rate of 2.27 kHz (once every 440 $\mu$ s) which is the default of the PMAC. The voltage input to the simulated C-Axis motor from the PMAC and its output position in radians were recorded. The recorded input to the motor and its output are shown in

Figure 29. A power spectrum of the input is shown in Figure 30. This is the power spectrum of the white noise input into the PMAC controller. The power spectrum from the PMAC output into the motor is shown in Figure 31. This demonstrates that the necessary frequency content is present in the exciting signal for identification purposes.

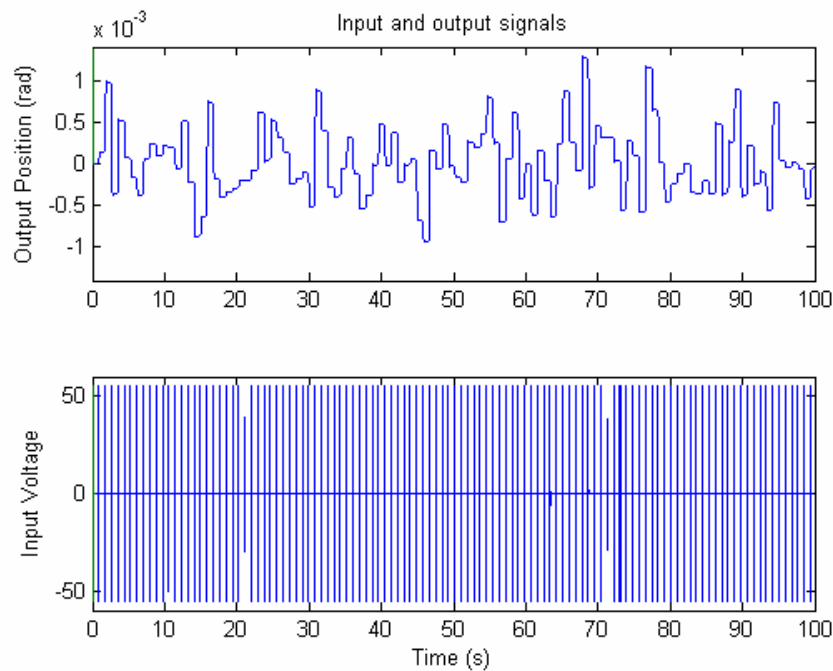


Figure 29: Identification data for C-axis

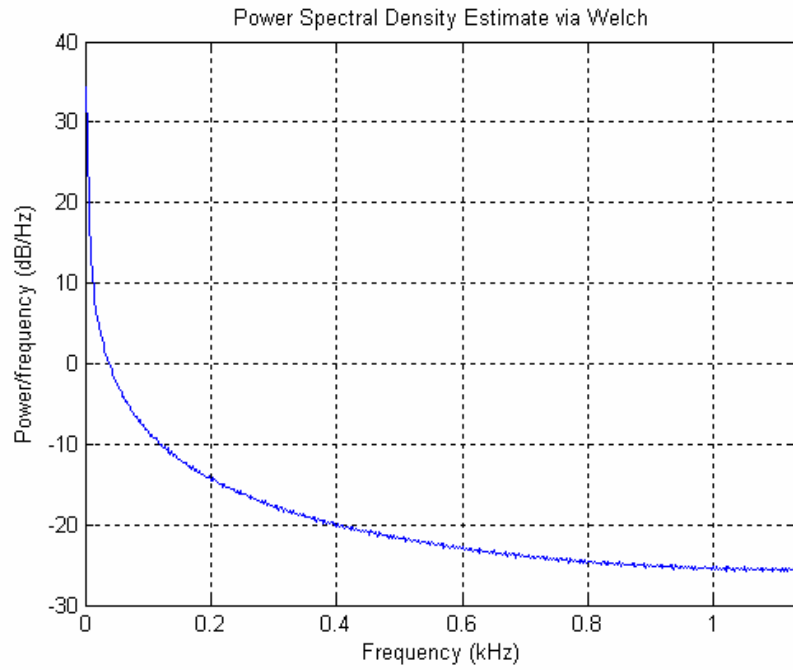


Figure 30: Power spectrum of white noise input to PMAC

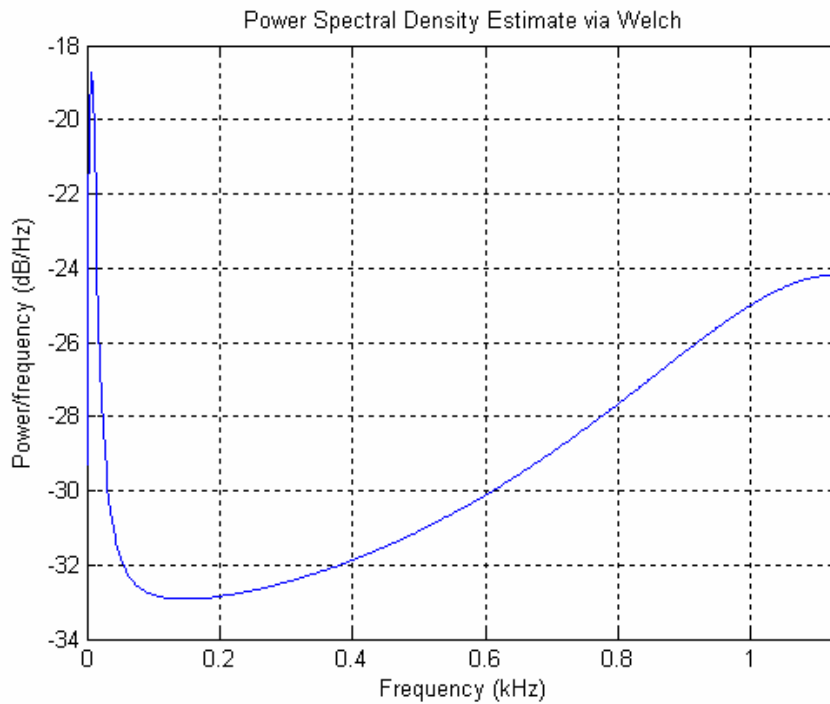


Figure 31: PSD of motor input from PMAC

The model identified by selecting an ARX structure with the orders and a delay of both two and three as determined above is shown in equations (4.5) and (4.6).

$$ARX_{322} = e^{(-0.00044s)} \frac{-6.742 \times 10^{-7} s^2 - 3.42 \times 10^{-5} s + 20.9}{s^3 + 101.4s^2 + 128.1s + 0.0002721} \quad (4.5)$$

$$ARX_{323} = e^{(-0.00088s)} \frac{1.334 \times 10^{-6} s^2 + 0.00916 s + 20.9}{s^3 + 101.4 s^2 + 128.1 s - 0.00447} \quad (4.6)$$

These equations represent the deterministic part of the ARX model which is necessary for model-based control. A Bode plot of the above two equations and the model of the C-axis used to generate the data is shown in Figure 32. The ARX322 corresponds to a third order numerator, second order denominator, and delay of two. The ARX323 signifies the same but with a delay of three. The Bode plot shows that the model of the C-axis is correctly identified by both ARX models which both assume that the noise enters early in the system and shares the systems dynamics.

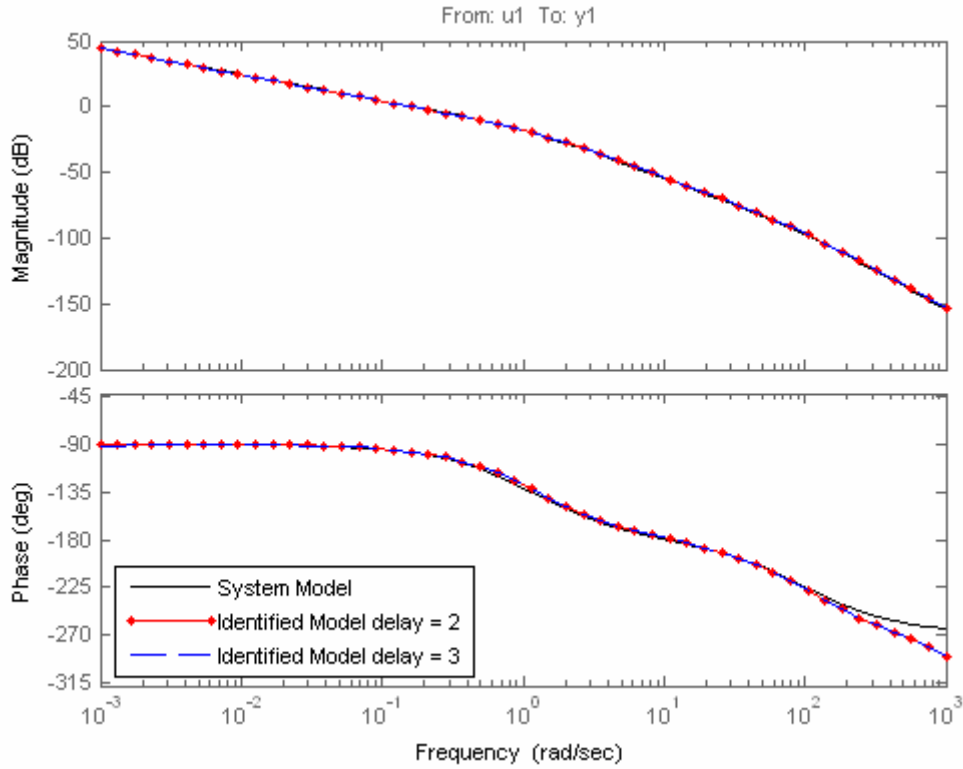


Figure 32: Bode plot of system model and identified models

Equations (4.5) and (4.6) indicate that both choices of delay identify the system dynamics, or poles, correctly. The roots from the identified ARX models above and the continuous time transfer function of the C-axis from Equation (3.12) are presented in Table 9.

Table 9: Roots of C-axis model and identified ARX models

C-axis roots	[-98.09, -1.16, 0]	% Error	[-, -, -]
Equation (4.5) roots	[-99.72, -1.28, 0]	% Error	[1.6%, 10.3%, 0%]
Equation (4.6) roots	[-99.72, -1.28, 0]	% Error	[1.6%, 10.3%, 0%]

As seen in the table, the poles of the system are identified with less than 11% error. Inspection of the time response of both ARX models compared to the simulated and actual data is shown in Figure 33. The similar responses of the identified models demonstrate that identification has been performed with high quality. Another measure of the quality of the models is inspection of the residuals left by the identification. Residual analysis is done by correlation of the residual with itself and the input.

