Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

DYNAMIC FRICTION MEASUREMENT, MODELING, AND COMPENSATION FOR PRECISE MOTION CONTROL

by
Simon Cohn

In this thesis, measurements of dynamic friction in a hydrodynamic journal bearing were performed for varying sinusoidal velocity excitations, loads, and lubricants. The results indicate that the friction data displays a negative slope in the mixed region of friction vs. velocity (f-v) curves, and also shows that the dynamic friction is not just a function of current velocity, but also a function of velocity history (hysteresis). These results are in agreement with previous experimental investigations by other investigators in lubricated friction.

Secondly, a dynamic friction model is fully explored and partially extended to provide quantitative agreement to measured friction values. A contribution to friction modeling was made by reducing the model from a fourth to a second order equation. Parameters were determined for one lubricant and two normal loads, and the model output is compared to experimental data.

Finally, model-based friction compensation was successfully performed. The dynamic friction model is used as a basis for velocity and position control of an apparatus with high friction by incorporating a function to constantly learn two parameters of the model. Results demonstrate the feasibility of using a rich friction model in real time, and its ability to greatly reduce the tracking errors caused by friction.

This thesis was supported by the National Science Foundation under Grant MSS-9215636.

DYNAMIC FRICTION MEASUREMENT, MODELING, AND COMPENSATION FOR PRECISE MOTION CONTROL

by
Simon Cohn

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Department of Mechanical Engineering

January 1998

APPROVAL PAGE

DYNAMIC FRICTION MEASUREMENT, MODELING, AND COMPENSATION FOR PRECISE MOTION CONTROL

Simon Cohn

Dr. Avraham Harnoy, Thesis Advisor — Date
Professor of Mechanical Engineering, NJIT

Dr. Bernard Friedland, Committee Member — Date
Distinguished Professor of Electrical Engineering, NJIT

Dr. Zhiming Ji, Committee Member — Date
Associate Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Simon Cohn

Degree: Master of Science in Mechanical Engineering

Date: January 1998

Undergraduate and Graduate Education:

- Master of Science in Mechanical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1998

- Bachelor of Science in Aerospace Engineering,
  Boston University, College of Engineering, Boston, MA, 1994

Major: Mechanical Engineering

To my father and mother, without whose sacrifice
and support I would not have been able to achieve all that I have

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor, Dr. Avraham Harnoy, for his guidance, advise, suggestions, compassion, and patience in all matters which related to my research work, including the writing of this thesis.

Special thanks to Professor Bernard Friedland for giving me insight and guidance with the controls engineering aspect of this thesis and for serving as a committee member. Thanks also to Associate Professor Zhiming Ji for serving as a member of my thesis committee on such short notice.

An earnest thank you goes out to Jayesh Amin, who helped me with the controls and electrical side of my work, and who selflessly helped me even after his own graduation.

I am also appreciative to the Department of Mechanical Engineering for providing me with financial support during my Masters studies and research work. Furthermore, this investigation was supported by the National Science Foundation under Grant MSS-9215636.

Finally, I would like to thank both friends and family for their moral support and understanding during the past several years; especially during the times when many were neglected while I concentrated on this thesis.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background and Objective

Friction is a force which exists at all interfaces where two surfaces are in contact with one another. In many cases, friction can be a stabilizing force; imagine what it would be like to slide across a hardwood floor while wearing socks if there was no friction. However, for an engineer who wishes to control the movement of a machine precisely, friction is usually an obstacle which must be overcome. This is because friction either causes errors in tracking, or completely destabilizes a system.

Much research has been done into the nature of friction, as evidenced by the large number of papers dealing with the subject in the tribology and control literature. Much work has also been done by controls engineers in solving the problems associated with friction. A survey paper of the friction phenomenon, and its effect on precise motion control, is presented by Armstrong-Hélouvry et al. (1994), and is an excellent starting point for information on the subject.

The objective of this thesis is threefold. First, experiments were conducted to explore the nature of dynamic friction in hydrodynamic journal bearings. Quantitative examination included measurement of dynamic friction for two normal loads with one lubricant, as well as the use of another lubricant and no lubricant. Tests were conducted for sinusoidal velocity inputs in both uni-directional and bi-directional modes in order to create accurate plots of friction vs. velocity.

Second, a dynamic friction model for hydrodynamic journal bearings - developed by Harnoy and Friedland (1993) - was simulated for the purpose of determining proper parameters, and to investigate the basic accuracy of the model. Modifications to the model were also suggested and incorporated to give the model greater flexibility, and to make the model parameters easier to determine. Also, a reduced order (second order) model was developed from the original fourth order model to speed calculation for incorporation in a control program.

Lastly, this thesis ties together some tribology with controls engineering by investigating the feasibility of using the above mentioned second order dynamic friction model to estimate the forces of friction in a system on-line, and thereby offset friction's detrimental effects by feedback compensation. In other words, the purpose was to measure the improvement in performance due to the friction compensation with respect to some baseline control system.

1.2 Outline

Chapter 2 is a brief survey of the friction phenomenon. It describes friction from the simplified models first proposed in the earlier stages of understanding, to the fuller models currently available. Chapter 2 also lists some differing approaches used in the control engineering field to cancel frictional forces in machines.

Chapter 3 describes the dynamic friction model proposed by Harnoy and Friedland (1993) in its original form. It further goes on to extend the model to better fit experimental results, and to promote an easier determination of model parameters. This chapter also describes the steps necessary for successful simulation.

Chapter 4 is a presentation of experimental measurements of dynamic friction in hydrodynamic plain bearings when using 10W-40 motor oil. Research was conducted for varying normal loads and velocity inputs. Also, the friction model simulations are compared to experimental results, and parameters chosen for best fit.

Chapter 5 introduces experimental measurements of dynamic friction when using a low viscosity lubricant. This chapter also presents results when using no lubricant at all.

Chapter 6 deals with the use of the friction model for the purpose of friction cancellation in real time. The model was incorporated into a control program written in the C programming language, and two of the model parameters were constantly updated in order to offset the effects of inexact friction cancellation. Experiments for velocity control resulted in a remarkable improvement compared to the baseline control system with no friction compensation. Experiments for position control also showed a dramatic reduction in error compared to the baseline control system.

Chapter 7 summarizes the work in this thesis and gives some conclusions. It also discusses some of the improvements which are possible, and suggests a direction for future work on the subject.

CHAPTER 2

A SURVEY OF FRICTION AND FRICTION MODELS

Friction is present in all machines in which there is relative movement. It exists in machine guideways, screw drives, motors, belts, gears, rolling element and journal bearings, seals, etc. All these forces of friction serve to impede the precise control of machinery. For example, this is the case in the positioning of tank turrets, milling machines, robots, and deflective surfaces in aircraft. The deleterious effects of friction can be reduced through the use of appropriate lubricants and by mechanical means which increase costs. On the other hand, friction may be compensated through the use of a control scheme. This is where the development of a good friction model is most important.

2.1 Behavior and Characteristics of Friction

The nature of friction has been studied for over 400 years, with the first known study performed by Leonardo Da Vinci. Since then, numerous experimental investigations have uncovered the rich, dynamic nature of friction, and this has led many to propose models - both based on physical principles and on experimental results - to imitate and predict the forces of friction.

2.1.1 The Classical Friction Model

A study of the friction phenomenon was performed by Leonardo Da Vinci (1519). Da Vinci knew that friction is a force which always opposes the direction of motion and is

proportional to the normal load. This phenomenon was rediscovered by Amontons (1699), and then further developed by Coulomb (1785). This model of friction has been termed Coulomb friction, and can be written as:

$$F = -\mu N \operatorname{sgn}(V)$$

where F is friction force, μ is the friction coefficient, N is normal force, and V is velocity. Since then, two other facets of the classical friction model have been added; static friction was introduced by Morin (1833), and viscous friction was proposed by Reynolds (1886). Together, these three parts came together as the static + Coulomb + viscous friction model, which is still taught as the de facto friction phenomenon in most college level physics courses, even though it is a simplification of the true nature of friction. (*see* Fig. 2.1)

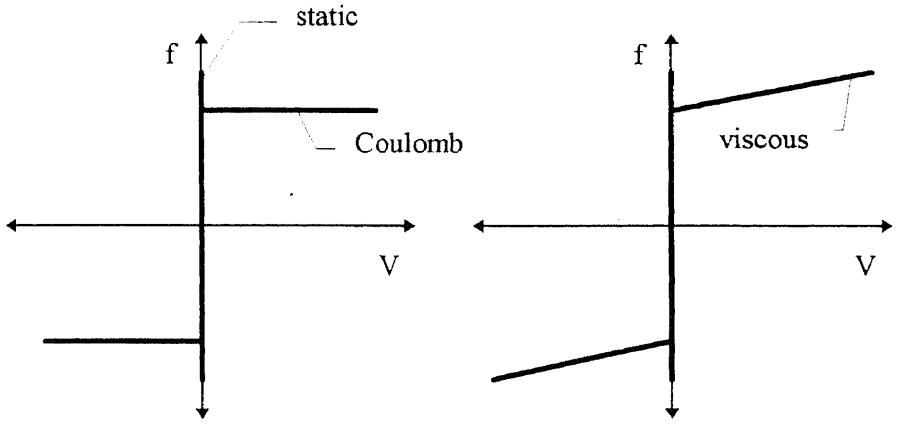


Figure 2.1 Classical friction models; (a) static + Coulomb friction, (b) static + Coulomb + viscous friction.

2.1.2 Stribeck Friction

The above mentioned classical friction model fails to predict an important part of the friction - velocity (f-v) curve in lubricated contacts. Stribeck (1902) was first to notice

that in very low velocities, after static friction has been overcome, friction force decreases with increasing velocity. This has been termed negative viscous friction, or the Stribeck effect.

The physical explanation of this can be reasoned as follows: When velocity is rising from zero, the lubricant is trapped between the asperities, and the normal load is supported entirely by the solid metal contacts (*see* figure 2.3). The velocity region where the entire load is carried by the surface asperities is termed the boundary lubrication regime. As velocity rises, the fluid film slowly builds and begins to support the weight of the normal load up until a critical transition velocity, U_{tr}, where the normal load is just fully supported by the fluid film. During this regime, termed the mixed lubrication regime, the friction force decreases because of the lower shearing forces of the lubricant compared with the deformation of the solid contacts. Above U_{tr} is the fully developed hydrodynamic regime. Here, the friction slowly rises in proportion to the velocity, and is entirely due to viscous shearing forces. Fig. 2.2 shows the Stribeck curve with both limited and substantial boundary lubrication.

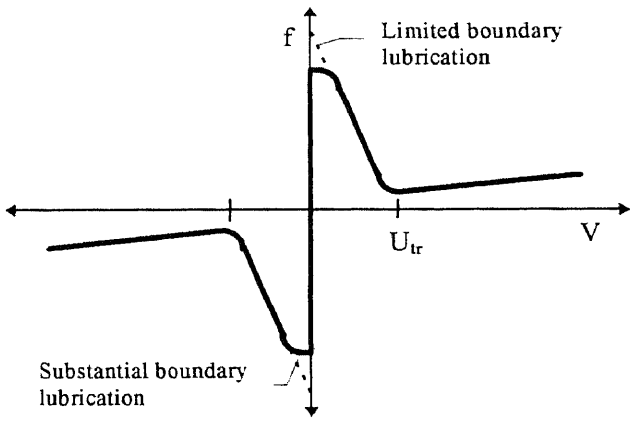


Figure 2.2 The Stribeck curve.

A key issue amongst the tribology community is the nature of boundary lubrication. Stribeck (1902), Biel (1920), Czichos (1978), Hamrock and Dowson (1981), and Fuller (1984) show constant friction at very low velocities as delineated by substantial boundary lubrication. In contrast, the works of Bell and Burdekin (1969), Rachoor (1996), and the present work show experimental measurements of friction-velocity curves with limited boundary lubrication.

From the controls standpoint, it is the negative slope of the *f-v* curve in the mixed region which causes instability and promotes limit cycles in the form of stick-slip. For example, take a block on a lubricated plate which needs to be moved a very small distance by a proportional-integral (PI) control system. When the force is applied from the side by the control, the block will not move until the side force integrates to a large enough value to overcome static friction. However, as the block begins to move, the friction force drops very rapidly, and the side force becomes greater than the frictional force. This causes the block to move too far, and now the control system will try to move the block in the other direction to compensate. Theoretically, the block will never reach its desired position, but will instead enter a stable limit-cycle, continuously hunting back and forth looking for the desired position.

2.1.3 Pre-sliding Displacement

While studying the behavior of friction during small displacements of ball bearings, Dahl (1977) observed that at zero velocity, friction is not a constant force, but is instead a spring-like force which opposes motion and is approximately proportional to the displacement. This Dahl effect can be explained by the microscopic asperities which

form the contact between two objects as shown in Figure 2.3. At zero velocity, as force is applied, the asperities deform without sliding relative to one another up to a critical force, at which breakaway occurs. (*see* also Futami et al. 1990)

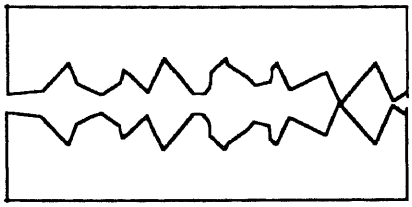


Figure 2.3 A close-up view of surface asperities.

2.1.4 Frictional Memory

The static + Coulomb + viscous curve and the Stribeck curve both illustrate the behavior of friction during static conditions. It is reasonable to predict that for a given system, friction is a single-valued function of velocity, but experimental results by Bell and Burdekin (1969), Hess and Soom (1990), Polycarpou and Soom (1992 and 1995), Harnoy et al. (1994), Rachoor (1996), Amin (1996), and Amin et al. (1996) have shown that changes in the force of friction lag any change in velocity. For velocities in the mixed lubrication regime - the range of velocity in which the normal load is supported partly by solid contact and partly by hydrodynamic lubrication film - the results are hysterisis loops, in which the force of friction for a given velocity is lower when approached from a higher velocity that when approached from a lower velocity (*see* Fig. 2.4).

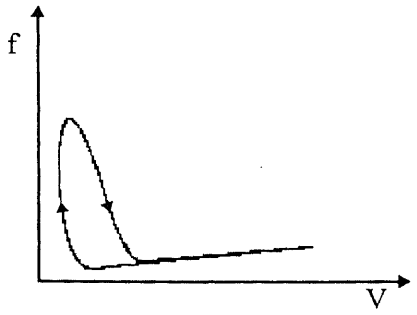


Figure 2.4 Friction vs. Velocity curve for lubricated contacts of the character observed by Hess and Soom (1990), Harnoy et al. (1994), etc. showing hysterisis.

Within the mixed lubrication regime, as velocity increases, it takes a finite time for the fluid film to form, and the friction force slowly decreases. When the velocity is decreasing, the film is already at least partially formed, and it takes some time for the lubricant to be squeezed out. For sinusoidal velocity excitations, experimental results show that the width of the hysterisis loops increases with increasing frequency.

2.1.5 Rising Static Friction

Rising static friction is a phenomenon where static friction, f_s, the maximum frictional force at zero velocity, rises with increasing time at zero velocity. This behavior was observed in Bell and Burdekin (1969) in the measurement of friction in machine tool guideways. The authors observed rising static friction for both lubricated and non-lubricated contacts.

What is very interesting is that, at least for non-lubricated conditions, Bell and Burdekin (1969) noticed that when static friction rose with increasing time of stick, t_s, the kinetic friction force, f_k, also rose by an equal amount. Therefore, they determined that although static friction rises with increasing time of stick, the value $f_s - f_k$ remains

constant. This is a very important conclusion for the formulation of dynamic friction models. For example, contrast a rising static friction model with constant f_s - f_k, with another model in which only f_s rises with t_s, and f_k is held constant. For the former model with constant f_s - f_k, when a very low desired velocity is imparted, a stable limit-cycle is predicted. However, for the second model with f_k held constant, motion would start with stick-slip, but as time progressed, the time at stick would continuously drop, leading f_s to drop as well, until f_s was equal to f_k, and steady sliding was achieved.

2.1.6 A Full Description of Lubricated Friction

The following is adapted from Armstrong-Hélouvry (1991) and Armstrong-Hélouvry and Dupont (1993), and summarizes all of the important characteristics of lubricated friction.

1. The Four Velocity Regimes:

   I. Static Friction, Steady State Velocity = 0: Friction is a function of position, not velocity. Friction is a constraining force up until breakaway.

   II. Boundary Lubrication, Steady State Velocity << Transition Velocity (U_{tr}): Friction relies on surface and lubricant properties. Note that no boundary lubrication was evident in the experimental portion of this work.

   III. Partial Fluid Lubrication, Steady State Velocity > Boundary Lubrication, but < U_{tr}: Friction decreases with increasing velocity unless special way lubricants are used, which promote lower static friction than Coulomb friction.

   IV. Full Fluid Lubrication, Steady State Velocity > U_{tr}: Friction is proportional to velocity and is entirely caused by shearing of the fluid lubricant.

2. Two Time Dependent Properties:

I. Rising Static Friction: Static friction increases with dwell time at zero velocity. For example, Kato et al. (1972) proposed an empirical model relating static friction to dwell time of the form:

$$F_s(t) = F_{s\infty} - (F_{s\infty} - F_k)e^{-\gamma t^m}$$

where $F_{s\infty}$ is the ultimate static friction; F_k is the kinetic friction at the moment of arrival in the stuck condition; and γ and m are empirical parameters.

II. Frictional Lag: In partial fluid lubrication, changes in friction lags changes in velocity. Note that there are different ways of modeling this phenomenon, ranging from a pure time delay between changes in velocity and changes in friction, to physical principle models, such as described in the next chapter.

2.2 Static Friction Models

The simplest of friction models, Coulomb friction, can be mathematically modeled as

$$F = -C\,\mathrm{sgn}(V).$$

For rolling element bearings, when surfaces which are rolling are separated by a lubricant film, SKF (1991) suggests that frictional moment can be calculated from $M = M_0 + M_1 + M_3$, where M_0 is the load-independent term which is dependent mostly on lubricant quality and viscosity; M_1 is the load-dependent frictional moment, and arises from both elastic deformations and partial sliding in the contacts; and M_3 is the frictional moment due to seals (if applicable). Equations for M_0, M_1, and M_3 can be found in SKF (1991), and are highly based on the empirical results for specific bearing arrangements.

The Stribeck curve, and variants of it (*see* fig 2.2), can be modeled by an exponential model proposed by Bo and Pavelescu (1982), mostly based on Tustin's (1947) model, with a viscous term added by Armstrong-Hélouvry et al. (1994), and the absolute value and sgn(v) term added by this author.

$$F(v) = [F_C + (F_s - F_C)e^{-(v/v_c)^\delta} + F_v|v|]\,\mathrm{sgn}(v)$$

Here, F_s is the level of static friction; F_C is the minimum level of Coulomb friction; and v_c and δ are empirical parameters. Bo and Pavelescu (1982) found δ to range from 0.5 to 1, while Armstrong-Hélouvry (1991) takes δ equal to 2.

2.3 Dynamic Friction Models

There are two approaches to creating dynamic friction models. The first way involves experimentally measuring friction, and then finding a mathematical model to fit the data. The second method involves the modeling of physical occurrences which contribute to frictional forces, and subsequent tuning of physical parameters. Examples of models which stem from both approaches are presented in this section.

2.3.1 Empirical Models

The Seven Parameter Friction Model proposed by Armstrong-Hélouvry and Dupont (1993) attempts to capture the friction aspects of pre-sliding displacement, sliding, and rising static friction. This model can, at least qualitatively, predict all of the phenomena observed by Polycarpou and Soom (1992) in their measurements of friction. The model is given by:

Pre-Sliding Displacement:

$$F_f(x) = -k_t x$$

Sliding (Coulomb + viscous + Stribeck with frictional memory):

$$F_f(\dot{x},t) = -\left(F_C + F_v|\dot{x}| + F_S(\gamma,t_2)\frac{1}{1+\left(\frac{\dot{x}(t-\tau_L)}{\dot{x}_S}\right)^2} \right)$$

Rising Static Friction (Friction at Breakaway):

$$F_S(\gamma,t_2) = F_{S,a} + (F_{S,\infty} - F_{S,a})\frac{t_2}{t_2+\gamma}$$

where:

$F_{f(\bullet)}$ is the instantaneous friction force

F_C (*) is the Coulomb friction force

F_v (*) is the viscous friction force

F_S is the magnitude of the Stribeck friction (frictional force at breakaway is $F_C + F_S$)

$F_{S,a}$ is the magnitude of the Stribeck friction at the end of the previous sliding period

$F_{S,\infty}$ (*) is the magnitude of the Stribeck friction after a long time at rest (with a slow application of force)

k_t (*) is the tangential stiffness of the static contact

$\dot{x}_S$ (*) is the characteristic velocity of the Stribeck friction

τ_L (*) is the time constant of frictional memory

γ (*) is the temporal parameter of the rising static friction

t_2 is the dwell time, time at zero velocity

(*) marks friction model parameters, other variables are state variables

To the best of this author's knowledge, no one has yet attempted to use the Armstrong-Hélouvry & Dupont model as a basis for feedforward friction compensation.

