



Room 14-0551  
77 Massachusetts Avenue  
Cambridge, MA 02139  
Ph: 617.253.5668 Fax: 617.253.1690  
Email: docs@mit.edu  
<http://libraries.mit.edu/docs>

## **DISCLAIMER OF QUALITY**

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available. If you are dissatisfied with this product and find it unusable, please contact Document Services as soon as possible.

Thank you.

**Dahl Friction Modeling**

by

Danielle Chou

Submitted to the Department of Mechanical Engineering  
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science

at the

Massachusetts Institute of Technology

June 2004

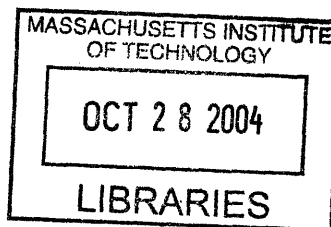
© 2004 Danielle Chou  
All Rights Reserved.

The author hereby grants permission to MIT to reproduce and publicly distribute  
paper and electronic copies of this thesis document in whole or in part.

Signature of Author: \_\_\_\_\_  
Department of Mechanical Engineering  
7 May 2004

Certified by: \_\_\_\_\_  
David L. Trumper  
Associate Professor of Mechanical Engineering  
Thesis Supervisor

Accepted by: \_\_\_\_\_  
Ernest Cravalho  
Chairman, Undergraduate Thesis Committee  
Department of Mechanical Engineering



ARCHIVES

# **Dahl Friction Modeling**

by

Danielle Chou

Submitted to the Department of Mechanical Engineering  
on 7 May 2004 in Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Mechanical Engineering

## **Abstract**

The drive behind improved friction models has been better prediction and control of dynamic systems. The earliest model was of classical Coulomb friction; however, the discontinuity during force reversal of the Coulomb friction model has long been a point of contention since such a discontinuity does not accurately portray the behavior of real systems. Other models have been suggested, but variations of the Dahl solid friction model remain some of the simplest yet most useful. Dahl's original theory proposed that friction behaved as a stress acting upon the quantum mechanical bonds at the interface. Thus, the relationship between frictional force and position would be analogous to a stress-strain curve, complete with hysteresis should there be permanent displacement akin to plastic deformation in materials. This project reviews the variations of Dahl friction models popular in the literature and then demonstrates it both analytically via Matlab and Simulink simulations and experimentally by observing the behavior of a limited angle torque motor.

Thesis Supervisor: David L. Trumper

Title: Associate Professor of Mechanical Engineering

## Table of Contents

1.0 Introduction.....	5
1.1 Purpose.....	5
1.2 Project Overview.....	5
1.3 Organization.....	5
1.4 Acknowledgements.....	5
2.0 Classical Friction Models: A History.....	6
2.1 The Coulomb Model.....	6
2.2 The Dahl Model.....	7
2.3 Dahl’s Formulation.....	8
2.4 The LuGre Model.....	9
2.5 Applications and Current Research.....	10
3.0 Test Apparatus.....	11
3.1 Tests by Computer Simulation.....	11
3.2 Tests by Hardware.....	13
3.2.1 Hardware.....	13
3.2.2 Simulink Real-Time Interface.....	15
4.0 Controller Implementation.....	17
4.1 System Requirements.....	17
4.2 Controller Design.....	17
5.0 Discussion of Experimental Results.....	18
5.1 Bode plots.....	18
5.2 Force-displacement Plots.....	19
6.0 Conclusions.....	21
References.....	22
Appendix A: Matlab script for digital signal analyzer.....	23
Appendix B: Simulink Simulation Model and Code.....	30
Appendix C: Matlab ODE45 Simulation Code.....	31
Appendix D: Motor Encoder Specifications.....	32
Appendix E: 2.737 Mechatronics Laboratory Assignment 4: Brushless Motor Control.....	38
Appendix F: Online Reference—K.J. Aström, “Friction Phenomena and Friction Models”.....	51

## List of Figures

2.1 Friction force versus sliding velocity—Stribeck curve.....	6
2.2 Bristle simulation of non-linear friction.....	7
2.3 Force-deflection curves while varying $i$ .....	8
2.4 LuGre model.....	10
3.1 Dahl modeled in Simulink.....	11
3.2 Dahl simulation result by Matlab ODE45.....	11
3.3 Dahl simulation result by Matlab ODE45 (varying time until force reversal).....	12
3.4 Dahl simulation result by Simulink (t=0 to 10,000).....	13
3.5 Hardware System Block Diagram.....	13
3.6 Schematic.....	14
3.7 Interface circuit for analog to digital conversion of output signals.....	14
3.8 Circuit for creating dSPACE reference.....	15
3.9 Computer control loop.....	15
4.1 Root locus a) before lead compensator, b) after lead compensator..	17
4.2 Bode plot for controller implementation.....	17
5.1 Position/Torque Bode plot.....	18
5.2 Amplifier Input/Position Bode plot.....	18
5.3 Input Sine/Position Bode plot.....	19
5.4 Torque-displacement at 10Hz.....	20
5.5 Torque-displacement at 1Hz.....	20

## **1.0 Introduction**

### **1.1 Purpose**

This paper discusses the Dahl friction model and subsequent non-linear friction models and their applications. The aims of this project are twofold: to explore and better understand the available literature on non-linear friction and to experimentally observe the said-effects in a Winchester limited-angle torque motor using Simulink as the control environment.

### **1.2 Project Overview**

This project involves observing Dahl frictional effects in a limited-angle torque motor. A controller is implemented so that displacements are within the encoder resolution while gain is minimized to maintain stability. Simulink's visual block diagram format is clear and concise and therefore used to create the control environment.

Two experimental parameters can be altered; these are frequency and amplitude of the input signal. This gives rise to two types of test data. The first is a swept-sine response focusing on gain and phase; control commands are given and responses recorded using the Matlab file `dsa_dahl.m` appended to this report. This set of experiments demonstrates the resonance owing to elastic deformation behavior at low amplitude displacements.

The aforementioned tests are conducted at various input amplitudes. A Bode plot of the swept-sine response exhibits shifts in the output gain as the input amplitude is altered. This clear amplitude dependence is further explored in the second set of data, in which force-displacement behavior at constant frequencies is compared for different input amplitudes.

### **1.3 Organization**

This report comprises two main themes: the theory behind solid friction and the experimental observation of it. Section 2 gives a brief history of friction models since Coulomb and an account of Dahl's solid friction theory as well as more recent modifications to and applications of the Dahl model. Section 3 discusses tests of the Dahl model in both a computer simulation environment as well as in the frictional behavior of a real system. This section also describes the hardware setup and the software used to conduct the tests. Section 4 covers the control loop of the system and briefly summarizes the choice and implementation of the controller. Section 5 analyzes the experimental data, and Section 6 relates features of the data to those described qualitatively by Dahl and other researchers. The computer code used to run the simulations and tests are included in the appendices.

### **1.4 Acknowledgements**

I am grateful to the many people who helped me with this project. Many thanks to Professor Trumper for his ever-helpful suggestions, advice, and encouragement of this endeavor; to Katie Lilienkamp, Matlab guru among many other things, who helped me tons by providing the code and helping me decipher it; to Evencio Rosales, whose project was a major launch point for mine and who made my job a lot easier; and to all the guys in the Precision Motion Control Lab—Augusto Barton, Marty Byl, Xiaodong Lu, Aaron Mazzeo, and Rick Montesanti—whose help prevented me from burning down the lab.

## 2.0 Friction Models: A Brief History

Friction phenomena are remarkably complicated, and despite the best efforts of many researchers, it continues to elude complete and comprehensive quantification. Different models better suit different set-ups under subtly different operating conditions. As one author wrote, “more than one servo functions because of a badly balanced fan in the vicinity”—an illustration of the sensitivity of frictional behavior to operating conditions such as position, velocity, surface finish, material temperature, contact geometry, lubricant viscosity, frictional memory (time delay between change in state and change in friction magnitude), and dwell time (time at zero velocity) [1].

Given the multiple applications and various analytical approaches taken to tackle the challenges of those applications, there are numerous proposed solutions to the modeling problem. These solutions may be categorized into four accepted friction regimes, as shown in Figure 2.1. The first regime is static friction/pre-sliding displacement, the second boundary lubrication (low velocity sliding, solid-to-solid contact), the third partial fluid lubrication (support by both solid asperity contact and fluid lubrication), and the fourth full fluid (hydrodynamic) lubrication [1]. This paper concerns itself primarily with the first regime of pre-sliding displacement, where fluid lubrication plays no role.

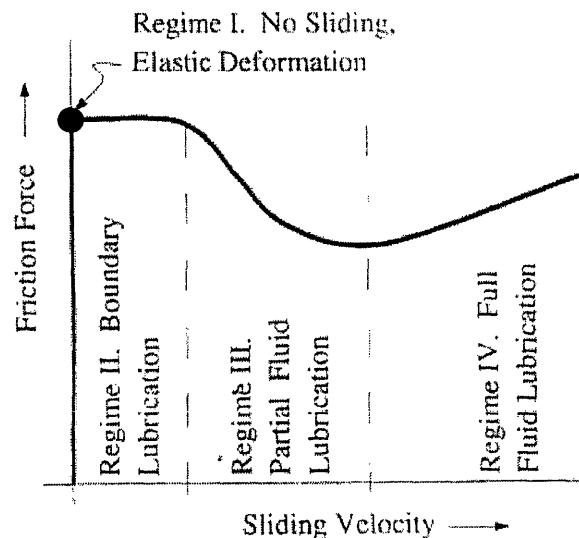


Figure 2.1: Friction force versus sliding velocity—Stribeck curve (from [1])

### 2.1 The Coulomb Model

For nearly 200 years, scientists modeled friction according to the 1785 description given by Coulomb—that there are two regimes of friction, static and kinetic. Static friction occurs at zero velocity when the force can take on any of a range of values. Kinetic friction occurs for non-zero velocities and is constant in magnitude and equal to the normal force multiplied by a constant coefficient times the sign of the velocity [1].

Coulomb’s model accurately portrays kinetic friction, but the transition from static to kinetic friction is unrealistic as the discontinuity at zero displacement implies an infinite rate of change of force that is not physically reasonable. Despite its incomplete description of frictional

behavior at low amplitude inputs, the model suffices for larger-amplitude sliding friction. A classic example is a block sliding down an inclined plane.

## 2.2 The Dahl Model

In 1968 P.R. Dahl proposed an alternative to the Coulomb friction model. While observing the behavior of ball bearings, he noted that very small amplitude input forces were reacted against by small elastic restoring forces. From this observation, he developed his theory of solid friction in which he describes friction as the macroscopic result of quantum mechanical bonds between two contact surfaces. Intermolecular bonds keep the surfaces together, but shearing or tensile forces cause them to break. This rupture behavior is demonstrated more familiarly in material deformation tests, and Dahl found that stress-strain behavior serves as a good analogy for the continuous friction force-displacement relationship [2].

Dahl likens the transition from static to kinetic friction to that of elastic to plastic deformation in ductile materials. The quantum mechanical bonds at a surface are strong enough at low tensile loads that the behavior is elastic and spring-like. Hence, upon relaxation of the load, the bonds return to their original, unstressed state. If the load is larger, the bonds experience permanent displacement analogous to plastic deformation, and if the load is then decreased, the bonds will not return to exactly their previous state. Therefore, the bonds experience hysteresis similar to that found in plastic material deformation. If the load is eventually increased beyond the strength of the bonds, the bond will break. This limit is analogous to the rupture stress of a ductile material or the ultimate tensile stress of a brittle one [2].

A commonly-used analogy to convey this idea is the bristle model; initially, an applied load will not move the intersection point of the bristle to the surface but merely deform the bristles elastically. If the load is removed, the bristles will return to their original positions. However, when the load exceeds that of only bristle deformation, the entire brush moves. The stiffness of the bristle is equivalent to the elasticity of the contact surfaces, and the non-linear behavior at the tip of the bristle is the point of interest [3].

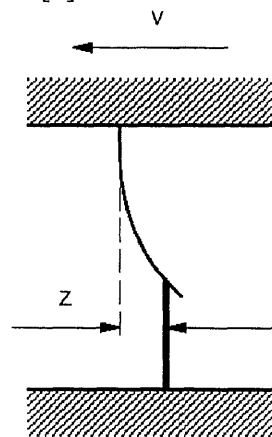


Figure 2.2: Bristle simulation of non-linear friction (from [5])

Dahl also addresses the issue of different types of friction. Rolling friction is modeled as alternating compressive and tensile stresses on the quantum mechanical bonds: compressive at



the front of the contact patch and tensile at the back. Likewise, sliding friction involves continuous shearing of bonds [2].

### 2.3 Dahl's Formulation

Dahl made his initial observation of non-linear friction effects while studying the behavior of ball bearings acted on by low amplitude input forces. At low amplitudes and low frequencies, ball bearings provide a certain amount of elastic resistance to input forces before they finally are permanently displaced. This elasticity results in hysteresis; at loads below  $F_c$  (sliding Coulomb friction), release of the load allows the balls to return to their initial position, yet if a larger load is applied, there will be a permanent displacement after the load is released. Dahl was interested in modeling this load  $F_c$  and the resulting elastic behavior, and he sought to include them in his friction formulation,

$$\frac{dF(x)}{dx} = \sigma \left| 1 - \frac{F}{F_c} \operatorname{sgn} \dot{x} \right|^i \operatorname{sgn} \left( 1 - \frac{F}{F_c} \operatorname{sgn} \dot{x} \right) \quad (1)$$

where  $\sigma$  is the slope of the force-deflection curve at  $F = 0$ ,  $F_c$  is the friction force at “yield” or the kinetic friction force, and  $i$  defines the order of the slope [4], as illustrated in Figure 2.3. Note that regardless of order, the slope of the force-displacement curve is always positive for positive  $\dot{x}$ . For  $0 \leq i \leq 1$ , the force-deflection behavior is similar to that of brittle materials. For  $i \geq 1$ , the behavior is more like that of ductile materials and therefore are of greater interest to modelers since it is precisely this non-linear behavior which is not accounted for by the Coulomb model.

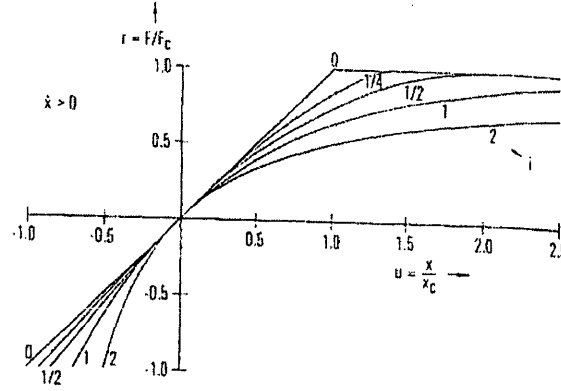


Figure 2.3: Force-deflection curves while varying  $i$  (from [2])

For the simplest case of  $i = 1$ , Dahl's equation may be written as the following time derivative [5]:

$$\frac{dF}{dt} = \frac{dF}{dx} \cdot \frac{dx}{dt} = \sigma \dot{x} - \frac{F}{F_c} \sigma |\dot{x}| \quad (2)$$

Notice that for  $F = F_c$ , the slope  $dF/dx = 0$ . Moreover, at force reversal, the force may change sign instantaneously, but the velocity will not. Thus, the rate of force change  $dF/dt$  at reversal may be twice as large as that at force initiation ( $F = 0$ ). At the steady state limit when  $F = F_c$ ,  $dF/dx$  goes to zero, the friction force  $F_c$  is constant thereafter, and Dahl's model becomes the Coulomb model.

Dahl also assumed that  $F = \sigma z$ , so the elastic deflection  $dz/dt$  may be calculated as:

$$\dot{z} = \frac{1}{\sigma} \cdot \dot{F} = \dot{x} - \frac{\sigma z}{F_c} |\dot{x}| \quad (3)$$

When there is close to no load,  $dz/dt = \dot{x}$ . To borrow the bristle analogy, all movement is contained within the bristle deformation. When the elastic limit of the bristle  $\sigma z$  reaches  $F_c$ , there is no longer any bristle velocity, i.e., the entire brush moves [5].

#### 2.4 The LuGre Model

The previous two models assume friction force is dependent on displacement only. Many models dispute that premise and emphasize the importance of velocity effects, which are especially dominant in the friction behavior of lubricated contact surfaces. The most basic velocity-dependence is viscous damping. Another experimentally observed effect is the Stribeck effect, as shown in Figure 2.1. Stribeck states that for zero velocity, the friction force is higher than for low, non-zero velocities [1]. Eventually, as the velocity increases, the friction force will likewise increase.

An analogous feature within the positional domain is stiction, a phenomenon in which the contact surfaces stick together because the load at which there is some initial displacement always exceeds the load required to maintain motion. This effect is not accurately portrayed by the Dahl model either. These shortcomings are addressed in the LuGre model, which was developed by Canudas de Wit, Olsson, and Aström [6]. The LuGre model expands on the Dahl model to accommodate the Stribeck effect. It is not the focus of this paper, and therefore only a cursory explanation will be given to acquaint the reader with this other relatively popular model.

The LuGre model modifies Dahl by assuming that friction force varies with not only bristle deflection  $z$ , but also brush and bristle deflection velocities  $\dot{x}$  and  $\dot{z}$ :

$$F = \sigma_o z + \sigma_1 \dot{z} + \sigma_2 \dot{x} \quad (4)$$

Like Dahl, the LuGre model couples bristle deflection velocity  $\dot{z}$  to the velocity  $\dot{x}$  of the entire brush. However, the relationship is less straightforward since the friction force  $F$  is longer equal to only  $\sigma z$ . The relationships  $dF/dx$  and  $dz/dF$  also need to be taken into account. One description of  $\dot{z}$  that successfully matches experimental curves is:

$$\dot{z} = \dot{x} - \frac{\sigma_o z}{g(\dot{x})} |\dot{x}| = \dot{x} - \frac{\sigma_o z}{F_c + (F_s - F_c) e^{-(\dot{x}/\dot{x}_s)^2}} |\dot{x}| \quad (5)$$

where  $F_s$  is the stiction (breakaway) force,  $F_c$  the Coulomb friction, and  $\dot{x}_s$  the constant Stribeck velocity.  $g(\dot{x})$  is only one possible description of the Stribeck curve.  $\sigma_o$ ,  $\sigma_1$ , and  $\sigma_2$  are the bristle stiffness, damping coefficient, and viscous coefficient, respectively [6].

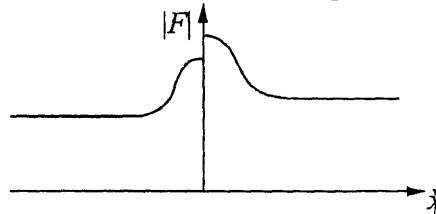


Figure 2.4: LuGre model (from [5])

As shown by Aström's force-velocity curve of Figure 2.4, the force required for initial motion is greater than for any other velocity. The asymmetry at  $\dot{x} = 0$  is commonly observed; it stems from the bristle deflection being recovered immediately after force reversal. At steady state, the LuGre model too becomes the Coulomb model.