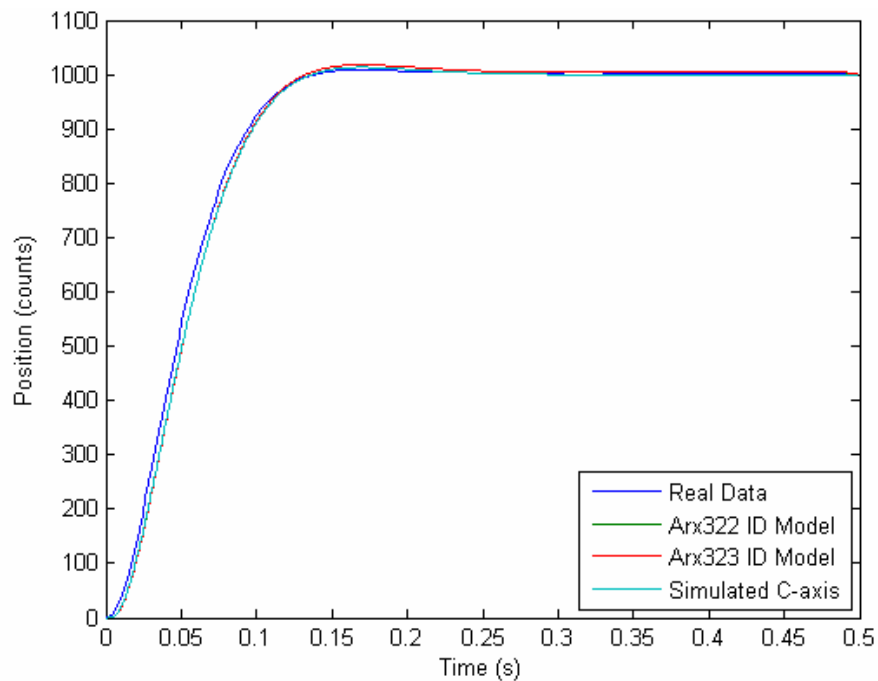


Figure 33: Comparison of models

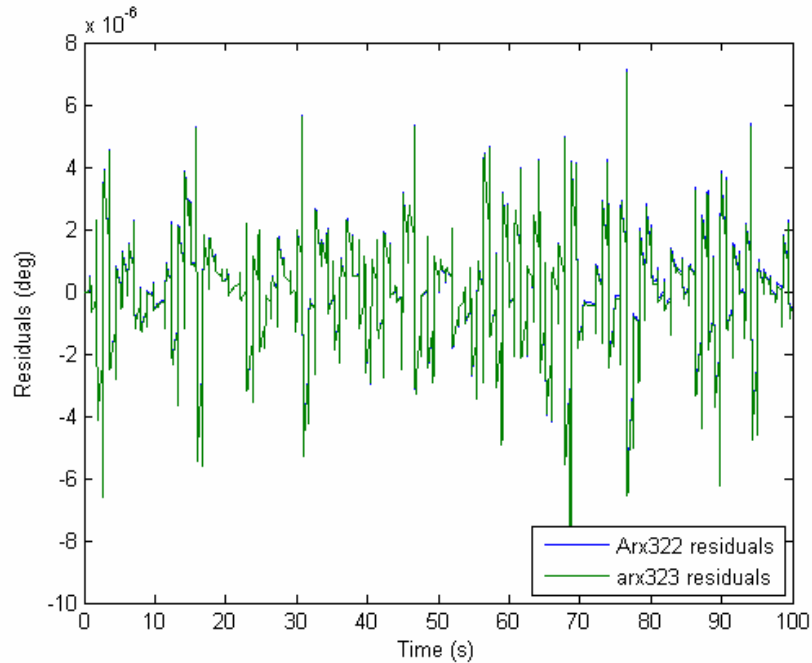


Figure 34: Residuals from identified ARX models

However, residual analysis by correlation will not accurately demonstrate the model quality for data identified under closed loop. Closed loop data will necessarily have a correlated error and input meaning that the correlation analysis cannot be used to measure the model quality. The residuals themselves are informative and are shown in Figure 34. This figure shows that the residuals are three orders of magnitude less than the signal. The max value of the residual error for the ARX322 model is  $7.1423 \times 10^{-6}$  and for the ARX323 it is  $7.0214 \times 10^{-6}$ . The average residual for the ARX322 model is  $7.2321 \times 10^{-7}$  and for the ARX323 model it is  $7.1782 \times 10^{-7}$ . Therefore, the average residual is four orders of magnitude smaller than the output which indicates that the model is very accurate.

### **Effect of Noise and Disturbance for C-axis Identification**

The data used above to calculate the ARX models was simulated without noise or disturbance. In all real systems noise and disturbances exist and how they enter the system will determine which model structure is used. This section describes how the identified model structure changes when noise enters early in the model as an input and late in the model as additive output noise. If the noise enters early in the model, then the equation error models such as the ARX model can still be used to describe the systems dynamics.

A continuous normally disturbed random number with zero mean and variance of one was added to the voltage input in the motor to evaluate noise that enters the system early. This input and corresponding output are shown in Figure 35. The identified model using an ARX322 model structure with input noise is shown in Equation (4.7). The roots of the denominator are [-99.72, -1.28, 0] which are exactly the same as the models identified without the noise input. This demonstrates how the noise is effectively fit into the stochastic part of the ARX structure while the deterministic part remains effectively unchanged.

$$ARX_{322} = e^{(-0.00044 s)} \frac{-6.742 \times 10^{-7} s^2 - 3.42 \times 10^{-5} s + 20.9}{s^3 + 101.4 s^2 + 128.1 s + 0.005657} \quad (4.7)$$

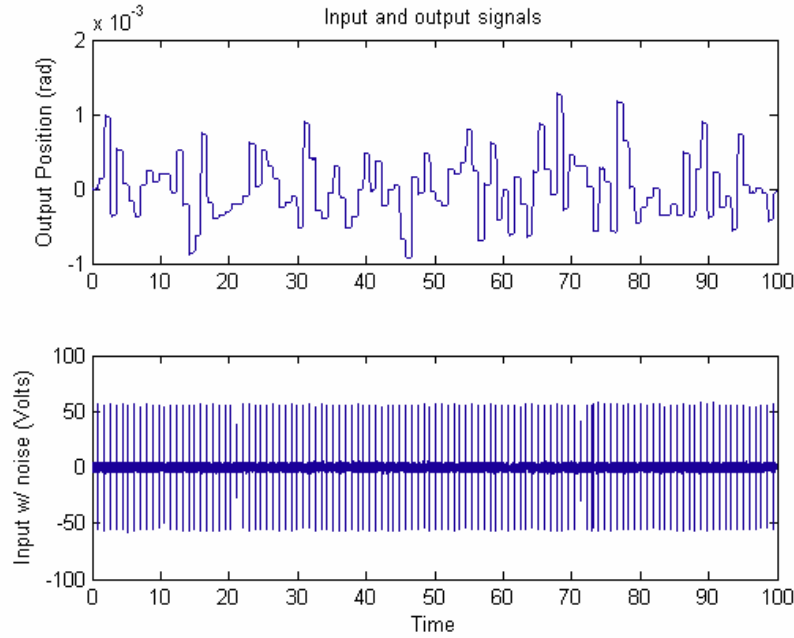


Figure 35: Identification data with noisy input into C-axis motor

When the noise is added to the output an output error model such as the OE or BJ of Table 1 must be used. To show this, data was created with a normally distributed random number added to output every 0.00044 seconds with variance of  $10^{-12}$  for 10 seconds. A plot of the input to the motor and the output with the added noise is shown in Figure 36. A Box-Jenkins model was selected to represent this system with same structure as the ARX models above for the deterministic part. A comparison of the performance of the ARX models with the same structure as above and the Box-Jenkins model is shown in a Bode plot of the corresponding models is Figure 38. Figure 37 shows a view of the added noise to the output. The deterministic part of the Box-Jenkins model is shown in Equation (4.8). Although the Bode plot shows close agreement of the Box-Jenkins model is



much more complex than necessary and the ARX model above should be used for control design.

$$BJ = \frac{-2.217 \times 10^{-6} s^5 - 0.007295 s^4 + 20.17 s^3 + 1610 s^2 + 2.322 \times 10^4 s + 745.2}{s^6 + 178.7 s^5 + 9102 s^4 + 1.227 \times 10^5 s^3 + 1.461 \times 10^5 s^2 + 4551 s - 0.6993} \quad (4.8)$$

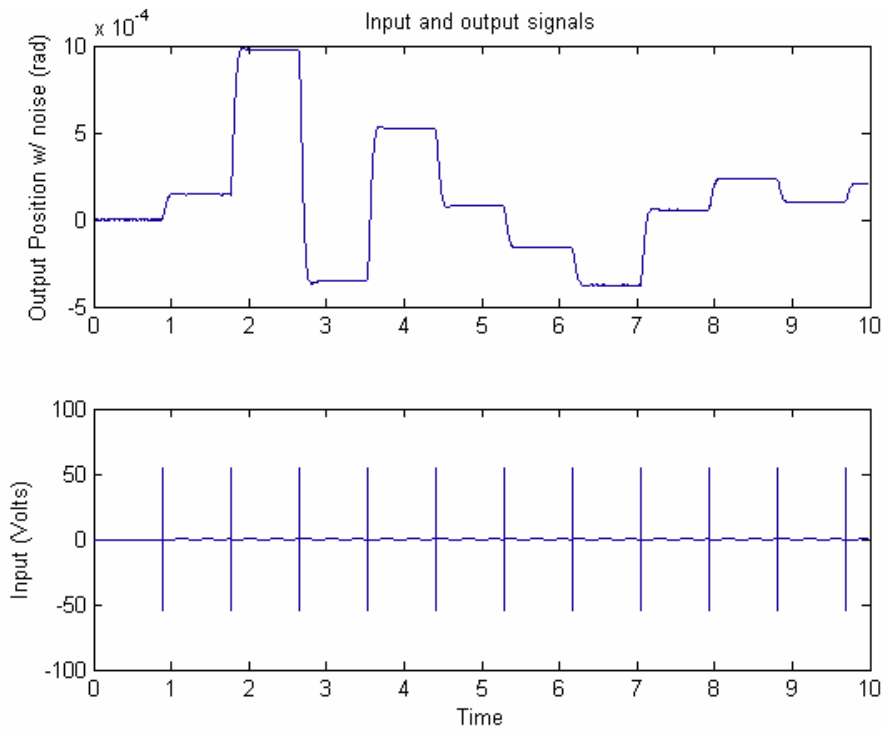


Figure 36: Identification data with added noisy output

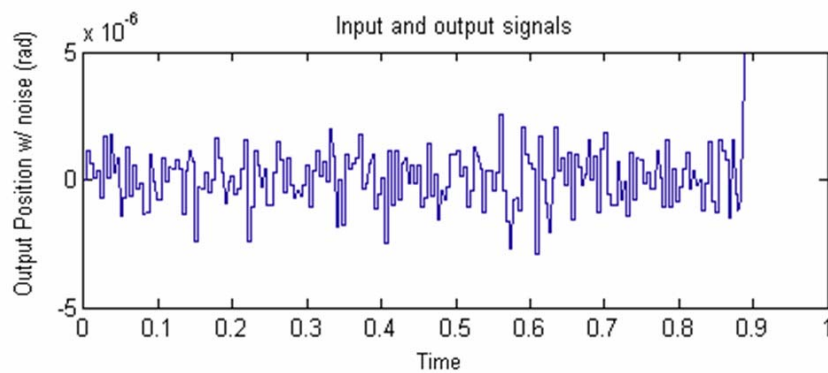


Figure 37: Close up of output noise

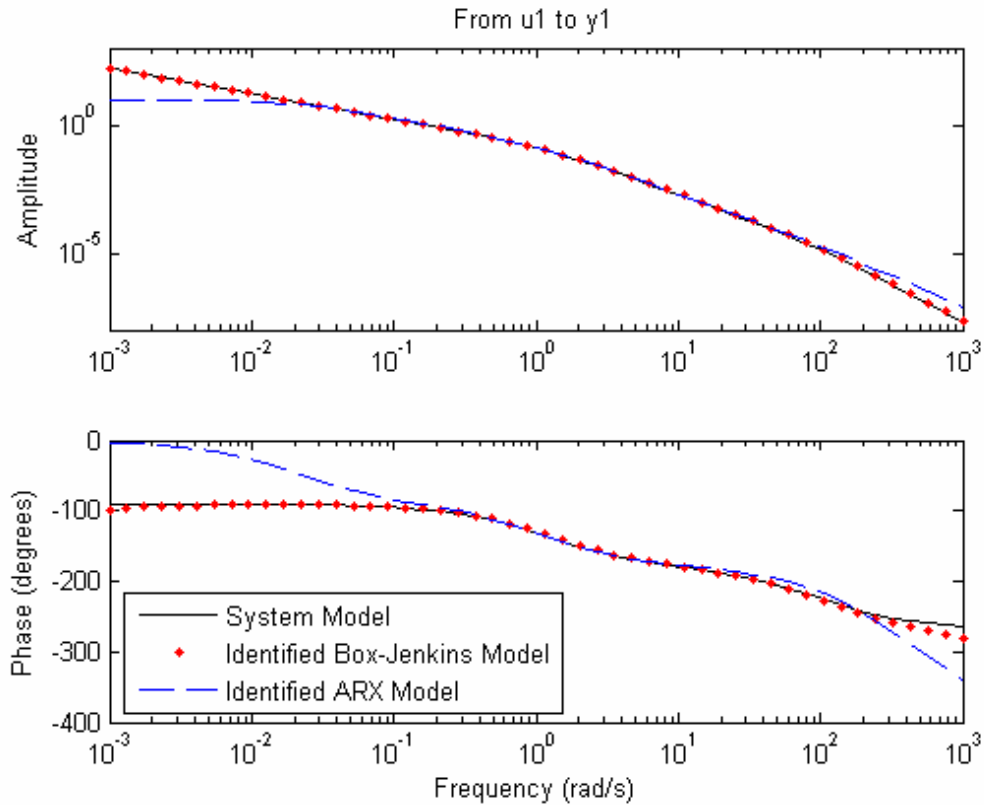


Figure 38: Comparison of Actual, Box-Jenkins, and ARX models with output error

In summary, the ARX model with either a delay of two or three samples identified the C-axis dynamics accurately. The accuracy of the identified ARX models implies that the effort required to determine all the parameters as discussed in the previous chapters can be replaced by proper input-output data if the disturbances and share the same dynamics as the system. The identified models are also suitable for control design because the dynamics of the models in the frequency range of interest is accurately captured. Even in the case of error in the input to the system the ARX model was able to identify the system correctly. Even when a different model structure must be used to account for additive output

error, such as the Box-Jenkins model above, the system dynamics can still be captured. However, the identified model deviates from the model of the true system in this case and better data is needed. The identified ARX model is ideal for control design because it accurately captures the dynamics of the system. The control design from the identified model is discussed in the next Chapter.