In Olsson (1996), a new friction model is introduced in which the frictional forces from the solid-to-solid contact of the surface asperities is considered. Here, the contact is modeled as elastic bristles which deform during motion and small displacements.

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

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

$$g(v) = \frac{1}{\sigma_0}\left(F_C + (F_S - F_C)e^{-\left(\frac{v}{v_s}\right)^2}\right)$$

$$\sigma_1(v) = \sigma_1 e^{-(v/v_d)^2}$$

Here, F is the total dynamic friction force, $\sigma_0 > 0$ is the stiffness of an average bristle, $\sigma_1(v) > 0$ a velocity dependent damping coefficient, z is the deflection of the bristle, and $f(v)$ is the viscous friction term.

2.3.2 A Physical Model

Models to predict friction based on physical principles are more rare than models based on experimental results of friction force. However, physical models have perhaps the most promising potential for use in friction compensation.

A dynamic model for lubricated short journal bearings was developed by Harnoy and Friedland (1993). This model, which covers the full regime of velocities, including boundary, mixed, and hydrodynamic regimes, takes into account the deformation of the contact asperities, as well as the load carrying capacity of the fluid film. The total, dimensionless friction force is of the form:

$$\overline{F}_f = f_m \overline{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{tr})\Delta + \frac{CR}{L^2}\frac{2\pi}{(1-\varepsilon^2)^{0.5}}\overline{U}$$

where the second term describes the friction arising from hydrodynamic lubrication, and is proportional to dimensionless velocity, $\overline{U}$. The first term is the friction force arising from the contact with the metal asperities, and only makes a contribution to the total friction at that time. Much of this thesis is based on this friction model, and more details will be presented in the next chapter.

2.4 Friction Compensation Examples

The following section serves to give some examples of the various techniques in use today and proposed for future controllers of systems with friction. Providing an exhaustive survey on friction compensation is obviously out of the scope of this text, but this section gives insight into the many methods of dealing with friction in systems.

2.4.1 Non-Model Based Friction Compensation

A look-up table was used by Armstrong-Hélouvry (1991) to control the motion at joint one of a PUMA 560 industrial robot. In his experiments, Armstrong-Hélouvry collected data on break-away friction torque as it related to position within a five radian span of the arm. He used 2,000 position bins per radian of arm rotation, and collected 60,000 data points to allow averaging and improved accuracy. The break-away data was then low pass filtered to 20 cycles per radian and compressed into a look-up table consisting of only 200 bins per radian.

The results of the look-up table, in conjunction with kinetic friction compensation, were used for velocity control of the arm at a moderate rate of 1 radian per second. Results show remarkable improvement of accuracy in velocity obtainable, from a peak error of approximately 0.08 rad/s using constant torque control, to approximately 0.03 using the look-up table.

Repetitive control, a subset of learning control theory was used by Tung et al. (1993) to compensate for the stiction forces arising during the machining of circular shapes with an end milling machine. The workpiece being machined is clamped onto a X-Y bed which moves by way of two orthogonally oriented ball screw drives. While following a circular contour, the bed must reach very low velocities during velocity reversals. Essentially, four evenly spaced tracking errors, called *quadrant glitches*, are produced 90 degrees apart.

Repetitive control is a method by which the machine improves its motion through practice. It is useful for machines which perform repetitive tasks, like an industrial robot performing a pick-and-place operation. Tung et al. (1993) used repetitive control to get nearly perfect tracking through velocity reversals. Then, by analyzing the feedforward portion of the control, they were able to construct a look-up table for future use in order to eliminate the learning of the controller for each new application.

A digital feedforward controller incorporating the Zero Phase Error Tracking Controller (ZPETC) and feedforward friction compensation was used successfully in Tung et al. (1996) on a CNC end milling machine. They were able to diminish errors to ± 3 μm over a 26 mm trajectory at a maximum speed of 2 m/min, and achieve a maximum contouring error of less than 4 μm for a 26 mm diameter circle.

Yao and Tomizuka (1995) also considered adaptive motion and force control for the application of a robot manipulator in the presence of parametric uncertainties, including friction, and simulated their results for a two degree of freedom (DOF) direct drive planer SCARA robot.

Canudas de Wit and Seront (1990) consider the topic of inexact friction cancellation, and propose a new method for designing linear compensators to increase the robustness of the closed-loop system.

2.4.2 Model Based Friction Compensation

To the present day, model based friction compensation alone is usually not sufficient for position or velocity control. Although many models of friction exist, their use has primarily been for the purpose of system identification during controller design. Much of the problem is that friction models exhibiting all of the relatively important aspects of dynamic friction contain numerous parameters which are not known. For the reason of simplicity, Coulomb friction is the most widely used friction model for compensation. This sub-section gives one example of its use.

Coulomb friction by itself does not capture the downward slope of the f-v curve at low velocities. In Amin (1996), a nonlinear observer from Friedland and Park (1992) was used to estimate the Coulomb friction coefficient on-line. The observer design is given as:

$$\hat{F} = \hat{a}\,\mathrm{sgn}(\hat{v})$$
$$\hat{a} = z_F - k_F|\hat{v}|^{\mu}$$
$$\dot{z}_F = k_F \mu |\hat{v}|^{\mu-1}(u - \hat{F})\,\mathrm{sgn}(\hat{v})$$

where:

F	Frictional force
a	Coulomb friction constant
v	Velocity
u	Control input
k_F	Friction observer gain
μ	Experimental parameter
(^)	Denotes an estimate

This friction observer makes use of the estimated velocity of the system. The velocity estimate is derived from the measured position by a separate observer fully described in Tafazoli et al. (1995) and presented in Figure 2.5. In essence, this velocity observer acts as a low-pass differentiator, removing the unwanted noise usually associated with taking derivatives in real systems. The value, Kv, is the velocity observer gain, with higher values of Kv increasing the bandwidth of the low-pass.

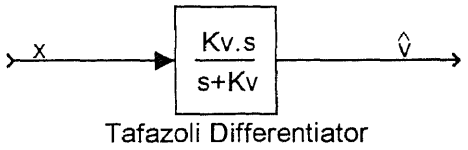


Figure 2.5 Block diagram of the low-pass differentiator proposed in Tafazoli et al. (1995)

Although the Coulomb friction model is simple, when matched with the friction observer, it was shown to track both the negative slope in the boundary lubrication regime, and the hysterisis of the friction during sinusoidal velocity excitations. Results using this type of compensation showed as much as a three-fold reduction in RMS error

for position control, and as much as eight-fold reduction in RMS error for velocity control.

CHAPTER 3

A DYNAMIC FRICTION MODEL

This chapter describes the dynamic friction model presented in Harnoy and Friedland (1993) for short hydrodynamic journal bearings. After introduction of the model in its original form, modifications are proposed to facilitate simulation, and to better fit the experimental results presented in the next chapter. Also, the method of simulation and parameter identification is explained in this chapter.

3.1 Original Form of the Dynamic Friction Model

The dynamic friction model for short hydrodynamic journal bearings used for simulation and comparison to experimental results was proposed by Harnoy and Friedland (1993) and was also extensively studied in Rachoor (1996). In this section, a condensed presentation of the model in its original form is made.

An illustration of a short journal bearing is presented in Figure 3.1. In this friction model, it is assumed that at higher rotational speeds, the entire normal load is supported by a hydrodynamic fluid film. As velocity is lowered into the mixed lubrication regime, part of the load is carried by a solid contact made between surface asperities of the shaft and journal sleeve (*see* figure 2.3 for a close-up sketch of the surface asperities). The deformation of the surface asperities is in the direction going through the point of contact and the center of the bearing (X-direction in figure 3.1). The deformation of the asperities, δ, can be rewritten as:

$$\delta = C(\varepsilon - \varepsilon_{tr})$$

where C is the average circumferential clearance between shaft and bearing; ε is the relative eccentricity, defined as $\varepsilon = e/C$; e is the journal eccentricity defined in figure 3.1; and ε_{tr} is the relative eccentricity at the transition between full fluid lubrication and the mixed lubrication regime. The velocity at which ε is equal to ε_{tr} is called the transition velocity, U_{tr}.

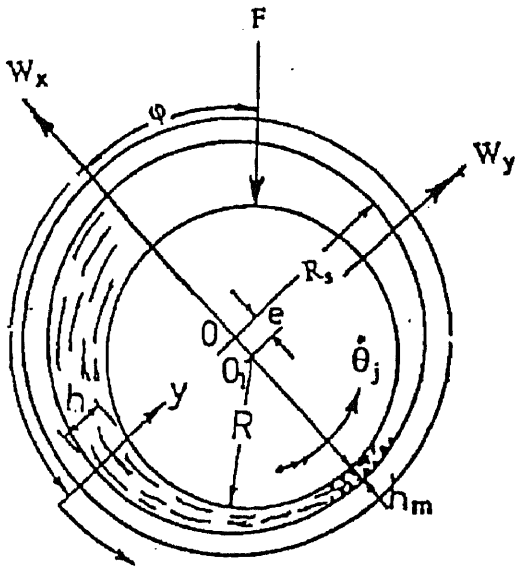


Figure 3.1 Cross-section of a hydrodynamic journal bearing

The elastic reaction force in the X-direction, W_e, of the surface asperities is $W_e = K(\delta)\delta$. This can be rewritten in terms of C, ε, and ε_{tr} as:

$$W_e = \kappa(\varepsilon)C(\varepsilon - \varepsilon_{tr})\Delta$$

where $\kappa(\varepsilon) = K(\delta)$, and Δ is defined as:

$$\Delta = \begin{cases} 0, for \varepsilon < \varepsilon_{tr} \\ 1, for \varepsilon > \varepsilon_{tr} \end{cases}$$

In order to present the modeled friction coefficient, it is first necessary to introduce several dimensionless parameters:

$$\overline{F} = \frac{C^2}{\mu U_{tr} L^3} F \qquad \text{Dimensionless normal load} \tag{3.1}$$

$$\overline{\kappa}(\varepsilon) = \frac{C^3}{\mu U_{tr} L^3} \kappa(\varepsilon) \qquad \text{Dimensionless equivalent stiffness of the surface asperities} \tag{3.2}$$

$$\overline{U} = \frac{U}{U_{tr}} \qquad \text{Dimensionless shaft velocity} \tag{3.3}$$

$$\overline{m} = \frac{C^3 U_{tr}}{\mu L^3 R^2} m \qquad \text{Dimensionless rotating mass} \tag{3.4}$$

where: C is the average journal clearance; μ is the viscosity of the lubricant, assumed constant; U_{tr} is the circumferential velocity of the shaft when $\varepsilon = \varepsilon_{tr}$; L is the journal length; F is the external normal load; $\kappa(\varepsilon)$ is the equivalent stiffness of the surface asperities; U is the tangential shaft speed; and m is the rotating mass.

During static conditions, it is possible to relate the external load, F, to the reaction forces W_x and W_y in Figure 3.1. The component W_x is equal to the sum of the hydrodynamic force component due to the fluid film pressure, and the elastic reaction, W_e, due to contact of the asperities. W_y^* is due to hydrodynamic pressure only. The equations describing this force balance, in dimensionless form, were found to be:

$$\overline{F}\cos\varphi = \overline{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{tr})\Delta - 0.5 J_{12}\varepsilon\overline{U} \tag{3.5}$$

$$\overline{F}\sin\varphi = 0.5 J_{11}\varepsilon\overline{U} \tag{3.6}$$

Finally, two differential equations in dimensionless form were derived for ε and φ (from Figure 3.1), describing the journal motion during dynamic conditions:

$$\overline{F}\cos\varphi = \overline{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{tr})\Delta - 0.5J_{12}\varepsilon|\overline{U}| + J_{12}\varepsilon\dot{\varphi} + J_{22}\dot{\varepsilon} + \overline{m}\,\ddot{\varepsilon} - \overline{m}\,\varepsilon\dot{\varphi}^2 \tag{3.7}$$

$$\overline{F}\sin\varphi = 0.5J_{11}\varepsilon|\overline{U}| - J_{11}\varepsilon\dot{\varphi} - J_{12}\dot{\varepsilon} - \overline{m}\,\varepsilon\ddot{\varphi} - 2\overline{m}\,\dot{\varepsilon}\dot{\varphi} \tag{3.8}$$

The dimensionless frictional force for dynamic conditions was derived as:

$$\overline{F}_f = f_m\overline{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{tr})\,\mathrm{sgn}(\overline{U})\Delta + \frac{CR}{L^2}\frac{2\pi}{(1-\varepsilon^2)^{0.5}}\overline{U} \tag{3.9}$$

In equations 3.7 and 3.8, the absolute value sign around the $\overline{U}$, and the sgn($\overline{U}$) term in equation 3.9 are from a correction to Harnoy and Friedland (1993) presented by Haessig (1995). The dynamic friction coefficient is then defined as:

$$f = \frac{\overline{F}_f}{\overline{F}} \tag{3.10}$$

Equations 3.7 - 3.9 characterize the dynamic friction model as described in Harnoy and Friedland (1993). To simulate the model, equations 3.7 and 3.8 are solved for $\ddot{\varepsilon}$ and $\ddot{\varphi}$, respectively, and then integrated twice for each time step of the simulation. The three variables J_{11}, J_{12}, and J_{22} in equations 3.5 - 3.8 are defined as:

$$J_{11} = \int_0^{\pi}\frac{\sin^2\theta}{(1+\varepsilon\cos\theta)^3}d\theta = \frac{\pi}{2(1-\varepsilon^2)^{3/2}} \tag{3.11}$$

$$J_{12} = \int_0^{\pi}\frac{\sin\theta\cos\theta}{(1+\varepsilon\cos\theta)^3}d\theta = \frac{-2\varepsilon}{(1-\varepsilon^2)^2} \tag{3.12}$$

$$J_{22} = \int_0^{\pi}\frac{\cos^2\theta}{(1+\varepsilon\cos\theta)^3}d\theta = \frac{\pi}{2}\frac{(1+2\varepsilon^2)}{(1-\varepsilon^2)^{5/2}} \tag{3.13}$$

3.2 Modifications to the Model

When the model was simulated and compared to experimental results, it was found that in order to fit the slope of the *f-v* curve in the fully lubricated region, an unrealistically high value of lubricant viscosity, μ, needed to be chosen. Unfortunately, choosing a high viscosity also changed the characteristic of the model curve in the mixed lubrication regime to such a degree that a good match between theory and experiment could not be achieved. To overcome this, a factor, γ, was introduced in equation 3.9 to yield:

$$\overline{F}_f = f_m \overline{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{tr}) \operatorname{sgn}(\overline{U})\dot{\Delta} + \gamma \frac{CR}{L^2} \frac{2\pi}{(1-\varepsilon^2)^{0.5}} \overline{U} \tag{3.14}$$

Actually, the factor γ is more than just an arbitrary variable to fit the model to experimental results. In its original derivation, the model was derived for a very short bearing. Therefore, it is not illogical that an additional factor be added to account for the medium length of the bearing (0.01905 m for the apparatus used in this thesis). For a given lubricant viscosity, the slope factor, γ, was easily determined from experimental measurements of friction, and stayed essentially constant.

Next, a modification in the calculation of the equivalent stiffness coefficient of the surface asperities, $\overline{\kappa}(\varepsilon)$, was made. As originally described in the paper, $\overline{\kappa}(\varepsilon)$ is a function of four variables:

$$\overline{\kappa}(\varepsilon) = \overline{\kappa}_0 \frac{\varepsilon - \varepsilon_{tr}}{\varepsilon_b - \varepsilon_{tr}} \tag{3.15}$$

where ε_b is the relative eccentricity at the border between mixed and boundary lubrication; and $\overline{\kappa}_0$ is a constant dimensionless coefficient of the asperities stiffness. While simulating, it was noted that having $\overline{\kappa}(\varepsilon)$ as a function of four variables was an

unfavorable fact. Once a value was chosen for $\overline{\kappa}_0$ and ε_b, it would be advantageous to have $\overline{\kappa}(\varepsilon)$ a function of ε (i.e. δ) only. However, as shall be seen shortly, $\varepsilon_{tr} = f(\overline{F})$, and $\overline{F} = f(C, \mu, U_{tr}, L, F)$, which leads to $\overline{\kappa}(\varepsilon) = f(\overline{\kappa}_0, \varepsilon, \varepsilon_b, C, \mu, U_{tr}, L, F)$. Evidently, equation 3.15 allows unnecessary coupling of many parameters, which made it quite difficult to find a set of parameters which fit experimental results.

It is important to note that the variables $\overline{\kappa}_0$, ε_b, and ε_{tr}, are held constant throughout the simulation. Therefore, it is beneficial to combine these variables into one, K_0, which leads to:

$$K_0 = \frac{\overline{\kappa}_0}{\varepsilon_b - \varepsilon_{tr}} \tag{3.16}$$

$$\overline{\kappa}(\varepsilon) = K_0(\varepsilon - \varepsilon_{tr}) \tag{3.17}$$

Now the new variable K_0 can be modified independently of the above mentioned parameters, while ε_b and $\overline{\kappa}_0$ are eliminated. Also, once a suitable value for K_0 has been determined, it is possible to find any combination of ε_b and $\overline{\kappa}_0$ to get the same results. Also, experimental results showed very limited boundary lubrication (*see* figure 2.2), which made a good choice for ε_b unclear anyway.

Next, a simplification to the model was made by neglecting all second derivative and crossed derivative terms in equations 3.7 and 3.8. These neglected terms correspond to the rotating mass inertia. Although this simplification was not necessary when simulating the model alone - this because Simulink, the simulation software used, allows an Adams/Gear integration technique which is stable for this fourth order model - it became necessary when the model was included in the control program described in

Chapter 6. The reason for the simplification is because the control program uses a simple but fast-calculating first-order Euler integration technique, which is unstable with the full model. The first-order Euler technique is stable for the simplified model, and it was found that the difference in predicted friction was found to be negligible between the full and the simplified model.

Neglecting all second derivative and crossed derivative terms in equations 3.7 and 3.8, and solving for $\dot{\varepsilon}$ and $\dot{\varphi}$ yields:

$$\dot{\varepsilon} = \frac{\overline{F}\cos\varphi - \overline{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{ir})\Delta + \dfrac{\overline{F}J_{12}\sin\varphi}{J_{11}}}{J_{22} - \dfrac{J_{12}^2}{J_{11}}} \tag{3.18}$$

$$\dot{\varphi} = \frac{0.5J_{11}\varepsilon|\overline{U}| - J_{12}\dot{\varepsilon} - \overline{F}\sin\varphi}{J_{11}\varepsilon} \tag{3.19}$$

where the $\dot{\varepsilon}$ term in equation 3.19 is taken from equation 3.18. For reference, the method of calculating ε and φ by first-order Euler integration is presented here:

$$\varepsilon_{i+1} = \dot{\varepsilon}_i \Delta t$$

$$\varphi_{i+1} = \dot{\varphi}_i \Delta t$$

where Δt is the sampling period, taken as 0.002 sec. in the control program of Chapter 6.

3.3 Procedure to Simulate the Model

There are several steps which must be taken, and parameters chosen in order to successfully simulate the friction model described in sections 3.1 and 3.2. All simulation was done on a PC with the software Simulink, an extension of Matlab. This section describes the procedure for simulation.

There are two types of parameters used in simulation: physically determined and experimentally determined. Table 3.1 outlines all of the parameters needed to be chosen before simulation can begin.

Table 3.1 Parameters needed for simulation

Physically Determined Parameters		Experimentally Determined Parameters	
F	Normal load	f_m	Maximum friction coefficient
m	Rotational mass	U_{tr}	Transition velocity
μ	Lubricant viscosity	K_0	Asperities stiffness constant
C	Circumferential clearance	γ	Slope factor
R	Journal radius		
L	Journal length		

After the parameters of Table 3.1 are chosen, the dimensionless constants $\overline{F}$ and $\overline{m}$ of equations 3.1 and 3.4, respectively, are calculated. Finally, the transition eccentricity, ε_{tr}, needs to be calculated.

It is important to understand the meaning of ε_{tr} fully in order to justify its method of calculation: The dynamic friction model should be able to predict friction for any velocity inputs. All simulations were done for inputs of sinusoidal velocity at various frequencies of oscillation. Therefore, if the frequency of oscillation is reduced to a very small number, the model should reduce to the Stribeck curve, and in fact, it does. At these small frequencies, the system can be described as quasi-static, and it is for this reason that the static equations 3.5 and 3.6, and not dynamic system equations are used to calculate ε_{tr}.