## 2.5 Applications and Current Research

In the modern era of precision engineering, increasingly accurate descriptions of physical models are required in order to create better tracking and friction compensation, and much research has been done in this field within the past ten years. The ramifications of better friction modeling include better control over the dynamics of nano-scale motion control systems and higher fidelity haptics applications. These include machine tools, servomechanisms for telescopes and antennas, traction control in automobiles, and haptics actuators [5]. In all cases, the hysteretic nature of friction can result in unacceptable drift and even instability as the positional steady state errors accumulate.

Although the Dahl model is imperfect, it has nonetheless been a useful tool for estimating and compensating for friction. In 1990, Futami et al reported they could produce movement of nano-scale accuracy with simple ball bearings by using the Dahl and Coulomb friction models alone. Observation of such fine motion reveals there are three different regimes of friction behavior: regime I occurs for a displacement of less than 100nm and displacement behaves elastically as Dahl predicted; regime II, for displacements between 100nm and 100 $\mu$ m, is characterized by the bearings acting as non-linear springs and corresponds to Dahl's description of hysteretic plastic deformation; and regime III covers displacements greater than 100 $\mu$ m and is described by the traditional Coulomb friction in which force remains constant regardless of displacement [7].

Gene Franklin and Feei Wang of Stanford University and Hewlett-Packard, respectively, also explored a version of the Dahl model combined with viscous damping in their 1994 analysis of the movement of disk drive actuator pivots. They found that though the Dahl model is adequate for describing rolling and pre-rolling, it does not describe velocity reversal or behavior near zero displacement as well as other models such as the discontinuous two-slope spring [8].

Beginning in 1997, the Dahl model has also been applied toward haptics research. In 2000, Hayward and Armstrong proposed a modified version that was also strictly position-dependent. Their experimental results revealed that their new model attenuates the computational drift associated with Dahl's model [9].

Characterizing the fundamental behavior of friction at low frequencies and low amplitudes still remains a difficult feat, and many researchers continue to study and compare simulation results with experimental observations. A basic experiment consists of obtaining the frequency response for a system such as a motor. The response alone can reveal some of the interesting features of frictional behavior. This project takes that approach.

### 3.0 Test Apparatus

Testing is conducted via both computer simulation and experimentation. Simulations are based on the traditional Dahl model of 1968.

### 3.1 Tests by Computer Simulation

To determine the expected behavior of the system, simulations are first run using Matlab ODE45 and Simulink. Figure 3.1 illustrates the Simulink model, and Figures 3.2 through 3.4 show the simulation results. In the Simulink model, shown in Figure 3.1, the MATLAB function block rewrites Dahl's equation (1) as:

$$(s \cdot (\text{abs}(1-u(2)/F_c \cdot \text{sign}(u(1))))^i) \cdot u(1) \cdot \text{sign}(1-u(2)/F_c \cdot \text{sign}(u(1))) \quad (6)$$

where  $u(1)$  is the first input, velocity;  $u(2)$  is the second input, force; and  $s$  is equivalent to  $\sigma$ . The Simulink model is run with  $F_c=5$ ,  $s=10$ , and  $i=1$ .

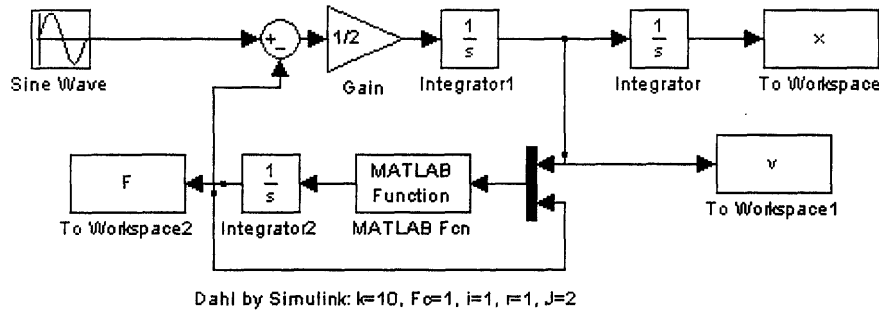


Figure 3.1: Dahl modeled in Simulink

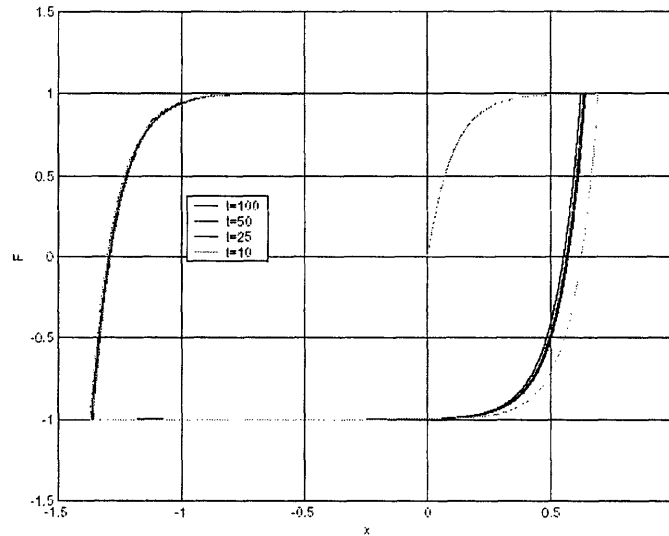


Figure 3.2: Dahl simulation result by Matlab ODE45

The Matlab ODE45 solver finds solutions for the equation

$$dF = (s \cdot \text{abs}(1-F/F_c \cdot \text{sign}(v)))^i \cdot v \cdot \text{sign}(1-F/F_c \cdot \text{sign}(v)) \quad (7)$$

by integrating in discrete steps. The model is run with initial conditions of zero position, zero velocity, and zero force and  $F_c=1$ ,  $s=10$ , and  $i=1$ . Further details of the simulation may be

referenced in Appendix C, and results from this ODE45 simulation are shown in Figures 3.2 and 3.3. The results predict a relatively steep approach to the  $F_c$  limit. Thus, prior to force reversal, the slope is nearly flat. Moreover, there is asymmetry about zero displacement. This is an artifact of the program in which velocity reversals are programmed to occur at every  $n^{\text{th}}$  time step. Figure 3.3 illustrates the effect of varying the number of time steps until force reversal. A large  $n$  means more time at the steady state reference value. If  $n$  is less than 1, velocity reversal occurs so quickly that the force-displacement curve remains along the linear section. The asymmetry arises because  $n$  time steps are allotted for allowing the force to increase to  $F$ , but thereafter, the force reversal requires a change of  $2|F|$  in the same  $n$  time steps.

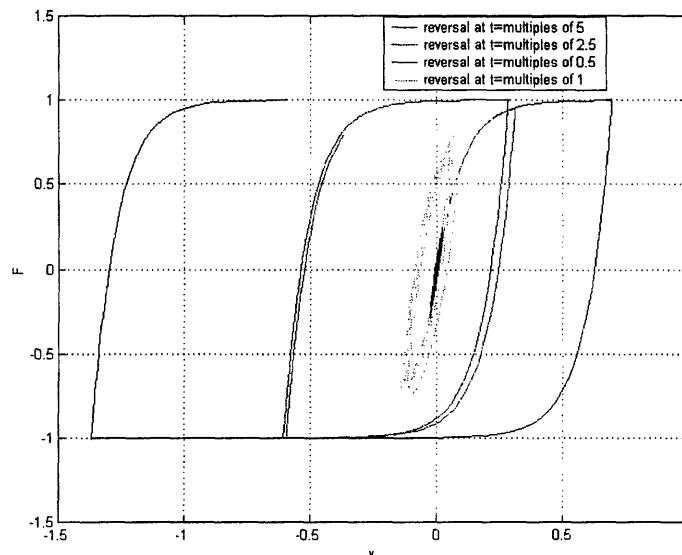


Figure 3.3: Dahl simulation result by Matlab ODE45 (varying time until force reversal)

The results of the Simulink model are shown in Figure 3.4. The hysteresis in Simulink's prediction is not as obvious. The torque-displacement slope is never flat, though it too eventually reaches a steady value. There is also a considerable shift toward the negative  $x$ -axis as time passes; this might be explained by the accumulation of errors due to the numerous integrators within the model. It is interesting to note that for initial data points (up to  $t=3000$ ), there are large hysteresis loops with no consistent maximum value. The output amplitude reaches a steady value only after about  $t=3000$  time steps.

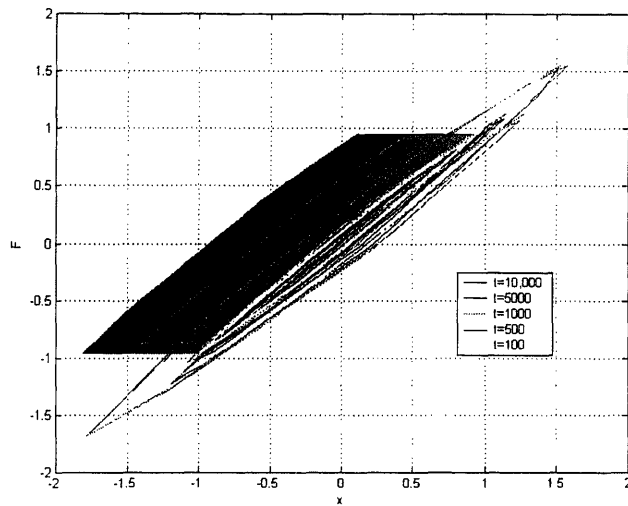


Figure 3.4: Dahl simulation result by Simulink (t=0 to 10,000)

### 3.2 Tests by Hardware

After computer simulations are run, hardware is tested to see how theoretical and experimental results compare. The test set-up is detailed in the ensuing paragraphs.

#### 3.2.1 Hardware

A Tektronix PS280 DC power supply provides 2A to the current amplifier. The amplifier, which has a gain of 2, in turn powers the motor. Motor encoder data is then relayed to the dSPACE board, which both makes control decisions that are sent to the amplifier and processes data to be sent to the computer for storage. The dSPACE control commands are created based on the computer program `dsa_dahl.m` downloaded to the board before testing; this program may be found in Appendix A.

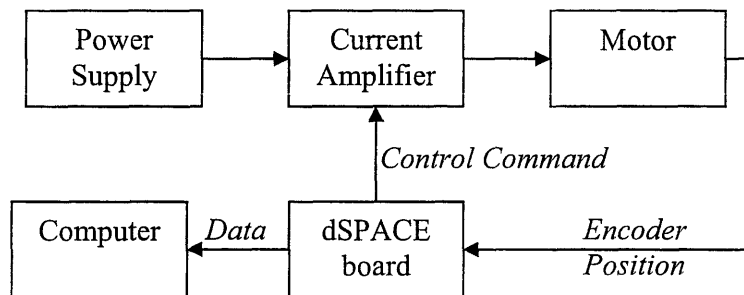


Figure 3.5: Hardware System Block Diagram

Tests are conducted on a limited angle brushless DC motor taken from a 5 ¼" Winchester disk drive. An encoder is encased on the backside of the motor; encoder resolution is 4400 quadrature states per revolution. Specifications are given for a motor operating at 25° C. Torque is generated over about ±60 degrees of motion only, and continuous torque is supplied at  $6.07 \times 10^{-2}$  N-m and peak torque at  $9.18 \times 10^{-2}$  N-m. The motor's torque constant  $K_t = 3.885 \times 10^{-2}$  N-m/A, its rotor inertia  $J = 1.2 \times 10^{-5}$  kg-m<sup>2</sup>, and its viscous damping constant  $b = 4.45 \times 10^{-4}$  N-m-s/rad. Since

the motor is current-driven by the amplifier, back emf and armature resistance are irrelevant at this level of description.

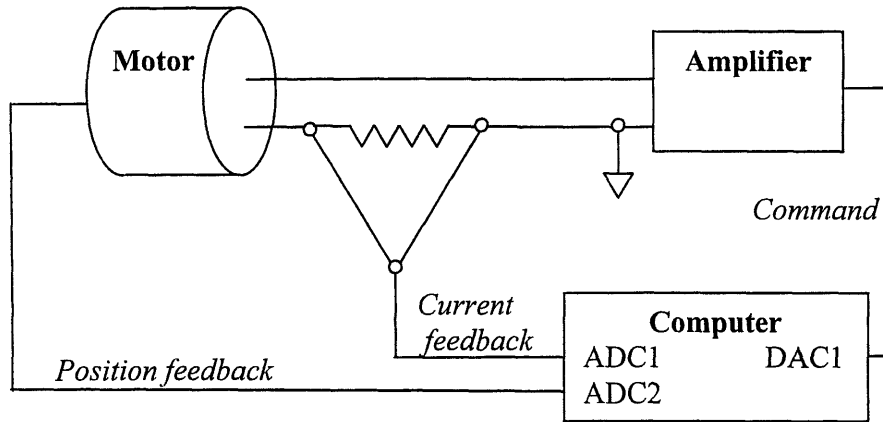


Figure 3.6: Schematic

A  $1\Omega$  power resistor is connected between the motor and the ground connection of the amplifier in order to determine the voltage drop and hence current flowing through the circuit. The resistance is considered small enough that it does not seriously impact the behavior of the rest of the circuit. The voltage drop across the resistor comprises the first output channel to the computer: ADC1 shown in Figure 3.6.

The second output channel is motor encoder position data, shown as ADC2 in Figure 3.6. The data coming out of the motor encoder is an analog combination of sine and cosine waves. In order to send a digital signal to the dSPACE encoder, these waves need to be squared up and presented at proper TTL levels. Two interface circuits are prepared, one each for the sine and cosine outputs. These are shown in Figure 3.7.

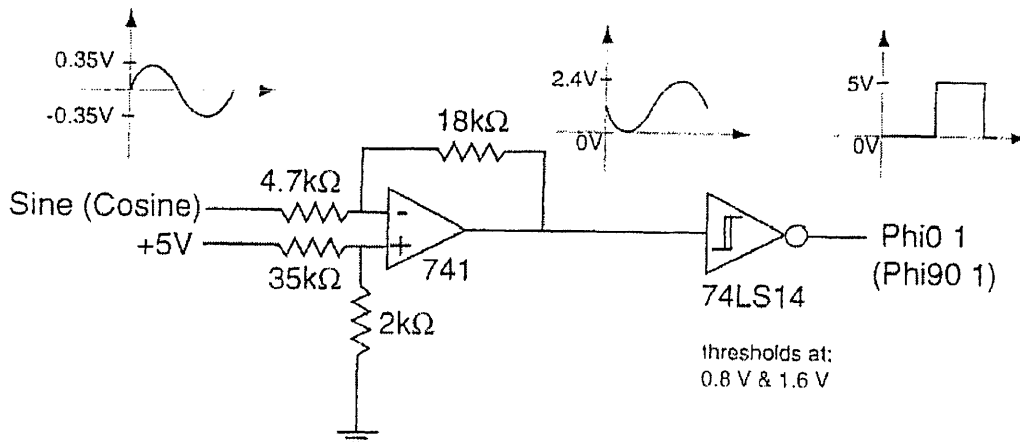


Figure 3.7: Interface circuit for analog to digital conversion of output signals (from Appendix E)

The dSPACE board also needs reference values with which to compare the digitized signals from the motor encoder. Thus, a 1.5V reference value and an index to locate the zero position of the encoder are provided to dSPACE encoder channel 2 by the circuit shown in Figure 3.8.

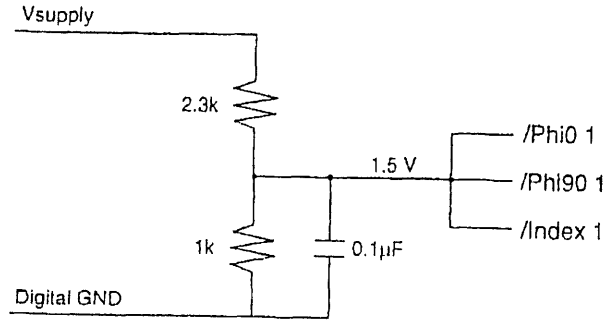


Figure 3.8: Circuit for creating dSPACE reference (from Appendix E)

A more detailed explanation of the motor specifications and circuitry is included in Appendices D and E.

### 3.2.2 Simulink Real-Time Interface

This project uses Simulink Real-Time Interface for a dSPACE 1102 real-time control board for implementation and observation of the motor. Simulink is preferred over other programs because of its convenient visual block diagram representation.

The control loop used in Simulink is shown in Figure 3.9. The dynamic signal analyzer of the control loop consists of two input channels that are sent to two displays. The output of the signal analyzer, Sine Out, is actually independent of incoming signals. Sine Out is merely a signal generator that outputs sine waves at incremental frequencies.

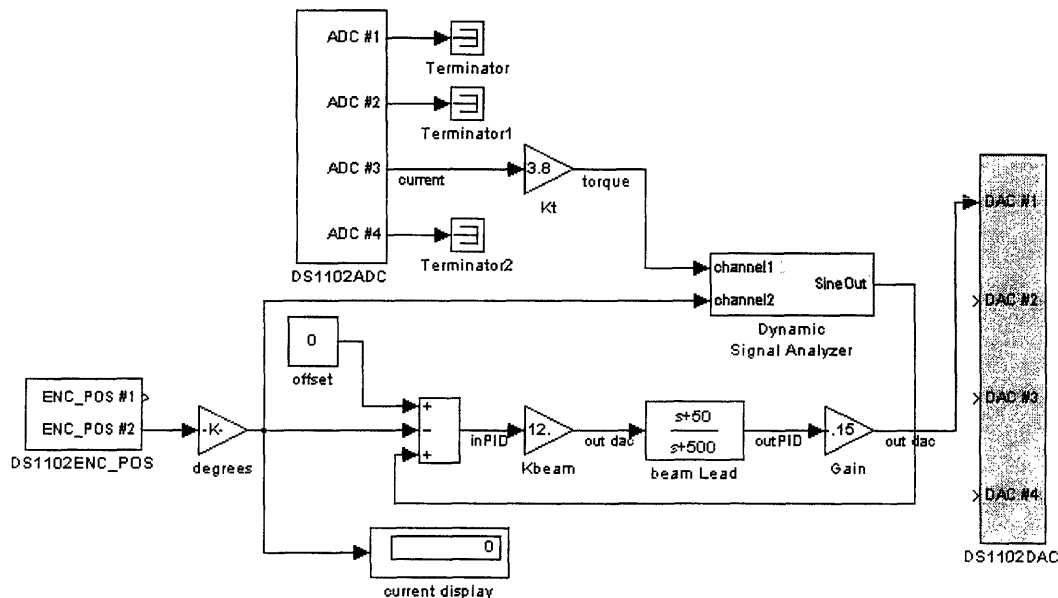


Figure 3.9: Computer control loop

The signal analyzer is run by a code adapted from a thesis by Katie Lilenkamp of MIT [10]. Two sets of data are collected simultaneously. The first is gain and phase for swept-sine



frequency responses with constant amplitude input. The second is the averaged input torques and corresponding output displacements for the swept-sine response. The experiment was run four times at input amplitudes of 0.5, 1.0, 3.0, 4.0, and 4.5; these are scaled by a factor of 0.1 to get the voltage, so the equivalent input voltages are 0.05, 0.10, 0.30, 0.40, and 0.45 volts.

At each frequency, there are over 9000 torque-displacement data points; when plotted on a force-displacement graph, these 9000+ points presumably retrace the same force-displacement trajectory several times. The repetition of data points results in increased processing time, so only the first 900 or so points at each frequency are retained for data analysis. Since the raw data remains rather noisy, each data point is averaged with the five points immediately following to obtain mean torques and displacements. This averaging is detailed in Appendix A.