## CHAPTER V

### CONTROL DESIGN

This chapter presents how the model identified in Chapter IV by system identification is used to analytically determine controller gains. The identified model and controller are combined to form a single transfer function. The poles of this transfer function are placed using pole placement and the gains necessary to achieve these pole locations are determined.

An accurate plant model determined by system identification can be used to determine the necessary controller gains to achieve the desired performance of the system without tuning. These gains are obtained deterministically by obtaining the closed loop transfer function of the system and placing the dominant poles in a location that yields the desired response. From the documentation of the PMAC the output of the controller is governed by Equation (3.1).

$$\text{CMDout}(n) = 2^{-19} \text{Ix30} \left[ \begin{array}{l} \left\{ \text{Ix08} \left[ \text{FE}(n) + \frac{\text{Ix32} \cdot \text{CV}(n) + \text{Ix35} \cdot \text{CA}(n)}{2^7} + \frac{\text{Ix33} \cdot \text{IE}(n)}{2^{23}} \right] \right\} \\ - \frac{\text{Ix31} \cdot \text{Ix09} \cdot \text{AV}(n)}{2^7} \end{array} \right]$$

In terms of PID and feed-forward Gains,

$$\text{CMDout}(n) = 2^{-19} \cdot K_p \cdot \left[ \begin{array}{l} \left\{ \text{Ix08} \cdot \left[ \text{FE}(n) + \frac{K_{\text{vff}} \cdot \text{CV}(n) + K_{\text{Aff}} \cdot \text{CA}(n)}{2^7} + \frac{K_1 \cdot \text{IE}(n)}{2^{23}} \right] \right\} \\ - \frac{K_D \cdot \text{Ix09} \cdot \text{AV}(n)}{2^7} \end{array} \right] \quad (5.1)$$

Where, the parameters for the equation are shown in Table 4. The schematic of the system setup is shown in the appendix. The following definitions are used to re-write Equation (5.1) above with difference equations.

$$\text{FE}(n) = \text{CP}(n) - \text{AP}(n) \quad (5.2)$$

$$\text{CV}(n) = \text{CP}(n) - \text{CP}(n-1) = (\text{CP})(1 - z^{-1}) \quad (5.3)$$

$$\begin{aligned} \text{CA}(n) &= \text{CV}(n) - \text{CV}(n-1) = (\text{CP}(n) - \text{CP}(n-1)) - (\text{CP}(n-1) - \text{CP}(n-2)) \\ &= \text{CP}(1 - z^{-1}) - \text{CP}(z^{-1} - z^{-2}) = (\text{CP})(1 - 2z^{-1} + z^{-2}) \end{aligned} \quad (5.4)$$

$$\text{IE}(n) = \sum_{i=1}^n \text{FE}(n)_i = \sum_{i=1}^n (\text{CP}(n) - \text{AP}(n))_i = \frac{1}{1 - z^{-1}} \cdot (\text{CP} - \text{AP}) \quad (5.5)$$

Inserting these definitions into Equation (5.1) yields the following expression,

$$\text{CMDout} = 2^{-19} K_p \left[ \begin{array}{l} \text{Ix08} \left( \text{CP} - \text{AP} + \frac{K_{\text{vff}} \text{CP}(1 - z^{-1}) + K_{\text{Aff}} \text{CP}(1 - 2z^{-1} + z^{-2})}{2^7} \right) \\ + \frac{K_1 (\text{CP} - \text{AP})}{2^{23} (1 - z^{-1})} \end{array} \right] - \frac{K_D \text{Ix09} \text{AP}(1 - z^{-1})}{2^7} \quad (5.6)$$

If the command position and actual position terms are collected equation (5.6) becomes,

$$\text{CMDout} = \left[ 2^{-19} K_p \text{Ix08} \left( 1 + \frac{K_{\text{vff}} (1-z^{-1}) + K_{\text{Aff}} (1-2z^{-1} + z^{-2})}{2^7} + \frac{K_I}{2^{23} (1-z^{-1})} \right) \text{CP} \right. \\ \left. - 2^{-19} K_p \left( \text{Ix08} + \frac{K_I \text{Ix08}}{2^{23} (1-z^{-1})} + \frac{K_D \text{Ix09AP} (1-z^{-1})}{2^7} \right) \text{AP} \right] \quad (5.7)$$

For transparency, the following definitions are used to simplify equation (5.6).

$$K_{\text{Ps}} = 2^{-19} K_p \quad (5.8)$$

$$K_{\text{vffs}} = \frac{K_{\text{vff}}}{2^7} \quad (5.9)$$

$$K_{\text{Affs}} = \frac{K_{\text{Aff}}}{2^7} \quad (5.10)$$

$$K_{\text{Is}} = \frac{K_I}{2^{23}} \quad (5.11)$$

$$K_{\text{Ds}} = \frac{K_D}{2^7} \quad (5.12)$$

Using the above definitions yields,

$$\text{CMDout} = \left( \begin{array}{l} K_{\text{Ps}} \text{Ix08} \left( 1 + K_{\text{vffs}} (1-z^{-1}) + K_{\text{Affs}} (1-2z^{-1} + z^{-2}) + \frac{K_{\text{Is}}}{(1-z^{-1})} \right) \text{CP} \\ - K_{\text{Ps}} \left( \text{Ix08} + \frac{\text{Ix08} K_I}{(1-z^{-1})} + K_{\text{Ds}} \text{Ix09} (1-z^{-1}) \right) \text{AP} \end{array} \right) \quad (5.13)$$

The identified motor in equation (4.5) can be simplified by ignoring negligible terms as,

$$\frac{20.9}{s^3 + 101.4s^2 + 128.1s} \quad (5.14)$$

The amplifier is treated a simple low pass filter with a gain. The transfer function for combined motor and amplifier is:

$$\frac{AP}{CMDout} = Res_C \cdot DAC \left( \frac{K_A \cdot 2\pi f}{s + 2\pi f} \right) \left( \frac{20.9}{s^3 + 101.4s^2 + 128.1s} \right). \quad (5.15)$$

Where,  $Res_C$  is the resolution of the C-axis, DAC is the conversion factor of the digital to analog converter,  $f$  is the cutoff frequency of the amplifier (523 Hz), and  $K_A$  is the amplifier gain. Equation (5.15) must be converted to discrete form to combine with equation (5.13). Conversion by a zero-order hold with a sample time of  $440\mu s$  yields and rewriting in terms of CMDout yields,

$$CMDout = AP \left( \frac{z^4 - 3.192z^3 + 3.609z^2 - 1.642z + 0.2253}{7.587 \times 10^{-8}z^2 + 3.001 \times 10^{-7}z + 7.42 \times 10^{-8}} \right) \quad (5.16)$$

In order to eliminate the CMDout term, equation (5.16) is combined with (5.13) to yield the following,



$$\begin{aligned}
& \text{AP} \left( \frac{z^3 - 2.957z^2 + 2.915z - 0.9573}{8.949 \times 10^{-8} z^2 + 3.541 \times 10^{-7} z + 8.755 \times 10^{-8}} \right) \\
& = K_{Ps} \text{Ix08} \left( 1 + K_{Vffs} (1-z^{-1}) + K_{Affs} (1-2z^{-1} + z^{-2}) + \frac{K_{Is}}{(1-z^{-1})} \right) \text{CP} \quad (5.17) \\
& - K_{Ps} \left( \text{Ix08} + \frac{\text{Ix08} K_{Is}}{(1-z^{-1})} + K_{Ds} \text{Ix09} (1-z^{-1}) \right) \text{AP}
\end{aligned}$$

Collecting terms for AP yields,

$$\begin{aligned}
& \text{AP} \left\{ \left( \frac{z^3 - 2.957z^2 + 2.915z - 0.9573}{8.949 \times 10^{-8} z^2 + 3.541 \times 10^{-7} z + 8.755 \times 10^{-8}} \right) + K_{Ps} \left( \text{Ix08} + \frac{\text{Ix08} K_{Is}}{(1-z^{-1})} + K_{Ds} \text{Ix09} (1-z^{-1}) \right) \right\} \\
& = K_{Ps} \text{Ix08} \left( 1 + K_{Vffs} (1-z^{-1}) + K_{Affs} (1-2z^{-1} + z^{-2}) + \frac{K_{Is}}{(1-z^{-1})} \right) \text{CP} \quad (5.18)
\end{aligned}$$

Therefore the transfer function for the controller and plant together is given by

$$\begin{aligned}
\frac{\text{AP}}{\text{CP}} &= \frac{K_{Ps} \text{Ix08} \left( 1 + K_{Vffs} (1-z^{-1}) + K_{Affs} (1-2z^{-1} + z^{-2}) + \frac{K_{Is}}{(1-z^{-1})} \right)}{\left\{ \left( \frac{z^3 - 2.957z^2 + 2.915z - 0.9573}{8.949 \times 10^{-8} z^2 + 3.541 \times 10^{-7} z + 8.755 \times 10^{-8}} \right) + K_{Ps} \left( \text{Ix08} + \frac{\text{Ix08} K_{Is}}{(1-z^{-1})} + K_{Ds} \text{Ix09} (1-z^{-1}) \right) \right\}} \\
& \quad (5.19)
\end{aligned}$$

Equivalently,

$$\frac{AP}{CP} = \frac{K_{ps} \text{Ix08} \left( 1 + K_{vfb} \left( \frac{z-1}{z} \right) + K_{afb} \left( \frac{z^2 - 2z + 1}{z^2} \right) + K_{is} \left( \frac{z}{z-1} \right) \right)}{\left\{ \left( \frac{z^3 - 2.957z^2 + 2.915z - 0.9573}{8.949 \times 10^{-8} z^2 + 3.541 \times 10^{-7} z + 8.755 \times 10^{-8}} \right) + K_{ps} \left( \text{Ix08} + \text{Ix08} K_{is} \left( \frac{z}{z-1} \right) + K_{ds} \text{Ix09} \left( \frac{z-1}{z} \right) \right) \right\}}$$

This is the overall transfer function of the system. Simplifying the above expression and examining the pole locations as functions of the controller gains will lead to the gains that place the poles at the desired location.

The desired locations of the poles are determined by the desired steady state and transient response characteristics. The desired settling time is no greater than 0.2 seconds, the desired system damping must be at least 0.8 so that the overshoot is less than 2% and to ensure no amplification at the systems natural frequency. This corresponds to a rise time of 0.166 sec. The specification for the settling time in the Laplace domain is,

$$t_s = \frac{4}{\zeta \omega_n} = \frac{4}{0.8 \omega_n} = 0.2 \text{ sec} \quad (5.20)$$

Therefore the natural frequency should be 25 rad/sec. The settling time performance specifications is mapped to the Z-domain by  $z = e^{(-\zeta \omega_n T)}$  where T is sampling time. Therefore the poles must lie inside the circle defined by the radius,

$$r \leq e^{-\zeta \omega_n T d} = e^{-(0.8)(25)(0.00044)} = 0.9912 \quad (5.21)$$

The imaginary part of the dominant poles lies at an angle  $\theta$  defined by,

$$\theta = \omega_d T_p = \left( \omega_n \sqrt{1 - \zeta^2} \right) 0.00044 = 0.0066 \quad (5.22)$$

Therefore, the poles in the z-plane must be located at values less than  $0.9912 \pm 0.00653j$ . When simplified, the denominator of Equation (5.19) is a seventh order polynomial in z and has a pole at 0 and 1 that is due to the digitization. Although, the pole at 1 would yield a marginally stable system it is cancelled by a zero at 1 in the numerator. The denominator is in terms of the proportional gain,  $K_P$  the integral gain,  $K_I$  and the derivative gain,  $K_D$ .