For the Stribeck curve, when $U = U_{tr}$, by definition, $\overline{U} = 1$ and $\varepsilon = \varepsilon_{tr}$. Then, plugging in these values into equations 3.5 and 3.6, squaring each equation, and adding the two together yields:

$$\overline{F}^2 = \frac{1}{4}\varepsilon_{tr}^2\left(J_{11tr}^2 + J_{12tr}^2\right)$$

Next, plugging in the definitions of J_{11} and J_{12} from equations 3.11 and 3.12, and using ε_{tr} instead of ε (to make J_{11tr} and J_{12tr}) yields:

$$4\overline{F}^2 = \varepsilon_{tr}^2\left[\frac{\pi^2}{4\left(1-\varepsilon_{tr}^2\right)^3} + \frac{4\varepsilon_{tr}^2}{\left(1-\varepsilon_{tr}^2\right)^4}\right] \tag{3.20}$$

in which only ε_{tr} is an unknown. Obviously, this is not a simple equation to solve for ε_{tr}. However, it can be calculated with iterative techniques made possible with the Matlab function fsolve. Consequently, a Matlab routine named jordat3.m, listed in Appendix A, was written to automate the task of loading and changing simulation parameters, calculating the dimensionless constants $\overline{F}$ and $\overline{m}$, and calculating ε_{tr}.

After running the routine jordat3.m, the simulation was run in the Simulink environment. The block diagram for Simulink is displayed in Appendix B, and all simulation was carried out with the full model of equations 3.7 and 3.8. The function journ.m referenced in the block diagram is listed in Appendix C.

CHAPTER 4

DYNAMIC FRICTION MODEL SIMULATION AND COMPARISON WITH EXPERIMENTAL RESULTS

This chapter covers two basic topics. First, experiments were performed to measure the frictional forces produced in hydrodynamic journal bearings that were under known load, and lubricated with 10W-40 oil. Results were obtained with an apparatus specifically designed to measure dynamic friction in hydrodynamic journal bearings.

Second, the dynamic friction model explained in the previous chapter is simulated by the use of the computer program Simulink, and plotted together with the corresponding experimental measurements. The purpose of simulation was to find a set of parameters for which the model most closely approximated experimental results, and to determine the overall accuracy of the model.

4.1 Experimental Apparatus

In the past, apparatuses to measure dynamic friction in journal bearings did not exist. There were, however, devices that were suitable for static friction measurements in journal bearings. One such example was an apparatus which used a pendulum attached to the bearing, where the angle the pendulum made with the vertical was indicative of the friction torque. This apparatus is not effective at measuring dynamic friction because of the inertial forces induced by the pendulum.

An experimental device has been designed (see Harnoy et al. ,1994) which overcomes the drawbacks associated with this other bearing friction measurement device.

Figure 4.1 shows a cross-sectional view of the new apparatus, while figure 4.2 is a photograph of said apparatus. This apparatus is designed to isolate and measure the dynamic frictional torque in four journal bearings which are approximately evenly loaded.

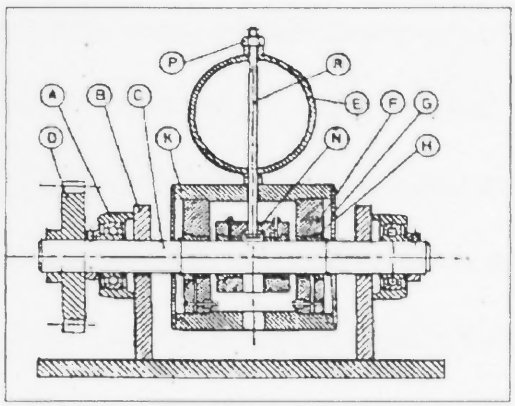


Figure 4.1 Cross-sectional view of the mechanical apparatus

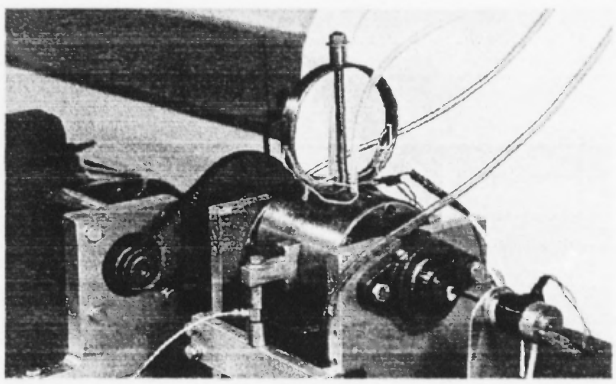


Figure 4.2 Photograph of the mechanical apparatus

The apparatus will be described with the aid of figures 4.1 and 4.2. The friction of interest is generated in four brass sleeve bearings (H). These bearings are supplied with lubricant by gravity feed through four plastic tubes (S) which lead into the bearing supply channels (I).

The normal load is transferred to all four bearings by means of a thin elastic ring (E). When the nut (P) is tightened on the elastic ring, the ring pushes the outer housing (K) down against the shaft (C) while simultaneously pulling the inner housing (N) up against the shaft with an approximately equal force. This normal load is then equally split between each of the two bearings of the inner and outer housings. The compressive force on the ring is measured by means of four strain gauges mounted on the ring, with the output of the strain gauges fed to a digital strain indicator.

The main shaft of the apparatus (C) is driven through a no-slip timing belt and pulley system (U) with a reduction ratio of 3.75:1. The driven pulley attached to the shaft is shown as component (D). Rotation is imparted by a DC servomotor (V) driven by an IBM compatible, 486DX-33 computer running a C program written by Amin (1996), with a LabWindows front end. All inputs and outputs from the computer are routed through an IBM Data Acquisition and Control Adapter (DACA) board. The control program is able to excite the shaft with varying sinusoidal velocities to a high accuracy by implementation of a Coulomb friction observer (Amin, 1996; Friedland and Park, 1992; Friedland and Mentzelopoulou, 1992; Tafazoli et al., 1995).

When the shaft is turned, approximately equal frictional torque is produced in all four loaded bearings. The friction moment from the outer two bearings is transmitted directly to the outer housing. The friction moment from the inner two bearings is transmitted indirectly to the outer housing through the compression rod (R). Therefore, the friction torque from all four bearings is in the same direction and additive.

If the outer housing were not held in place, the frictional torque generated in the bearings would make it rotate with the shaft. However, there exists an arm (T), which is

connected to the outside of the outer housing. This arm transmits the frictional torque as a force, located a known distance from the center of the shaft. The force is measured by a very stiff (20 x 10^{-8} in./lb.) piezoelectric load cell (W) which simultaneously keeps the outer housing from rotating. By recording the low-pass filtered output from the load cell, a record of dynamic friction is available.

Explanation for the calculation of frictional torque is made with the aid of Figure 4.3. The total tangential frictional force, F_t, is produced at a radius, R_S, of the shaft. However, the frictional force, F_P, is measured at a distance, R_P, the distance from the center of the shaft to the center of the piezoelectric load cell. Nonetheless, it is possible to calculate the tangential force, F_t, because the torque at the two distances is equal. Setting the two moments equal to each other, and solving for F_t yields:

$$F_t = \frac{R_P}{R_S} F_P \tag{4.1}$$

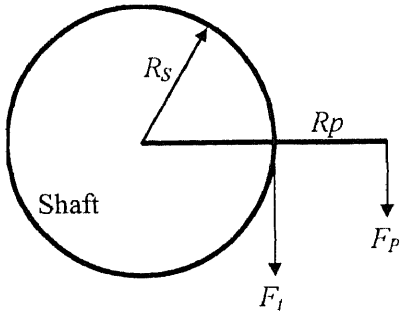


Figure 4.3 Free body diagram used to calculate frictional torque

As mentioned previously, the normal load created at the elastic ring is only approximately equally proportioned between the inner and outer housings of the

apparatus. A static load balance was performed, and the friction coefficient was calculated as:

$$f = \frac{F_t}{2F_r + W_O - W_I + W_R - W_{ROD}} \tag{4.2}$$

where:

f Dimensionless friction coefficient

F_t Total tangential friction force at the shaft, calculated from equation 4.1

F_r Compressive force on the thin ring

W_O Weight of the outer housing assembly, including the two outer bearings

W_I Weight of the inner housing assembly, including the two inner bearings

W_R Weight of the elastic ring

W_{ROD} Weight of the connecting rod and nut

Specific physical dimensions of the apparatus are given in Table 4.1.

Table 4.1 Dimensional data for test apparatus

Diameter of the bearing/shaft $(D = 2R_S)$	D= 0.0254 m
Length of each bearing	L= 0.01905 m
Weight of outer housing, with bearings	W_O= 8.92 N
Weight of inner housing, with bearings	W_I= 4.46 N
Weight of elastic ring	W_R= 0.45 N
Weight of rod and nut	W_{ROD}= 0.22 N
Radial distance from center of shaft to piezotron	R_P= 0.0915 m

4.2 Model Parameters

The model of Chapter 3 has many parameters that need to be found experimentally. A list of these parameters can be found in Table 3.1. The objective was to obtain a plot of the static Stribeck curve[1], find the model parameters which fit it, and use these same parameters for all dynamic friction experiments.

Table 4.2 is an example of the parameters which fit well for the Stribeck and bi-directional f-v curves. The values for F, m, μ, and C are physically based. That is, F was the actual load supplied to each bearing, and was also used for the model. The value used for m in the model simulation was also the actual rotating mass. The lubricant used for all experiments in this chapter was a multi-grade SAE 10W-40 engine oil. It was assumed that the viscosity change due to running of the apparatus was negligible, and so the value of μ reflects a reasonable value for viscosity of 10W oil at room temperature. Finally, the circumferential clearance, C, between shaft and sleeve was originally specified as 0.001" (2.54×10^{-5} m). However, because the apparatus had already been extensively used prior to the experimental investigations presented in this thesis, it was presumed that there had already been some wear. As a result, a value for C of 0.002" (5.08×10^{-5} m) was taken for model simulations.

---

[1] The Stribeck curve is a plot of friction vs. velocity for static conditions. That is, obtain a given velocity for a long enough period for the friction to become stable, and that becomes the single-valued friction for that velocity. However, the same results were obtained in this thesis by lowering the frequency of oscillation in sinusoidal velocity excitations to a level low enough that no hysteresis was measured. The resulting f-v curve of quasi-static conditions was then termed the Stribeck curve.

4.3 Experimental Results and Model Simulation for Stribeck and Bi-directional f-v Curves

Friction measurement for bi-directional velocities is very important for control system design. However, there is very little previous data concerning measured friction for velocities passing through zero. Most of the previous experiments, such as those performed by Hess and Soom (1990), dealt only with unidirectional velocities to avoid the complication of stiction. This section presents friction measurement for bi-directional velocities.

4.3.1 Stribeck Curve and Bi-directional Friction vs. Velocity Curves with Normal Load of 104 N

Presentation of results will begin with the Stribeck curve. As formerly mentioned, the Stribeck curve was obtained through sinusoidal velocity inputs with very small frequencies of oscillation. In this case, a frequency of 0.0055 rad/s was found sufficiently small to eliminate any signs of hysteresis.

Following the Stribeck curve, bi-directional *f-v* plots are shown which progress to ever-higher frequencies of oscillation, terminating with a maximum frequency of 2.0 rad/sec. All experimental data (dots) is shown compared to the friction model (curves) using the parameters of Table 4.2.

Table 4.2 Model parameters for dynamic bi-directional velocities and static conditions

f_m	0.26	K_0	7.5×10^5	μ	0.002 kg/(m-sec)
U_{tr}	0.06 m/s	F	104 N	C	5.08e-5 m
ε_{tr}	0.9727	m	2.27 kg	γ	33

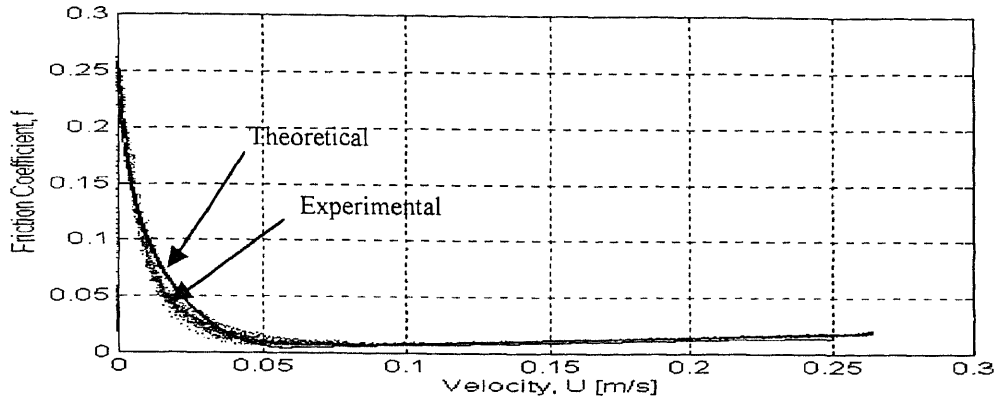


Figure 4.4 Stribeck curve: $U = 0.1323\sin(0.0055t) + 0.1323$ m/s. Simulation parameters from Table 4.2

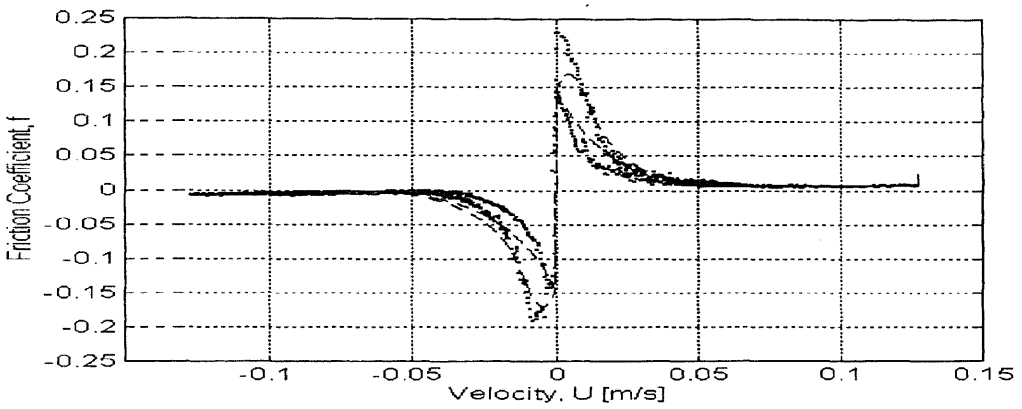


Figure 4.5 $U = 0.127\sin(0.0225t)$ m/s. Simulation parameters from Table 4.2

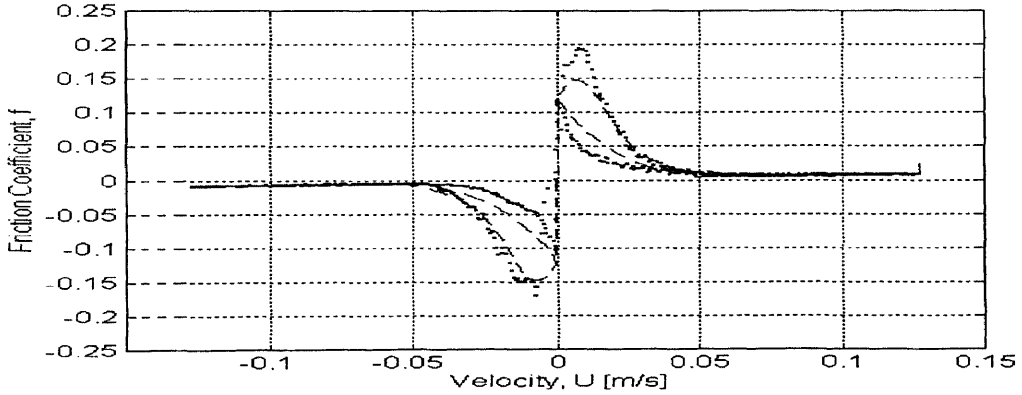


Figure 4.6 $U = 0.127\sin(0.045t)$ m/s. Simulation parameters from Table 4.2

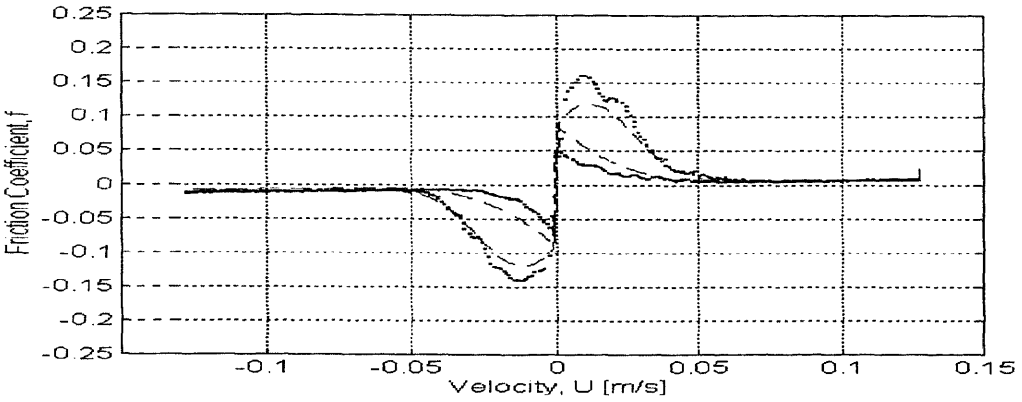


Figure 4.7 $U = 0.127\sin(0.1t)$ m/s. Simulation parameters from Table 4.2

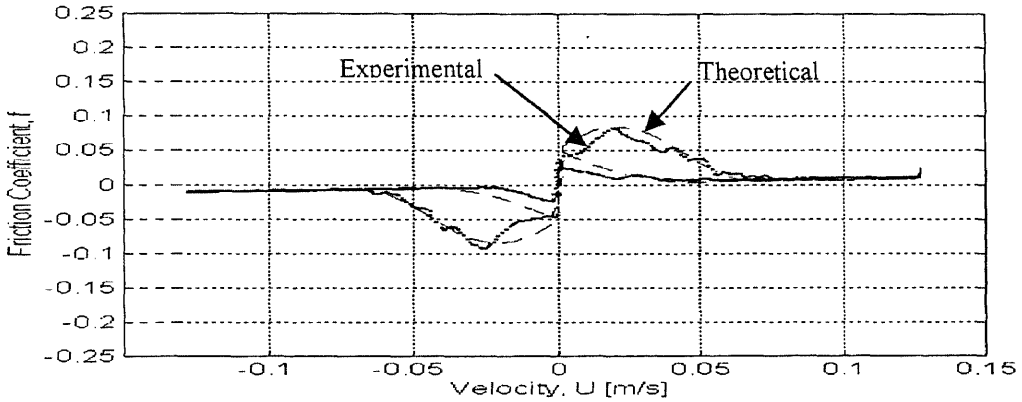


Figure 4.8 $U = 0.127\sin(0.25t)$ m/s. Simulation parameters from Table 4.2

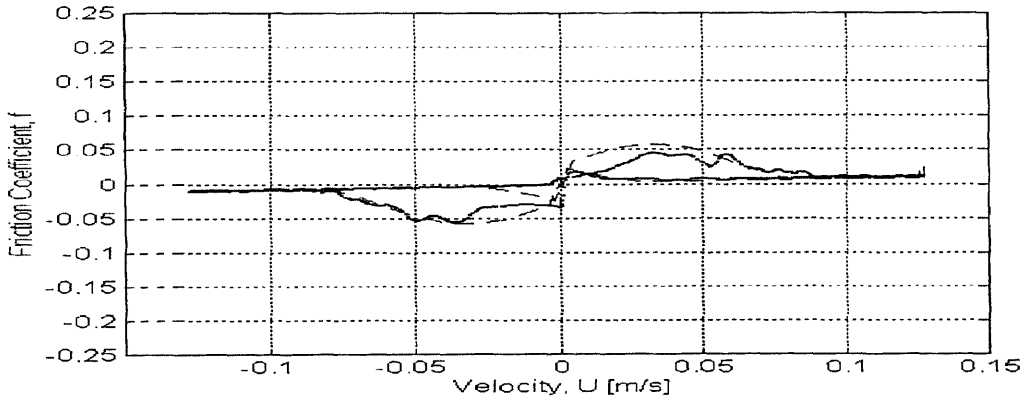


Figure 4.9 $U = 0.127\sin(0.5t)$ m/s. Simulation parameters from Table 4.2

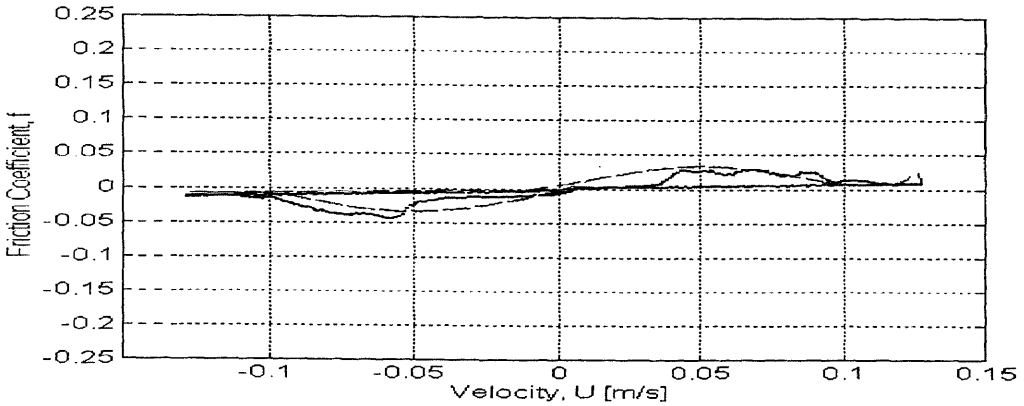


Figure 4.10 $U = 0.127\sin(t)$ m/s. Simulation parameters from Table 4.2

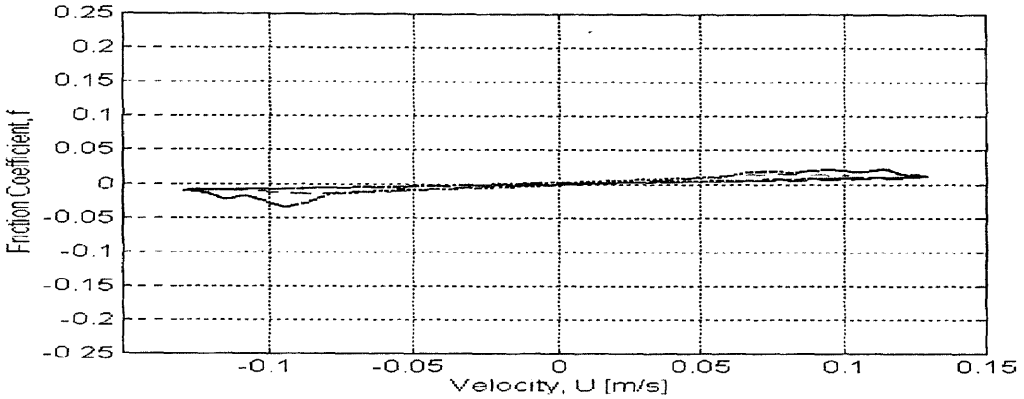


Figure 4.11 $U = 0.127\sin(2.0t)$ m/s. Simulation parameters from Table 4.2

4.3.2 Bi-directional Friction vs. Velocity Curves with Normal Load of 84 N

Static and bi-directional *f-v* curves have been presented for a journal normal load of 104N in sub-section 4.3.1. The current sub-section displays the results of *f-v* curves at a lower normal load of 83.5 N. Results show similar behavior for all frequencies of oscillation.