## 4.0 Controller Implementation

A digital controller is required to maintain stability within the system.

### 4.1 System Requirements

The motor system has essentially only rotor inertia; therefore, the open loop transfer function is:

$$\frac{\Theta}{T} = \frac{1}{Js^2} \quad (6)$$

A study of the root-locus plot reveals the system controller requirements. The open loop system has two poles at the origin only. In order to improve stability, the two poles need to be drawn toward the left half-plane of the root-locus plot. Otherwise, an increase in gain can easily lead the poles to move unstably into the right half plane (shown in Figure 4.1a). This suggests a PD controller is advisable as it will introduce a zero.

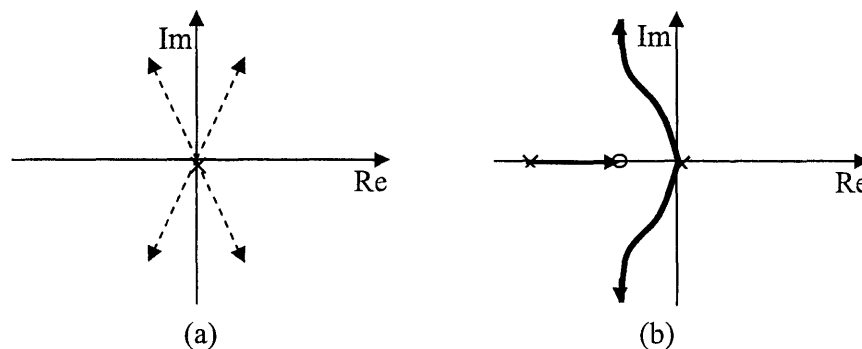


Figure 4.1: Root-locus a) before lead compensator, b) after lead compensator

### 4.2 Controller Design

A lead compensator with one zero with a breakpoint at  $\omega = 50$  rad/s and one pole with a breakpoint at  $\omega = 500$  rad/s is used. On the root-locus plot, this controller would draw the real component of the two poles at the origin toward the zero on the negative real axis, as illustrated in Figure 4.1b.

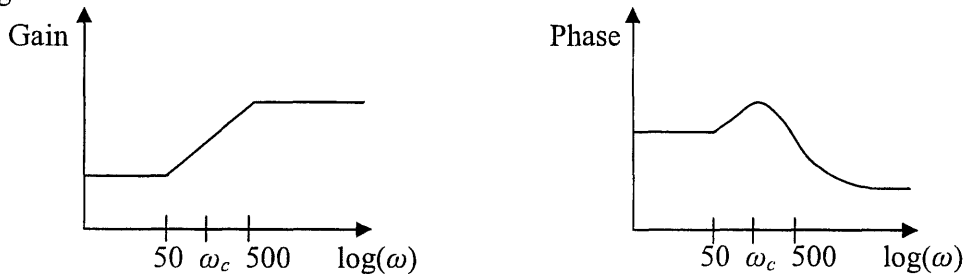


Figure 4.2: Bode plot for controller implementation

The compensator pole is necessary to limit the increase in magnitude introduced by the zero. This is most evident when examining the Bode plot of gain versus  $\log(\omega)$ . The phase plot is another indication of the utility of the lead compensator. Because of the zero, the phase margin increases and thereby increases the range of permissible crossover frequencies before instability. Instability is marked by phase exceeding 180 degrees of lag at the crossover frequency. A high crossover frequency exactly between the logarithms of the compensator gains is mildly desirable for the sake of expediency in experimentation.

## 5.0 Discussion of Experimental Results

### 5.1 Bode Plots

A study of the Bode plot in Figure 5.1 reveals break frequencies around 5Hz for input amplitudes of 1, 3, 4, and 4.5, and mild resonances occur at these frequencies. In contrast, for an input amplitude of 0.5, the break frequency is closer to 8Hz, and there is a significant resonant spike of about 15dB. This is possibly because 8Hz is likely the resonant frequency for the spring-like, pre-sliding elastic behavior of the contact surfaces.

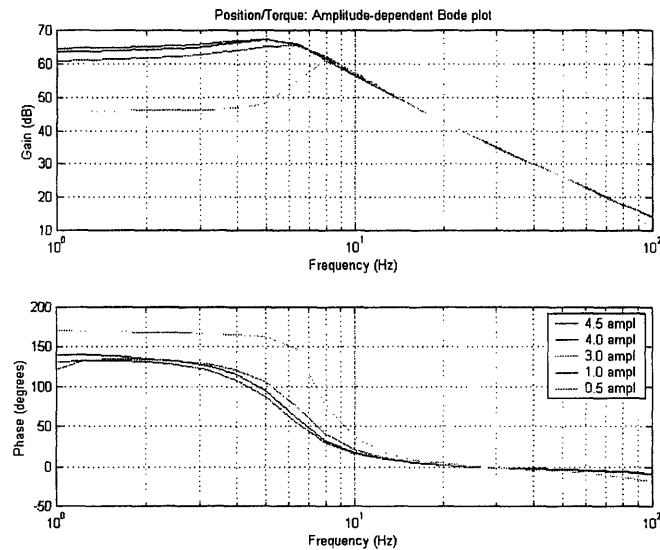


Figure 5.1: Position/Torque Bode plot

The trend seems to be a small increase in gain as the amplitude increases, but at low amplitude, there is a significant difference of 15dB in gain compared to the gain at higher amplitudes.

Amplitude-dependent behavior turns out to be relatively common for motors. Stephen Ludwick of MIT observed similar results in his experiments with an Anorad LEB-S-4 linear motor actuator [11]. His Bode plot is shown in Figure 5.2.

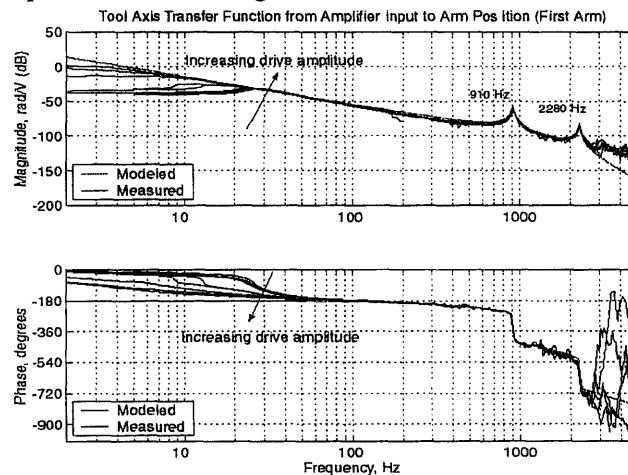


Figure 5.2: Amplifier Input/Position Bode plot (from [11])

The frequency response for an Aerotech BMS60 motor is shown in Figure 5.3. Again, larger input amplitudes drive the system closer to a free mass response while smaller amplitudes show the spring-like behavior.

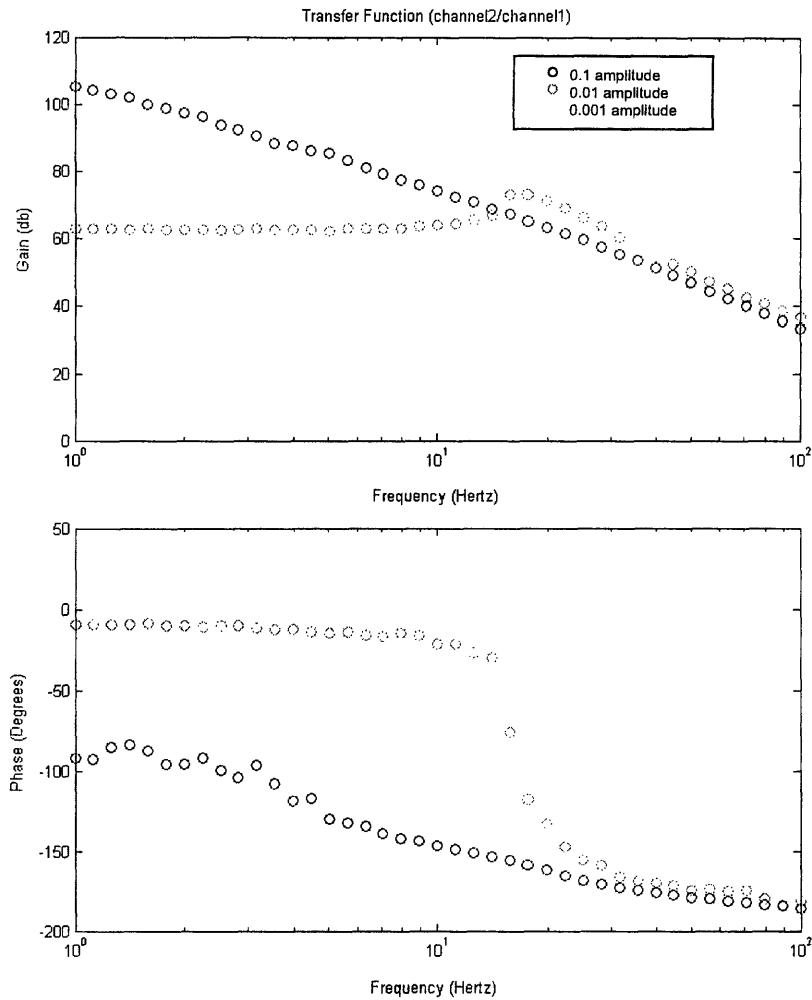


Figure 5.3: Input Sine/Position Bode plot

## 5.2 Force-displacement Plots

At frequencies of 1 and 10Hz, the torque-displacement graphs are analyzed to generalize about a few trends. The behavior at 10Hz is most similar to that predicted by Dahl, so this discussion will begin there.

The torque-displacement response at 10Hz is shown in Figure 5.4. The frequency is slightly above the breakpoint frequency, so rotor inertia begins to play a part. However, the 10Hz motor response is included in this discussion nonetheless because at lower frequencies, the encoder resolution is not adequate so the low frequency data are quite noisy. At a frequency of 10Hz and an input amplitude of 0.5, the output is still noisy, and the data trends are not apparent. However, by the time the amplitude is 3.0, the hysteretic behavior is obvious, and this conclusion is further confirmed by greater-amplitude inputs. At torque reversal, the displacement is greater than at

initial loading, and the curve is more or less monotonic as predicted by Dahl. The response is relatively symmetric about zero torque-displacement.

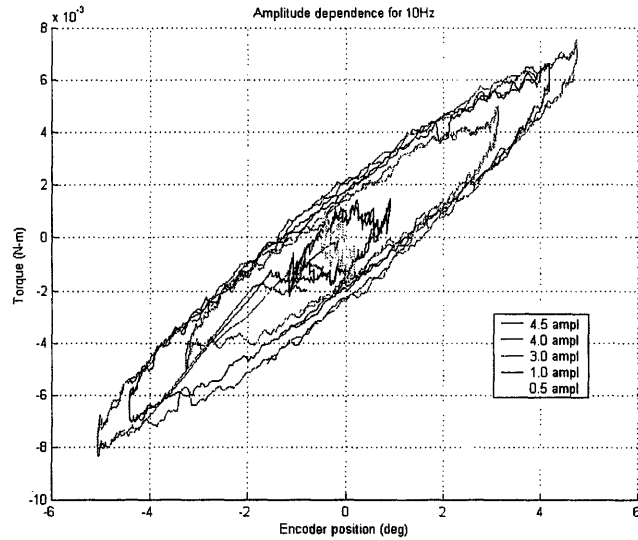


Figure 5.4: Torque-displacement at 10Hz

At 1Hz, illustrated in Figure 5.5, the signal is very noisy, but the trend of the slope is relatively consistent at various amplitude inputs. The response is also very asymmetric about zero torque-zero displacement, with positive displacements and negative torques favored. This may be attributable to an offset in the zero-value for current, but this bias toward positive torque-positive displacements is not seen in responses to higher amplitude inputs. There is also the possibility of anisotropy in the bearings; anisotropy has been known to result in different friction forces for different directions of motion [1], but in these experiments, there is no asymmetry in higher frequency responses. Also highly unusual is the negatively-sloped trend of the curves. This is not in accord with Dahl's proposal that the slopes of torque-displacement curves are always positive, but the experimental response is likely affected by limited encoder resolution.

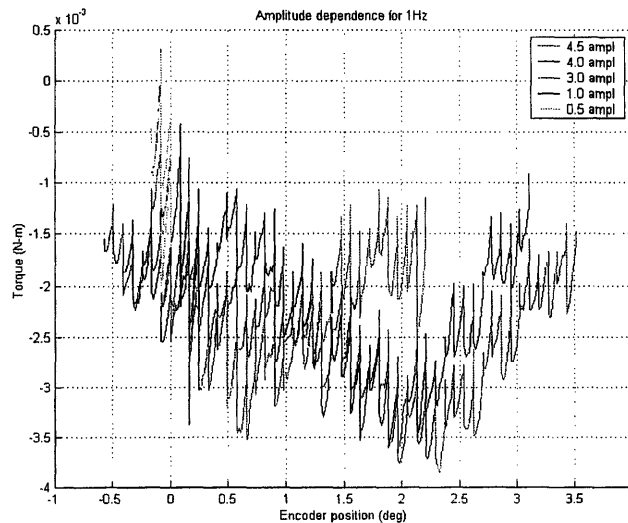


Figure 5.5: Torque-displacement at 1Hz

## 6.0 Conclusions

Dahl's model, in which position is the single state, incompletely describes friction under all conditions, but it nonetheless is a useful tool for improving friction compensation. As expected, experimental responses to low frequency and low amplitude input are noisy and difficult to qualify. Still, certain trends can be ascertained, and a comparison of the computer simulations and hardware experimentation shows that Dahl's model has both its strengths and weaknesses.

Among its competencies is its account of the hysteresis and force-reversals present in small-amplitude friction. It is interesting that the Wang et al study criticized Dahl's model for its inaccuracy during velocity reversals and for its greater displacement compared to experimental results. Wang et al analyzed a Dahl-plus-viscosity model, and their experimental force-displacement hysteresis curves had very similar slopes/stiffnesses in both directions of loading. Thus, the curves looked more like parallelograms. In comparison, their theoretical curves were more elliptical [8]. For this project, the Dahl model seems to be more sufficient; experimental curves are more elliptical and similar to Dahl's predictions. However, at velocity reversals, the trend of the experimental data tends to agree with Wang's generalization that Dahl's prediction is not accurate: unlike Dahl's model in which displacement immediately decreases as the load is released, experimental data suggest the displacement behavior lingers at the instant of velocity reversal. However, because this observation is made at a frequency slightly above resonance, it is difficult to ascertain whether it is strictly a mass-dependent effect or is partially an artifact of the elastic deflection within the intermolecular contacts.

The increase in gain and slightly lower breakpoint frequencies at higher input amplitudes may be due to the Stribeck effect; at very low frequencies, the frictional force is greater due to stiction, so the gain of displacement to input torque will be lower. The negative slope of force-displacement at low frequencies is another very odd feature, but data are debatable at such low amplitudes given the limitations of encoder resolution. Such challenges confirm that friction due to low amplitude and low frequency inputs is difficult to characterize.

Despite such shortcomings and decades of research into more complex models, the Dahl model survives. With the tradeoff between accuracy and simplicity, the attraction of single-state models cannot be underestimated, and the search for a simple, all-encompassing theory of friction, however unlikely, continues.

## References

- [1] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit, "A Survey of Models, Analysis Tools, and Compensation Methods for the Control of Machines with Friction," 1994, *Automatica*, Vol. 30, No. 7, pp. 1083-1138.
- [2] P.R. Dahl, "A Solid Friction Model," TOR-0158 (3107-18-1), May 1968, The Aerospace Corporation, El Segundo, CA.
- [3] P. Dupont, V. Hayward, B. Armstrong, "Single State Elasto-Plastic Models for Friction Compensation," December 1999, *IEEE Transactions on Automatic Control*.
- [4] P.R. Dahl, "Solid Friction Damping of Mechanical Vibrations," December 1976, *AIAA Journal*, Vol. 14, pp. 1675-1682.
- [5] K.J. Aström, "Friction Phenomena and Friction Models," [Online document], 1998, [cited April 2004], Available HTTP: [http://www-lag.ensieg.inpg.fr/canudas/publications/friction/\(2\)CCA\\_Models.ps](http://www-lag.ensieg.inpg.fr/canudas/publications/friction/(2)CCA_Models.ps)
- [6] J. Wang, S. S. Ge, and T.H. Lee, "Adaptive NN Control of Robot Manipulator with Unknown Dynamic Friction," June 2001, *Asian Conference on Robotics and Its Applications*.
- [7] S. Futami, A. Furutani, and S. Yoshida, "Nanometer Positioning and Its Micro-Dynamics," March 1990, *Nanotechnology*, Vol. 1, pp. 31-37.
- [8] F. Wang, T. Hurst, D. Abramovitch, and G. Franklin, "Disk Drive Pivot Nonlinearity Modeling Part II: Time Domain," June 1994, *Proceedings of the 1994 American Controls Conference*.
- [9] V. Hayward and B. Armstrong, "A New Computational Model of Friction Applied to Haptic Rendering," 2000, *IEEE Transactions on Automatic Control*.
- [10] K.A. Lilienkamp, "A Simulink-Driven Dynamic Signal Analyzer," February 1999, MIT mechanical engineering department bachelor's thesis.
- [11] S. Ludwick, "Calibration and Control of a Rotary Fast Tool Servo," October 1999, *Proceedings of ASPE Fourteenth Annual Conference*.