The controller gains that give the desired pole locations can be found by creating a polynomial with the desired poles and equating the coefficients of this polynomial to the coefficients of the seventh order polynomial denominator in Equation (5.19). Because both polynomials are monic there are six equations relating the coefficients. The coefficients of the denominator in Equation (5.19) are in terms of the three unknown gains,  $K_P$ ,  $K_I$ , and  $K_D$ . Four of the pole locations are specified by the desired performance characteristics and constraints of the system; the other three poles are unknown. Therefore, the three unknown poles and three unknown gains are determined by solving the six equations relating the coefficients of the polynomials. Therefore, only two of the poles can be uniquely placed, two are fixed at 0 and 1, and the other three are determined by solving the equations. The  $K_P$ ,  $K_I$ , and  $K_D$  gains are uniquely found by specifying

four of the pole locations. A MATLAB program was used to find the gains and remaining pole locations and is shown in the appendix.

The pole locations based on the transient specifications were at  $0.9912 \pm 0.00653j$ . The two other poles intrinsic to the system were at 0 and 1. The other three poles determined by solving the system of equations were 0.23448, 0.97469, and 0.000304537 indicating that the system is stable with the desired dominant poles. The gains determined by these poles where  $K_P = 84,666$ ,  $K_I = 1,586$ , and  $K_D = 22,829$ . The response of the simulated system to a step input of 1000 counts using these gains is shown in Figure 39. The figure shows that the desired settling time of 0.2 sec is met but the rise time is 0.192 sec which fails by approximately 14 % to meet the rise time specification of 0.166 sec. This is due to the fact that there is no control over the other three pole locations and they are placed by their relationship to the three gains and the other pole locations. However, the desired performance characteristics are close to being met and the step response indicates that the dominant poles have been placed as desired. Figure 39 also shows the step response from the gains determined by the experienced professional. In comparison, the gains determined by the experienced professional were where  $K_P = 200,000$ ,  $K_I = 1000$ , and  $K_D = 15,000$ . The dominant pole locations with these gains are at  $[0, 0.99994, 0.98840 + 0.014529j, 0.98840 - 0.014529j, 0.98840, 0.23406, 0.00042653j]$ . A similar response can be attained by pole placement if the desired poles are changed.

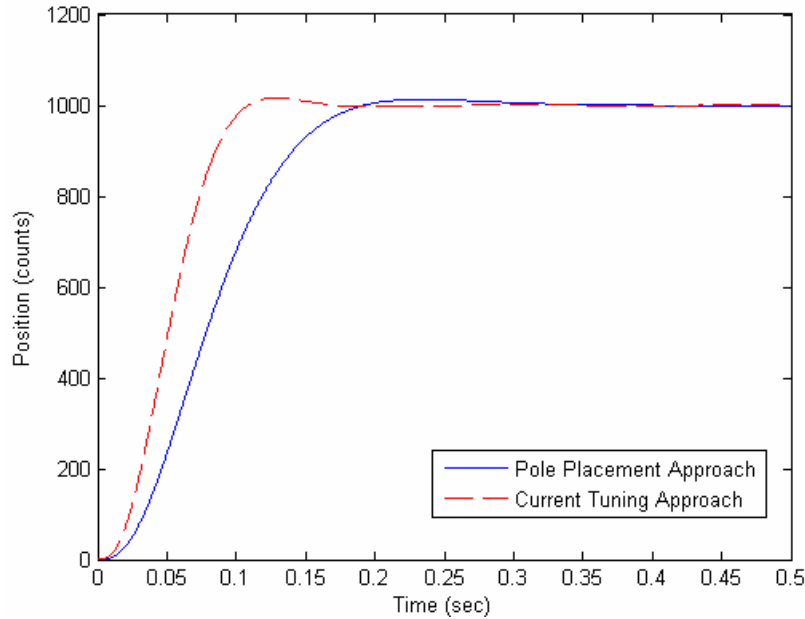


Figure 39: Step response using pole placement gains

A pole-zero map of the system with the gains determined by pole placement is shown in Figure 40 with the poles indicated by an 'x' and the zeroes indicated by a red 'o'. A close-up of the pole locations is shown in Figure 41. The figures show the pole locations at  $[0, 1, 0.9912+0.00653i, 0.9912-0.00653i, 0.23448, 0.97469, 0.000305]$  and that there is pole-zero cancellation at 0 and 1.

The feasible pole locations can be determined by observing the pole-zero map for a range of gains. To observe how specific ranges of gains affect the system the proportional, integral, and derivative gains were varied individually to see their effect on the overall pole locations. In addition, the effect of each gain on the maximum acceleration constraint of  $245^{\circ}/s^2$  is investigated.

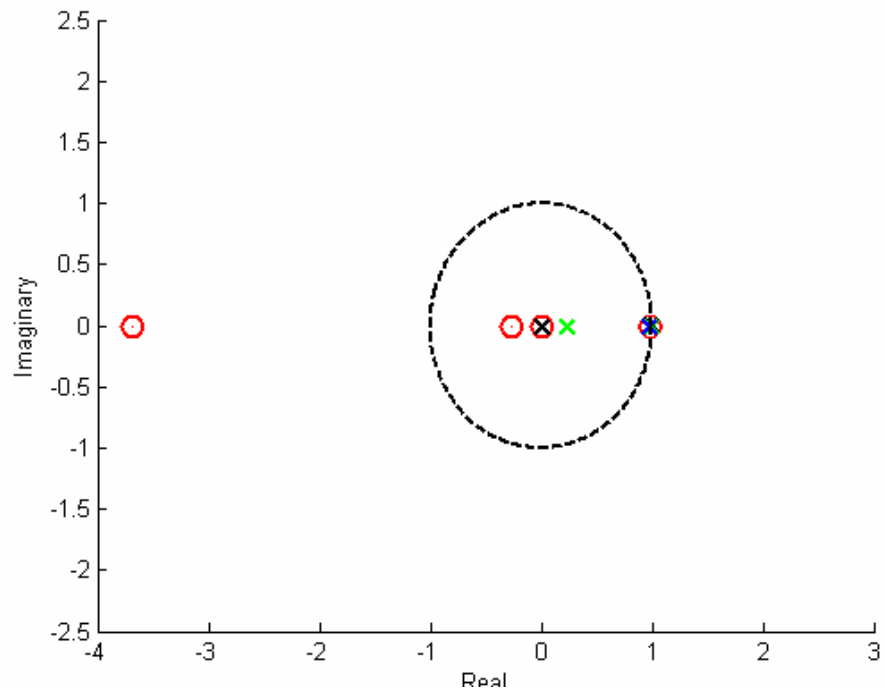


Figure 40: Pole-zero map of system with pole placement approach

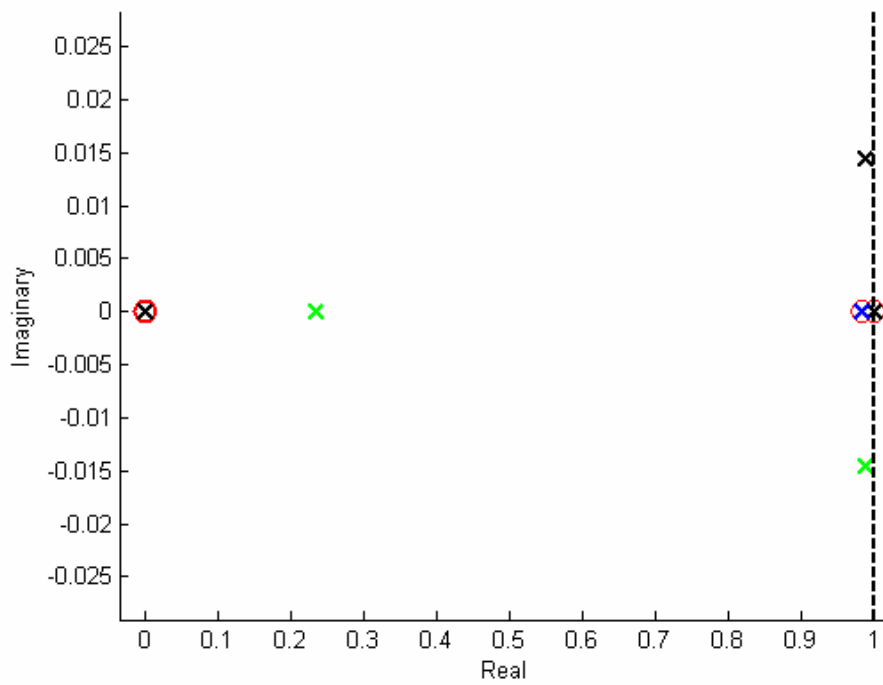


Figure 41: Close-up of pole-zero map

The proportional gain's effect on the pole locations was investigated by holding the integral and derivative gains constant at 1,000 and 15,000 respectively, while the proportional gain was varied. The result from varying the proportional gain from 1 to 1,000,000 in increments of 5,000 is shown in Figure 42. Close-ups, shown in Figure 43 and Figure 44, show that the poles near zero are only slightly affected but the poles near 1 are significantly affected by varying KP. Figure 42 through Figure 44 show that the locations of the zeros are not affected by the proportional gain. The effect of the proportional gain on the maximum acceleration is shown in Figure 45. Figure 45 shows that the proportional gain should be less than 475,000 so that the max acceleration limitation is not exceeded.

The effect of the integral gain was investigated by holding the proportional and derivative gains constant at 200,000 and 15,000 respectively while varying the integral gain from 1 to 1,000,000 in increments of 5,000. Plots of the pole and zero locations as  $K_I$  is varied are shown in Figure 46 through Figure 48. The plots show that  $K_I$  moves the location of the zeros and dominant poles significantly and can result in an unstable system for large values of  $K_I$ . The value of  $K_I$  where the system becomes unstable is approximately 150,000 with the proportional and derivative gains held constant. Figure 49 shows that the integral gain does not affect the max acceleration limits but this is because controller only integrates when the velocity is zero

The derivative gain's effect on the system was investigated by holding the proportional and integral gains constant at 200,000 and 1,000 respectively, while

varying the derivative gain from 1 to 1,000,000. Plots of the pole and zero locations as  $K_D$  is varied is shown in Figure 50 through Figure 52. These plots show that the derivative gain does not move the zero locations and has very little effect on the poles closest to 0. However, the poles near 1 are significantly affected by the derivative gain. The plots indicate that large values for  $K_D$  can result in an unstable system. The value of  $K_D$  where the system becomes unstable is approximately 350,000 with the proportional and integral gains held at the constant values. Figure 53 shows that the derivative gain must be greater than 7000 so that the acceleration limit is not exceeded.