Presentation of results will be for dynamic bi-directional velocities only, with frequencies of oscillation ranging from 0.05 rad/s to 1.0 rad/s in Figures 4.12 through 4.16, respectively. All simulation parameters are listed in Table 4.3.

Table 4.3 Model parameters for bi-directional velocities

f_m	0.26	K_0	6.25 x 10^5	μ	0.002 kg/(m-sec)
U_{tr}	0.05 m/s	F	83.5 N	C	5.08e-5 m
ε_{tr}	0.9718	m	2.27 kg	γ	33

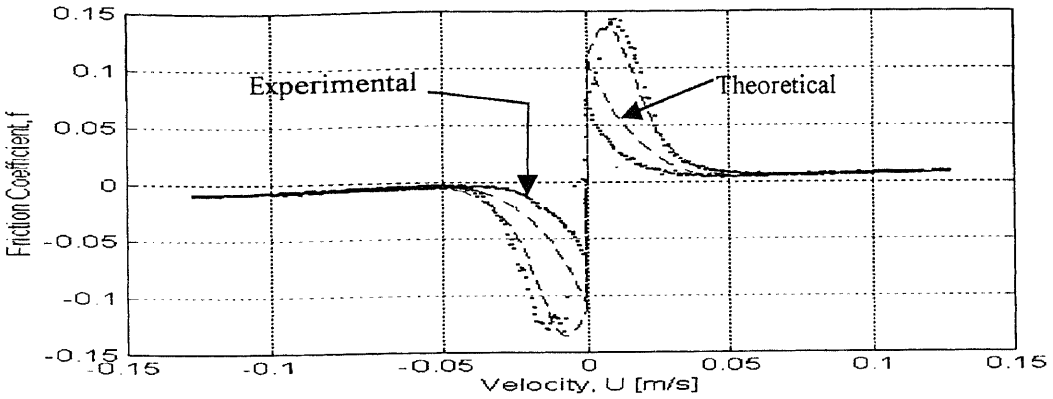


Figure 4.12 $U = 0.127\sin(0.05t)$ m/s. Simulation parameters from Table 4.3

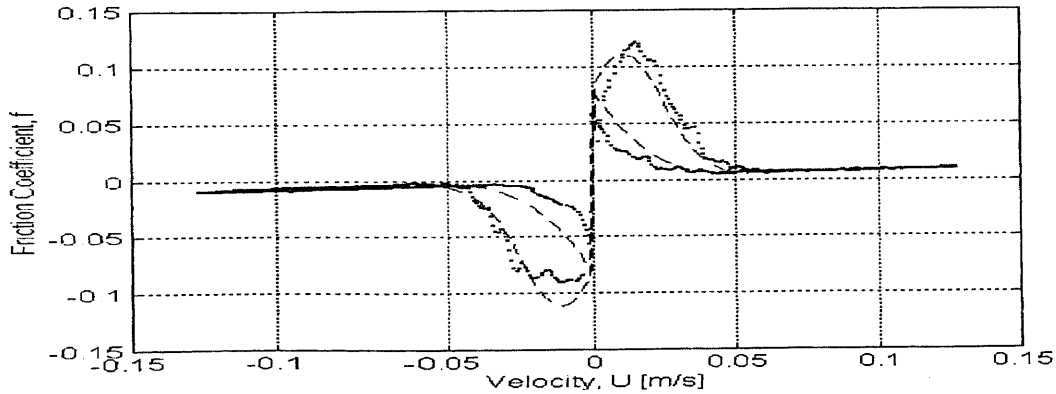


Figure 4.13 $U = 0.127\sin(0.1t)$ m/s. Simulation parameters from Table 4.3

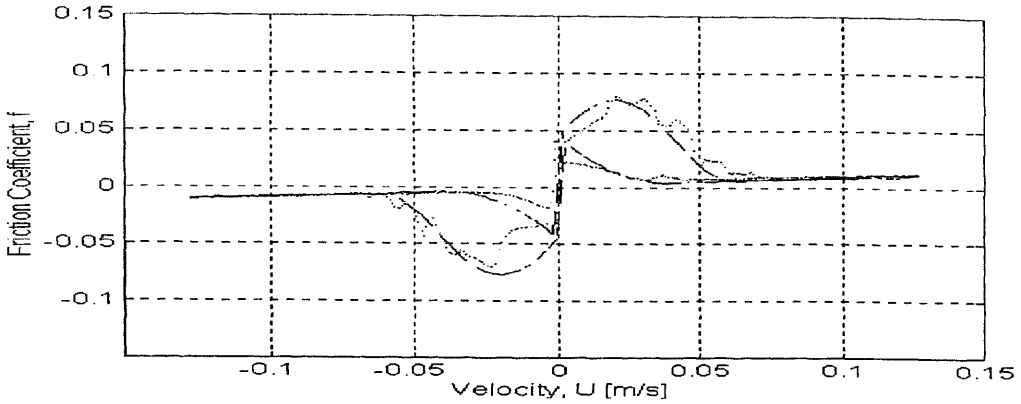


Figure 4.14 $U = 0.127\sin(0.25t)$ m/s. Simulation parameters from Table 4.3

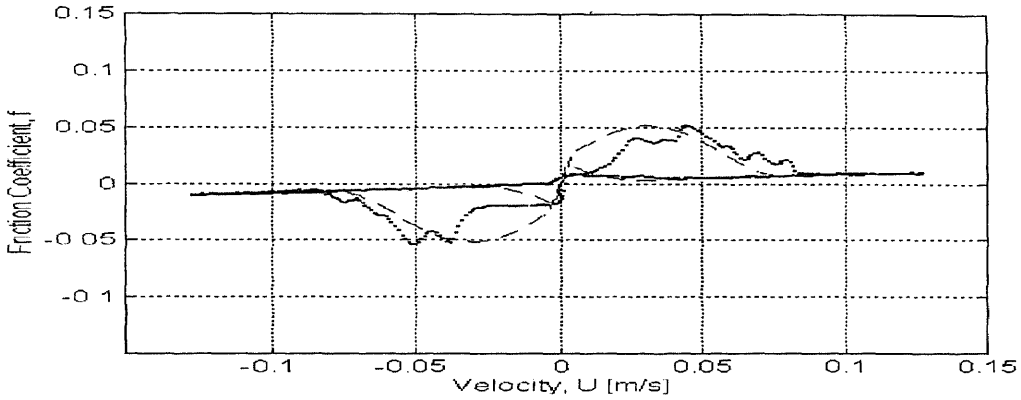


Figure 4.15 $U = 0.127\sin(0.5t)$ m/s. Simulation parameters from Table 4.3

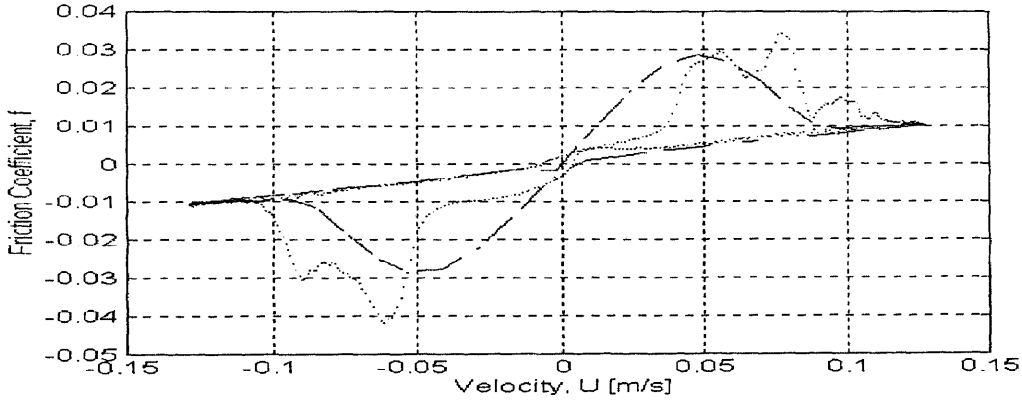


Figure 4.16 $U = 0.127\sin(t)$ m/s. Simulation parameters from Table 4.3

4.4 Experimental Results and Model Simulation for Uni-directional f-v Curves

In the previous two sections, data was presented for static and dynamic friction for bi-directional velocity excitations. This section outlines the behavior of friction during uni-directional velocity inputs. The purpose of this section is to compare these experiments with previous unidirectional experiments of line contact friction (Hess and Soom; 1990).

4.4.1 Uni-directional Friction vs. Velocity Curves with Normal Load of 104 N

This sub-section presents friction vs. velocity curves for uni-directional velocity excitations with a normal load of 104N. During comparison of model and experimental data, it became clear that the parameters of Table 4.2, which led the model to fit experimental results so well for both the Stribeck curve and dynamic bi-directional conditions, gave rise to excess predicted friction in uni-directional conditions. This suggests that the model still needs improvement in order for only one set of parameters to be requisite for both uni-directional and bi-directional conditions. Therefore, a new set of parameters was determined to best fit the model to the experimental data for uni-directional velocity inputs.

Although two sets of model parameters were used, when comparing the parameters for uni- and bi-directional *f-v* curves, it is seen that many of the values in each set are identical. The values for f_m, F, m, μ, C, and γ are the same for both sets. Only the values of U_{tr} and K_0 are different between the two sets. The value of ε_{tr} is also different, but recall that ε_{tr} is a calculated value, and it is altered with the different value of U_{tr}.

Presentation of data includes frequencies of oscillation ranging from 0.1 rad/s to 0.5 rad/s. Model parameters are listed in Table 4.4.

Table 4.4 Model parameters for dynamic uni-directional velocities

f_m	0.26	K_0	6.25 x 10^5	μ	0.002 kg/(m-sec)
U_{tr}	0.05 m/s	F	104 N	C	5.08e-5 m
ε_{tr}	0.9747	m	2.27 kg	γ	33

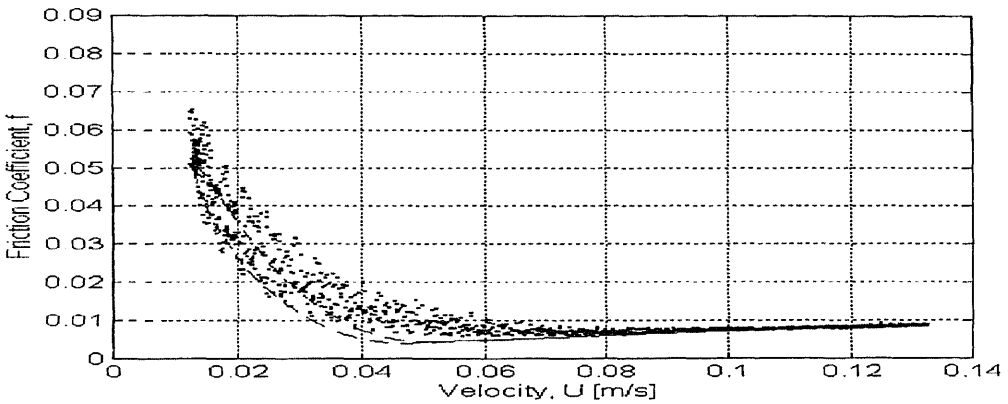


Figure 4.17 $U = 0.0595\sin(0.1t) + 0.0728$ m/s. Simulation parameters from Table 4.4

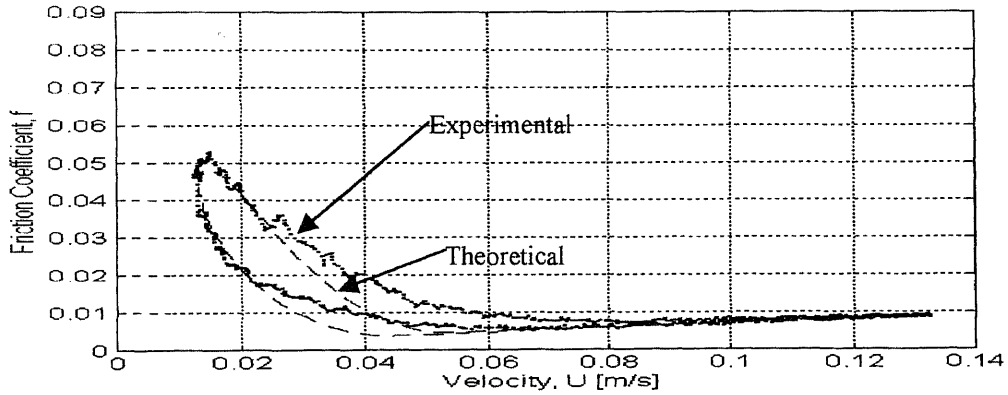


Figure 4.18 $U = 0.0595\sin(0.25t) + 0.0728$ m/s. Simulation parameters from Table 4.4

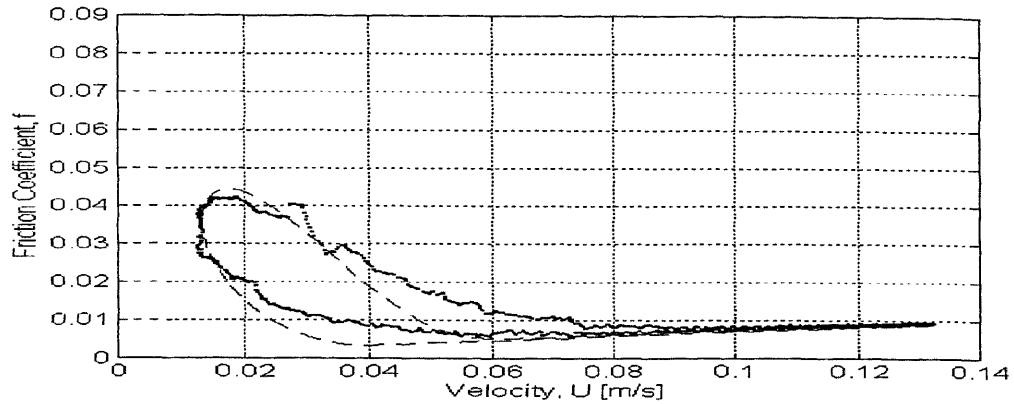


Figure 4.19 $U = 0.0595\sin(0.5t) + 0.0728$ m/s. Simulation parameters from Table 4.4

4.4.2 Uni-directional Friction vs. Velocity Curves with Normal Load of 84 N

In this sub-section, uni-directional friction vs. velocity curves are presented for a journal load of 84 N. Similar to the uni-directional results obtained for the normal load of 104N, K_0 needed to be reduced in comparison to the value used for bi-directional motion in order to make the model fit experimental results in the mixed lubrication region well. Presentation of data includes frequencies of oscillation ranging from 0.1 rad/s to 0.5 rad/s. The parameters used for this section are listed in Table 4.5.

Table 4.5 Model parameters for uni-directional velocities

f_m	0.26	K_0	4.5×10^5	μ	0.002 kg/(m-sec)
U_{tr}	0.05 m/s	F	83.5 N	C	5.08e-5 m
ε_{tr}	0.9718	m	2.27 kg	γ	33

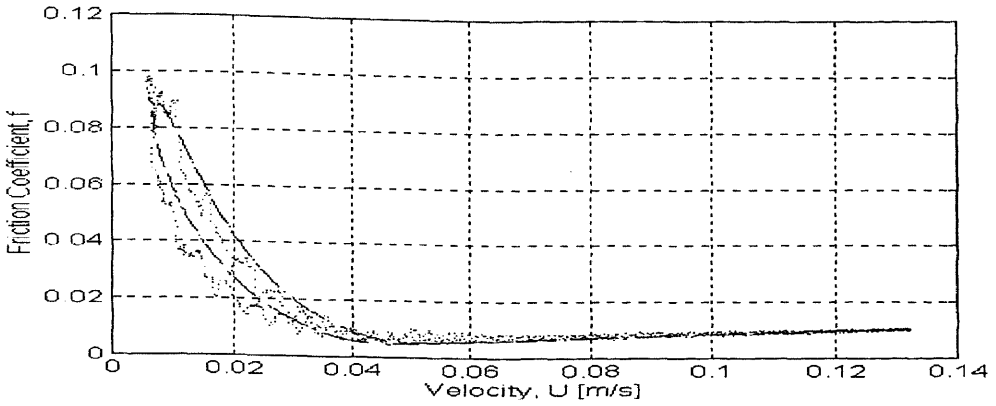


Figure 4.20 $U = 0.0628\sin(0.1t) + 0.0695$ m/s Simulation parameters from table 4.5

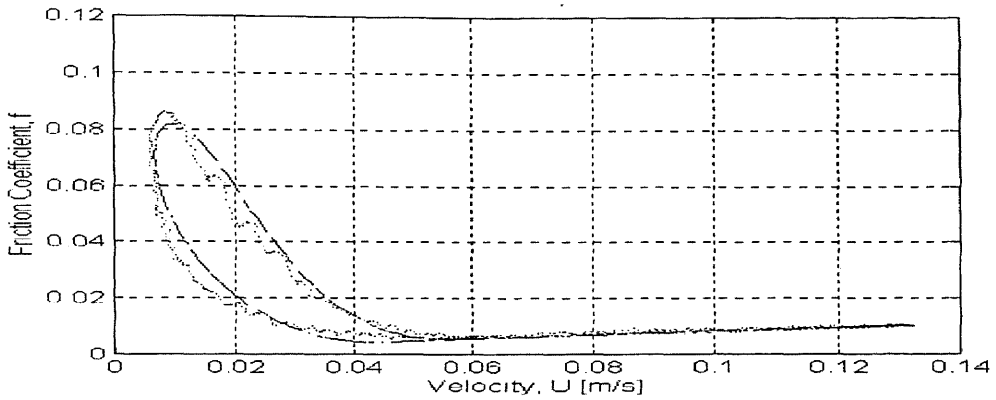


Figure 4.21 $U = 0.0628\sin(0.25t) + 0.0695$ m/s Simulation parameters from table 4.5

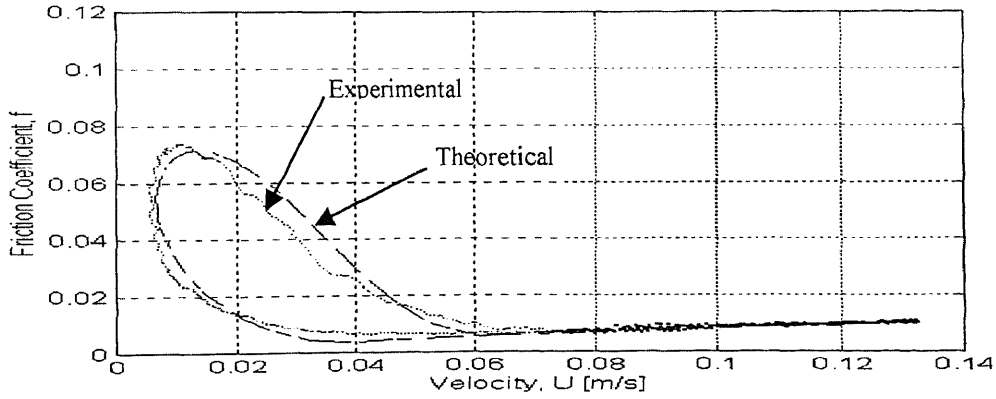


Figure 4.22 $U = 0.0628\sin(0.5t) + 0.0695$ m/s Simulation parameters from table 4.5

CHAPTER 5

EXPERIMENTAL RESULTS OF DYNAMIC FRICTION IN DRY CONDITIONS AND SUNOCO 104 OIL

This chapter presents experimental results of dynamic friction in journal bearings lubricated with a very low viscosity lubricant (Sunoco 104, viscosity 0.0005 kg/(m-sec)) and no lubricant. Measurements were obtained using the apparatus described in Chapter 4. Results suggest there is a large reduction of hysteresis in measured dynamic friction compared to measurements made using SAE 10W-40 oil.