## Appendix A: Matlab file dsa\_dahl.m

```
function [mytf] = dsa_tf3(w_list,amp_list,sval,datestamp)
% This program is a new version of dsa_tf2.m modified to analyze friction
% modeling data. 5-5-04 DC
%
% function [mytf] = dsa_tf3(w_list,amp_list,sval)
% w_list : optional vector of INPUT FREQUENCY values at which to find TF
% amp_list : optional vector of INPUT AMPLITUDE values at which SINE
%             output will sweep. (If amp_list is a scalar, the same
%             amplitude will be used at ALL frequencies...)
% sval : optional string value for the plot (e.g. 'r*' would plot
%        red '*'s at points on figure(1))
% datestamp: optional 4th argument (not shown above).
%            If a non-zero 4th argument is used in calling the
%            function dsa_tf, the program will 'time stamp' the
%            output figure. This will NOT WORK unless you have
%            the time-stamping MATLAB function 'get_date.m'
%            as well, however!!!
%-----
% OUTPUTs: The output is an (N x 3) matrix. 'N' is the length of w_list.
% Column 1: Returned FREQUENCY list (w_list values)
% Column 2: GAIN, as the ratio of channel2/channel1 (not in db)
% Column 3: PHASE, in degrees
%-----
% NOTES:
% (1) You must be running a SIMULINK model on the ds1102 board
%     for the MATLAB function dsa_tf() to work, and the simulink
%     model must include a special block called 'Dynamic Signal Analyzer'
% (2) This program will overwrite figures 1 and 2 (in matlab)!!
%     Before running dsa_tf(), make sure you do not have images/plots
%     in either figure which should not be destroyed.
%-----
% Section 0: Define number of samples and default siggen amplitude.
if ~exist('datestamp')
    do_date=0;
else
    do_date=(datestamp ~= 0);
end
N=10000;      % N: number of points to sample.
w_rad=0;      % indicates frequencies are NOT in rad/sec by default (Hz
default)
dsa_A=.1;     % default AMPLITUDE of Swept Sine from DSA
exitval=0;    % program will exit if this is non-zero.
%-----
% Section 1: Make sure all frequencies and amplitudes are set for this run:
fprintf(1,'\nNote: Make sure your (DSA) model is built and running.\n');
fprintf(1,'          This program will erase any images or plots in figure 1 and
figure 2.\n\n');
fprintf(1,'          If you would like to STOP this program now, enter 'q' to
quit,\n');
isok=input('          otherwise, just hit enter to continue : ','s');
if strcmpi(isok,'q')
    fprintf(1,'\nOK, bye.\n')
else
```



```

if ~exist('sval') % default symbol for Bode plot
    sval='bo-';
end
if ~exist('w_list')
    w0=1:(1/10):3.0;
    w_use=10.^w0;
    fprintf('\nRunning %d points.\n [Range = %.2f [Hz] (min) to %.2f [Hz]
(max)] \n',length(w_use),min(w_use),max(w_use));
else
    is_rad=input('\nThe list of frequencies you entered was in:\n HERTZ (h)
or RAD/SEC (r) [default to Hertz]? ', 's');
    if strcmpi(is_rad,'r')
        fprintf('\n--> using RAD/SEC\n');
        w_use=(1/(2*pi))*w_list;
        w_rad=1;
    elseif strcmpi(is_rad,'h')
        fprintf('\n--> using HERTZ\n');
        w_use=w_list;
    else
        fprintf('\n...hmmm, I don''t understand, so I''m going to use HERTZ
(by default).\n');
        w_use=w_list;
    end
end
end
if ~exist('amp_list')
    A=dsa_A; % initial SINE WAVE amplitude
    amp_list=A+(0*w_use);
else
    if length(amp_list)<length(w_use)
        fprintf('Using %.4f as SINE WAVE amplitude as ALL
frequencies...\n');
        amp_list=amp_list(1)+(0*w_use);
        A=amp_list(1);
    end
end
end
%-----
% Section 2: Get dSPACE board parameter addresses with mlib().
mlib('SelectBoard','ds1102');
mtrc31('SelectBoard','ds1102');
amp_addr=mlib('GetSimAddr','P[Model Root/Dynamic Signal Analyzer/Swept
Sine.Amplitude]');
freq_addr=mlib('GetSimAddr','P[Model Root/Dynamic Signal Analyzer/Swept
Sine.Frequency]');
phi_last=0; B=1;
%-----
% Section 3: Begin outputting the frame for a TABLE to the matlab screen.
fprintf('\n----- :: ----- \n');
fprintf('_____Frequency_____ _SineAmp__ :: __GAIN__
_PHASE_\n');
fprintf('    Hertz          [rad/sec]          ::          db
Degrees\n');
fprintf('----- :: ----- \n');
%-----
% Section 4: Run through all requested FREQUENCY values; find gain and
phase at each.
f1=figure(1); set(f1,'Position',[465,210,450,520]);
f2=figure(2); clf; set(f2,'Position',[10,210,450,520]);

```

```

patch([-0.5 1.5 1.5 -0.5 -0.5],[-0.5 -0.5 1.5 1.5 -0.5],[0 0 0])
patch([0 0 1 1 0],[0 1 1 0 0],[0 1 0])
axis([-0.5 1.5 -0.5 1.5])
t1=text(.5,.65,'COLLECTING'); set(t1,'HorizontalAlignment','center');
t1=text(.5,.5,'FIRST'); set(t1,'HorizontalAlignment','center');
t1=text(.5,.35,'DATA SET...'); set(t1,'HorizontalAlignment','center');
axis off
u1=icontrol(2,'Position',[10 10 150 30],...
            'String','HIT to BREAK',...
            'Callback','stopval=1;',...
            'Enable','on','Value',5);

i=1;
while (i<=length(w_use) & (exitval==0)
    w=w_use(i); % frequency for this data set
    A=amp_list(i); % NOTE: amplitude must be 'reasonable'...
                    % USER must visually check that OUTPUT is not
saturated!!!
    fprintf('%9.4f      [%10.4f]      %9.6f      :: ',w,2*pi*w,A);
    drawnow;
    mlib('WriteD',freq_addr,w); mlib('WriteD',amp_addr,A);
    tic % give system some time to
settle...
    while (toc<(10/w) & (get(u1,'Value')>1)
        drawnow
    end
    %-----
    % Section 4-1: Here is the actual procedure to get gain and phase at
    % a particular frequency. This would be more elegantly implemented
    % as a separate function. Instead, it is included within this loop
    % so that the dynamic signal analyzer can be run from a SINGLE FILE.
    % (Otherwise, the user has to worry about having all files needed.)
    A=mlib('Readd',amp_addr); w=mlib('Readd',freq_addr);
    T=1/w;
    y_addr=mtirc31('GetAddr','rti B[Model Root/Dynamic Signal
Analyzer/channel1]',...
                  'rti B[Model Root/Dynamic Signal Analyzer/channel2]',...
                  'rti B[Model Root/Dynamic Signal Analyzer/Swept Sine]');
    mtrc31('TraceVars',y_addr);
    samp_per=mlib('GetSimAddr','Task Info/Timer Task 1/sampleTime');
    dt=mlib('Readd',samp_per);

    if w>10 % by T.SATO 12/10/99
        ds=1;
    else
        ds=ceil(1/w)+1; % by T.SATO 12/10/99
    end

    ncyc=floor(w*dt*ds*N); % by T.SATO 12/10/99; Use INTEGRAL # of sine
waves!!
    Nlast=round(ncyc/(w*dt*ds)); % by T.SATO 12/10/99

    if ncyc>10 % Display no more than this # of sine waves
        Nplot=round(10/(w*dt*ds));
        % by T.SATO 12/10/99, for the user to observe during the run...
    else
        Nplot=Nlast;

```

```

end
tic
while (toc<2) & (get(u1,'Value')>1) % settle time pause...
    drawnow
end
if (get(u1,'Value')>1)
    % !!!! set frame in desired way!!!
    mtrc31('SetFrame', [], ds, 0, Nlast*ds); % by T.SATO 12/10/99
    mtrc31('SetTrigger', y_addr(3,:), 0, 1);
    mtrc31('LockProgram');
    mtrc31('StartCapture');
    while mtrc31('CaptureState')~=0
        drawnow;
    end
    my_data=mtrc31('FetchData');
    % Take an INTEGRAL number of sine waves, total:
    k=[0:(Nlast-1)];
    y_out=my_data(2,1:Nlast);
    y_in=my_data(1,1:Nlast);
    t_out=dt*ds*[1:Nlast];
    mtrc31('UnlockProgram');
end

for h=1:length(y_out)*0.1
    torq(h)=y_in(h);
    pos(h)=y_out(h);
end

for k=1:length(torq)-4
    avgTorq(k)=(torq(k)+torq(k+1)+torq(k+2)+torq(k+3)+torq(k+4))/5;
    avgPos(k)=(pos(k)+pos(k+1)+pos(k+2)+pos(k+3)+pos(k+4))/5;
end

avgData=[avgPos; avgTorq];
amplitude=A*10;
fname=['torqPos' num2str(w, '%3.0f') 'hz' num2str(amplitude) 'ampl.m'];
fid=fopen(fname, 'w');
fprintf(fid, '%8.6f %9.6f\n', avgData);
fclose(fid);

set(u1, 'Enable', 'off'); % do not allow break until new data presented
if (get(u1, 'Value')<1)
    i=max(1, (i-1));
    w=w_use(i); % frequency for this data set
    A=amp_list(i); % NOTE: amplitude must be 'reasonable'...
    % USER must visually check that OUTPUT is not saturated!!!
    figure(2);
    fprintf(1, '\nBREAK: going back to previous frequency.\n');
    fprintf(1, ' * Use COCKPIT to reset Amplitude.\n');
    fprintf(1, ' * USE TRACE to view resulting sine waves.\n');
    fprintf(1, 'Restart DSA by hitting RESTART button in figure 2.\n');
    drawnow;
    mlib('Writed', freq_addr, w); mlib('Writed', amp_addr, A);
    startval=0; exitval=0;
    subplot(3,1,3); cla;
    patch([0 30 30 0 0], [2 2 7 7 2], [0 1 0]);
    axis off; axis([0 30 2 7]);

```

```

text(.5,6,'You can use COCKPIT and TRACE to reset');
my_str=num2str(w_use(i),'%.1f');
text(.5,5,['Amplitude and then CONTINUE at ' my_str ' Hz...']);
text(.5,4,'...or you can EXIT to completely rerun DSA. ');
text(.5,3,['choose EXIT or CONTINUE']);
set(u1,'String','HIT to EXIT','Callback','exitval=1;');
u2=uicontrol('Position',[200 10 150 30],'Value',5,...
    'String','CONTINUE','Callback','startval=1;');
set(u1,'Enable','on','Value',5);
while (get(u1,'Value')~=0) & (get(u2,'Value')~=0)
    drawnow; % wait for user to hit a button...
end
if get(u1,'Value')==0
    exitval=1; % exit the dsa
else
    startval=1; % restart the dsa
    A=mlib('Readd',amp_addr);
    amp_list=A+(0*amp_list);
    % reset amp_list for current and future freqs
    figure(2); clf;
    patch([-0.5 1.5 1.5 -0.5 -0.5],[-0.5 -0.5 1.5 1.5 -0.5],[0 0 0])
    patch([0 0 1 1 0],[0 1 1 0 0],[0 1 0])
    axis([-0.5 1.5 -0.5 1.5])
    t1=text(.5,.65,'COLLECTING');
    set(t1,'HorizontalAlignment','center');
    t1=text(.5,.5,[my_str ' Hz']);
    set(t1,'HorizontalAlignment','center');
    t1=text(.5,.35,'DATA SET...');
    set(t1,'HorizontalAlignment','center');
    axis off
    u1=uicontrol(2,'Position',[10 10 150 30],...
        'String','HIT to BREAK',...
        'Callback','stopval=1;',...
        'Enable','on','Value',5);
end
else
    % user did NOT request a break, so analyze this data set:
    if exist('y_out')
        if (max(y_out)<0.95) & (min(y_out)>-.95) & (max(y_in)<0.95) &
(min(y_out)>-.95)
            warncolor=[1 1 0]; % Amplitude not 'saturated.' Data OK to use
        else
            warncolor=[1 0 0];
        end
        y_sin=((w*2*pi)*t_out); % Ideal SINE at this freq
        Bc=(2/(length(y_out)))*sum((y_out).*cos(y_sin));
        Bs=(2/(length(y_out)))*sum((y_out).*sin(y_sin));
        B_=sqrt(Bc^2+Bs^2); % Amplitude of OUTPUT at this freq
        Ac=(2/(length(y_out)))*sum((y_in).*cos(y_sin));
        As=(2/(length(y_out)))*sum((y_in).*sin(y_sin));
        A_=sqrt(Ac^2+As^2); % Amplitude of INPUT at this freq (check)

        Gain=B_/A_; % TF of torque to position

        phi_out=atan2(Bc,Bs);
        phi_in=atan2(Ac,As);
        phi=phi_out-phi_in; % Difference in phase (input->output) (rad)
    end
end

```

```

figure(2); clf
subplot(311)
plot(t_out(1:Nplot),y_out(1:Nplot),'r. ');
hold on; grid on; title('RED: channel2'); %xlabel('seconds')
subplot(312)
plot(t_out(1:Nplot),y_in(1:Nplot),'b. ');
hold on; grid on; title('BLUE: channel1 (input)');
xlabel('seconds')
subplot(313)
patch([0 30 30 0 0],[2 2 7 7 2],warncolor);
axis off; axis([0 30 2 7]);
text(.5,6,'Check that the OUTPUTS above are : ');
text(1.5,5,'* NOT SATURATED. ');
text(1.5,4,'* Not buried in noise. ');
text(.5,3,'otherwise, BREAK and run with new AMPLITUDES. ');
ui=icontrol(2,'Position',[10 10 150 30],...
    'String','HIT to BREAK',...
    'Callback','stopval=1;',...
    'Enable','on', 'Value',5);
stopval=0; % insure 'stopval' is reset to show 'OK' state
end
if phi>phi_last
    while (phi-phi_last)>(pi)
        phi=phi-(2*pi);
    end
else
    while (phi-phi_last)<(-pi)
        phi=phi+(2*pi);
    end
end
mytf(i,:)=[w Gain phi];
fprintf(1,'%8.3f      %9.2f\n',20*log10(Gain),(180/pi)*phi);
figure(1)
subplot(2,1,1); semilogx(mytf(1:i,1),20*log10(mytf(1:i,2)),sval);
hold on; axis auto; grid on;
xlabel('Frequency (Hertz)');
ylabel('Gain (db)'); title('Transfer Function (channel2/channel1)');
subplot(2,1,2); semilogx(mytf(1:i,1),(180/pi)*mytf(1:i,3),sval);
hold on; axis auto; grid on;
if do_date==1
    s_xlabel=get_date;
else
    s_xlabel=' ';
end
s_xlabel2=' ';
s_xlabel2=s_xlabel2(1:length(s_xlabel));
xlabel([s_xlabel2 ' Frequency (Hertz) ' s_xlabel]);
ylabel('Phase (Degrees)');
drawnow
phi_last=phi;

if rem(i,5)==1
    color=['r'];
end
if rem(i,5)==2
    color=['b'];
end

```

```

        end
        if rem(i,5)==3
            color=['m'];
        end
        if rem(i,5)==4
            color=['k'];
        end
        if rem(i,5)==0
            color=['g'];
        end

        figure(3); plot(avgPos,avgTorq,color); hold on; grid on;
        figure(2); % PUT FIGURE 2 ON TOP TO FORCE USER TO LOOK AT SINE DATA
        i=i+1;
    end
end

figure(3);
xlabel('Average Encoder Position (deg)');ylabel('Average Torque (N-m)');
title('Force-displacement for different frequencies (Hz)');grid on;

freqs=transpose(w_use);
legFreq=[num2str(freqs,'%6.2f')];
legend(legFreq,-1);

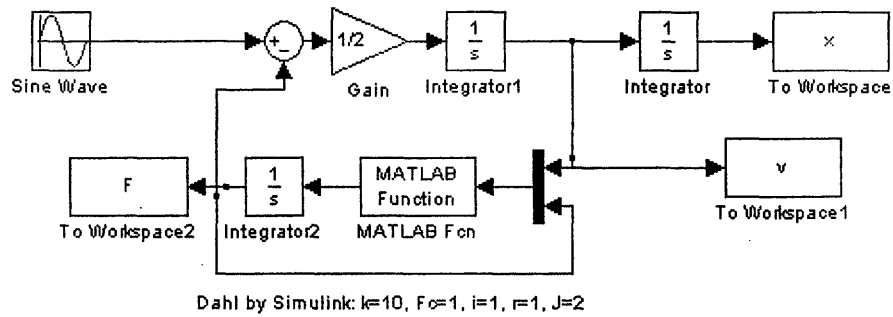
fid=fopen('test.m','w');
fprintf(fid,'%10.6f %9.6f %9.6f\n',transpose(mytf));
fclose(fid);
% saves frequency, gain (absolute--NOT dB), and phase (rad) to file test.m

if exitval==0
    if w_rad==1
        fprintf(1,'\nThe TF created has 3 columns: Frequency (rad/sec),
Magnitude (absolute), Phase (radians)\n')
        mytf(:,1)=(pi/180)*mytf(:,1);
    else
        fprintf(1,'\nThe TF created has 3 columns: Frequency (Hz),
Magnitude (absolute), Phase (radians)\n')
    end
    fprintf(1,'To replot GAIN in Bode format:
semilogx(tf(:,1),20*log10(tf(:,2))\n');
    fprintf(1,'To replot PHASE in Bode format:
semilogx(tf(:,1),(180/pi)*tf(:,3)\n');
    else
        fprintf(1,'\n\n *****\n');
        fprintf(1,' ** Program exited by user during run.... bye. **\n');
        fprintf(1,' *****\n\n');
    end % ends user query if 'ok' to continue
    mlib('Writed',freq_addr,w_use(1));
    mlib('Writed',amp_addr,0);

end
return
function y = setstop()
y=1;
return

```

## Appendix B: Simulink Simulation Model and Code



Matlab Function =  $(s.*(abs(1-u(2)/F_c*\text{sign}(u(1))))^i)*u(1)^*\text{sign}(1-u(2)/F_c*\text{sign}(u(1)))$ ;

### startup.m

```
s = 1;
Fc = 5;
i = 1;
r=1; %reference input; eg r=sin(pi*t)
J=2;

% TO RUN: x0=getfield(x,'signals');
% x1=squeeze(x0.values);
```

## Appendix C: Matlab ODE45 Simulation Code

```
function dy=dahl(t,y)

% y1 = position, y2 = velocity, y3 = force

s = 10;
Fc = 1;
i = 1;

n=t/5;
if mod(round(n),2)==0
    r=1;
    else r=-1;    % r=reference input, eg, r=sin(pi*t)
end

[t r];

J=2;
% these can be modified/add'l inputs

F = y(3);
v=y(2);
x=y(1);

dy=0*y; %goes to default column, in case it was a row of data
dy(3)=(s*(abs(1-F/Fc*sign(v)))^i)*v*sign(1-F/Fc*sign(v));

dy(2)=(r-F)/J;
dy(1)=v;

% to run: [t,y]=ode45('dahl',[0 10],[0 0 0])
```




DASH NO.	APPLICATION		REVISIONS S.O. 305166			
	NEXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED
	80299-002	SPC21-1100G2M	A	DCN-17261	3-13-84	<i>HW</i>

ENVELOPE DRAWING

REV		A	A	A	/																								
SHT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

REVISION STATUS 80299-002

<small>UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES AND APPLY AFTER PLATING</small> <small>TOLERANCES ON</small> <small>XX ± .01 ANGLES ± 0°30'</small> <small>XXX ± .005</small> <small>ALL DIA TO BE 0° WITHIN .010</small> <small>IT SCALE DRAWING</small>	CONTR NO.			ENCODER DIVISION		20745 NORDHOFF ST. CHATSWORTH, CA 91311	
	PREP <i>Gene Kinca</i> 12-22-83	CHK		ENGR <i>[Signature]</i> 1/24/83	MODEL ENCODER		SPC21-1100G2M
MATERIAL	SIZE	FSCM NO.	DRAWING NO.				
	A	17080	SPC21-1100G2M				
	SCALE			SHEET 1 of 6			

1.0 DESCRIPTION: This specification defines an encoder to be coupled to a Limited Angle Brushless DC Motor in a 5 $\frac{1}{2}$  Winchester Disk Drive

The encoder is to provide three output signals. One output is Channel A; second is Channel B; and the third is an Index or Reference Pulse. Channels A and B exhibit a relationship such that A is a Sine function, while B performs the Cosine function.

The outputs from the Sine and the Cosine are analog and represent the actual resolution on the code disk.

Any reference of CW or CCW rotation in the following is assumed to be seen from the encoder side of the motor.