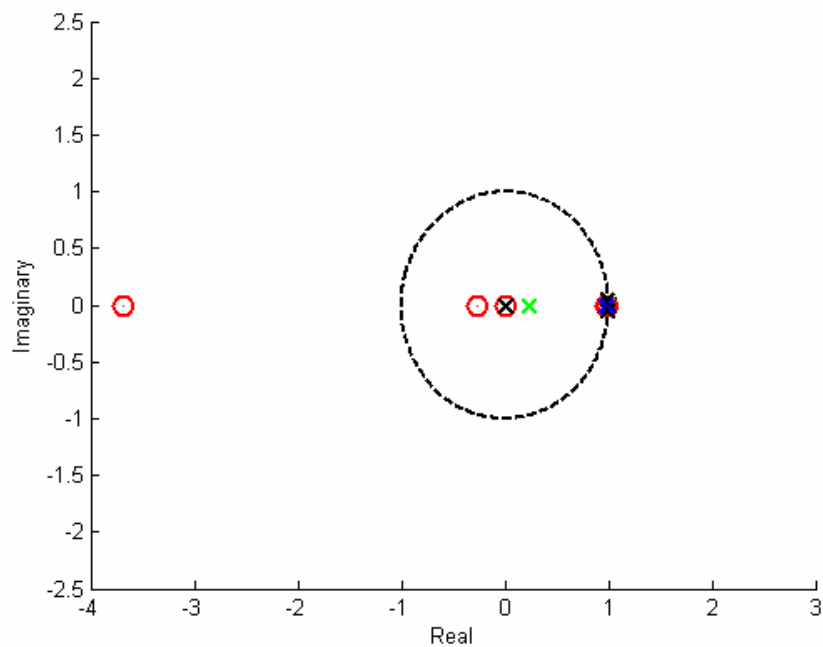


Figure 42: Pole locations as  $K_p$  is varied from 1 to 1,000,000



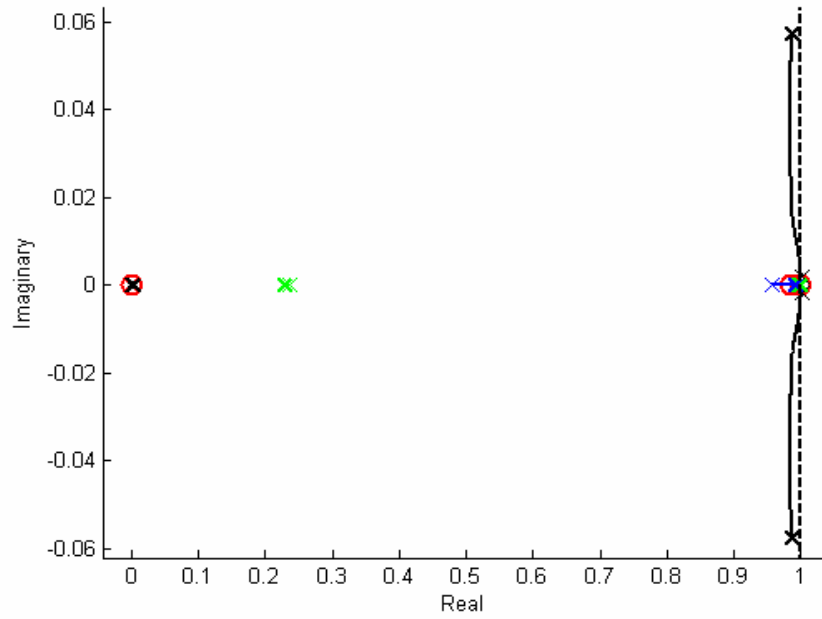


Figure 43: Close up of pole locations as  $K_p$  is varied from 1 to 1,000,000

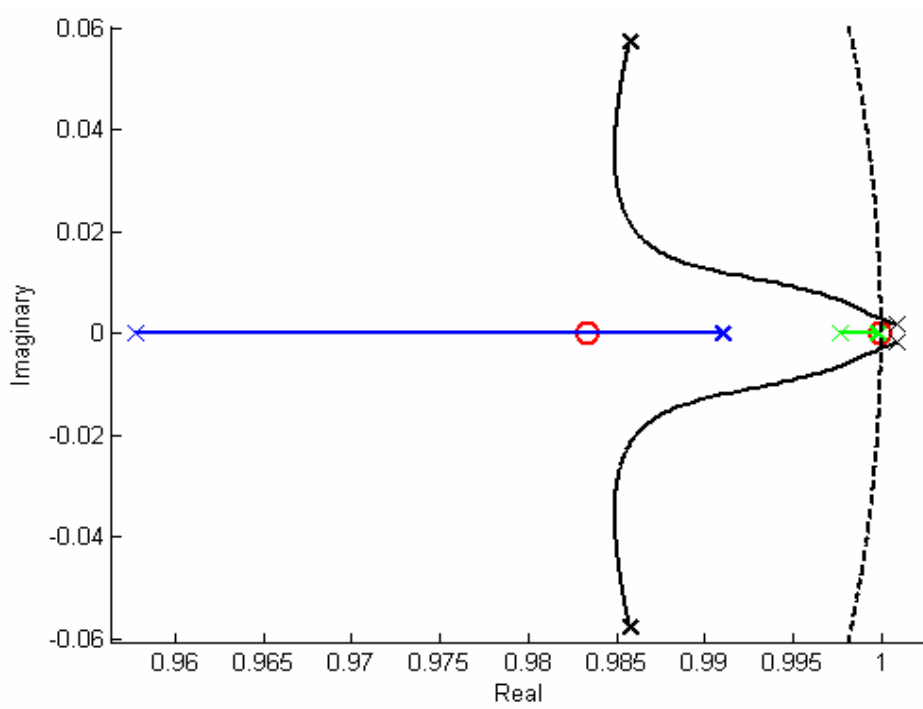


Figure 44: Dominant poles affected by  $K_p$

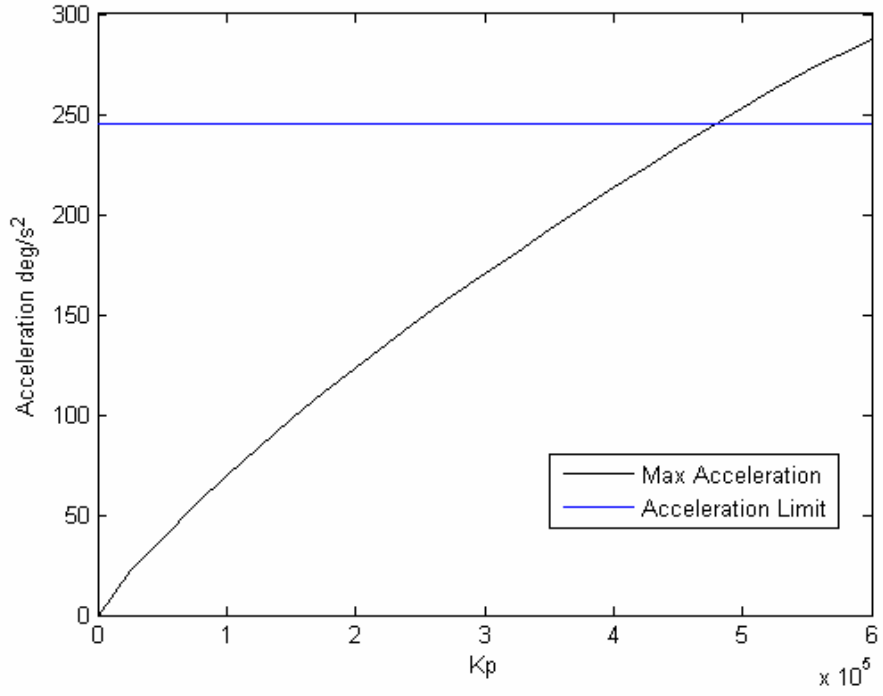


Figure 45: Acceleration limit related to K<sub>p</sub>

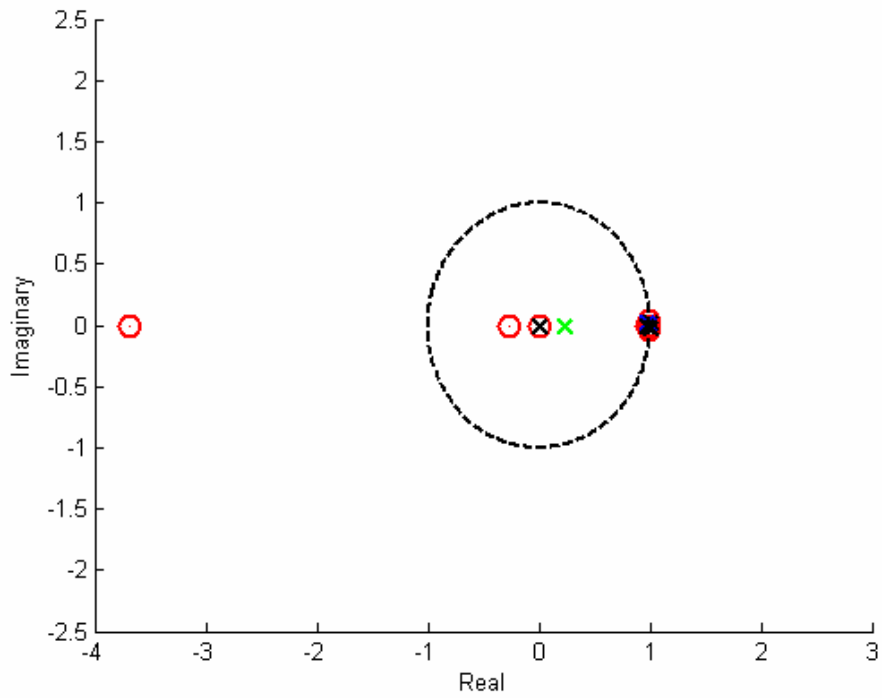


Figure 46: Pole locations as K<sub>I</sub> is varied from 1 to 1,000,000

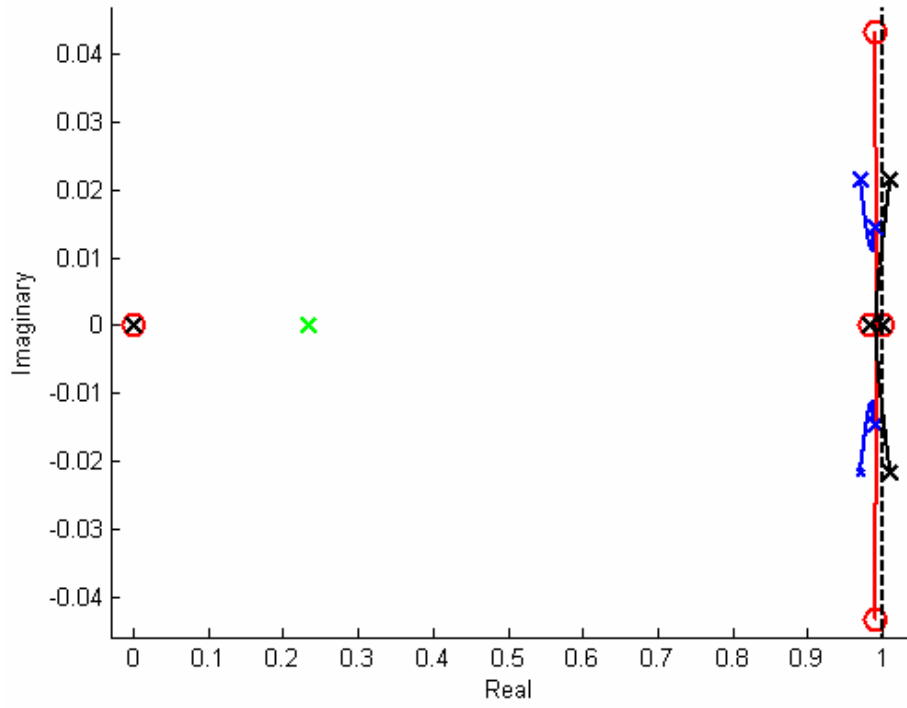


Figure 47: Close up of pole locations as  $K_I$  is varied from 1 to 1,000,000

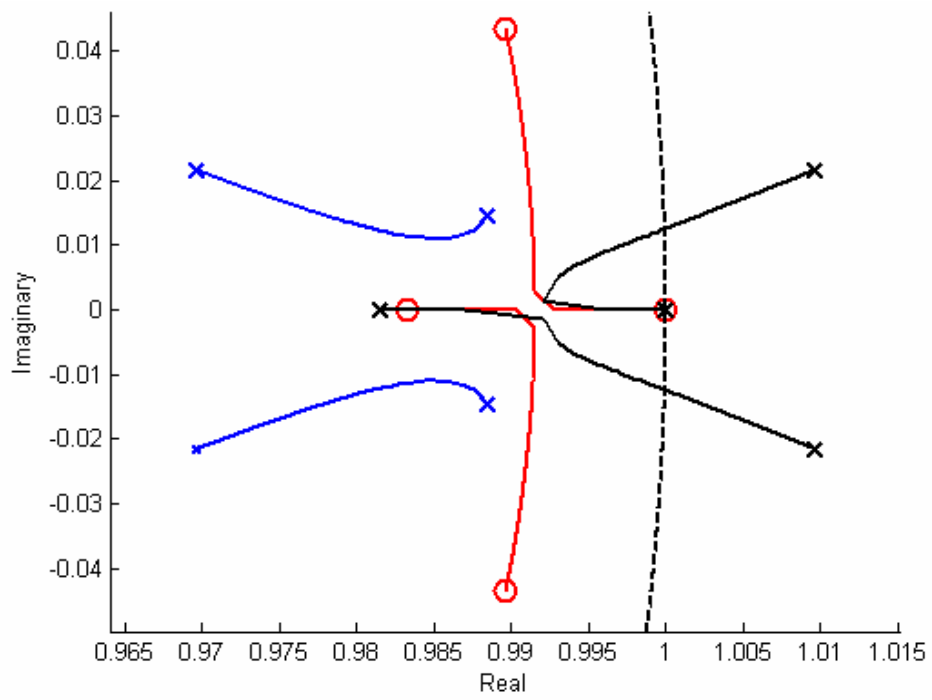


Figure 48: Dominant poles affected by  $K_I$

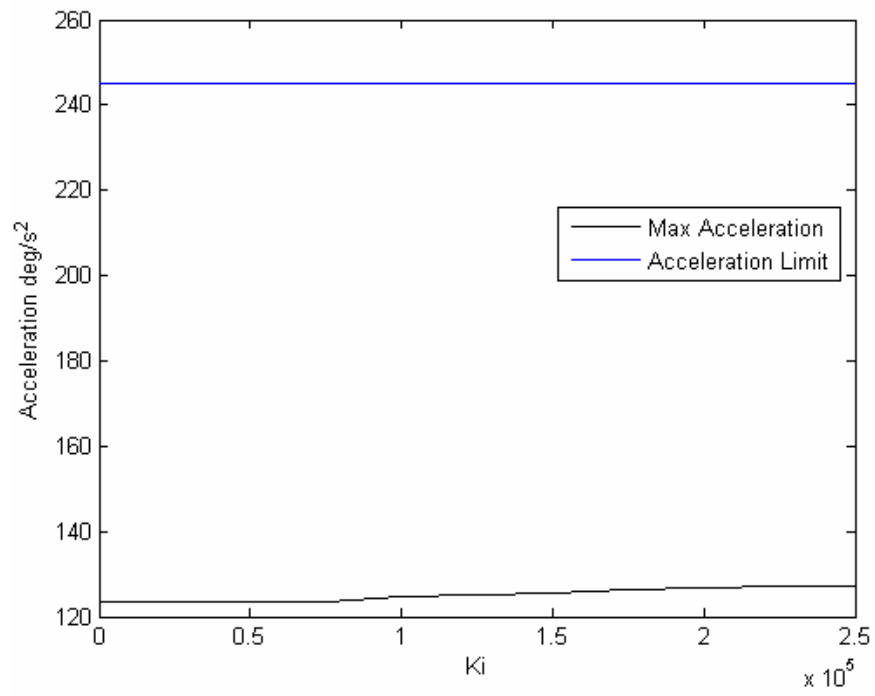


Figure 49: Acceleration limit related to  $K_I$

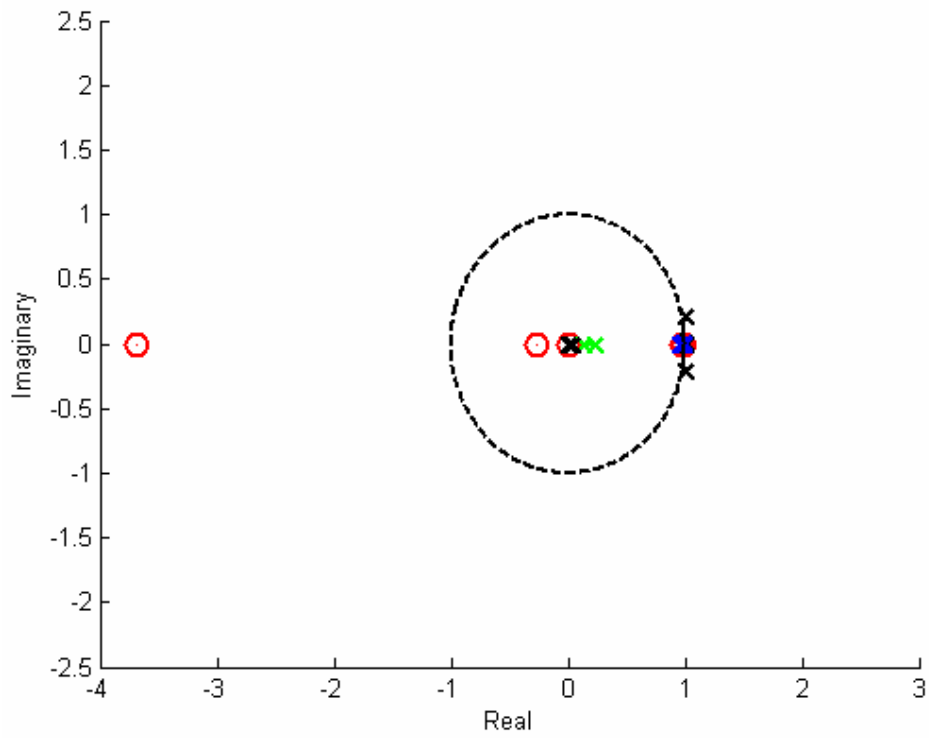


Figure 50: Pole locations as  $K_D$  is varied from 1 to 1,000,000