5.1 Dynamic Friction Measurement with Sunoco 104 Oil and Load of 104N

Figures 5.2 through 5.4 present dynamic friction measurements for a journal bearing under a load of 104N lubricated with 104 oil. The three figures are arranged with increasing frequencies of oscillation. The results show that hysteresis is greatly diminished at this load compared with results obtained in Chapter 4 for measurements made with 10W-40 motor oil. This is consistent with theory, which states that hysteresis is more prominent when higher viscosity lubricants are used.

Figure 5.4 apparently shows that at a frequency of three rad/s, the friction is opposite to what is expected. In other words, near the velocity reversals, when the rotation of the shaft passes through zero velocity, the friction actually accelerates the shaft. This is obviously an impossible physical phenomenon. This paradox can be explained by elasticity and inertial effects in the drive system of the measuring apparatus which make themselves apparent during the high accelerations encountered in high frequencies of oscillation. This phenomenon was studied theoretically in Harnoy et al.

(1994). A theoretical graph is shown from the paper of Harnoy et al. (1994) in Figure 5.1. It bears a remarkable similarity to the phenomenon experienced in Figures 5.4, 5.7, and 5.10.

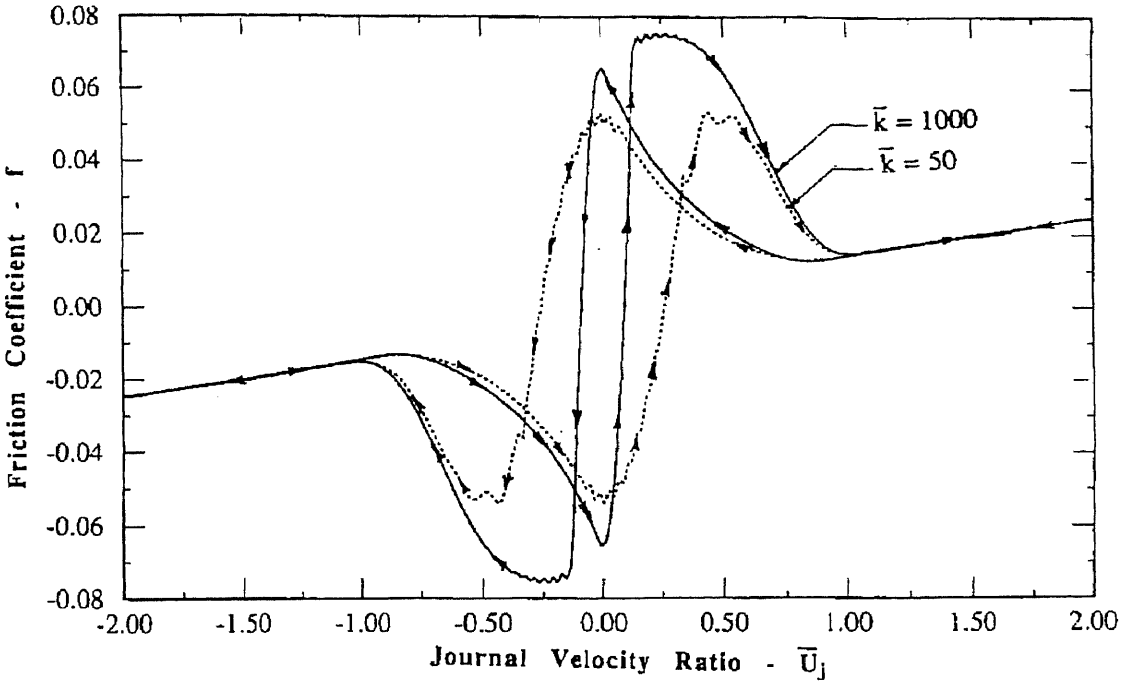


Figure 5.1 Theoretical graph of friction vs. dimensionless velocity for a system with elasticity. From Harnoy et al. (1994).

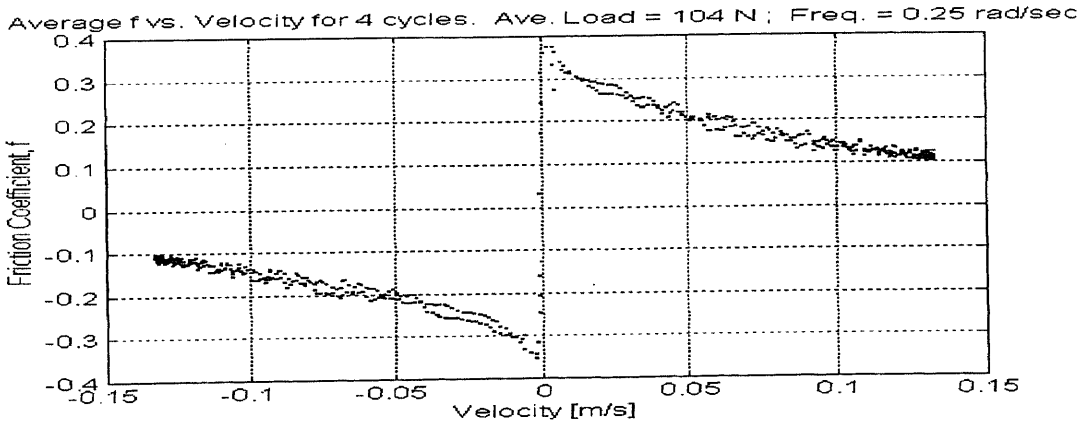


Figure 5.2 Experimental results with Sunoco 104 oil and bearing load of 104N. $U=$ $0.127\sin(0.25t)$ m/s.

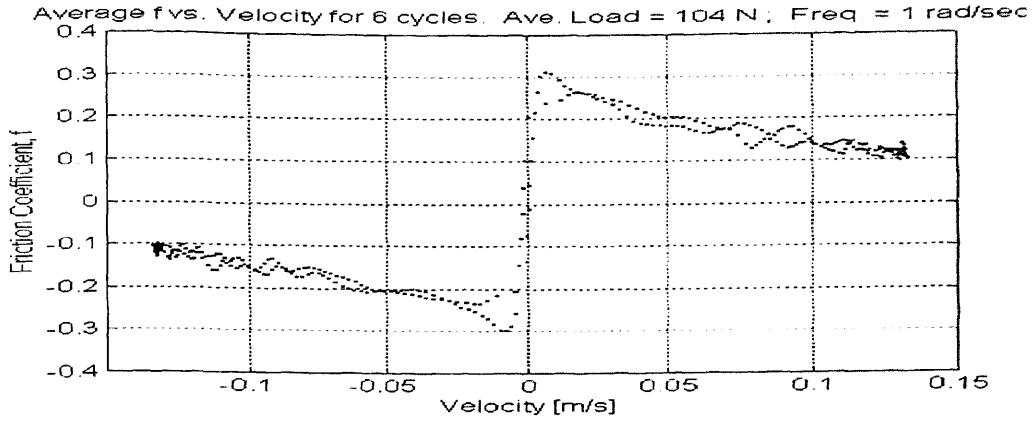


Figure 5.3 Experimental results with Sunoco 104 oil and bearing load of 104N. $U = 0.127\sin(t)$ m/s.

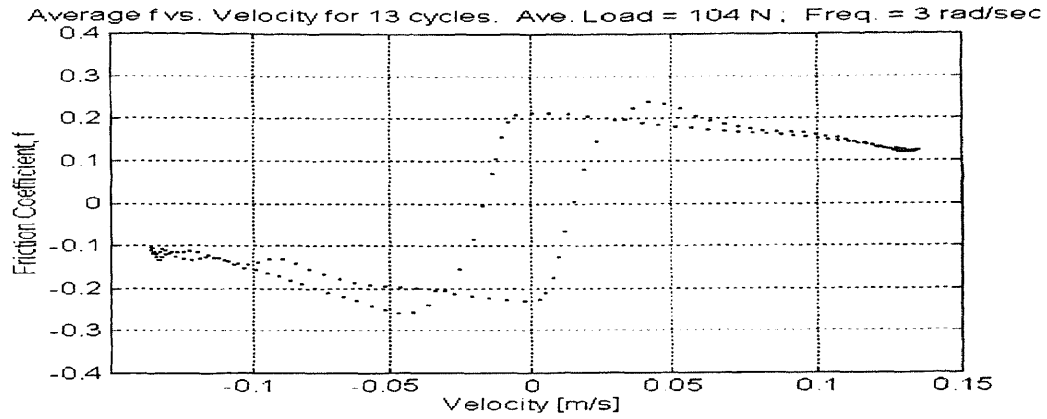


Figure 5.4 Experimental results with Sunoco 104 oil and bearing load of 104N at high frequency of oscillation. Shows gap near zero velocity due to elasticity in the system. $U = 0.127\sin(3t)$ m/s.

5.2 Dynamic Friction Measurement with Sunoco 104 Oil and Load of 37N

Figures 5.5 through 5.7 present dynamic friction measurements in a journal bearing under a load of 37N lubricated with Sunoco 104 oil. At this lower load, a fair amount of hysteresis is revealed. This is a reasonable result. Although the lubricant is very thin, a hydrodynamic film is able to form because of the low load. Figure 5.7 again exhibits signs of elasticity in the drive system of the apparatus.

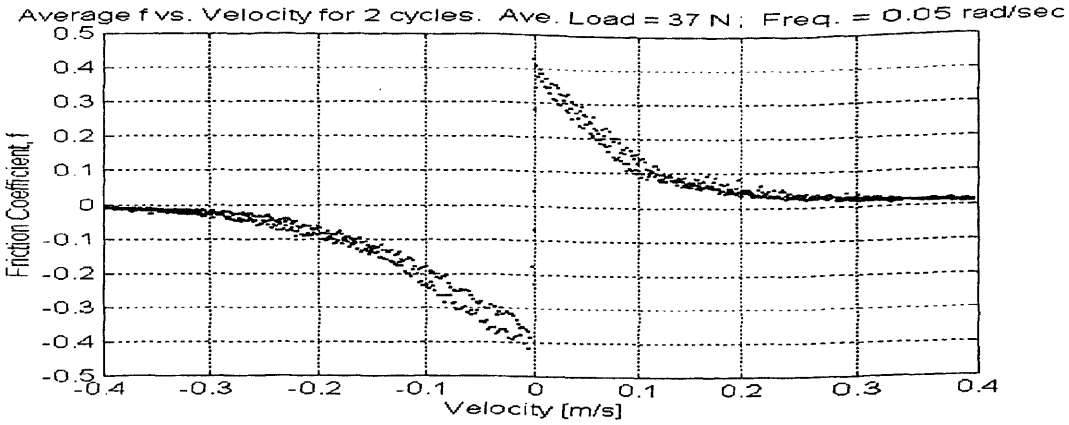


Figure 5.5 Experimental results with Sunoco 104 oil and bearing load of 37N. $U = 0.381\sin(0.05t)$ m/s.

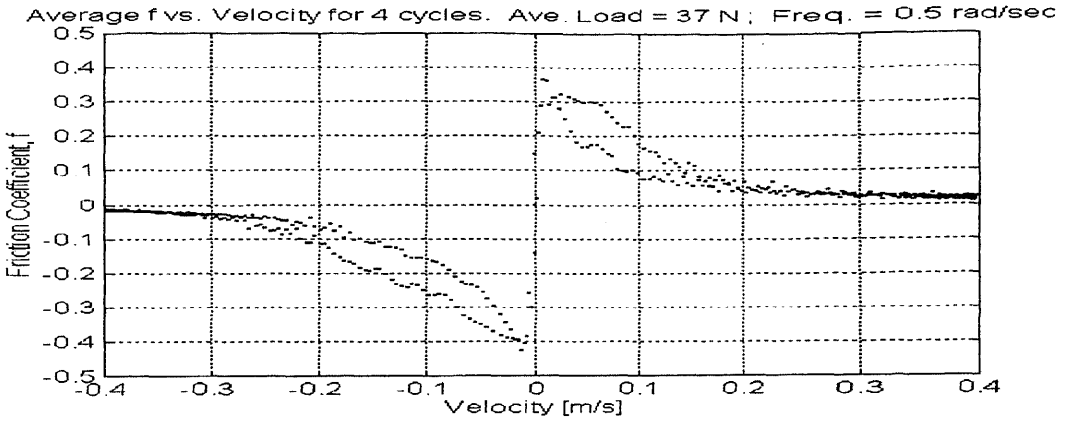


Figure 5.6 Experimental results with Sunoco 104 oil and bearing load of 37N. $U = 0.381\sin(0.5t)$ m/s.

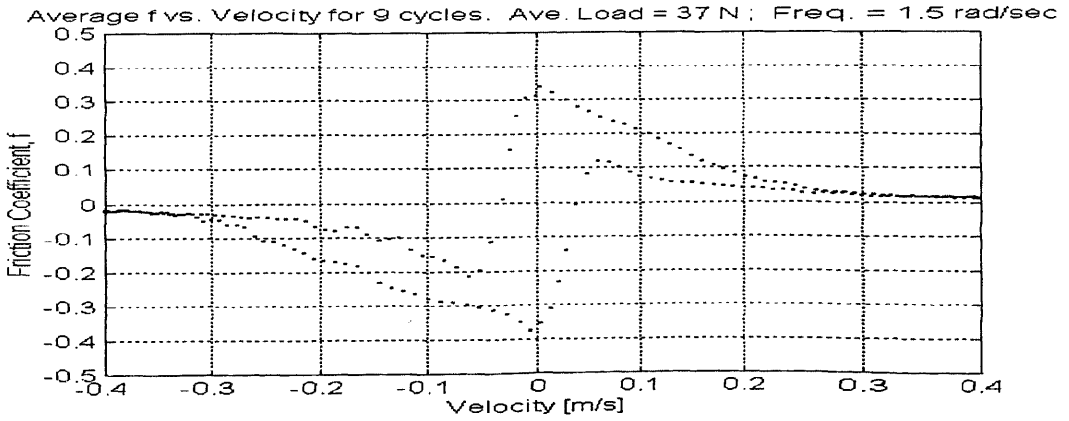


Figure 5.7 Experimental results with Sunoco 104 oil and bearing load of 37N at high frequency of oscillation. Shows gap near zero velocity due to elasticity in the system. $U = 0.381\sin(1.5t)$ m/s.

5.3 Dynamic Friction with No Lubricant and Load of 53N

Figures 5.8 through 5.10 present dynamic friction measurements in a journal bearing running with no lubrication and a normal load of 53N. It is expected that no hysteresis will be seen. The theory of Chapter 3 states that hysteresis in dynamic friction is attributable to the time delay associated with the changing thickness of a hydrodynamic fluid film. Therefore, it stands to reason that if there were no fluid, there would be no hysteresis. Nonetheless, Figure 5.8 clearly shows a small amount of hysteresis. The cause is unknown. Figure 5.10 again exhibits signs of elasticity in the drive system of the apparatus.

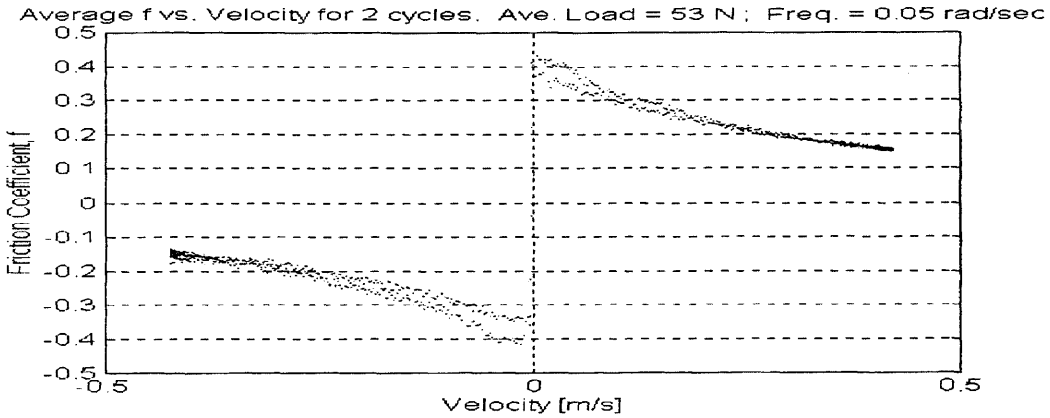


Figure 5.8 Experimental results with no lubricant and bearing load of 53N. $U = 0.381\sin(0.05t)$ m/s

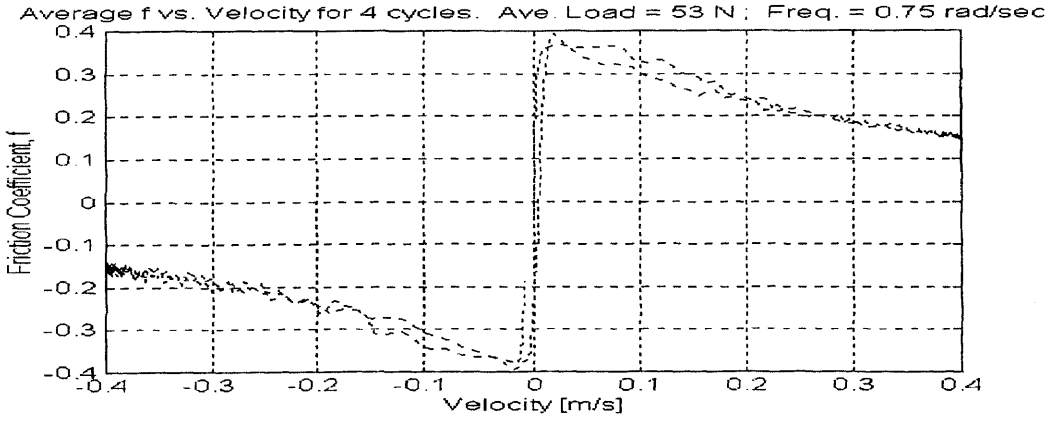


Figure 5.9 Experimental results with no lubricant and bearing load of 53N. $U = 0.381\sin(0.75t)$ m/s

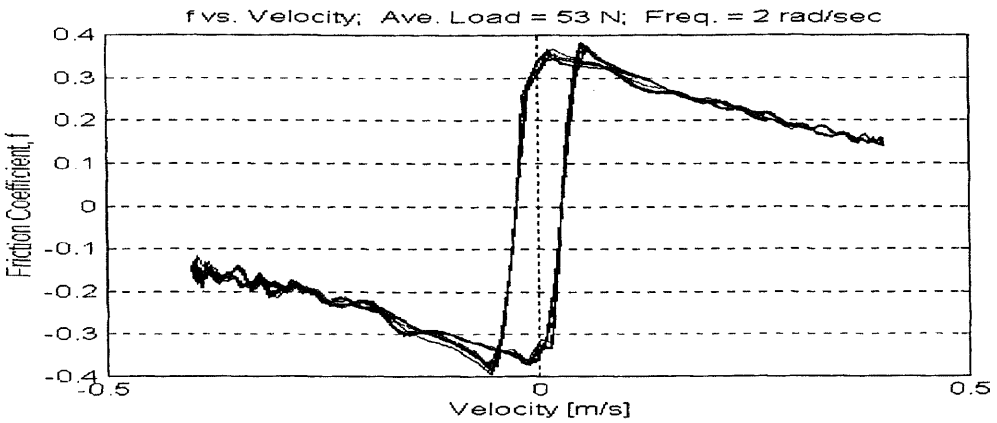


Figure 5.10 Experimental results with no lubricant and bearing load of 53N at high frequency of oscillation. Shows gap near zero velocity due to elasticity in the system. $U = 0.381\sin(2t)$ m/s

CHAPTER 6

MODEL BASED FRICTION COMPENSATION

In many dynamic systems where friction exists, a regular control scheme, such as proportional-integral-derivative (PIR), is implemented for precise motion control. A more advanced control technique to counteract friction is one in which a model of the friction in the system is incorporated, such that the control system can anticipate the friction, and account for it by adding an extra input to counteract it. Such a technique was incorporated in Amin (1996). In that study, Coulomb friction and velocity observers developed by Friedland and Park (1992) and later extended by Tafazoli (1995) were implemented. In this chapter, the more accurate friction model described in Chapter 3 (Harnoy and Friedland, 1993) – henceforth, termed the dynamic friction model - is used in conjunction with a the same velocity observer while estimating two of the model parameters online.