2.0 MECHANICAL CHARACTERISTICS:

2.1 MOTOR/ENCODER DIMENSIONS	1.545" MAX (REF FIG 1) SHT 5
2.1.1 MOTOR	1.006" MAX
2.1.2 ENCODER	.539" MAX
2.1.3 SHAFT DIAMETER (LOAD SIDE)	.2500 +.0000" -.0005"
2.1.4 SHAFT LENGTH (LOAD SIDE)	.82 MAX
2.2 COMM/HUB INERTIA	200 (10 <sup>-6</sup> ) OZ-IN SEC <sup>2</sup> MAX
2.3 RESOLUTION	1100 CYCLES/REV
2.3.1 INDEX	1 PULSE/REV 5 CYCLES LONG
2.4 SPEED	600 RPM MAX
2.5 ACCELERATION	6000 RAD/SEC <sup>2</sup> MAX

3.0 MOTOR CHARACTERISTICS @20°C:

3.1 EXCURSION ANGLE	+60°	O
3.2 TORQUE SENSITIVITY	5.5 OZ-IN/AMP	Kt
3.3 BACK EMF CONSTANT	.0388 VOLTS/RADS/SEC	Ke
3.4 RESISTANCE	3.7 $\Omega$	R
3.5 MOTOR CONSTANT	2.86 OZ-IN/ $\sqrt$ WATTS	Km
3.6 PEAK TORQUE	13 OZ-IN	Tp
3.7 CONTINUOUS TORQUE	8.6 OZ-IN	Tc
3.8 ROTOR INERTIA	1.7 (10 <sup>-3</sup> ) OZ-IN-SEC <sup>2</sup>	Jr
3.9 VISCOUS DAMPING (ZERO IMPEDANCE)	.063 OZ-IN/RAD/SEC	D

<b>ENCODER DIVISION</b>	SIZE	FSCM NO.	DRAWING NO.
	<b>A</b>	<b>17080</b>	SPC21-1100G2M
	SCALE		SHEET 2 OF 6

24123 DIETRICH-POST CLEARPRINT 1000H  
 0014 0000 ON 10 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00

4.0 ENCODER ELECTRICAL CHARACTERISTICS:

4.1 INPUT POWER REQUIRED

- 4.1.1 VOLTAGE +5VDC  $\pm 5\%$   
 +12VDC  $\pm 5\%$   
 -12VDC  $\pm 5\%$
- 4.1.2 CURRENT (+5VDC) 150mA MAX  
 (+12VDC) 200mA MAX  
 (-12VDC) 20mA MAX

4.2 OUTPUT (25°C & OVER SUPPLY VOLTAGE RANGE) (REF FIG 2)

- 4.2.1 SINE & COSINE (5KΩ LOAD) 700mV AVERAGE pp  $\pm 5\%$
- 4.2.2 AMPLITUDE VARIATION WITH TEMPERATURE (0 - 60°C, 5KΩ LOAD)  $\pm 10\%$  OF THE AVERAGE (REFERENCE AT 25°C)
- 4.2.3 INDEX LOGIC 1 3.0VDC - 5.25VDC  
 LOGIC 0 0.8VDC MAX
- 4.2.4 MODULATION 10% (OVER MOTOR ACTIVE RANGE, SEE NOTE 1 BELOW)
- 4.2.5 QUADRATURE  $\pm 10$  ELEC DEGREES MAX
- 4.2.6 OPERATING FREQUENCY 11.0 KHz
- 4.2.7 FREQUENCY RESPONSE THE AVERAGE Vpp VALUE AT 11.0 KHz SHALL NOT DROP BY MORE THAN 10% OF ITS AVERAGE VALUE AT 10Hz
- 4.2.8 DC OFFSET 0V  $\pm 5\%$  OF THE AVERAGE Vpp
- 4.2.9 DC OFFSET VARIATION WITH TEMPERATURE (0 - 60°C)  $\pm 5\%$  OF THE AVERAGE Vpp
- 4.2.10 SYMMETRY  $\pm 5\%$  FOR BOTH SINE & COSINE

4.3 DIRECTION OF ROTATION IN THE CCW DIRECTION - SINE LEADS COSINE

NOTE 1: THE MOTOR ACTIVE RANGE IS DEFINED AS THE MOTOR SHAFT ANGLE STARTING AT INDEX POSITION AND ENDING 115.4° MECHANICAL GOING CW AS SEEN FROM THE ENCODER END

ENCODER DIVISION	SIZE	FSCM NO.	DRAWING NO.
	A	17080	SPC21-1100 G2M
	SCALE		SHEET 3 OF 6

#### 4.4 INDEX PERAMETERS

##### 4.4.1 SETUP

##### 4.4.1.1 REFERENCE POSITION

THE MOTOR IS FIRST POSITIONED AT ITS REFERENCE POSITION. THE MOTORS REFERENCE POSITION IS DEFINED AS THE FINAL POSITION OF THE MOTOR SHAFT WHEN A POSITIVE VOLTAGE IS APPLIED TO THE MOTOR BLACK LEAD WITH RESPECT TO THE RED.

##### 4.4.1.2 INDEX LOCATION

WITH THE MOTOR LOCKED AT THE REFERENCE POSITION AND THE COMMUTATOR FREE TO ROTATE, THE INDEX MARK ON THE COMMUTATOR SHOULD BE LOCATED  $32.4^{\circ}$  ( $-0^{\circ} - +1^{\circ}$ ) MECHANICAL CCW FROM THE INDEX PHOTO SENSOR.

#### 5.0 ENCODER ENVIROMENTAL CHARACTERISTICS:

##### 5.1 TEMPERATURE

5.1.1 OPERATING  $0^{\circ}\text{C}$  TO  $60^{\circ}\text{C}$

5.1.2 STORAGE  $-40^{\circ}\text{C}$  TO  $80^{\circ}\text{C}$

5.2 HUMIDITY 5% to 95%  
(NO CONDENSATION)

5.3 SHOCK 50G's FOR 11mSEC  
(ALL AXIS)

5.4 VIBRATION 40 TO 2000 CPS @ 16G's

#### 6.0 RELIABILITY & LIFE CHARACTERISTICS:

##### 6.1 SOURCE OF ILLUMINATION

TYPE LED (GALLIUM-ARSENIDE)

LIFE 100,000 HOURS MIN

6.2 AMPLITUDE VARIATION AMPLITUDE SHALL NOT VARY  
BY MORE THAN  $\pm 5\%$  OVER  
40,000 HOURS OPERATION



ENCODER DIVISION

SIZE

A

FSCM NO.

17080

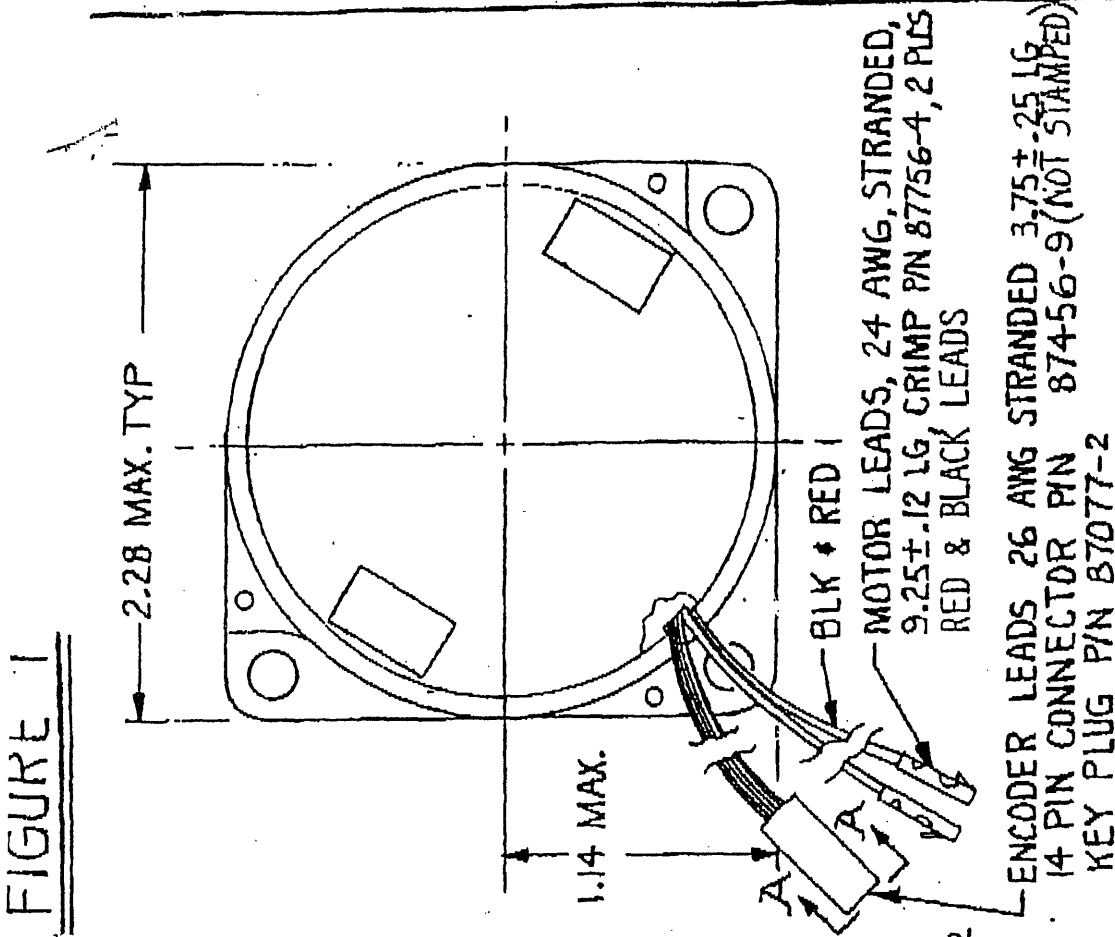
DRAWING NO.

SPG21-1100G2M

SCALE

SHEET 4 OF 6

**FIGURE 1**



ENCODER OUTPUT WIRING	
COLOR	FUNCTION
WHT	1 SINE
BLU	2 COSINE
KEY PLUG	11
GRN	4 ZERO INDEX
RED	5 +5V
BLK	6 GROUND
ORN	7 +12V
YEL	8 -12V



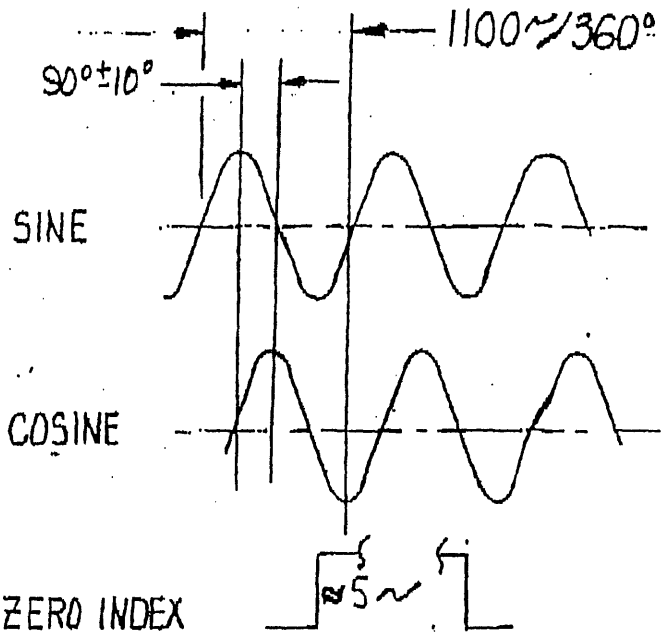
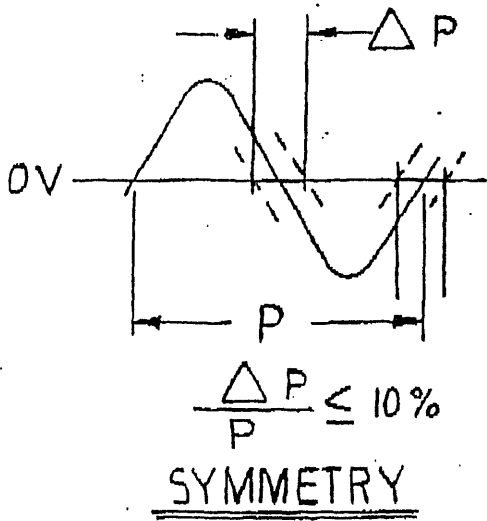
**ENCODER DIVISION**

SIZE **A** FSCM NO. **17080**  
SCALE

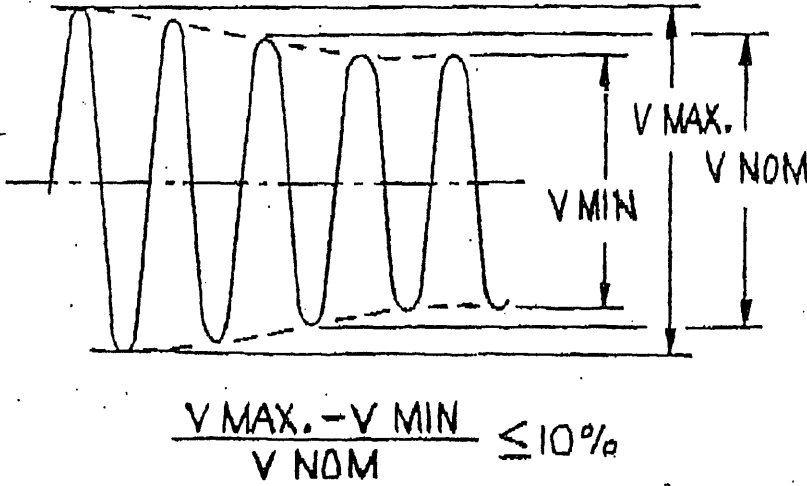
DRAWING NO. **SPC21-1100G2M**

SHEET 5 OF 6

# FIGURE 2



C.C.W. ROTATION AS VIEWED FROM  
 ENCODER END OF MOTOR  
 ZERO POSITION NOT RELATED TO  
 SINE-COSINE  
PHASING



AMPLITUDE MODULATION

ENCODER DIVISION	SIZE	FSCM NO.	DRAWING NO.
	A	17080	SPC21-1100G2M
	SCALE		SHEET 6 OF 6

16392 8559  
16414 C 180

## 2.737 Mechatronics

### Laboratory Assignment 4: Brushless Motor Control

Assigned: 4/5/99

Pre-lab due: 4/14/99

Reports due: 4/28/99 in checkoffs

Readings: Winchester actuator specifications, *Mastering Simulink 2* 7.1,7.2,7.4.1

Animations: <http://mot-sps.com/motor/mtrtutorial/mtr.html> (Motor Principles-Brushless DC Motors & Brushless AC Motors)

## 1 Introduction

In this lab you will learn about electromagnetic actuators and brushless motor commutation. We will first model an electromagnetic actuator and understand how it works. These actuators were used in old Winchester hard drives and have an encoder attached on the back. These actuators are designed to provide controlled torque over a range of about  $\pm 60^\circ$  but are not capable of controlling torque over a full rotation. To overcome this limitation, in the lab setups we have attached the shafts of two such actuators with a flexible coupling. By properly applying voltage to the two actuators, i. e., commutating them, it is possible at any angle to control the torque applied to the common shaft. This commutation is analogous to what would be done to spin a 2-phase (i. e. two-coil) brushless motor. Real brushless motors are usually 3-phase, but the underlying concepts for commutation are identical to those explored in this lab.

## 2 Prelab

Please read and understand the entire lab handout and attached actuator specifications. Write your solutions for all of section 3, conduct the quick experiment, and show us the results. Also, answer the questions on the interface circuit in section 4.1.2 and build it. Display the encoder angle in degrees in Cockpit on the PC's.

View the animations in Motorola's tutorial website, "Motors in Motion," <http://mot-sps.com/motor/mtrtutorial/mtr.html>. The animations relevant for this lab are the Brushless DC Motor and Brushless AC Motor animations under the "Motor Principles" heading. These animations provide an excellent way to understand motor commutation.

This is a long lab with many challenges and some open-ended sections. Pace yourself so that you will have time to work through the entire lab.

## 3 Winchester Actuator Modeling

### 3.1 Electromagnetic Actuators

There are three main types of electromagnetic principles used in actuators: reluctance, Lorentz force, and induction.

- In a reluctance actuator, the air gap in the magnetic circuit is allowed to change. An example of a reluctance actuator is an electromagnet used to suspend a steel ball. When the ball moves, the air gap in the magnetic circuit changes. High forces are possible with reluctance type actuators, but they are often nonlinear devices.
- When a wire carrying a current is placed in a magnetic field, the field exerts a force on the current in the wire. This Lorentz force is given by

$$\vec{F} = I\vec{l} \times \vec{B}. \quad (1)$$

Here,  $I$  [A] is the current in the wire,  $\vec{B}$  [T] is the magnetic field, and  $\vec{l}$  [m] is the length of wire and points in the direction of the current. Lorentz force actuators thus produce force linearly with current and with magnetic field.

- Induction actuators use a primary coil to induce currents in a secondary coil by transformer action. Since the currents in the secondary coil are in the magnetic field of the first coil, force is produced. The most common induction actuator is the induction motor such as is widely used in household appliances, washing machines, dishwashers, and air conditioners.

### 3.2 Actuator Geometry

In this lab you will use a Lorentz-type actuator taken from an old hard disk drive. These drives were known as *Winchester* type. The Winchester actuator is shown in cross-section in Figure 1. It consists of a permanent magnet rotor which spins with the actuator shaft and a stationary coil of wire wound around an iron core. Only the interior part of the coil does useful work. The other part of the coil returns around the outside and produces no force but is required to complete the coil. The coil is wound in two alternate polarity sections with a small gap between them. We will neglect this gap and assume that the two halves of the coil meet each other as shown in Figure 1. The polarity of these two sections is reversed so that the torque generated by the north magnet and that generated by the south magnet reinforce rather than cancel each other. In the figure, a dot represents current coming out of the page, and an x represents current going into the page. This actuator is a Lorentz force actuator since the coil is immersed in the magnetic field generated by the permanent magnets. The gap between the magnet and the iron core inside the coil remains fixed as the rotor spins. There are a couple of actuators in the lab which have been taken apart for you to examine so you can understand this construction.



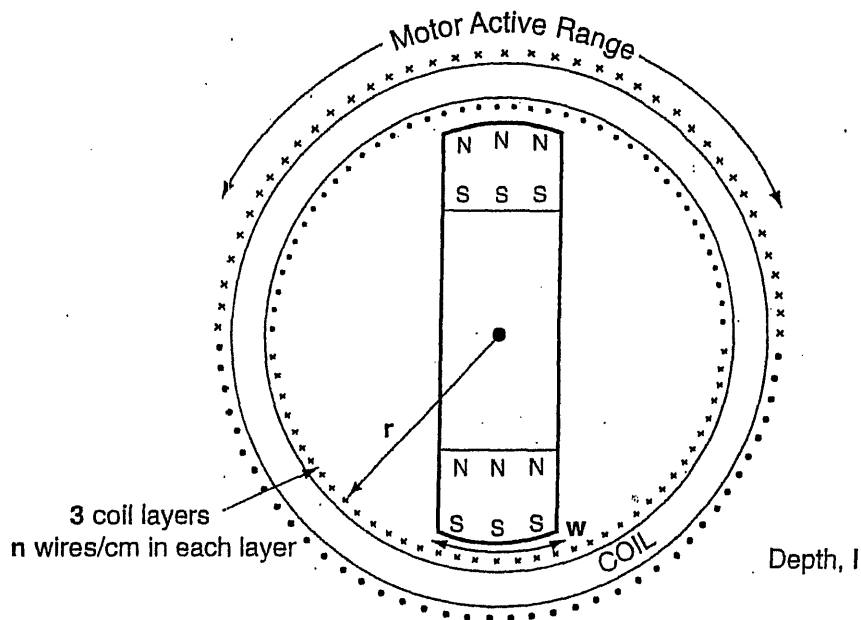


Figure 1: Geometry of the Winchester actuator. The radius at the tip of the magnet is  $r = 1.9$  cm. The width of the pole face is  $w = 1.7$  cm. The depth of the cylindrical actuator into the page is  $l = 1.52$  cm. The coil consists of three layers of wire with  $n = 20$  wires/cm in each layer for a total of 60 wires/cm in the coil.