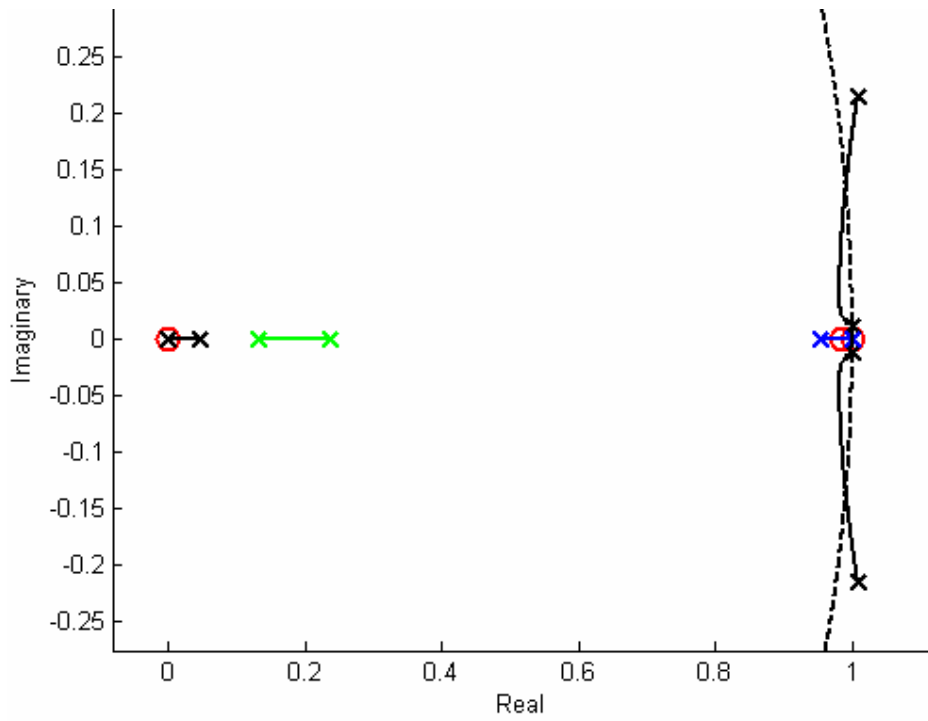


Figure 51: Close up of pole locations as  $K_D$  is varied from 1 to 1,000,000

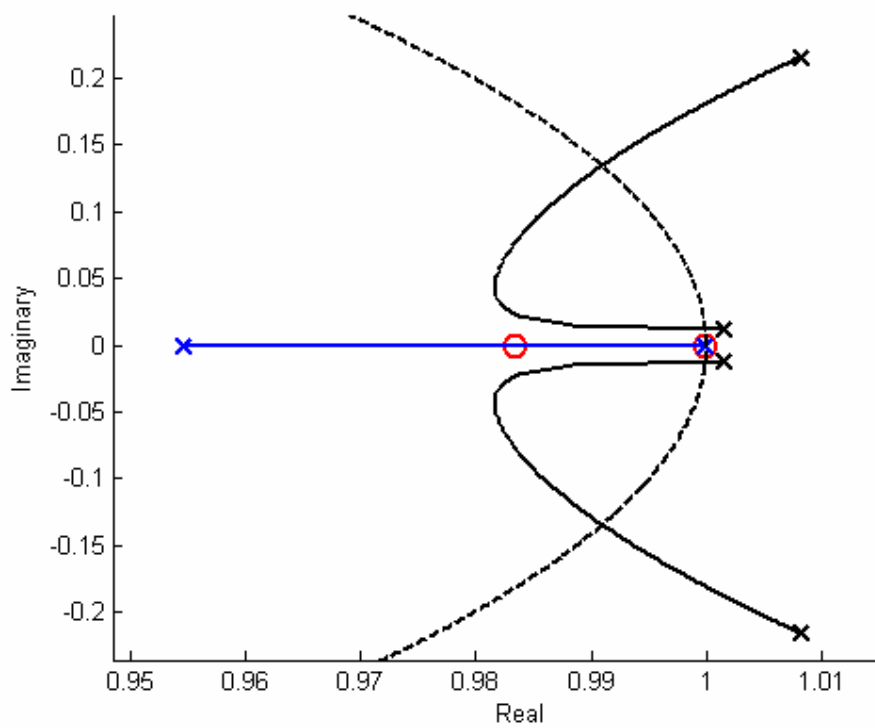


Figure 52: Dominant poles affected by  $K_D$

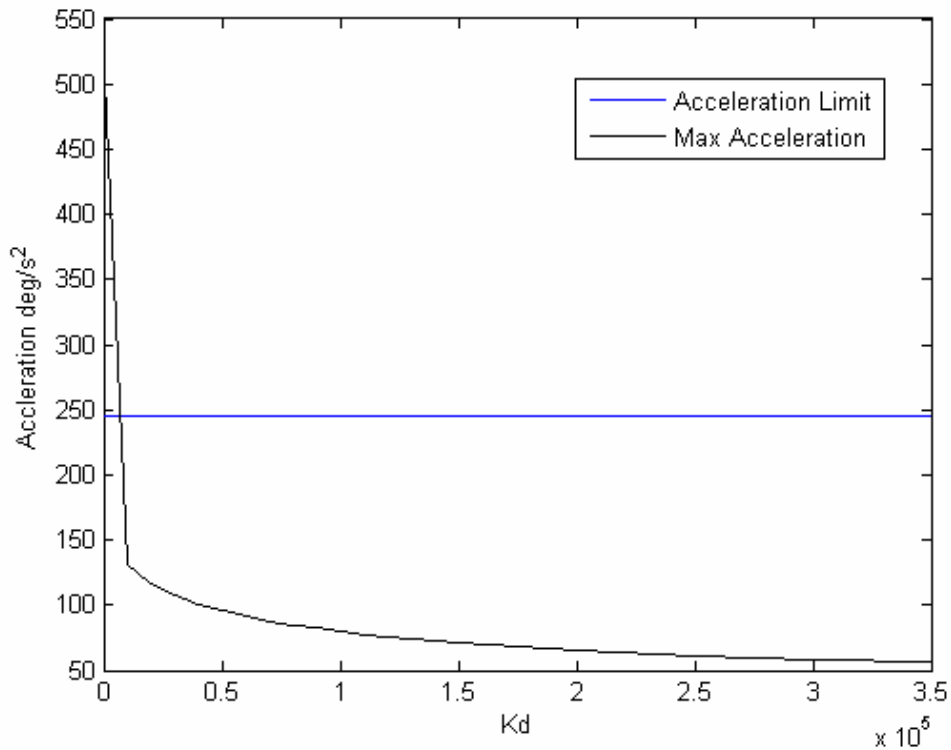


Figure 53: Acceleration limit related to  $K_D$

In summary, the plant identified by system identification was used to determine the closed loop transfer function of the controller-plant system. The dominant poles of this transfer function were placed by pole placement and the gains were determined. The system response using the pole placement approach matches the response to the current tuning approach well. This suggests that the tuning process can be replaced by proper system identification and pole placement. The pole-zero plots as functions of the proportional, integral, and derivative gains suggest that there is a wide range of possible gain combinations that will result in placing the dominant poles in desired locations.