6.1 The Concept of Model Based Friction Compensation

A single degree of freedom mechanical system with friction, acted upon by an external input, can be expressed by:

$$m\dot{v} = u - F + D$$

$$\dot{x} = v$$

where m is the mass of the system, x is the position, v is the velocity, u is the control input, F is friction, and D is any other disturbances or losses in the system. If the system considered did not have any friction, and very minimal disturbances, then the motion

produced would be entirely driven by the control input, u, and the movement of the system could be easily controlled.

However, in many systems, friction is present. Let us consider a system in which the disturbances are negligible, but in which friction is a relatively large force. The force of friction always opposes the direction of motion, so a system with large friction will have a tendency to lag behind the desired output. Model based friction compensation is a scheme whereby the force of friction is estimated by some means in real time. As such, if this estimated friction, $\hat{F}$, is a very close approximation to the actual friction in the system, and if it is added to the original control input, u, then the system can be made to behave like an ideal, frictionless system. This concept can be written as:

$$m\dot{v} = u + \hat{F} - F \approx u$$

6.2 Coulomb Friction Observer Based Compensation

A recent example of using a friction model to counteract friction is shown in Amin (1996). In that study, a Coulomb friction observer was implemented. The Coulomb friction model assumes a value of friction which equals a constant times the sign of velocity. However, in that study, the value of the constant, a, was constantly updated by an observer. The original Coulomb friction observer was devised by Friedland and Park (1992), and then extended by Friedland and Mentzelopoulou (1992). Later, Tafazoli et al. (1995) modified the velocity observer portion to give a better estimate of velocity, especially at low speeds. The best results were obtained by Amin (1996) and Tafazoli et al. (1995) using the extended Coulomb friction observer coupled with the modified velocity observer. The two observers are given as:

- Extended Coulomb Friction Observer:

$$\begin{aligned} \hat{F} &= \hat{a}\,\mathrm{sgn}(v) \\ \hat{a} &= z_F - k_F|\hat{v}|^{\mu} \quad where, \\ \dot{z}_F &= k_F \mu |v|^{\mu-1}(u - \hat{F})\,\mathrm{sgn}(\hat{v}) \end{aligned} \tag{6.1}$$

- Modified Velocity Observer

$$\begin{aligned} \hat{v} &= z_v + k_v x \\ \dot{z}_v &= -k_v \hat{v} \end{aligned} \tag{6.2}$$

The Coulomb friction observer was designed to compensate only for a simple, single-valued friction value, and its effectiveness in that role was verified by simulation in Friedland and Park (1992) and Mentzelopoulou (1994). However, further investigation by Friedland and Park (1992), Friedland and Mentzelopoulou (1992), Tafazoli et al. (1995), and Amin (1996) showed that the observer demonstrated the ability to follow friction even if it did not conform to the classical Coulomb friction. In fact, Tafazoli et al. (1995) and Amin (1996) showed that the Coulomb friction observer was able to improve the accuracy of physical systems with large amounts of varying friction to a great degree.

6.3 System Description

The experimental apparatus used to test the dynamic friction model compensation technique was the same as that described in Chapter 4, and the same as used by Amin (1996). The device is controlled by a computer which can also record dynamic friction in the journal bearings at the same time. Because the control technique estimates the friction in real time, a comparison of estimated and actual friction is possible.

In order to gauge the effectiveness of the new compensation technique, a baseline control law had to be established. That is, whether discussing position or velocity control, a simple control law was instituted, and its ability to trace a desired output was measured.

The baseline control laws instituted were identical to those designed and implemented by Amin (1996). For position control, the control law was designed as:

$$u = -k_1(x - x_0) - k_2 v$$

Here, x is the actual measured angular position, x_0 is the desired position, v is the rotational speed, and k_1 and k_2 are gains which were chosen using pole placement method to obtain a desired damping and natural frequency. The gains were chosen to be k_1 = 0.43764 and k_2 = -0.25164. Basically, this baseline control law is a proportional control with an added term which is proportional to the measured velocity of the system.

For velocity control, the baseline control law was chosen as:

$$u = -g_1(v - v_0) + Cv_0$$

This control law is also a proportional design, with an added feedforward term for the reference velocity, v_0, as also used by Carli et al. (1994). The value of the gains were chosen by Amin (1996) to be g_1 = 1.0 and C = 0.295, where C was calculated from the system dynamics. Incidentally, the calculation of C is made for the system with no load, and hence very little friction, in the journal bearings. The gain, C, absorbs all other losses in the system, including the driveline losses, friction in the support bearings of the apparatus, and the friction inherent in the motor. Consequently, when a friction compensation scheme is added to the baseline control law, it only has to account for the friction in the journal bearings. Hence, a direct comparison between estimated and

measured friction is possible because the measured friction originates solely from the journal bearings.

6.4 Dynamic Friction Model Based Friction Estimation

In Chapter 4, it was seen that one set of parameters could be chosen for the dynamic friction model to fit most experimental friction measurements. However, some adjustment to the parameters was necessary when switching between bi-directional and uni-directional velocity excitations. This fact, plus the author's desire that any friction compensation technique be as flexible as possible, dictated that to use the dynamic friction model to estimate friction on-line, some of the parameters of the model needed to be updated automatically.

6.4.1 Algorithm Development for Friction Estimation

For reference, the modified, second-order dynamic friction model is presented here:

$$\hat{\bar{F}}_f = K_0 f_m (\varepsilon - \varepsilon_{tr})^2 \operatorname{sgn}(\bar{U})\Delta + \gamma \frac{CR}{L^2} \frac{2\pi}{(1-\varepsilon^2)^{0.5}} \bar{U} \tag{6.3}$$

$$\dot{\varepsilon} = \frac{\bar{F}\cos\varphi - \bar{\kappa}(\varepsilon)(\varepsilon - \varepsilon_{tr})\Delta + \dfrac{\bar{F} J_{12} \sin\varphi}{J_{11}}}{J_{22} - \dfrac{J_{12}^2}{J_{11}}} \tag{3.18}$$

$$\dot{\varphi} = \frac{0.5 J_{11} \varepsilon |\bar{U}| - J_{12}\dot{\varepsilon} - \bar{F}\sin\varphi}{J_{11}\varepsilon} \tag{3.19}$$

where $\hat{\bar{F}}_f$ is the estimated friction, and J_{11}, J_{12}, and J_{22} are taken from equations 3.11 through 3.13. Equation 6.3 is derived by combining equations 3.14 and 3.17. In the mixed region of hydrodynamic lubrication, the first term in equation 6.3 prevails, where

as only the second term is active during the fully hydrodynamic region. Both K_0 and γ act as linear scaling functions, and therefore have a predictable influence on the overall estimated friction, $\hat{\bar{F}}_f$. Therefore, it was a natural choice to pick these two parameters to adjust during friction compensation.

The first step was to choose all of the constant parameters shown in Table 3.1. The physically based parameters on the left side of Table 3.1 were kept the same as in the simulations of Chapters 4 and 5. The experimental constants were chosen as $f_m = 0.26$ and $U_{tr} = 0.05$, which led to a calculated value of $\varepsilon_{tr} = 0.9747$. Also, initial values of K_0 = 650,000 and γ = 33 were chosen.

The dynamic friction model requires an accurate measure of velocity in order to calculate expected friction values. A velocity observer was designed by Tafazoli et al. (1995) and was implemented by Amin (1996) for the apparatus to produce accurate estimates of velocity. The same velocity observer was used in this study, and the gains for the velocity observer were left as chosen by Amin (1996).

6.4.2 Velocity Control Algorithm Development

The method chosen to adapt K_0 and γ during dynamic friction estimation is one which is normally used in learning control theory. The idea is to use the current error in the system to ratchet the value of a parameter, based on its previous value. First, let us consider the learning of K_0.

At each step of the experiment, the algorithm checks to see if the dynamic friction model predicts friction to be within the mixed lubrication regime by comparing the calculated ε to ε_{tr}. If $\varepsilon > \varepsilon_{tr}$, then the friction is within the mixed region, and changing K_0

will have an effect. If $\varepsilon < \varepsilon_{tr}$, then the friction is within the fully hydrodynamic region, and any change in K_0 would have no effect. Therefore, learning of K_0 is limited to only the mixed region. Also, no learning takes place when the velocity is exactly zero because the algorithm is not set up to predict which direction the velocity will go next.

The learning of K_0 is made at each time step (0.002s) by the following routine:

$$\begin{aligned} K_{0,t+1} &= K_{0,t} + C_{K_0} e_t \qquad v < 0 \\ K_{0,t+1} &= K_{0,t} - C_{K_0} e_t \qquad v > 0 \end{aligned} \tag{6.4}$$

where $e_t = v_t - v_{t,ref}$ = velocity error at the current time step. The constant C_{K_0} is the learning gain for K_0, and was chosen by trial and error as 15,000 for velocity control.

If when checking ε, the program determined that $\varepsilon < \varepsilon_{tr}$, then the model predicts that friction is within the fully hydrodynamic regime. If this is the case, only the value of γ is adjusted at each time step. The routine for learning γ at each time step is:

$$\begin{aligned} \gamma_{t+1} &= \gamma_t + C_\gamma e_t \qquad v < 0 \\ \gamma_{t+1} &= \gamma_t - C_\gamma e_t \qquad v > 0 \end{aligned} \tag{6.5}$$

The constant C_γ is the learning gain for γ, and was chosen by trial and error as 2.0 for velocity control.

6.4.3 Position Control Algorithm Extension

Trial runs of the position control program showed a limitation of the above learning algorithms. The control program did a good job of compensating friction for sinusoidal and triangular reference inputs. However, square reference signals caused unacceptable levels of overshoot. The cause of the overshoot stemmed from the value of the error at the step. Even if the error was small immediately before the step (good tracking), the

error suddenly became large right after the step. This fooled the algorithm into thinking that the value of K_0 or γ was far from the correct value.

For example, if the friction was in the mixed region, and the step in the position reference signal was positive, then immediately after the step, the second term in equation 6.4 would be very large. This would cause K_0 to grow to an unrealistically high value. Then, once the system reached the correct position, the estimated friction would be too high. The system would overshoot, followed by several oscillations until it stabilized.

To combat this, a line was added to the program at the beginning of the learning section which checked the value of the error. If $|e| > 0.8$ rad, a value which was reached by trial and error, then no learning would take place. At the step in the reference position, the control would still send the system towards the correct position, but without changing K_0 (or γ) until it came to within 0.8 rad.

Next, the learning algorithm checked to see if $|e| > 0.3$ rad (also a value reached by trial and error). If it was, then learning would take place, but the actual error was replaced by 0.3 rad if the error was positive and –0.3 rad if the error was negative. This allows the learning to be two-tiered, and also helps to suppress overshoot as the actual position approaches the reference position.

As with velocity control, the learning of K_0 and γ for position control were identical, and the modifications to control overshoot were applied to the learning of both K_0 and γ. For position control, the learning gain values were chosen as C_{K_0} =12,500 and $C_\gamma = 0.5$. For clarification purposes, a flow chart of the position control algorithm is presented in Figure 6.1.

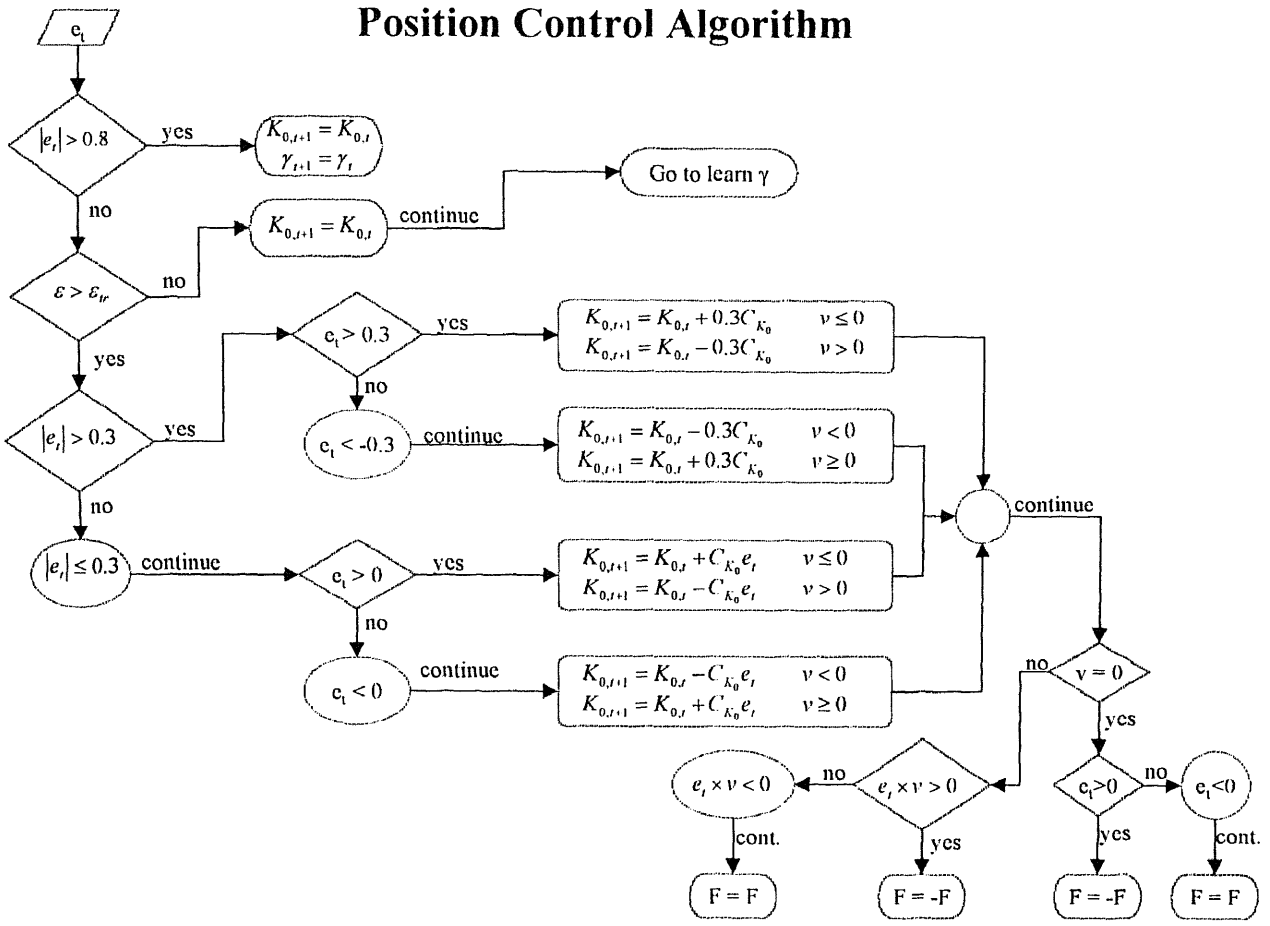


Figure 6.1 Flow chart for the learning of K_0 and γ for position control

Table 6.1 Position Control Experiments Summary

Figure Numbers	Range (rad/s) Low	Range (rad/s) High	Input Wave	Compensation Type	Frequency (rad/s)	Peak Error (rad/s)	% Peak Error of No Comp.	RMS Error (rad/s)	% RMS Error of No Comp.
6.14, 6.17,	0	10.4	Triangle	None	0.628	0.474	100.0 %	0.398	100.0 %
6.15, 6.18	0	10.4	Triangle	Dynamic Fric. Model	0.628	0.454	95.8 %	0.0894	22.5 %
6.16, 6.19	0	10.4	Triangle	Coulomb Observer	0.628	0.296	62.5 %	0.212	53.3 %
6.2, 6.5	-1.04	1.04	Square	None	0.250	n/a	n/a	0.232	100.0 %
6.3, 6.6	-1.04	1.04	Square	Dynamic Fric. Model	0.250	n/a	n/a	0.165	71.1 %
6.4, 6.7	-1.04	1.04	Square	Coulomb Observer	0.250	n/a	n/a	0.162	69.8 %
6.8, 6.11	-1.04	1.04	Sine	None	0.250	0.536	100.0 %	0.384	100.0 %
6.9, 6.12	-1.04	1.04	Sine	Dynamic Fric. Model	0.250	0.350	65.3 %	0.0595	15.5 %
6.10, 6.13	-1.04	1.04	Sine	Coulomb Observer	0.250	0.356	66.4 %	0.0615	16.0 %

Table 6.2 Velocity Control Experiments Summary

Figure Numbers	Range (rad/s) Low	Range (rad/s) High	Input Wave	Compensation Type	Frequency (rad/s)	Peak Error (rad/s)	% Peak Error of No Comp.	RMS Error (rad/s)	% RMS Error of No Comp.
6.20, 6.23	-1.04	1.04	Sine	None	0.250	0.703	100.0 %	0.468	100.0 %
6.21, 6.24, 6.26	-1.04	1.04	Sine	Dynamic Fric. Model	0.250	0.127	18.1 %	0.0304	6.5 %
6.22, 6.25, 6.27	-1.04	1.04	Sine	Coulomb Observer	0.250	0.223	31.7 %	0.0386	8.3 %
6.28, 6.31	-2.08	2.08	Triangle	None	0.500	0.563	100.0 %	0.359	100.0 %
6.29, 6.32, 6.34	-2.08	2.08	Triangle	Dynamic Fric. Model	0.500	0.205	36.4 %	0.0337	9.4 %
6.30, 6.33, 6.35	-2.08	2.08	Triangle	Coulomb Observer	0.500	0.296	52.6 %	0.0482	13.4 %
6.36, 6.38	-1.04	1.04	Square	None	0.250	n/a	n/a	0.451	100.0 %
6.37, 6.39	-1.04	1.04	Square	Dynamic Fric. Model	0.250	n/a	n/a	0.0733	16.3 %
	-1.04	1.04	Square	Coulomb Observer	0.250	n/a	n/a	Not Avail.	Not Avail.
6.40, 6.41	0.52	20.8	Sine	None	0.250	0.519	100.0 %	0.210	100.0 %
6.42, 6.44	0.52	20.8	Sine	Dynamic Fric. Model	0.250	0.168	32.4 %	0.025	11.9 %
6.43, 6.45	0.52	20.8	Sine	Coulomb Observer	0.250	0.0895	17.2 %	0.0162	7.7 %

6.5 Experimental Results – Position Control Using Dynamic Friction Model

In this section, experimental results are presented for position control using the dynamic friction model to estimate and cancel out friction. Results are included for sine, triangular, and square reference signals. The data shows that there is a remarkable improvement in position accuracy for all reference inputs with the dynamic friction model compensation when compared to the baseline control law with no friction compensation.

Data is also shown for position control using the Coulomb friction estimation technique used by Amin (1996). Comparison shows that the two friction estimation techniques give comparable results for all of the reference waveforms. Table 6.1 summarizes the data presented in the figures below. Shaded cells outline which control scheme gave the best results. In the column titled "Compensation Type", "Dynamic Fric. Model" refers to friction compensation using the adaptable dynamic friction model, and "Coulomb Observer" refers to the Coulomb friction observer as designed by Amin (1996). For all of the experiments, a load of 104 N was applied to the journal bearings of the apparatus.