### 3.3 Torque Constant Calculation

In the prelab, use the Lorentz force law (1) to calculate the expected torque constant  $K_t$  [N·m/A] for the motor when the rotor is in the motor active range. The average magnetic field in the gap between the magnet and iron core was measured with a gaussmeter and found to be  $B = 0.65$  T. The relevant motor dimensions for this calculation are given in Figure 1. Compare your calculated torque constant to the value  $K_t = 5.5$  oz·in/A specified by the manufacturer. The conversion factor, 0.2780 N/oz, may be helpful in doing this.<sup>1</sup>

### 3.4 Motor Detent Positions

#### 3.4.1 Theory

The location of the motor active range and index pulse are shown in Figure 2. Compare this with the information in the attached specifications. If a constant voltage is applied so that the currents are as shown in Figure 2, where will the rotor end up? Is this a stable or unstable equilibrium? Can you find another equilibrium point? Is it stable or unstable?

<sup>1</sup>The easiest way to do this conversion is to recognize that in SI units, the back EMF constant,  $K_e$ , has the same numerical value as the torque constant,  $K_t$ . Recall that this result can be derived by setting the electrical power in,  $iv$ , equal to the mechanical power out,  $\tau\omega$ .

What happens to these equilibria when you reverse the voltage? Draw some diagrams like Figure 2 which show the forces on the coils and magnets as the rotor is in different positions. Also draw a diagram with the voltage reversed. These diagrams will help you answer the questions above.

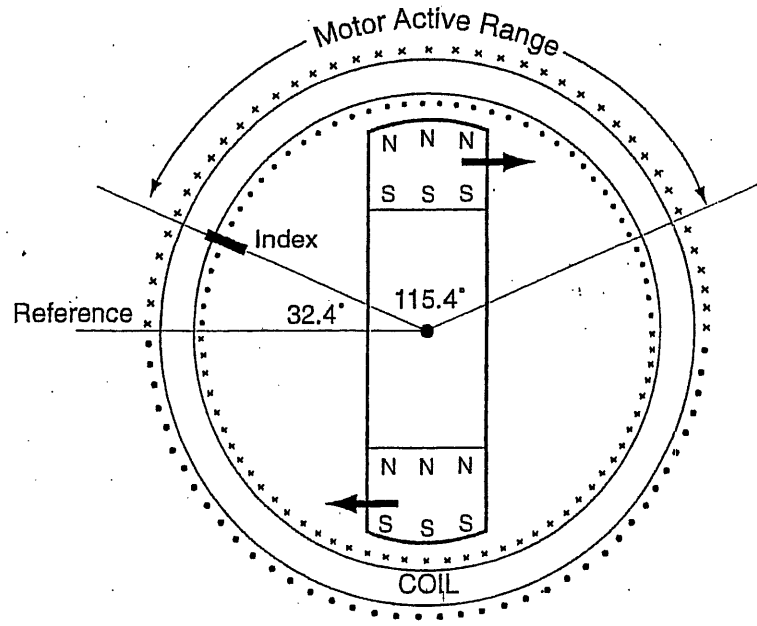


Figure 2: Schematic of the Winchester actuator as viewed from the encoder side. The forces shown on the magnets result from currents in the directions indicated.

### 3.4.2 Experimental Verification

It is easy to check your answers in the previous section experimentally. As part of the prelab, apply 8 V to Motor 1 by sending 4 V to the power amp from a DAC. The motor connections are described below and given in Figure 3. As shown in Figure 4 the fixed yellow mark on the frame near Motor 1 corresponds to its Reference position when aligned with the yellow mark on the flexible coupling. Observe the equilibria and their stability. Turn the motor with your hand while the constant voltage is applied. What happens as you increase and decrease the voltage? Now, reverse the sign of the voltage and explain what you observe.

### 3.4.3 Motor Connections

The motor connections are shown in Figure 3. The motor leads are brought out on a cable with a red wire, a black wire, and an uninsulated ground wire. The black leads from both motors are connected to the uninsulated wire. The red and black wires are connected to

the positive leads for the left and right motors, respectively. When connecting the motor to the power amp, connect the uninsulated wire to the black ground on the output of the power amp. The red wire for the left motor should be connected through a  $10\ \Omega$  power resistor to the yellow wire for output A of the power amp. The black wire for the right motor should be connected through a  $10\ \Omega$  power resistor to the yellow wire for output B of the power amp. Although they are marked, if you are not sure which yellow wire is A and which is B, carefully look at the bottom of the power amp board (turn it over) and follow the wires back to the solder pads "A Out" and "B Out."

Remember that the power supplies should be in series mode and the voltage should be set to no more than  $\pm 17$  volts to avoid damage to the op amp.

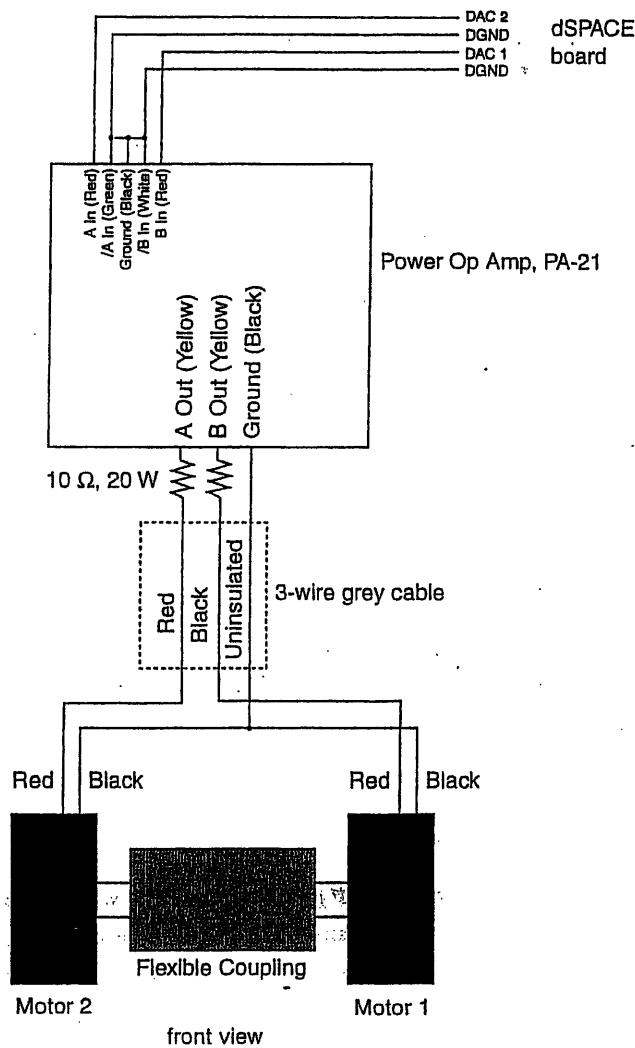


Figure 3: Motor Connections: The two motors are powered by the A and B channels of the power op amp, respectively. The  $10\ \Omega$  power resistors are connected to the terminal strip of the amplifier. Note that DAC 1 controls Motor 1 and DAC 2 controls Motor 2.

### 3.5 Torque vs. Angle Curves

The angular rotation of both Motors is defined using Encoder 1. Figure 4 defines this angle as CCW from the Reference position ( $0^\circ$ ). On the motor setups, this Reference position is indicated with a fixed yellow mark near Motor 1 and a fixed white mark near Motor 2. On the flexible coupling, the position of Motor 1's rotor is indicated with a yellow mark while the position of Motor 2's rotor is indicated with a white mark as shown in Figure 4. Note that Motor 2's rotor is  $-90^\circ$  from Motor 1's rotor.

In the prelab, plot the expected torque vs. angle curve of Motor 1 as its rotor changes from  $0^\circ$  through  $360^\circ$  when a constant positive voltage is applied. Recall from the actuator specifications that the Reference position is defined as the final position of the rotor when a negative voltage is applied. For simplicity, we will ignore the small gap inbetween the two halves of the coil. To carry out this computation, you will need to integrate (sum) the force per wire over the area faced by the permanent magnet. As the magnet spans the gap where the coil reverses polarity the net force on the rotor is reduced. Carry out this summation for rotor positions  $0^\circ \leq \theta \leq 360^\circ$  to develop the motor torque curve  $T_1(\theta)$ . You will find that the torque vs. angle curve is trapezoidal because of the crossing of coil polarity at the two halves of the coil.

Now plot the expected torque vs. angle curve for Motor 2,  $T_2(\theta)$ . The actuators are the same, but Motor 2's rotor is shifted by  $+90^\circ$  from Motor 1's rotor, and the sense of rotation is opposite since the actuators are facing each other (Figure 4). We will use these two curves to commutate the 2-phase motor.

### 3.6 Commutation

In order to control torque through the full  $360^\circ$  rotation we will need to use both actuators. The basic idea of commutation is that when one actuator is out of its active range, we can use the other actuator to produce torque. If both actuators are in their active ranges, we can use both of them.

Now that we have derived two trapezoidal torque constant vs. angle curves,  $T_1(\theta)$  and  $T_2(\theta)$ , we must figure out how to divide up the torque load between the two motors at each angle  $\theta$ . Let  $T_d(\theta)$  be our desired control torque. It is the sum of the torques produced by each motor,

$$T_d(\theta) = i_1(\theta)T_1(\theta) + i_2(\theta)T_2(\theta). \quad (2)$$

We have another degree of freedom in choosing  $i_1(\theta)$  and  $i_2(\theta)$  so we can also require power optimal operation of our motor. This additional constraint will lead to the commutation laws. The power  $P$  consumed by our two motor coils, each with resistance  $R$  is

$$P = (i_1^2(\theta) + i_2^2(\theta))R \quad (3)$$

We now substitute for  $i_2(\theta)$  in (3) using (2) to give

$$P = \left( i_1^2(\theta) + \left( \frac{T_d(\theta) - i_1(\theta)T_1(\theta)}{T_2(\theta)} \right)^2 \right) R \quad (4)$$

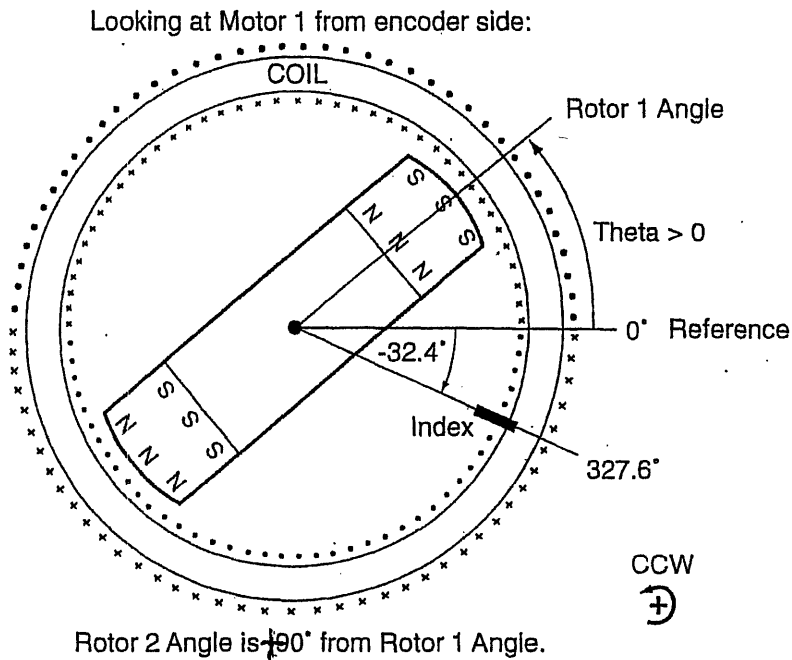
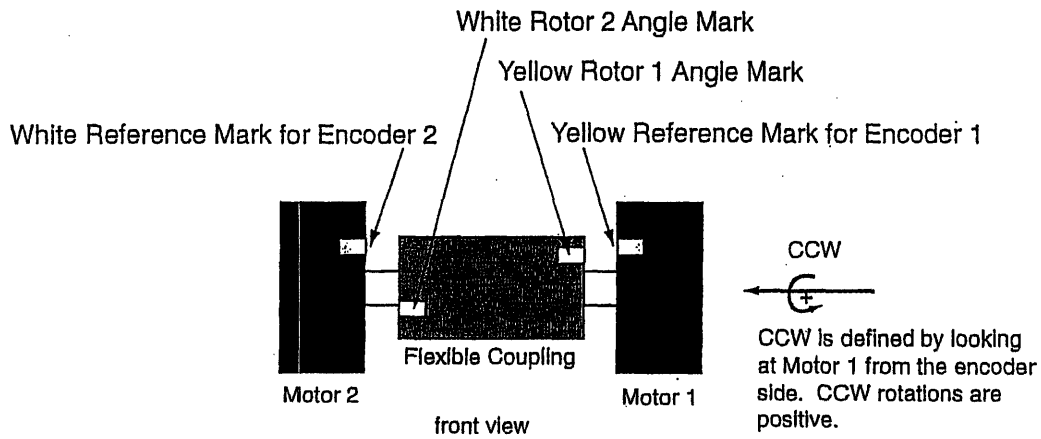


Figure 4: Our angle conventions are shown. The angles of Rotor 1 and Rotor 2 are measured CCW from the reference position of Motor 1.

To find the power optimal currents, we set  $\frac{\partial P}{\partial i_1} = 0$  which yields

$$\frac{i_1(\theta)}{T_1(\theta)} = \frac{i_2(\theta)}{T_2(\theta)} \quad (5)$$

Equation (5) says that the commanded currents,  $i_1(\theta)$  and  $i_2(\theta)$ , should be proportional to

the torque constants,  $T_1(\theta)$  and  $T_2(\theta)$ . In fact, using (2) again, we can arrive at

$$i_1 = \frac{T_d T_1(\theta)}{\sqrt{T_1(\theta)^2 + T_2(\theta)^2}} \quad (6)$$

$$i_2 = \frac{T_d T_2(\theta)}{\sqrt{T_1(\theta)^2 + T_2(\theta)^2}} \quad (7)$$

which are the commutation laws which we will implement on the dSPACE board. As before,  $T_d$  is a desired torque commanded by the controller. A block diagram representation of commutation is shown in Figure 5.

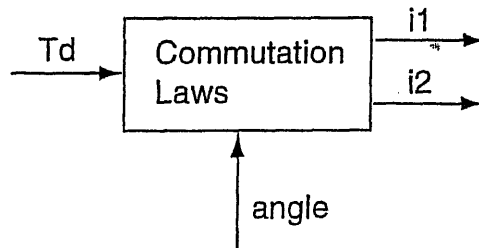


Figure 5: Commutation block.

To implement the commutation laws, (6) and (7), we would like to directly specify the current in each coil. In practice, this is often done with two minor closed-loop current control loops with high bandwidths such as the one you designed in Lab 2: Analog Feedback Systems. Here, however, we are fortunate that the inductance of the motor coils is small, experimentally determined to be 4.7 mH. The time constant  $\tau = L/R$  is therefore small, and we further reduce it by increasing the resistance by placing a 10  $\Omega$  power resistor in series with the motor coil. The time constant with this additional resistance is 0.35 ms, so our open-loop “natural” current control has a bandwidth of 2800 rad/s. As long as our outer position or velocity closed-loop bandwidth is significantly lower than this, we can ignore the inductance in the motor coil and model the coil as a simple resistor. Controlling a motor in this fashion is called “voltage mode” since we command voltages  $V_1$  and  $V_2$  rather than the currents  $i_1$  and  $i_2$  in (6) and (7) directly. In voltage mode, the back EMF generated by the motor adds damping, whereas in current mode, the closed-loop control eliminates the effects of the back EMF. For our motors, the back EMF damping is small compared to the mechanical viscous damping.

## 4 Commutating the Brushless Motor

### 4.1 Encoder

As long as we don't excite the torsional resonance of the coupling, the angles from the two encoders are very close so we only need to use one of them. We will thus use only the

encoder on Motor 1, the right-hand actuator (see Figure 3). Although we are using an incremental encoder, we need the absolute angle to be able to commutate the actuators. Thus, we will need to use the Reference pulse from the encoder to set the absolute angle.

#### 4.1.1 Encoder Connections

The encoder leads are soldered to a colored ribbon cable which is terminated in a DIP connector which you will plug into your protoboard. Each encoder is wired through one cycle of colors in the ribbon cable (ten colored wires). The encoder leads are soldered to identical colors in the ribbon cable. Note that the pins on one side of the DIP connector provide access to every 2nd wire in the ribbon cable, while the pins on the other side of the DIP connector provide access to the remaining wires: The black, grey, blue, . . . wires will be on one side of the DIP connector, while the white, purple, green, . . . wires will be on the other side. Please check your connections before applying power. We will use  $\pm 15$  V to power the encoder rather than the specified  $\pm 12$  V. Figure 6 shows the DIP connections and pin numbers.

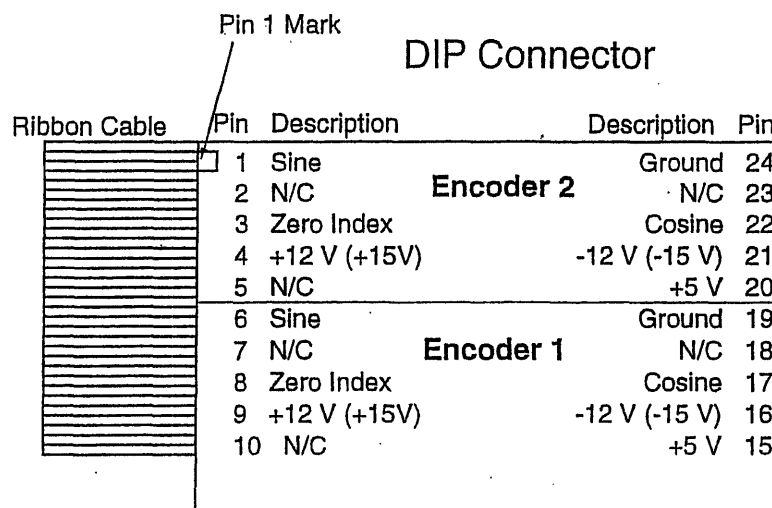


Figure 6: Encoder DIP Connector. In this lab we will use only Encoder 1.

#### 4.1.2 Encoder Interface Circuits

The encoders attached to the Winchester actuators provide analog sine and cosine signals. We would like to use the dSPACE board to keep track of the encoder position. To do this we must convert the analog sine and cosine signals into digital levels. Figure 7 shows an interface circuit that will accomplish this for each of the two encoder channels.

In the prelab, explain briefly how the interface circuit works. What is the function of the op amp? Can you see why these resistor values were chosen? What does the Schmitt trigger do? The encoder specifications and the voltage sketches in Figure 7 will be helpful in understanding this circuit.