## CHAPTER VI

### CONCLUSIONS AND RECOMMENDATIONS

#### **Conclusions**

The goal of this research was to use model-based control procedures to achieve the desired performance of the Universal Measuring Machine. In order to use to use model based control, a model was constructed by mathematically modeling the system and by using system identification. The model identified using system identification was used for model-based control via pole-placement.

The results of Chapter III demonstrate that a considerable level of effort is required to determine the parameters for this model. Theoretical and experimental methods were used to determine the unknown parameters of the R- and C-axes. A simulation of the C- and R- axes verifies the results of mathematical modeling and the theoretical and experimental parameter determination. The simulation and actual data from the R- and C-axes were compared and indicate that the parameters are identified accurately by this approach. The error in simulated data for the R-axis is approximately 6%. The error in simulated data for the C-axis is approximately 4% which shows that the simulation closely represents the actual data.

This approach requires the use of theoretical and assumed models, manufacturer data sheets, tailored experiments, post-processing of data, machine-down time, knowledge of gains, machine parameters, and the use of estimated values. Although the model is able to reproduce the actual data well, this approach is time consuming and can be replaced by proper system identification.

A deterministic model for the C-axis is determined using system identification in Chapter IV. An ARX model is assumed with orders attained from the *a priori* knowledge of the systems' dynamic model structure. The model built by using the experimental procedures of Chapter III and the model built via system identification in Chapter IV show extremely close agreement. This indicates that the model built by system identification can replace the model based on mathematical modeling if the proper input-output data is obtained. The model identified by system identification matches the actual data recorded from the C-axis well. These results are summarized in Figure 33. The ARX model with a delay of 2 samples results in a mean error of 0.41%. The ARX model with a delay of 3 samples results in a mean error of 0.40%. The model built by system identification in the noise-free simulation shows a 10.3% and 1.6% percent error in the identified poles when compared to those used to simulate the system. Control design using the identified model shows acceptable performance when implemented on the simulated model.

Use of system identification to determine a model for one of the axes requires considerably less effort and time than performing the manual parameter estimation approach of Chapter III. System identification requires input-output



data that is replete with system information. A white noise input is selected as the input to ensure that the necessary frequency content was present in the data. However, the real system may not lend itself well to this type of input and care must be taken to work within the limitations of the actuators and power supplies. Other options for inputs are discussed in Chapter II. System identification works well in the simulations and suggests that implementation on the actual system will show similar results.

The model built by system identification is used for analytical control design by pole-placement. In Chapter V the desired dominant poles are placed according to the desired damping ratio, settling time, and rise time. The transient response with these poles meets the damping and settling time constraints but the rise time error is approximately 14%. The response of the system using pole-placement and the response determined by expert tuning is shown in Figure 39. The results of Chapter V indicate that model-based control is achievable using a model built by system identification and that the tuning process can be replaced by pole-placement.

Pole-placement has some distinct advantages over manually tuning the system. Pole-placement is analytical and does not require the use of the actual machine. A range of pole locations are obtained by varying the gains or desired dominant pole locations as the pole-zero maps of Chapter V demonstrate. Simply using pole-placement with a set of desired dominant poles does not necessarily guarantee desired performance as demonstrated by the error in rise-time. The dominant poles of the system are placed in the desired locations but there is no

freedom to place the other poles. In addition, the gains resulting from selecting a set of desired poles are not always physically realizable due to machine limitations. Manually tuning the system by an expert with knowledge of the desired systems response and machine limitations results in the desired performance; however, this requires expert knowledge and is time-consuming. The pole-placement approach demonstrates that the entire control design process can be automated.

### **Recommendations**

This thesis demonstrates that model based control is not only feasible but suggests that it can be automated for the UMM. The modeling process is time consuming due to unknown system parameters and should be done by system identification. Accurate models of the UMM axes can be determined by system identification. The input and output data collected for system identification must contain enough information to identify the system accurately. The input signal to be used for identification must at least contain frequency content throughout the systems standard operation range and ideally cover the systems bandwidth. A variety of inputs are available to excite the system and their frequency content can be tested by the crest factor. Band-limited white noise should be used if possible but other inputs such as a pseudo-random binary signal can also be used and are easier to implement. A long enough data record should be collected so that the low frequency dynamics of the system are captured. The sampling rate must be selected so that there are enough data points during the transient response of the

system. The recommended sampling rate is 10 times faster than the systems bandwidth which ensures 5-8 data points over the rise-time in a step-response (Ljung and Glad, 1994). This corresponds to selecting a sampling rate that will accurately capture the dominant time constants of the system.

The current control approach of tuning the UMM can be replaced by pole placement. Ranges of acceptable gains should be determined as well as the limitations of the machines actuators, power supply, and acceleration limits so that these limitations are taken into account when placing the poles. The identification and pole-placement approach presented here can be automated and used on-line. An algorithm should be created to perform system identification on the systems axes and from the knowledge of the controller structure and system limitations place the dominant poles on-line. This would allow the machine to update appropriate gains for each axis without additional tuning and allow time-varying process parameters to be identified so that the machine performs optimally.