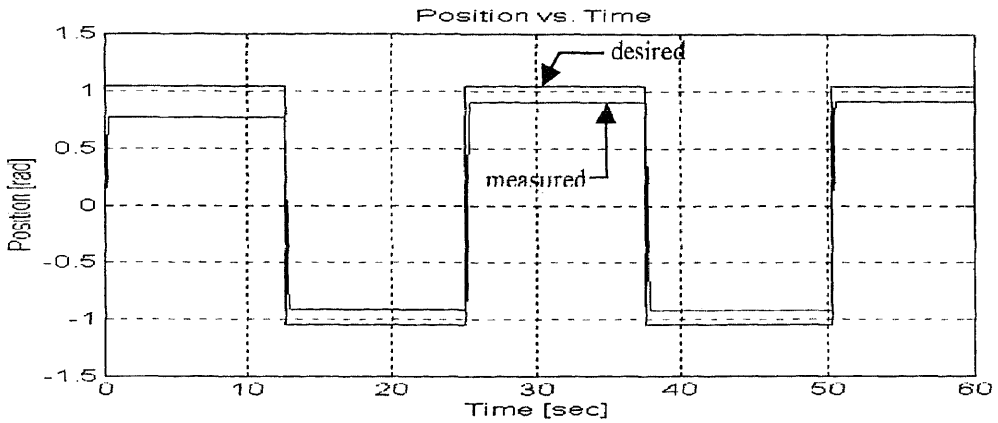


Figure 6.2 Experimental plot of position vs. time with square reference signal; No friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

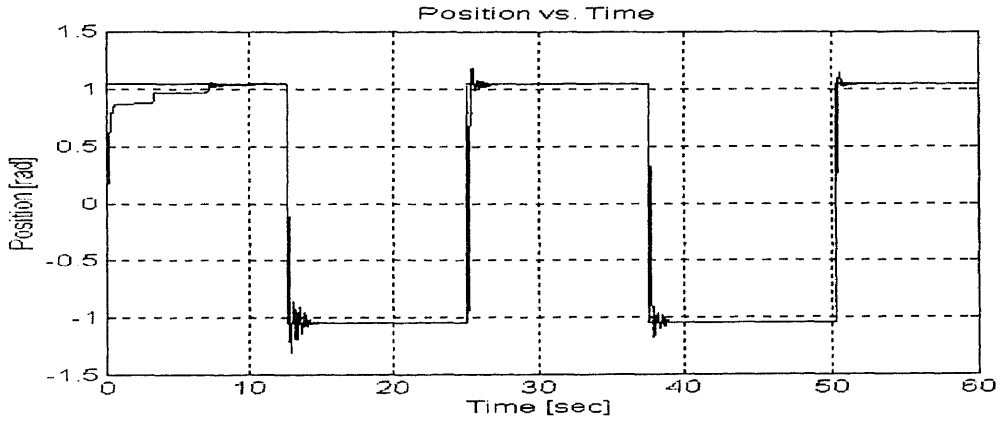


Figure 6.3 Experimental plot of position vs. time with square reference signal; Dynamic friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

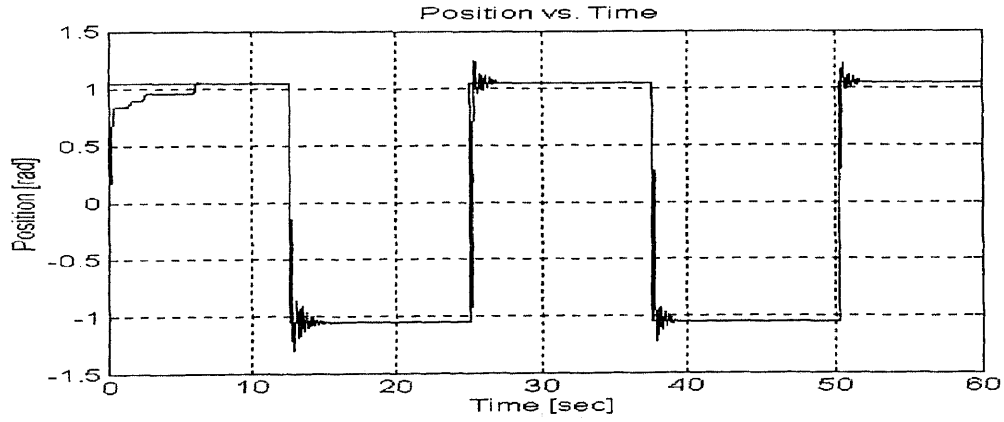


Figure 6.4 Experimental plot of position vs. time for square reference signal; Coulomb friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

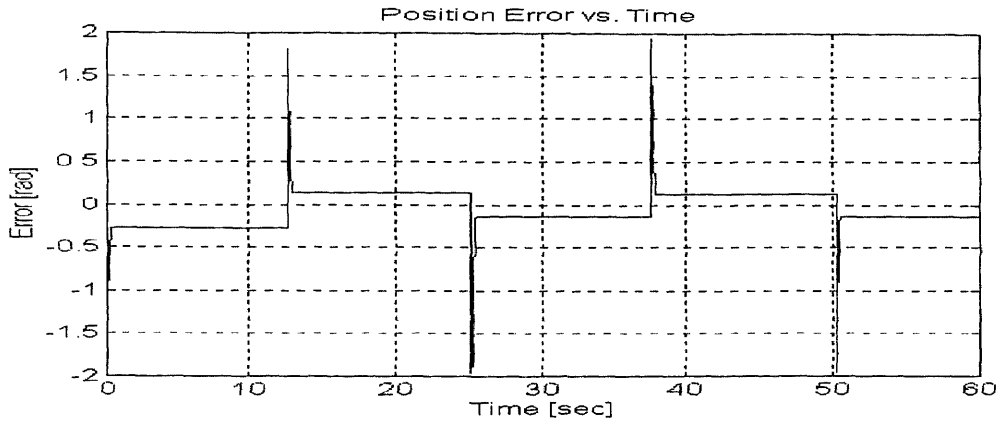


Figure 6.5 Experimental plot of position error vs. time for square reference signal; No friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

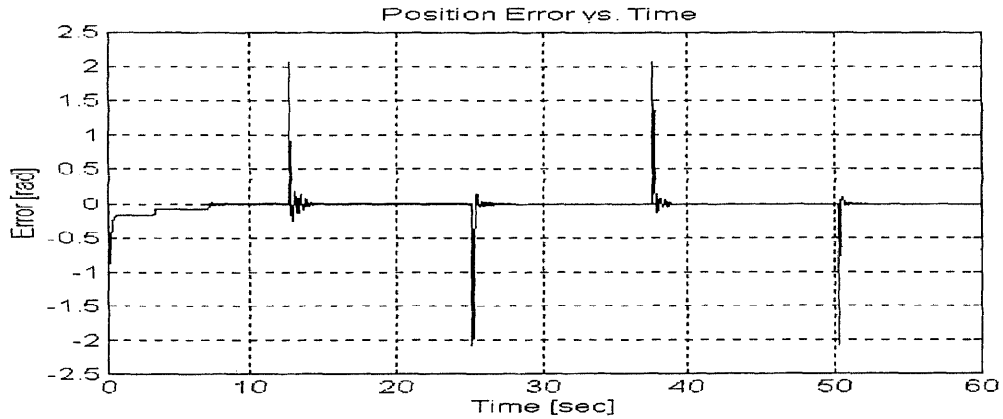


Figure 6.6 Experimental plot of position error vs. time for square reference signal; Dynamic friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

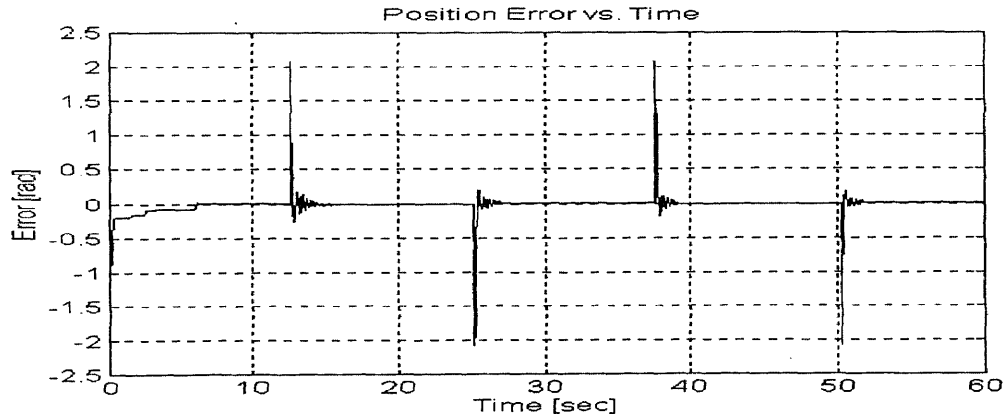


Figure 6.7 Experimental plot of position error vs. time for square reference signal; Coulomb friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

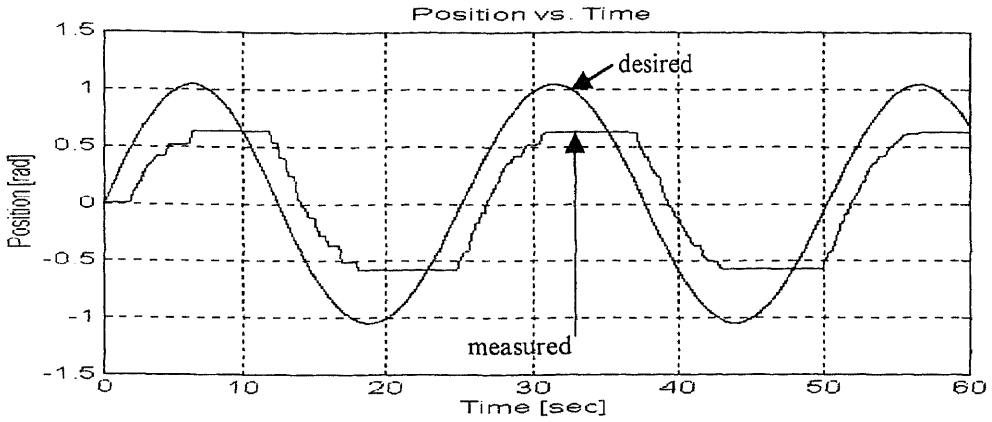


Figure 6.8 Experimental plot of position vs. time for sinusoidal reference input; No friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

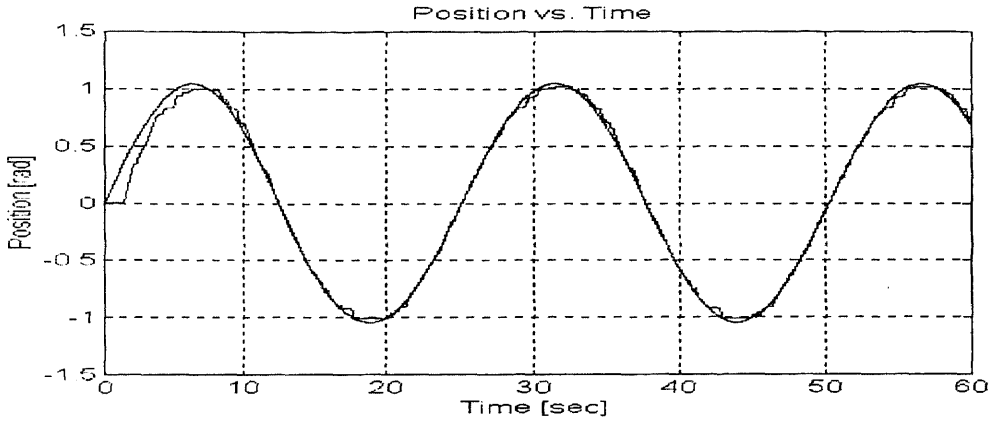


Figure 6.9 Experimental plot of position vs. time for sinusoidal reference input; Dynamic friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

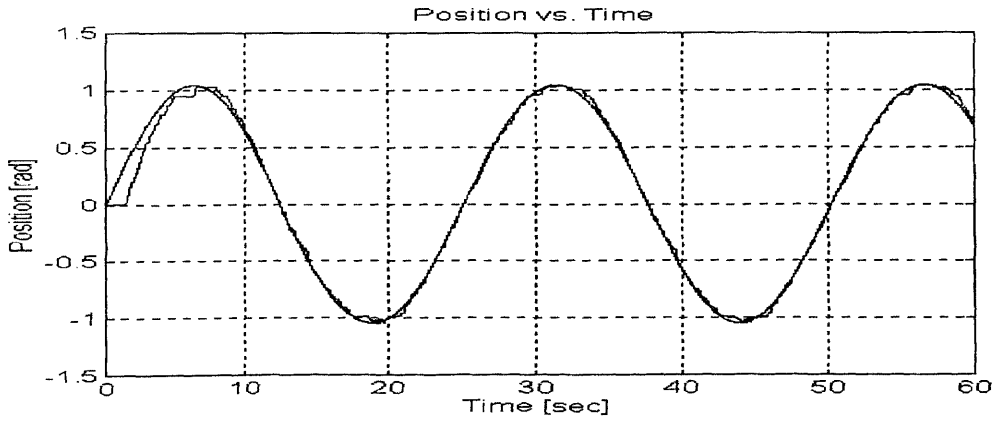


Figure 6.10 Experimental plot of position vs. time for sinusoidal reference input; Coulomb friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

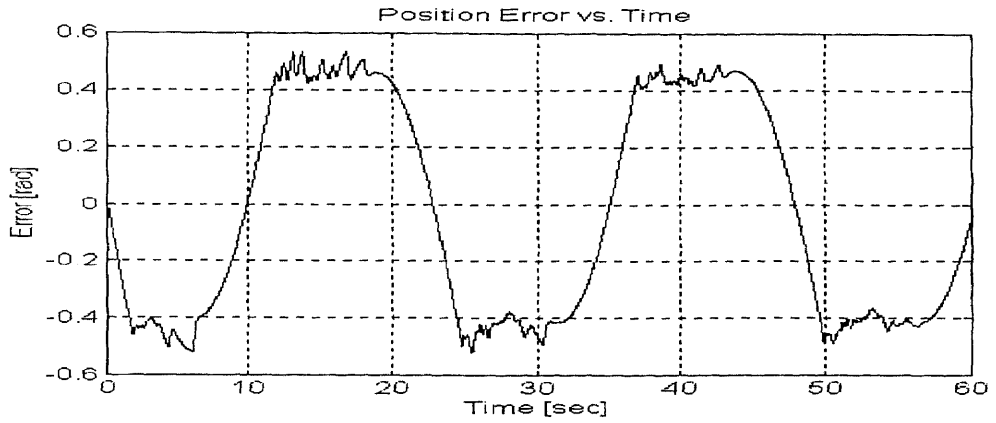


Figure 6.11 Experimental plot of position error vs. time for sinusoidal reference input; No friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

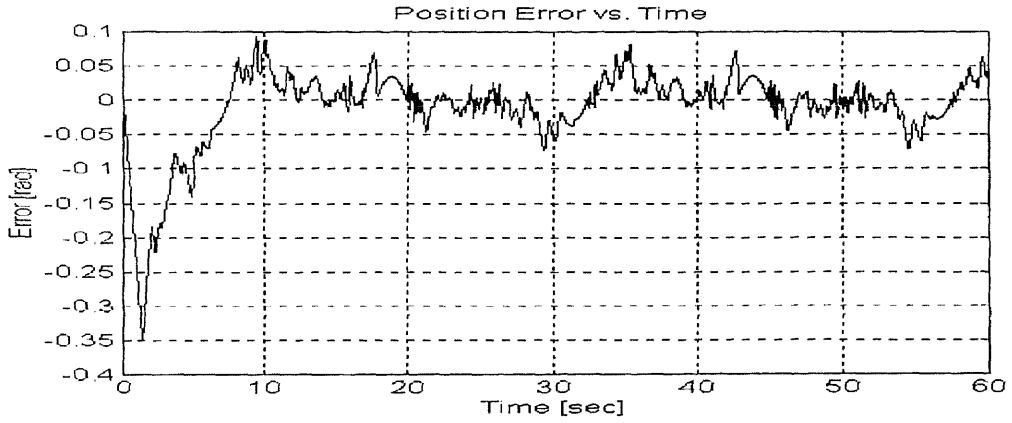


Figure 6.12 Experimental plot of position error vs. time for sinusoidal reference input; Dynamic friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

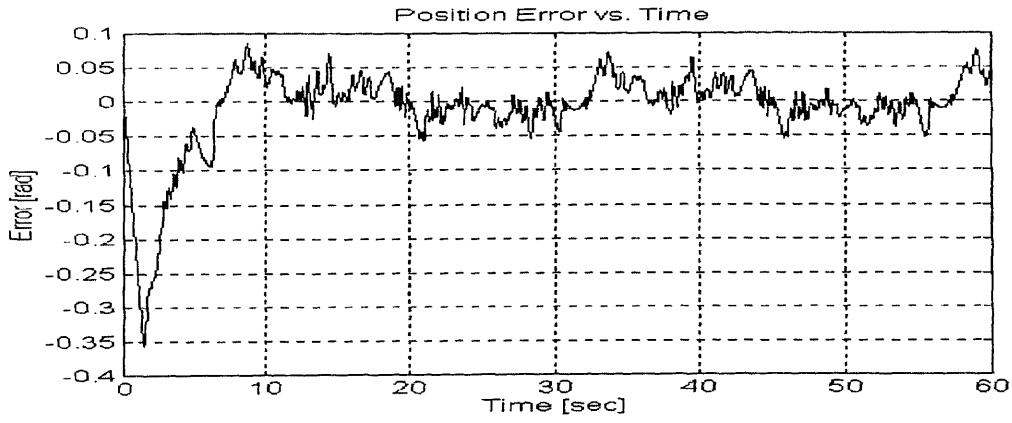


Figure 6.13 Experimental plot of position error vs. time for sinusoidal reference input; Coulomb friction compensation; range = ±1.04 rad; frequency = 0.25 rad/s.

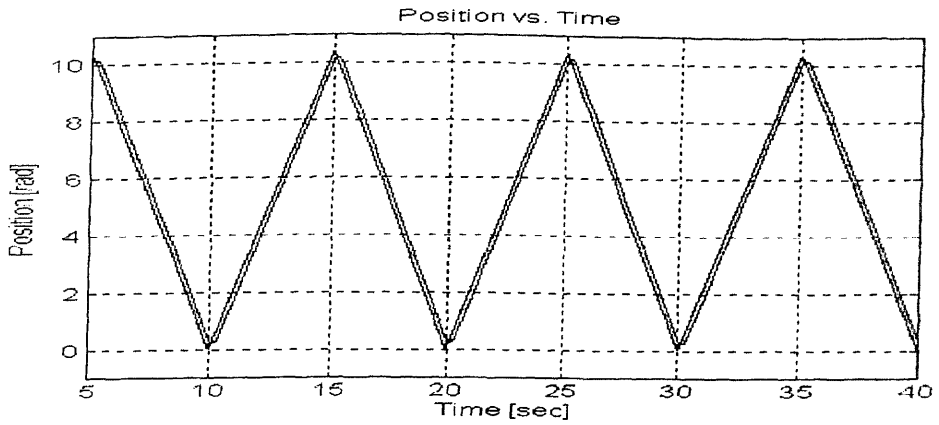


Figure 6.14 Experimental plot of position vs. time for triangular reference input; No friction compensation; range = 0 to 10.4 rad; frequency = 0.6283 rad/s.

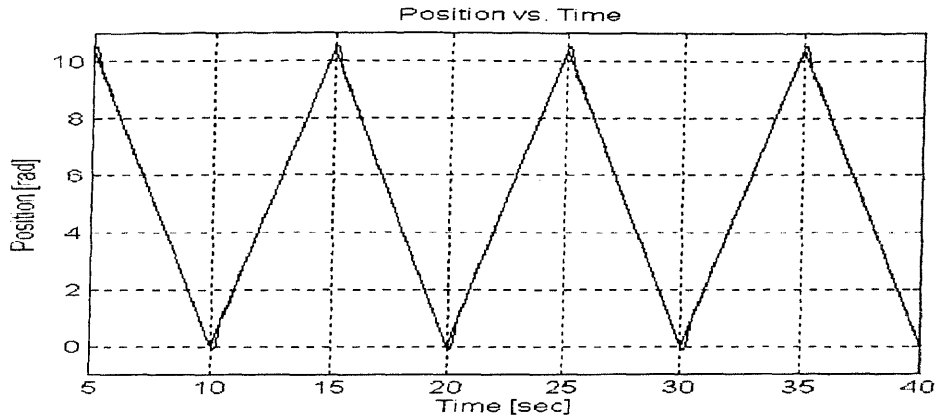


Figure 6.15 Experimental plot of position vs. time for triangular reference input; Dynamic friction compensation; range = 0 to 10.4 rad; frequency = 0.6283 rad/s.

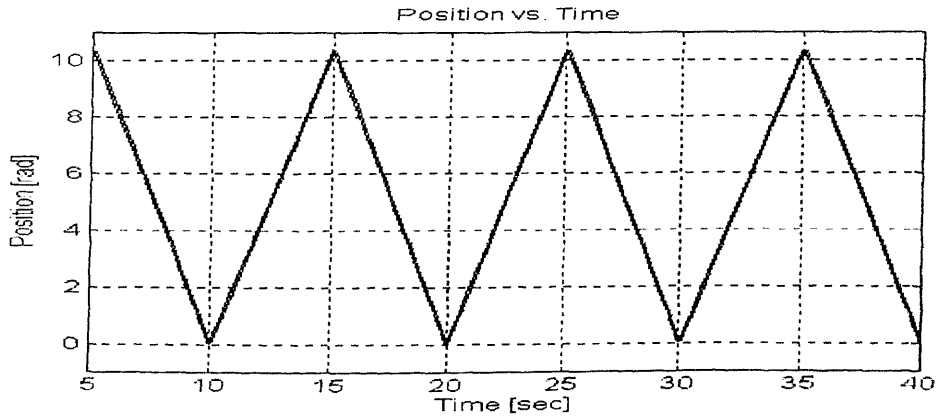


Figure 6.16 Experimental plot of position vs. time for triangular reference input; Coulomb friction compensation; range = 0 to 10.4 rad; frequency = 0.6283 rad/s.