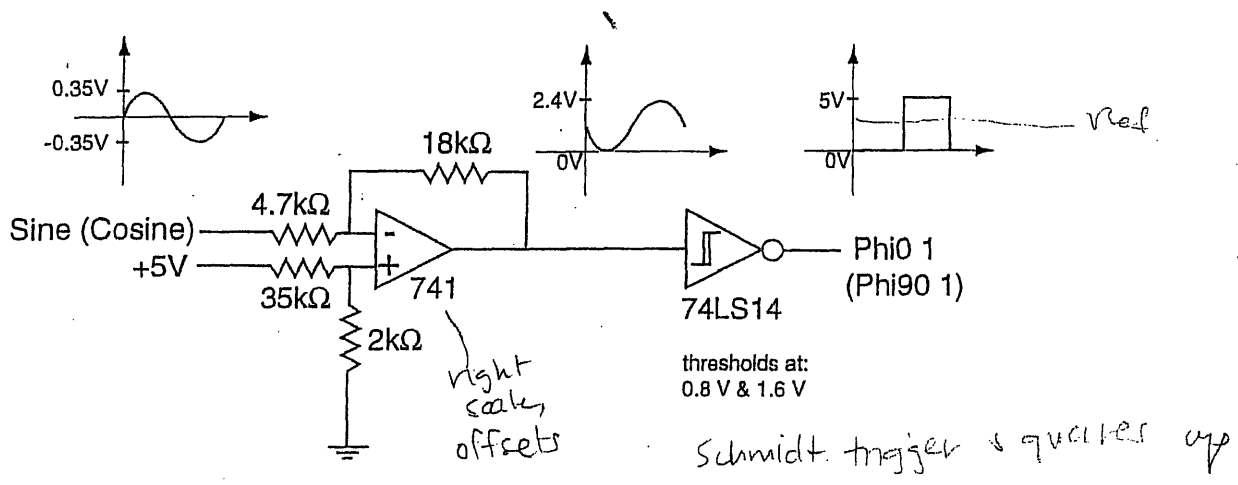


Figure 7: Interface circuit to convert analog sine and cosine signals from the encoder to digital levels that can be sent to the dSPACE encoder electronics. Two such circuits are required for the sine and cosine outputs of the encoder, respectively.

Since the dSPACE board provides for differential inputs, we will need to provide a 1.5 V reference to the complementary inputs /Phi0 1, /Phi90 1, and /Index 1. The "/" symbol indicates a complementary input. For example, Phi0 1 & /Phi0 1 are a differential input pair for one signal. The circuit to do this is shown in Figure 8.

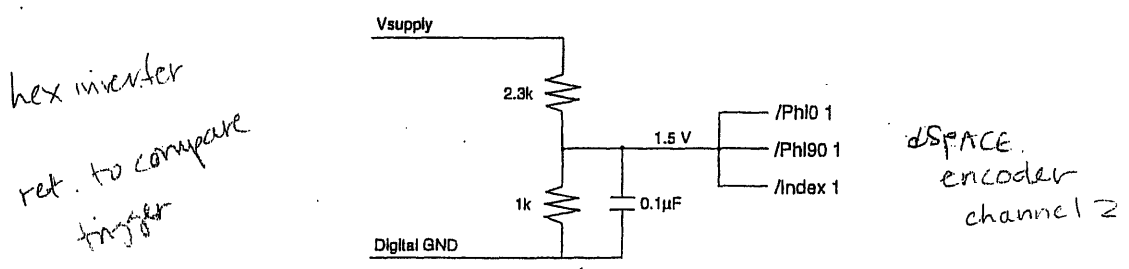


Figure 8: Circuit for generating a 1.5 V reference for the complementary inputs to the dSPACE encoder interface.

## 4.2 Software & System Integration

### 4.2.1 Encoder

First, make sure that the dSPACE system is reading the encoder through the interface circuit properly. Scale the angle so that it appears in degrees and look at the angle in a Cockpit display. Shift the angle in Simulink so that 0° corresponds to the Reference position, not the Index position. (using the offset angle shown in Figure 4). Does the angle change by 360° as you turn the shaft one revolution? Do the yellow and white marks painted on the flexible coupling (Figure 4) align at the appropriate angles?



Now, we need to use the reference pulse to give us an absolute zero angle as opposed to zero being the initial position of the shaft. To do this we will use the DS1102ENC\_INDEX\_C1 block in the dSPACE library to search for the reference pulse. Since we cannot commutate the motor until we have found the index, we will not want to run the rest of our code until the index is found first. A nice way to do this in Simulink is to use an enabled subsystem. We will enable the subsystem once the reference is found. When actually running the program, turn the shaft one revolution by hand until the encoder reference is found. You should monitor whether or not the Index has been found in Cockpit.

Note: If your encoder interface is working but then stops working later on as you add more code and you have not changed the interface circuit, try slowing down the sample time.

#### 4.2.2 Commutation

Build a commutation subsystem block that takes in as inputs desired Torque Command  $T_d$  & Angle in degrees. The outputs of the commutation subsystem should be Motor 1 and Motor 2 voltages that can be sent directly to two DAC channels. In this routine you should transform the angle back into the range  $0^\circ$ – $360^\circ$ , and use (6) and (7) to commutate the motor. In implementing these equations, use your predicted torque vs. angle curves from the Prelab. The Simulink Look-up Table block may be helpful to represent these curves.

When testing your commutation block, first hook up only one actuator at a time (in Simulink) and command a constant torque,  $T_d$ . What torque characteristics do you observe? Can you make the motor spin all the way around with only one motor? What's wrong with doing this? Try different initial conditions, i.e. motor angles. What happens if the rotation rate is slowed down? Now test the other actuator by itself. Make sure that both actuators are spinning the shaft in the same direction!

Connect both actuators and implement your commutation laws with a constant commanded torque,  $T_d$ . The shaft should spin smoothly. Look at and record the commutation signals on the scope. Explain their features. In Cockpit, display the Motor 1 and Motor 2 command signals with bar graphs. Keep these bar graph displays for use when you close the loop later on. Connect the desired Torque Command signal to a slider control in Cockpit. Stop the motor with your hand and turn it slowly. How do the command signals change with angle? How do they change with commanded torque?

## 5 Closing the Loop

### 5.1 Position Control

At this point you should have a working open loop commutation system. It is open loop since we are not feeding back the shaft angle to commanded torque although we are measuring it to commutate the motor. Figure 9 shows a block diagram of a closed-loop position control system for our 2-phase motor. We model the mechanical system as an inertia,  $J_t$ , with damping,  $b$ . The inertia of the rotors and encoder hubs are given in the specifications as  $1.7 \times 10^{-3}$  oz·in·s<sup>2</sup> and  $200 \times 10^{-6}$  oz·in·s<sup>2</sup> respectively. The flexible coupling's inertia

is calculated to be  $3.09 \times 10^{-6} \text{ kg}\cdot\text{m}^2$ . The total inertia is thus  $J_t = 1.65 \times 10^{-5} \text{ kg}\cdot\text{m}^2$ . The total damping is twice the value given for each motor, or  $b = 8.9 \times 10^{-4} \frac{\text{N}\cdot\text{m}}{\text{rad/s}}$ .

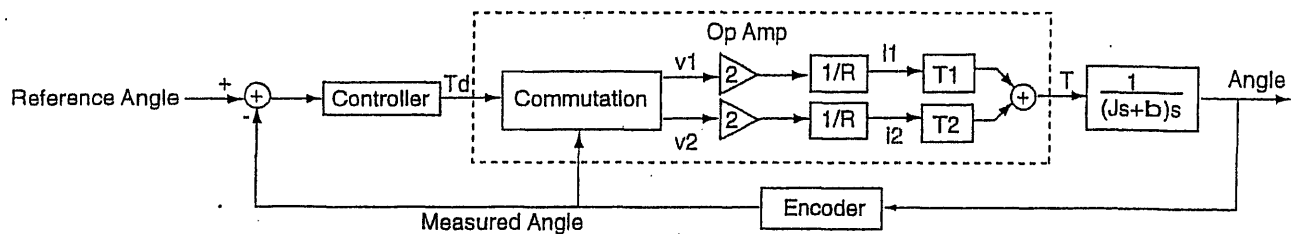


Figure 9: Block diagram for position control of a 2-phase motor.

Although the block diagram in Figure 9 shows that two voltages are used to actuate our motor, we can use SISO control system theory since we are only feeding back one variable (angle) and generating one command ( $T_d$ ). Draw a block diagram which you can use to design the controller. Include all gains in the system. Do not show the commutation in this diagram; treat the 2-phase motor as if it has only one equivalent current and a torque constant  $K_t$ . How is  $K_t$  related to your calculation in section 3.6? Explain your controller design.

Design and implement a closed-loop position controller for the motor. The closed-loop system should have a damping coefficient,  $\zeta = 0.5$ , and no steady state error to a step input.

- Connect a Cockpit slider control to the reference angle and use this control to move the motor to different positions.
- Starting at the Reference position,  $0^\circ$ , have the motor step in  $45^\circ$  increments counterclockwise up to  $180^\circ$  and then back to  $0^\circ$ .
- How fast can you get the motor to move and settle (2%) from  $0^\circ$  to  $-90^\circ$ ? Show Trace plots of this. Re-design your controller to minimize the transit time while satisfying the damping and steady-state error requirements. You may want to consider shaping the reference signal to minimize the move plus settle time.

## 5.2 Velocity Control

Although we do not have a physical tachometer, we can use the difference between successive encoder counts as a virtual tachometer. The dSPACE ENC\_DPOS block does just this. Design a velocity controller using this block with a 5kHz sampling rate and no steady-state error for a step in velocity. The closed-loop velocity control will be used for fast velocities as well as slow velocities (0.1 revolutions/s).

What problem do you notice with your first attempt at velocity control? Listen carefully to the motors. Uncover the root of the problem by looking at various signals in your Simulink code using Trace and/or Cockpit. Record some representative Trace plots

which show the problem, and explain why it occurs. Devise a solution which eliminates the problem: How do you make our implementation of a tachometer more like an ideal tachometer? Describe your solution in detail and demonstrate the improved performance at low velocities.

## Friction Phenomena and Friction Models

Karl J. Åström

1. Introduction
2. Friction Models
  - Static models
  - Dahl's model
  - The Bliman-Sorine model
  - The LuGre model
3. Effects of Friction on Control Systems
  - Stick slip motion
  - Servo systems
4. Conclusions

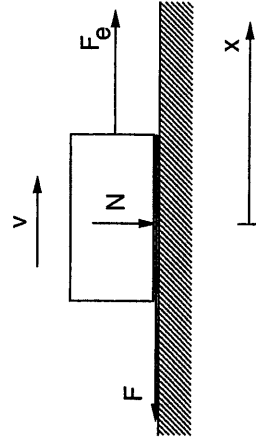
## Introduction

- Essential in Motion Control
- Classics
  - Leonardo da Vinci  $\approx 1452-1519$
  - Amontons 1699
  - Coulomb 1785
- Tribology
- Control
- Physics AFM
- Geology

## Amontons's Paradox

### Observations

- Friction is proportional to normal load
- Friction does not depend on the apparent area of contact



The classical friction law

$$F = \mu N$$

## Application Areas

- Robotics
  - Machine tools
  - Valves and actuators
  - Automobiles
  - Antennas
  - Telescopes
  - Mechatronics
  - Micro-mechanical systems
  - Geophysics
  - Surface physics
  - Physiology
- Tire-road interaction  
ABS  
Traction control

### Very Complex Phenomena

Reasonably well understood phenomenologically

- Stiction, elastic deformation

Some phenomena

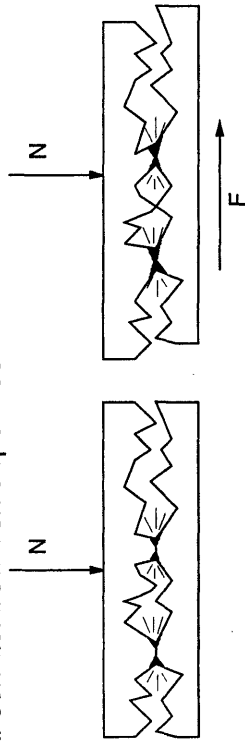
- Steady-state characteristics
- Pre-sliding displacement
- Hysteresis
- Varying break-away force
- Randomness—Repeatability

Much poorly explained

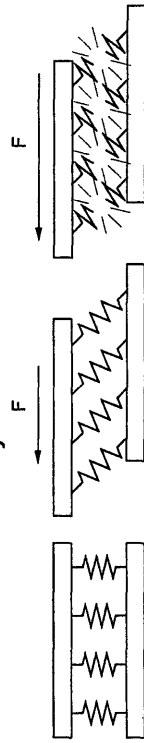
- Friction and surface roughness
- The Mica experiments
- The role of lubrication

### Simple Mechanisms

Metal contact between asperities

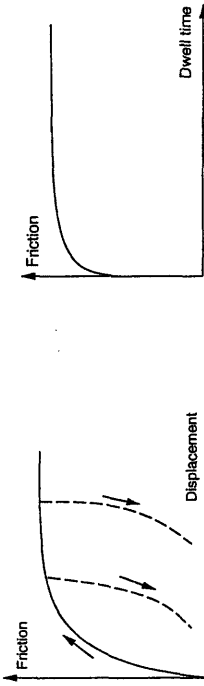


Visualization of break away

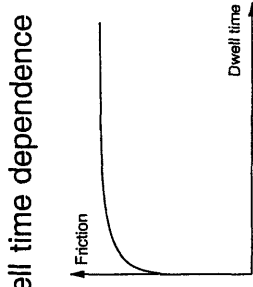
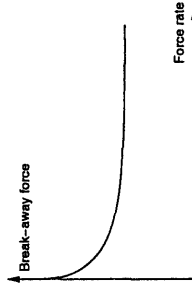


### Some Friction Phenomena

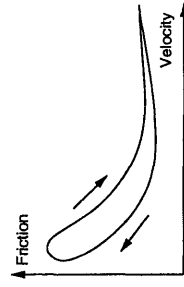
Pre-sliding displacement Dwell time dependence



Varying break away force

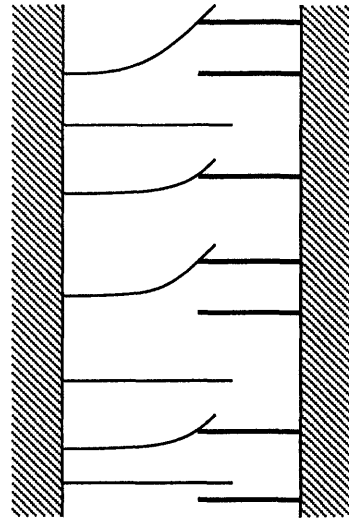


Hysteresis



### The Bristle Model

An abstraction of asperities



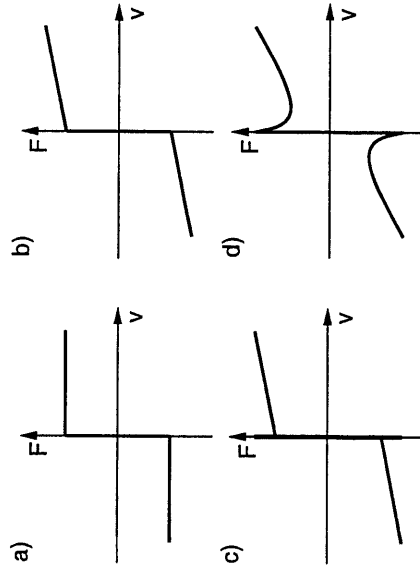
## A Control Perspective

- Understand the effects on friction on control systems
- Typical phenomena
  - Limit cycles
  - Stick slip motion
  - Hunting
  - Lack of precision in tracking
- Modeling and Simulation
- Friction compensation
- Modeling for control

## Friction models

- Classical static models
  - Coulomb friction
  - Viscous friction
  - Stiction
- Mechanics and fluid dynamics
  - First principles
  - Microscopical contact
  - Viscosity
- Empirical phenomenological models
  - The Dahl model
  - The Bliman-Sorine model
  - LuGre model

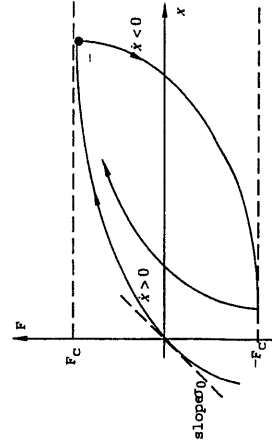
## Static Models



- a) Coulomb b) Coulomb + viscous c) stiction d) Stribeck effect
- In practice there are often asymmetries!

## Dahl's Model

- P. Dahl Aerospace Corporation
- Extensive use in simulations in military projects
- Inspired by solid friction



The stress-strain curve is modeled as

$$\frac{dF}{dx} = \sigma \left( 1 - \frac{F}{F_c} \operatorname{sgn} v \right)^\alpha$$

### Dahl's Model - Continued

The stress-strain curve

$$\frac{dF}{dx} = \sigma \left(1 - \frac{F}{F_c} \operatorname{sgn} v\right)^\alpha$$

Differentiate with respect to time

$$\frac{dF}{dt} = \frac{dF}{dx} \frac{dx}{dt} = \frac{dF}{dx} v = \sigma \left(1 - \frac{F}{F_c} \operatorname{sgn} v\right)^\alpha v.$$

For  $\alpha = 1$

$$\frac{dF}{dt} = \sigma v - \frac{F}{F_c} |v|.$$

Introduce  $F = \sigma z$  then

$$\frac{dz}{dt} = v - \frac{\sigma |v|}{F_c} z, \\ F = \sigma z.$$

### Dahl's Model - Steady State Properties

$$\frac{dz}{dt} = v - \frac{\sigma |v|}{F_c} z, \\ F = \sigma z.$$

In steady state

$$z = \frac{F_c}{\sigma} \operatorname{sgn} v \\ F = F_c \operatorname{sgn} v$$

The steady state version of Dahl's model is thus Coulomb friction.

### Properties of The Dahl Model

- Simple dynamic model
- Used extensively in simulation studies
- Captures many phenomena
  - Zero slip displacement
  - Hysteresis
- Friction depends only on displacement
- Friction does not depend on velocity
- Friction does not capture Stribeck effect
- Friction does not capture stiction

### The Bliman-Sorine Model

Idea:

- Generalize Dahl to obtain Stribeck effect
- Keep rate independence

The sliding variable

$$s = \int_0^t |v(\tau)| d\tau.$$

The model

$$\frac{dx_s}{ds} = Ax_s + Bv_s \\ F = Cx_s$$

$$A = \begin{pmatrix} -1/(\eta \epsilon_f) & 0 \\ 0 & -1/\epsilon_f \end{pmatrix}, \quad B = \begin{pmatrix} f_1/(\eta \epsilon_f) \\ -f_2/\epsilon_f \end{pmatrix}, \quad C = (1 \quad 1),$$

### Properties of the Bliman-Sorine Model

- Dahl models in parallel
- At least second order dynamics
- Captures many properties
- Captures stiction
- Friction depends on displacement
- Friction does not depend on velocity
- Related to hysteresis models

### The LuGre Model

Idea:

- Generalize Dahl to obtain Stribeck effect
- A first order model
- Rate dependent

The Dahl model

$$\frac{dz}{dt} = v - \frac{\sigma |v|}{F_c} z,$$

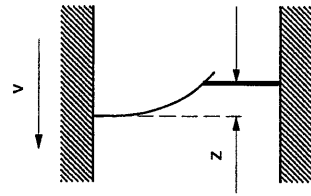
$$F = \sigma z.$$

The LuGre Model

$$\frac{dz}{dt} = v - \sigma_0 \frac{|v|}{g(v)} z,$$

$$F = \sigma_0 z + \sigma_1(v) \frac{dz}{dt} + f(v),$$

### Bristle Interpretation



$$\frac{dz}{dt} = v - \sigma_0 \frac{|v|}{g(v)} z,$$

$$F = \sigma_0 z + \sigma_1(v) \frac{dz}{dt} + f(v),$$

The variable  $z$  in the LuGre model can be interpreted as the average bristle deflection!