## APPENDIX

### MATLAB/SIMULINK PROGRAMS

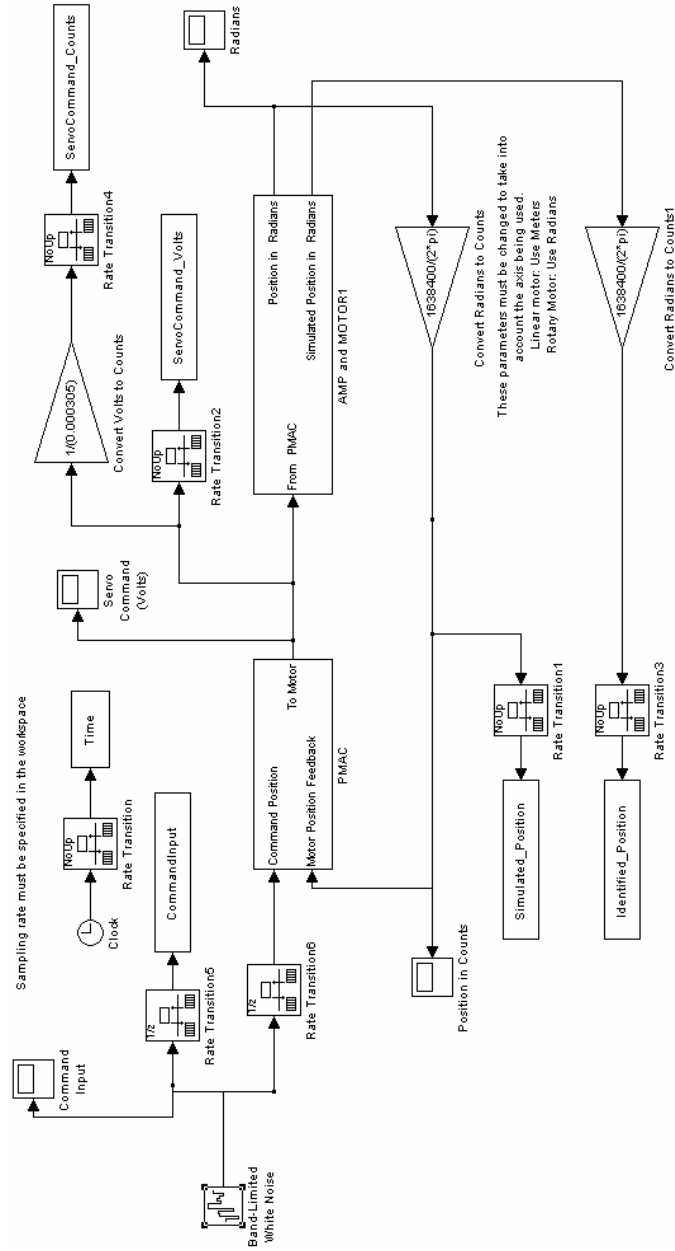


Figure 54: Diagram of UMM controller and motor

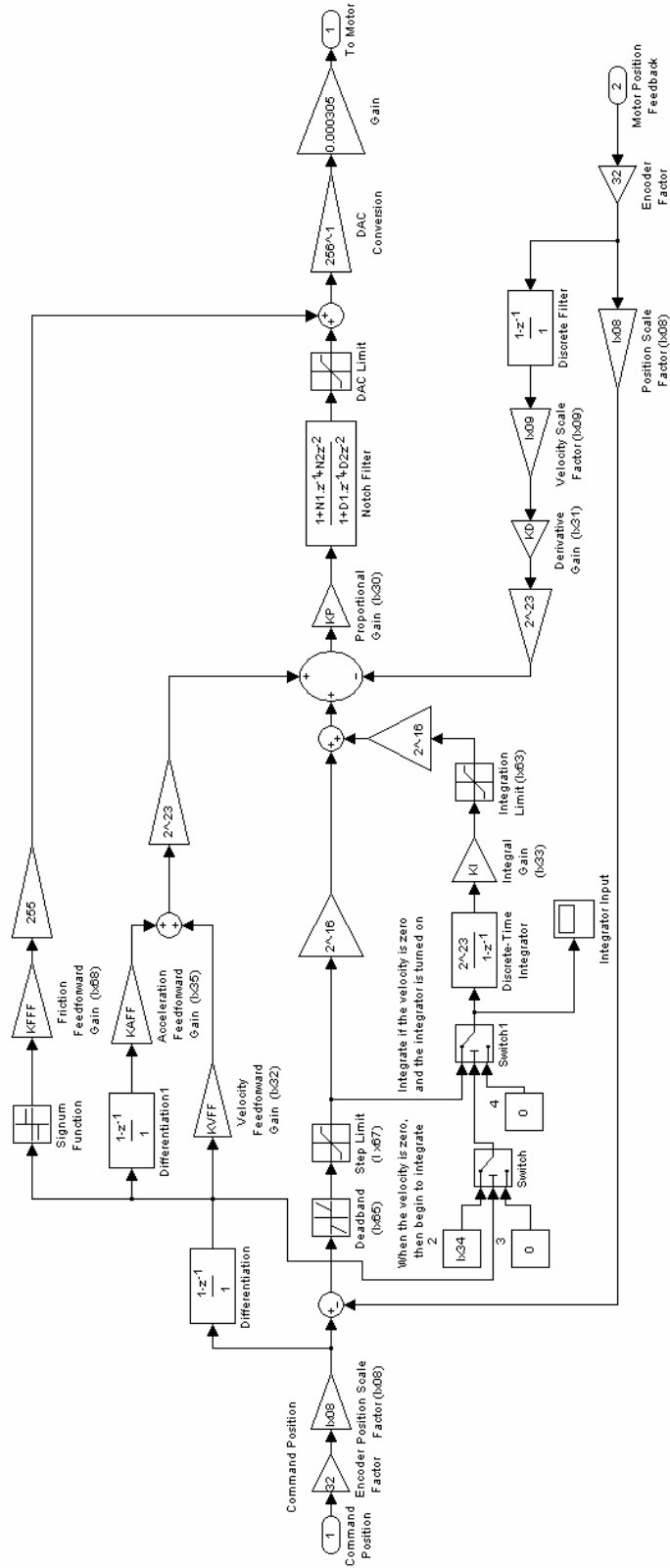


Figure 55: Diagram of PMAC

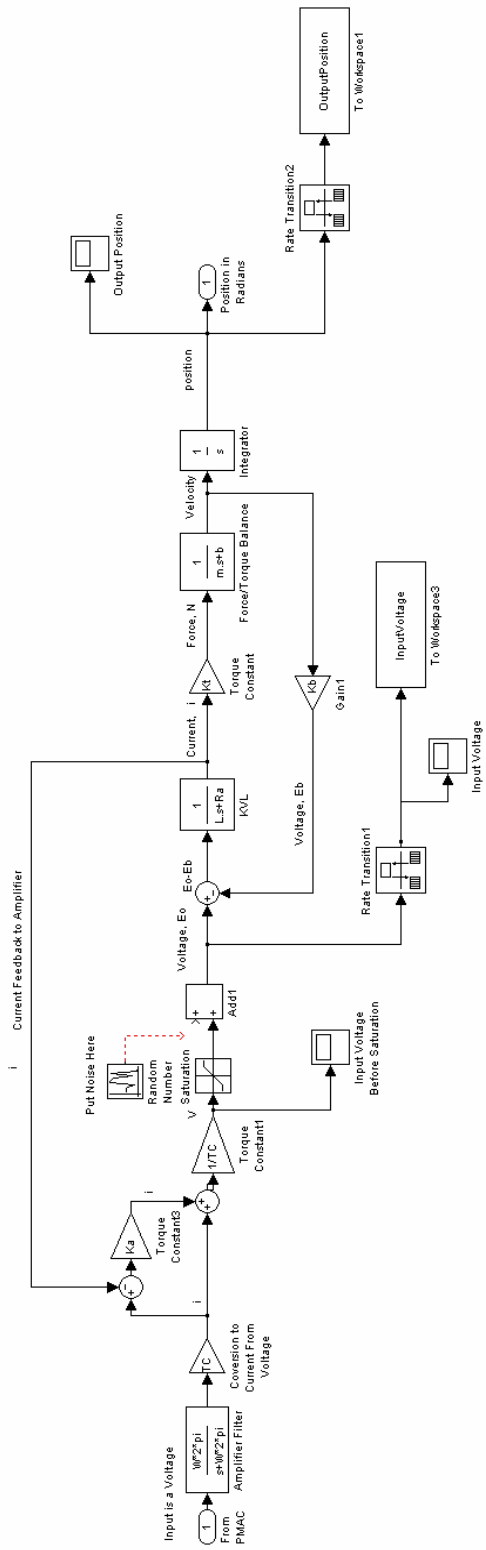


Figure 56: Diagram of motor model

```

% Calculation of poles and P,I,D gains
clear all
clc
syms f g h x Kps Kds Kis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Desired Poles
% Desired_Poles = [0 1 0.9912+0.00653i 0.9912-0.00653i f g h]
a = 0
b = 1
c = 0.9912+0.00653i
d = 0.9912-0.00653i
% These are the other calculated poles:
% roots: 0.00030453654173727358779964368097381
%         0.97469199955619182237116602553891
%         0.23448101090207090404103433078012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% These are the actual poles
% a = 0
% b = 0.999939038080093
% c = 0.98839786307770 + 0.01452941588949i
% d = 0.98839786307770 - 0.01452941588949i
% f = 0.98167666085888
% g = 0.23406217499354
% h = 0.00042652622514
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Pnml = (x-a)*(x-b)*(x-c)*(x-d)*(x-f)*(x-g)*(x-h)
collect(Pnml,x)
% This is the polynomial created from the four desired pole locations (roots)
% S = solve('(-1864/625-f-g-h) = -4.192777391',...
%       '(29649200809/10000000000+1864/625*f-(-1864/625-f)*g-(-1864/625-f-g)*h) = 0.1000000000e-
16*(0.2189103072e12*Kps+0.2189103072e12*Kps*Kis+0.6803819703e18+0.2189103072e12*Kps*Kds)',...
%       '(-9825200809/10000000000-29649200809/10000000000*f-
(29649200809/10000000000+1864/625*f)*g-(29649200809/10000000000+1864/625*f-(-1864/625-f)*g)*h)
= 0.1000000000e-16*(0.6472514509e12*Kps+0.8661617581e12*Kps*Kis-
0.5254778029e18+0.4283411437e12*Kps*Kds)',...
%       '(9825200809/10000000000*f-(-9825200809/10000000000-29649200809/10000000000*f)*g-(-
9825200809/10000000000-29649200809/10000000000*f-(29649200809/10000000000+1864/625*f)*g)*h) =
0.1000000000e-16*(-0.1299231825e13*Kps*Kds-
0.6519803747e12*Kps+0.2141813835e12*Kps*Kis+0.1869206515e18)',...
%       '(-9825200809/10000000000*f*g-(9825200809/10000000000*f-(-9825200809/10000000000-
29649200809/10000000000*f)*g)*h) = 0.1000000000e-16*(-0.2254707977e17+0.4377989910e12*Kps*Kds-
0.2141813835e12*Kps)',...
%       '9825200809/10000000000*f*g*h = 0.2141813835e-5*Kps*Kds')
% This is the IDENTIFIED TF Model denominator including amplifier
% 1.000000000*z^7
% -4.191877547*z^6
% +0.8000000000e-
16*(0.3034822324e11*Kps+0.3034822324e11*Kps*Kis+0.8501138251e17+0.3034822324e11*Kps*Kds)*z^5
% +0.8000000000e-16*(0.8970251888e11*Kps+0.1200507421e12*Kps*Kis-
0.6564309966e17+0.5935429565e11*Kps*Kds)*z^4
% +0.8000000000e-16*(-0.1800745452e12*Kps*Kds-
0.9037202632e11*Kps+0.2967871580e11*Kps*Kis+0.2334592912e17)*z^3
% +0.8000000000e-16*(-0.2815742638e16+0.6069331053e11*Kps*Kds-0.2967871580e11*Kps)*z^2
% +0.2374297264e-5*Kps*Kds*z

S = solve('(-1864/625-f-g-h) = -4.191877547',...
%       '(29649200809/10000000000+1864/625*f-(-1864/625-f)*g-(-1864/625-f-g)*h) = 0.8000000000e-
16*(0.3034822324e11*Kps+0.3034822324e11*Kps*Kis+0.8501138251e17+0.3034822324e11*Kps*Kds)',...
%       '(-9825200809/10000000000-29649200809/10000000000*f-
(29649200809/10000000000+1864/625*f)*g-(29649200809/10000000000+1864/625*f-(-1864/625-f)*g)*h)
= 0.8000000000e-16*(0.8970251888e11*Kps+0.1200507421e12*Kps*Kis-
0.6564309966e17+0.5935429565e11*Kps*Kds)',...
%       '(9825200809/10000000000*f-(-9825200809/10000000000-29649200809/10000000000*f)*g-(-
9825200809/10000000000-29649200809/10000000000*f-(29649200809/10000000000+1864/625*f)*g)*h) =
0.8000000000e-16*(-0.1800745452e12*Kps*Kds-
0.9037202632e11*Kps+0.2967871580e11*Kps*Kis+0.2334592912e17)',...
%       '(-9825200809/10000000000*f*g-(9825200809/10000000000*f-(-9825200809/10000000000-
29649200809/10000000000*f)*g)*h) = 0.8000000000e-16*(-0.2815742638e16+0.6069331053e11*Kps*Kds-
0.2967871580e11*Kps)',...
%       '9825200809/10000000000*f*g*h = 0.2374297264e-5*Kps*Kds')

Kps = S.Kps;
Kds = S.Kds;
Kis = S.Kis;
r1 = S.f
r2 = S.g
r3 = S.h

Kp = Kps*2^19
Kd = Kds*2^7
Ki = Kis*2^23

```



## REFERENCES

Åström, K. (1998). "Control of Systems with Friction". *Department of Automatic Control, Lund Institute of Technology*. Technical paper

Åström, K. J., and Eykhoff, P. (1971). "System Identification – A survey." *Automatica* **7**:123-162.

Åström, K. J. and T. Bohlin (1965). "Numerical Identification for linear dynamic systems from normal operating records." *IFAC symposium on Self-Adaptive Systems*.171-176.

Åström, K. J. and B. Wittenmark (1989). *Adaptive control*. Reading, Mass., Addison-Wesley.

Box, G. E. P., G. N. Jenkins, et al. (1994). *Time series analysis : forecasting and control*. Englewood Cliffs, N.J., Prentice Hall.

Cadwell, L. H. (1996). "Magnetic damping: Analysis of an eddy current brake using an airtrack." *American Journal of Physics* **64**:917–923.

Cooley, J. W. and S. W. Tukey (1965). "An algorithm for the machine calculation of Fourier series." *Math. Comput.* **19**:297-301.

Deistler, M. (2002). "System Identification and Time Series Analysis: Past Present and Future". *Stochastic Theory and Control*. B. Pasik-Duncan. Berlin, Springer. 97-108.

Ellis, G. (2004). *Control System Design Guide : A Practical Guide*. Amsterdam ; Boston, Elsevier Academic Press.

- Evans, W. R. (1948). "Graphical Analysis of Control Systems." *AIEE Transactions* **67**:547-551.
- Evans, W. R. (1950). "Control System Synthesis by Root Locus Method." *AIEE Transactions* **69**:66-69.
- Eykhoff, P. (1974). *System identification : parameter and state estimation*. London ; New York [etc.], Wiley-Interscience.
- Gauss, C. F. (1809). *Theoria motvs corporvm coelestivm in sectionibvs conicis solem ambientivm*. Hambvrgi,, Svmtibvs F. Perthes et I. H. Besser.
- Gevers, M. (2005). "Identification for control: from early achievements to the revival of experiment design." *European Journal of Control* **11**:1-18.
- Gevers, M. (2006). "A Personal View of the Development of System Identification". IEE Control Systems Magazine: 93-105
- Gevers, M. and L. Ljung (1986). "Optimal Experiment Design with Respect to the Intended Model Application." *Automatica* **22**:543-554.
- Ho, B. L. and R. E. Kalman (1965). "Effective construction of linear state-variable models from input-output functions." *Regelungstechnik* **12**:545-548.
- Hughes, S. B. (2000) Magnetic braking: finding the effective length over which the eddy currents form May 2000, Wooster, Ohio. Physics Department, The College of Wooster. Physics Junior Theses
- Juang, J.-N. and R. S. Pappa (1988). "A Comparative Overview of Modal Testing and System Identification for Control of Structures." *Shock and Vibration Digest* **20**:4-15.
- Kalman, R. E. (1960a). "A New Approach to Linear Filtering and Prediction Problems." *J. Basic Eng. Trans. ASME Ser. D.*:34-45.

Kalman, R. E. (1960b). "Contributions to the Theory of Optimal Control." *Boletín de la Sociedad Matemática Mexicana* **5**:102-119.

Kalman, R. E., Y. C. Ho, et al. (1963). "Controllability of Linear Dynamical Systems." *Contributions to Differential Equations* **1**:189-213.

Kurfess, T. R. (1996). "Ultra-high precision control". *The Control handbook*. W. S. Levine. Boca Raton, FL. New York, NY, CRC Press; IEEE Press. 1548 p.

Liaw, C. M., R. Y. Shue, et al. (2001). "Development of a linear brushless DC motor drive with robust position control". *Electric Power Applications IEEE Proceedings*. 111–118.

Ljung, L. (1999). *System Identification : Theory for the User*. Upper Saddle River, NJ, Prentice Hall PTR.

Ljung, L. and T. Glad (1994). *Modeling of dynamic systems*. Englewood Cliffs, N.J., PTR Prentice Hall.

Murray, R. M. (2003). "Control in an information rich world : report of the Panel on Future Directions in Control, Dynamics, and Systems". *Society for Industrial and Applied Mathematics*. 0898715288 (pbk.). 102 p.

Nelles, O. (2001). *Nonlinear system identification : from classical approaches to neural networks and fuzzy models*. Berlin ; New York, Springer.

Ogata, K. (1987). *Discrete-time control systems*. Englewood Cliffs, N.J., Prentice-Hall.

Ogata, K. (2002). *Modern control engineering*. Upper Saddle River, NJ, Prentice Hall.

Oppenheim, A. V., R. W. Schaffer, et al. (1999). *Discrete-time signal processing*. Upper Saddle River, N.J., Prentice Hall.

Oppenheim, A. V., A. S. Willsky, et al. (1997). *Signals & systems*. Upper Saddle River, N.J., Prentice Hall.

Rake, H. (1980). "Step and Frequency Response Methods." *Automatica* **16**:519-526.

Sodano, H., J. Bae, et al. (2004). "Improved Concept and Model of Eddy Current Damper." *ASME Journal of Vibration and Acoustics* **128**:294-302.

Stocia, P., T. Söderström, et al. (1985). "Optimal instrument variable estimates of the AR-parameters of an ARMA-process." *IEEE Trans. Automatic Control* **30**:1066-1074.

Van Overschee, P. and B. De Moore (1994). "N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems." *Automatica* **30**:75-93.

Wellstead, P. E. (1981). "Non-parametric methods of system identification." *Automatica* **17**:55-69.

Ziegler, J. G. and N. B. Nichols (1942). "Optimal Settings for Automatic Controllers." *ASME Trans* **64**: 759-68.

Ziegler, J. G. and N. B. Nichols (1943). "Process Lags in Automatic Control Circuits." *ASME Trans.* **65**:443-44.