### Steady State Properties of the LuGre Model

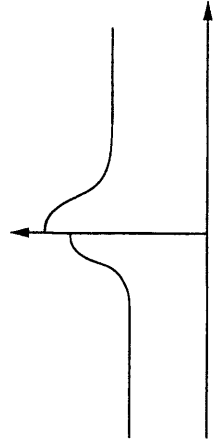
$$\frac{dz}{dt} = v - \sigma_0 \frac{|v|}{g(v)} z,$$

$$F = \sigma_0 z + \sigma_1(v) \frac{dz}{dt} + f(v),$$

In steady state

$$z = \frac{g(v)}{\sigma_0} \operatorname{sgn} v$$

$$F = g(v) \operatorname{sgn} v + f(v)$$





### Properties of LuGre Model

- Almost as simple as the Dahl model
- Captures many aspects of friction
  - Stiction
  - Stick slip
  - Stribeck
  - Hysteresis
  - Zero slip displacement
- But not all
  - Some hysteresis related phenomena
  - Item for research!
- Is passive if damping is velocity dependent
  - Very important for control design

### Effects of Friction in Motion Control

#### Examples

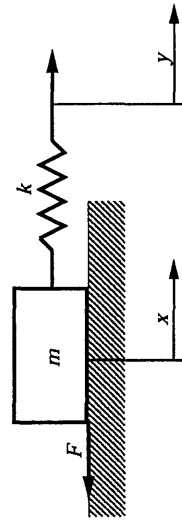
- Stick slip motion
- Inverted pendulum
- Servo systems

#### The basic mechanisms

- Stiction
- An instability mechanism
- Integral action
- Stretching of a spring
- Inverted pendulum

### Stick-slip Motion

A classic phenomena



Equation of motion

$$\frac{d^2 x}{dt^2} = k(y - x) - F$$

### Simulation Based on the LuGre Model

Parameters

- $k = 2$
- $\sigma_0 = 1000$
- $\sigma_1 = 20\sqrt{10}$
- $\sigma_2 = 0.4$
- $F_s = 1.5$
- $F_c = 1$
- $v_0 = 0.1$

$$\frac{d^2 x}{dt^2} = k(y - x) - F$$

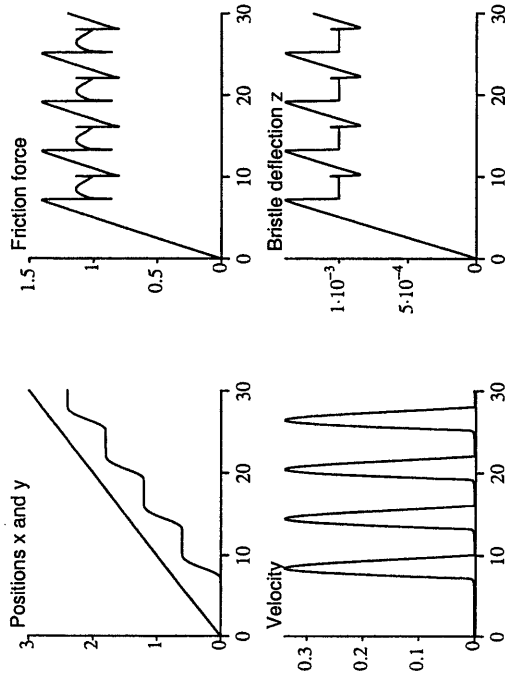
$$y = v_0 t$$

$$F = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 v$$

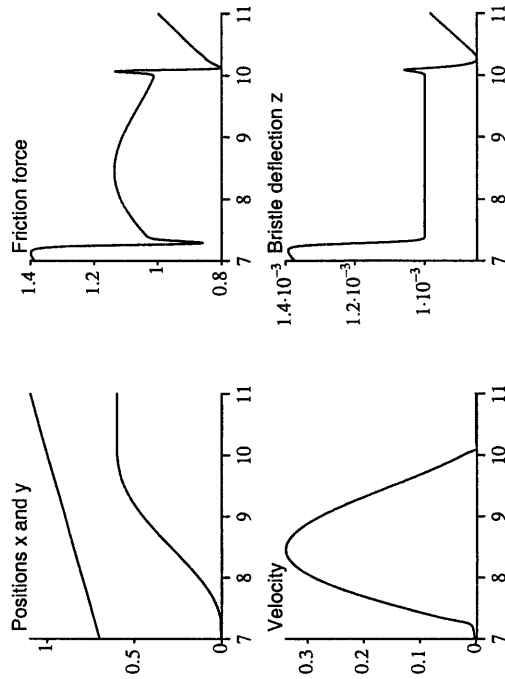
$$\frac{dz}{dt} = v - \frac{|v|}{g(v)} z$$

$$g(v) = \frac{F_c + (F_s - F_c) e^{-v^2/v_0^2}}{\sigma_0}$$

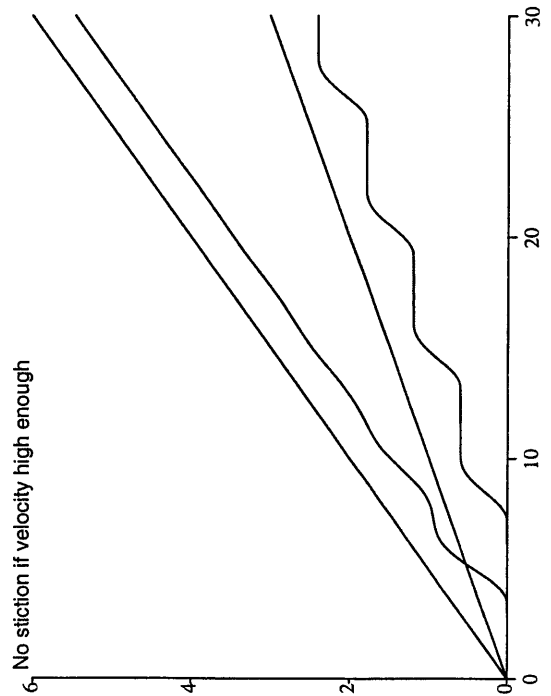
### Simulation Results



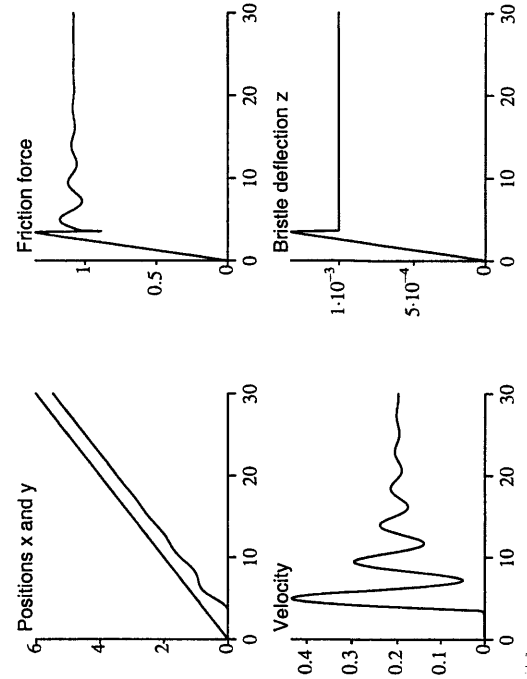
### Details



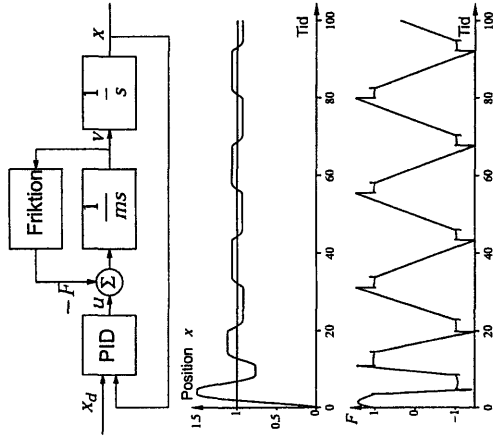
### No Stick-Slip if Tracking Velocity is Increased



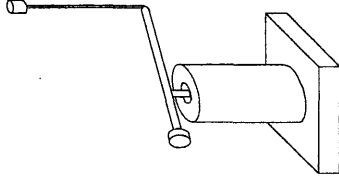
### Details



### Hunting

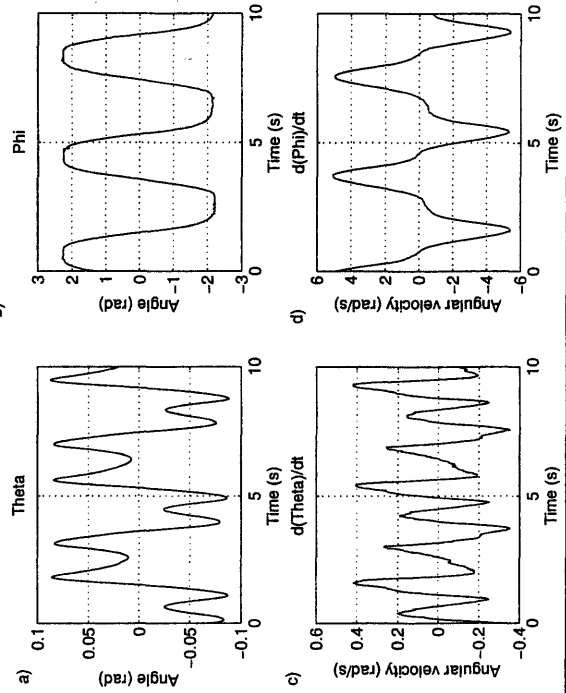


### An Inverted Pendulum

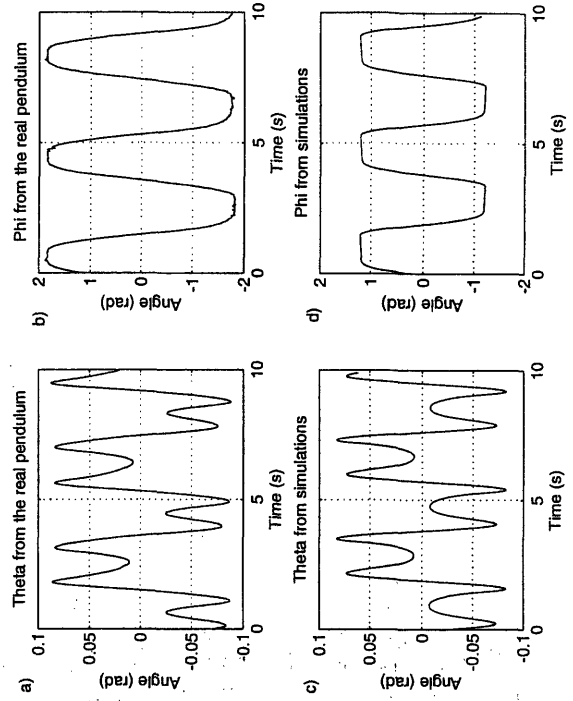


Arm angle  $\varphi$ , pendulum angle  $\theta$ .

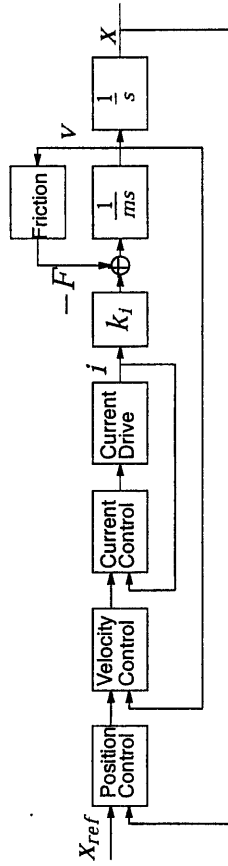
### Effect of Friction



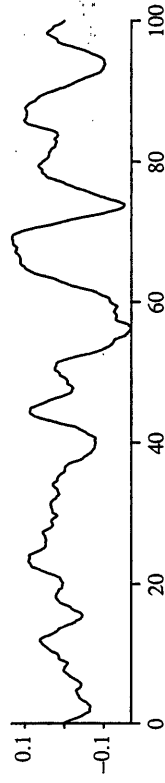
### Comparison with Simulations



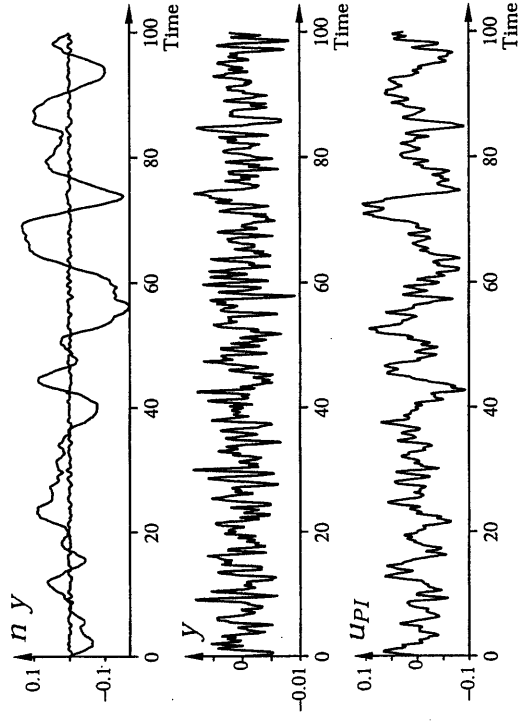
### Servo Systems



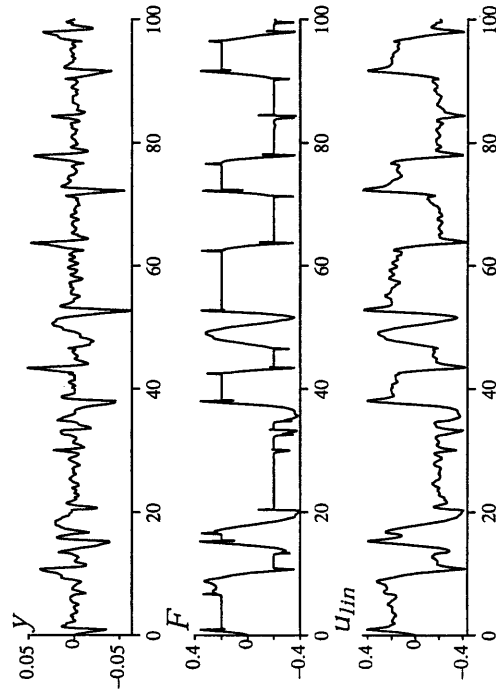
Disturbance signal



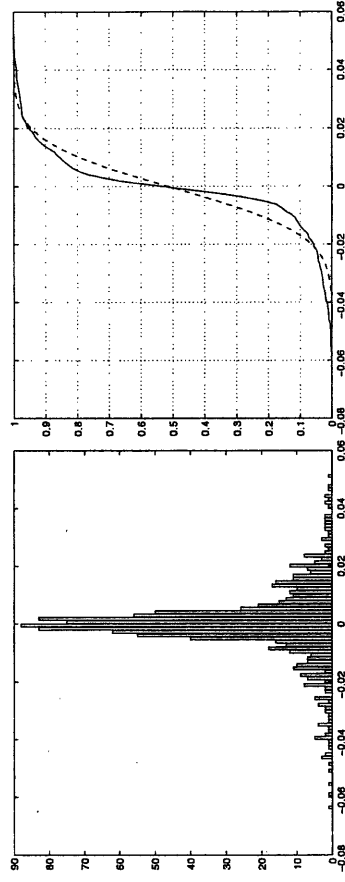
### Without Friction



### Performance Degradation due to Friction

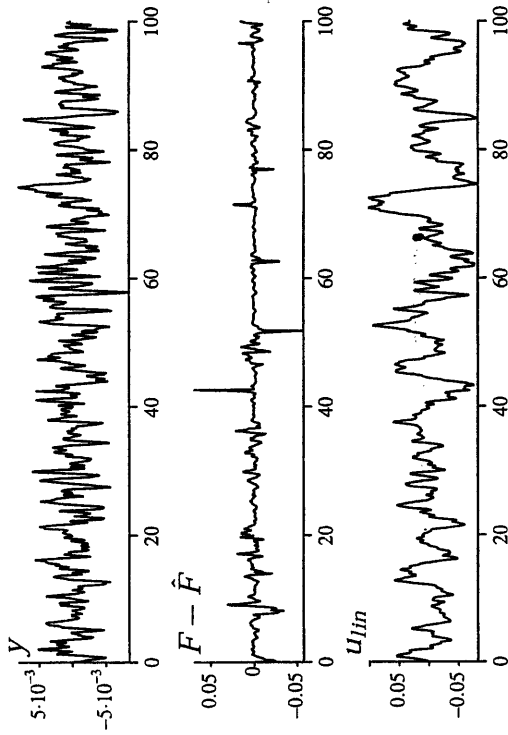


### Probability Distributions



- Non-gaussian distributions
- Fat tails
- Measures such as variance is questionable

### With Friction Compensation



### Conclusions

- A classical field
- Great interest in many disciplines
  - New measurement techniques
  - Essential in all motion control systems
  - Particularly micro-mechanical systems
  - Static and dynamic models
- Dahl, Bliman-Sorine, LuGre
  - Effects of friction on control systems
    - Deterioration of control performance
    - Steady state error
    - Stick slip motion

### References

- J. N. Bhushan, J. N. Israelachvili, and U. Landman. "Nanotribology: Friction, wear and lubrication at the atomic scale." *Nature*, **374**, pp. 607–616, 1995.
- F.P. Bowden and D. Tabor. *The friction and Lubrication of Solids*. Oxford Univ. Press, Part I and II Oxford, 1950 and 1964.
- P.-A. Bliman and M. Sorine. Friction modelling by hysteresis operators. application to Dahl, sticktion and Stribeck effects. In *Proceedings of the Conference "Models of Hysteresis"*, Trento, Italy, 1991.
- C. Canudas de Wit, H. Olsson, K. J. Åström, and P. Lischinsky. "A new model for control of systems with friction." *IEEE Transactions on Automatic Control*, **40:3**, 1995.

### References

- P. Dahl. A solid friction model. Technical Report TOR-0158(3107-18)-1, The Aerospace Corporation, El Segundo, CA, 1968.
- D. A. Haessig and B. Friedland. On the modelling and simulation of friction. *J Dyn Syst Meas Control Trans ASME*, **113(3)**:354–362, September 1991.
- Henrik Olsson. *Control Systems with Friction*. PhD thesis, Lund Institute of Technology, University of Lund, 1996.
- Ernest Rabinowicz. *Friction and wear of materials*. New York: Wiley, second edition, 1995.
- Nam P. Suh. *Tribophysics*. Englewood Cliffs, N.J.: Prentice-Hall, 1985.