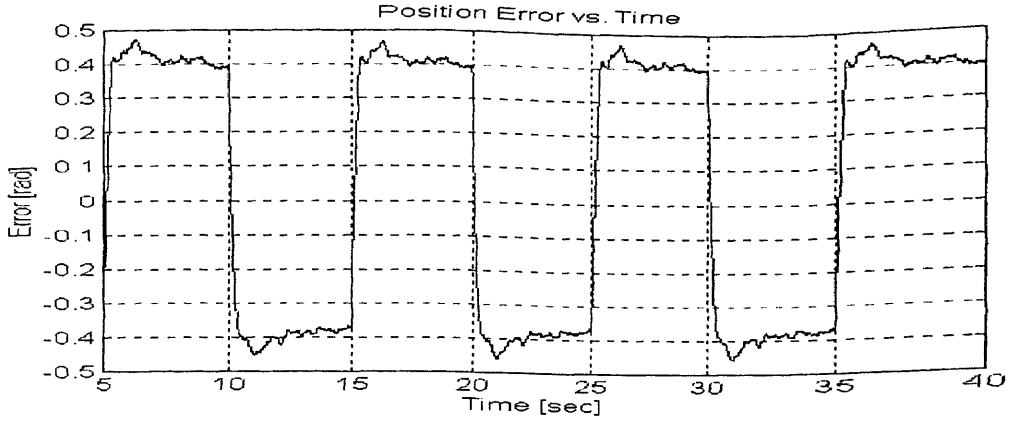


Figure 6.17 Experimental plot of position error vs. time for triangular reference input; No friction compensation; range = 0 to 10.4 rad; frequency = 0.6283 rad/s.

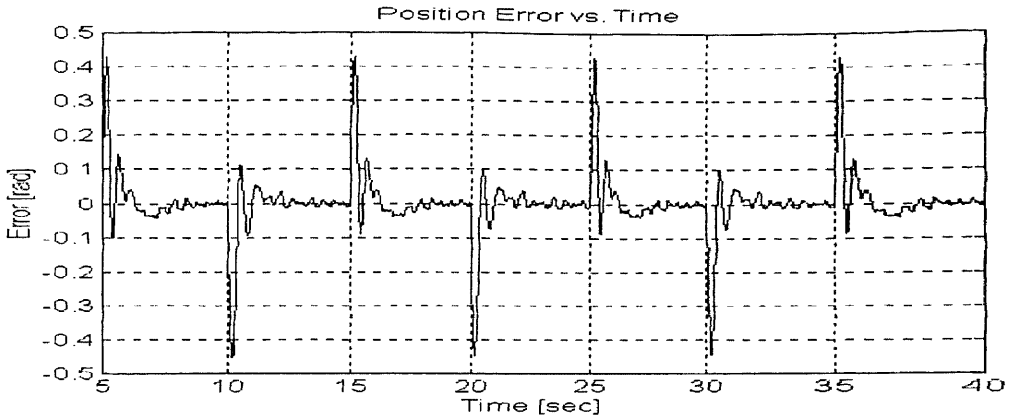


Figure 6.18 Experimental plot of position error vs. time for triangular reference input; Dynamic friction compensation; range = 0 to 10.4 rad; frequency = 0.6283 rad/s.

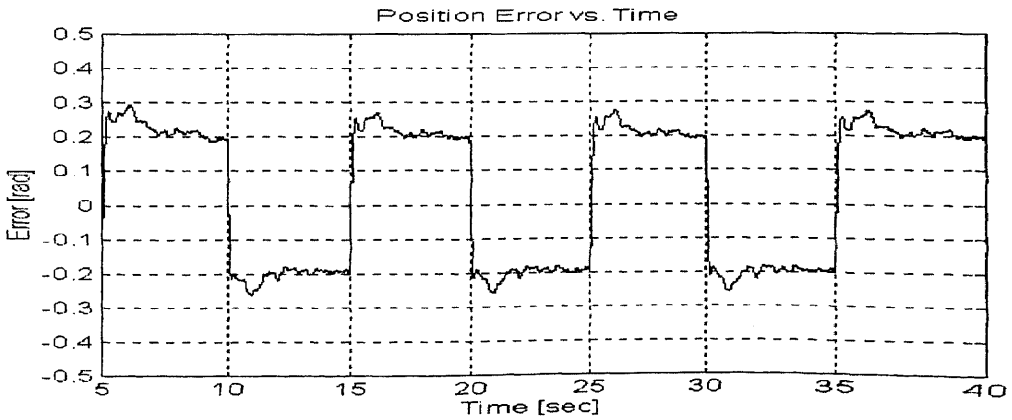


Figure 6.19 Experimental plot of position error vs. time for triangular reference input; Coulomb friction compensation; range = 0 to 10.4 rad; frequency = 0.6283 rad/s.

6.6 Experimental Results – Velocity Control Using Dynamic Friction Model

This section presents the results of the velocity control experiments. Data is presented for the baseline control law, the dynamic friction compensation, and Coulomb friction compensation, where available. The graphs include presentations of velocity vs. time, velocity error vs. time, and also estimated vs. measured friction. The results are summarized in Table 6.2. In the column titled "Compensation Type", "Dynamic Fric. Model" refers to friction compensation using the adaptable dynamic friction model, and "Coulomb Observer" refers to the Coulomb friction observer as designed by Amin (1996).

The third set of graphs, which compare experimentally measured friction to estimated friction, is particularly interesting because it gives insight into the accuracy of the friction estimation. The reader should take particular note, however, in the apparent error in estimated friction displayed in Figures 6.44 and 6.45 in comparison to the good friction estimate of, for example, Figures 6.26 and 6.27. For both sets of graphs, the reference velocity was a sine wave. However, in Figures 6.44 and 6.45, the velocity spans a much greater range. Apparently, both the Coulomb friction observer and the Dynamic friction estimate predict too much friction, similar in nature. Nonetheless, the reduction in error by both of these control algorithms is remarkable.

The author believes that the value of the friction estimate in Figures 6.44 and 6.45 is not only a measure of the friction in the journal bearings of the apparatus, but also of the other friction in the system, such as the friction in the support bearings, motor, and drive pulley. It is believed that this is the reason why an excess amount of friction is estimated by both the Coulomb friction observer and the dynamic friction estimator.

Moreover, it is seen that the Coulomb friction observer performs better at controlling velocity for high ranges of velocity. This is because the Coulomb friction observer does not have a disposition for any particular shape. It adapts to any force that is needed to decrease the error in the system. On the other hand, the dynamic friction estimator is tailored to compensate for only friction in hydrodynamic journal bearings. Accordingly, it has a harder time adapting to fit any shape necessary to reduce the error in the system. This is seen quite clearly by the large dip in the friction estimate shown in Figure 6.44, but not in Figure 6.45.

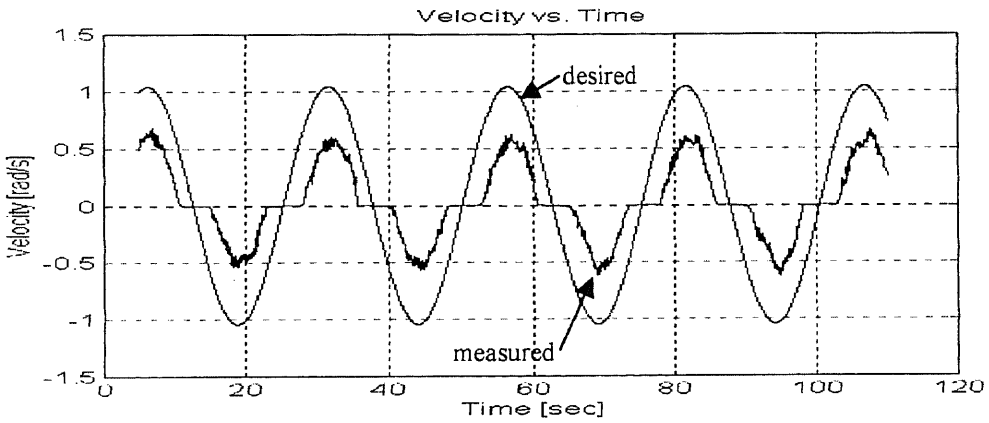


Figure 6.20 Experimental plot of velocity vs. time for sinusoidal reference input; No friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

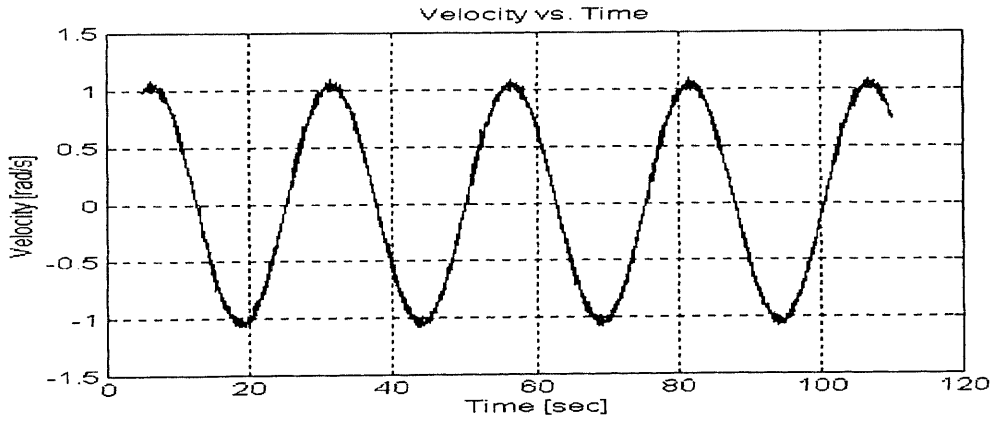


Figure 6.21 Experimental plot of velocity vs. time for sinusoidal reference input; Dynamic friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

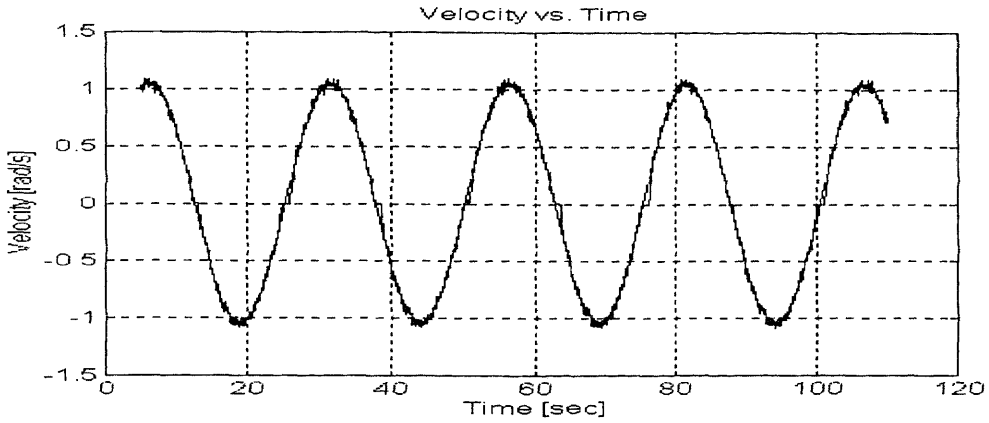


Figure 6.22 Experimental plot of velocity vs. time for sinusoidal reference input; Coulomb friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

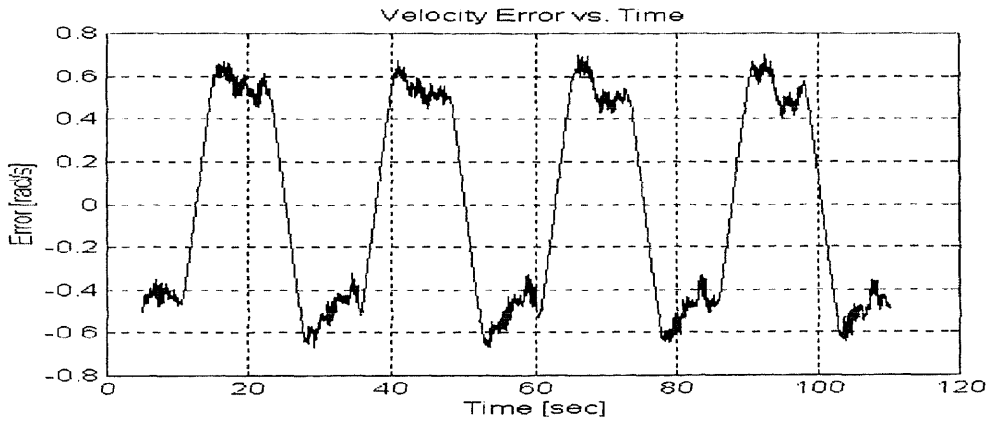


Figure 6.23 Experimental plot of velocity error vs. time for sinusoidal reference input; No friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

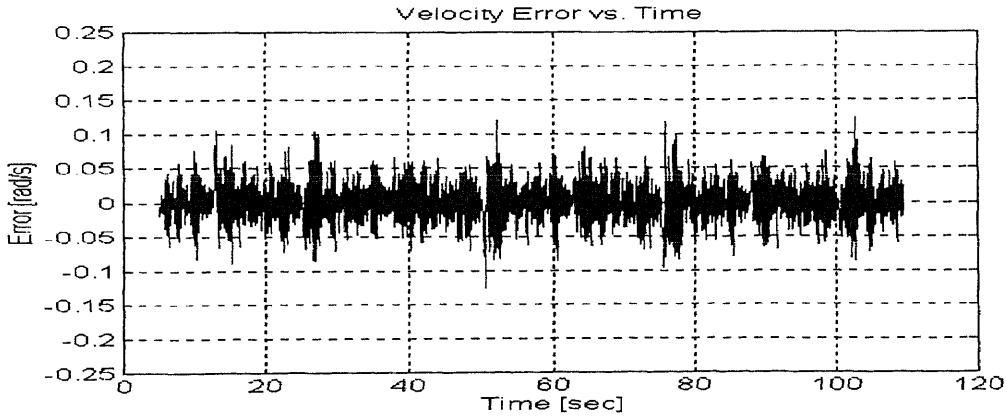


Figure 6.24 Experimental plot of velocity error vs. time for sinusoidal reference input; Dynamic friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

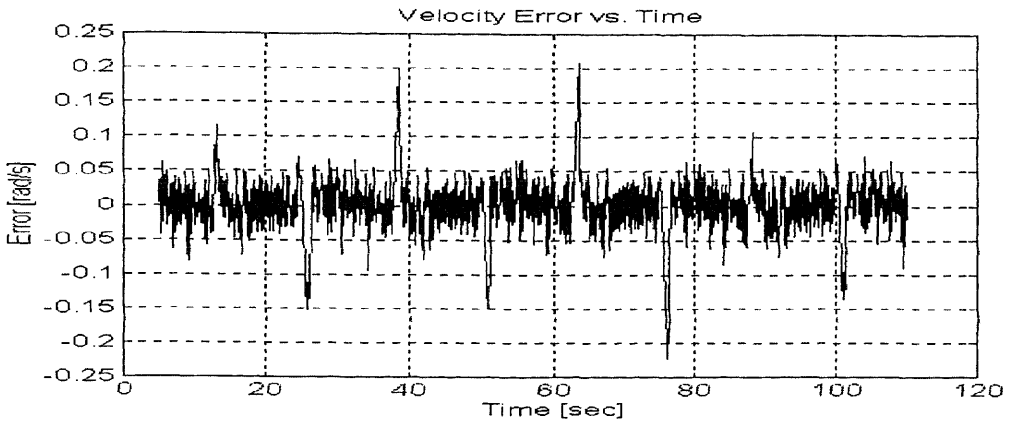


Figure 6.25 Experimental plot of velocity error vs. time for sinusoidal reference input; Coulomb friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

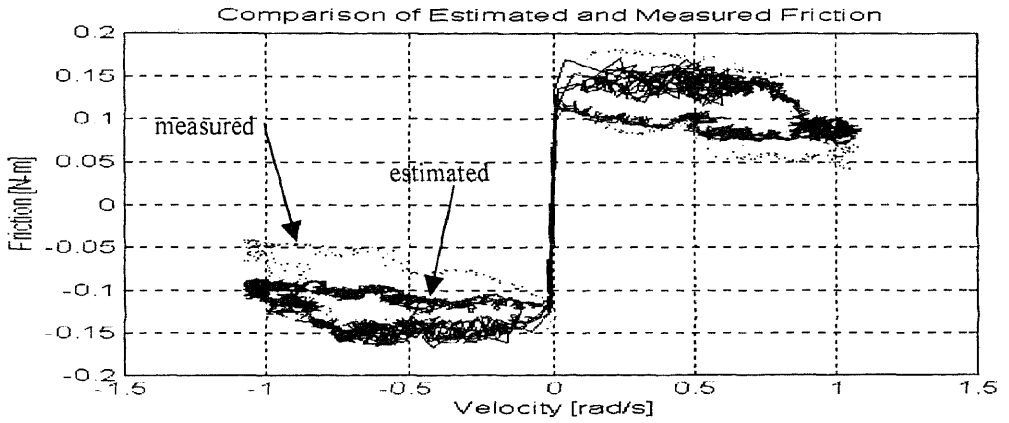


Figure 6.26 Experimental plot of estimated vs. measured friction for sinusoidal reference input; Dynamic friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

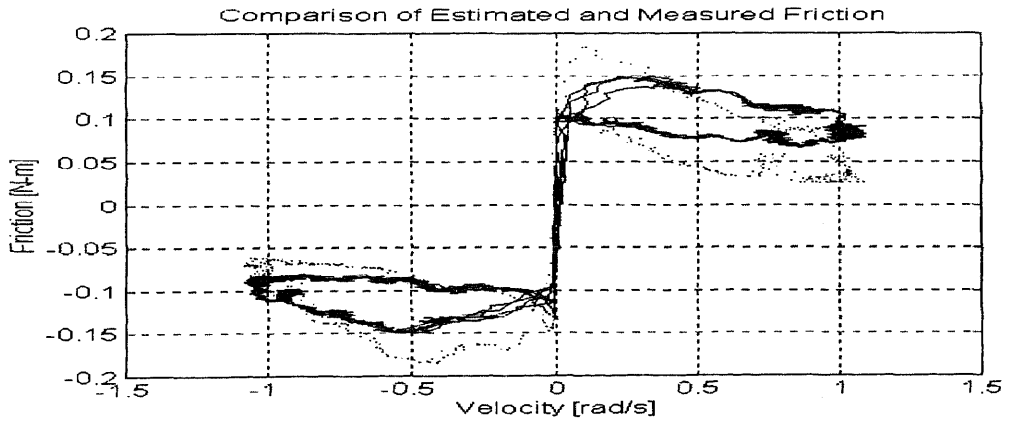


Figure 6.27 Experimental plot of estimated vs. measured friction for sinusoidal reference input; Coulomb friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

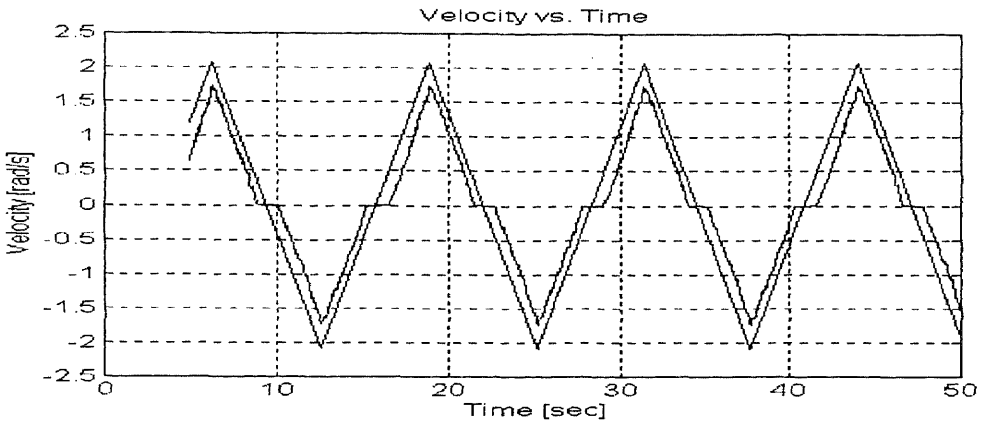


Figure 6.28 Experimental plot of velocity vs. time for triangular reference input; No friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

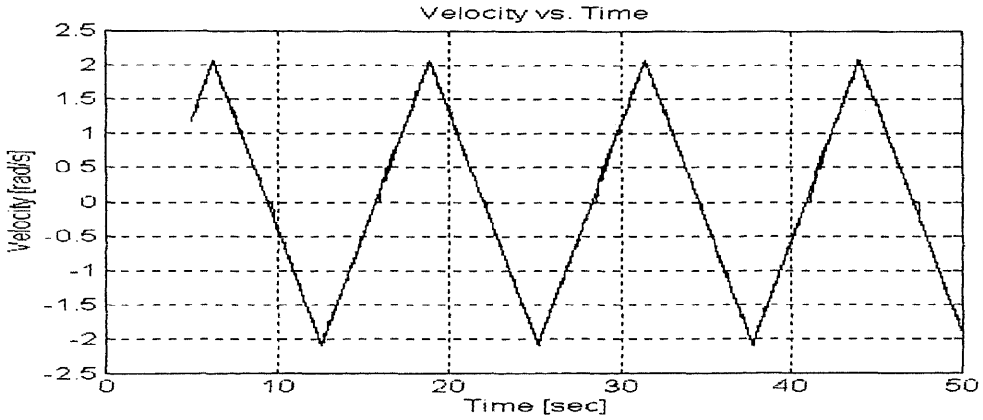


Figure 6.29 Experimental plot of velocity vs. time for triangular reference input; Dynamic friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

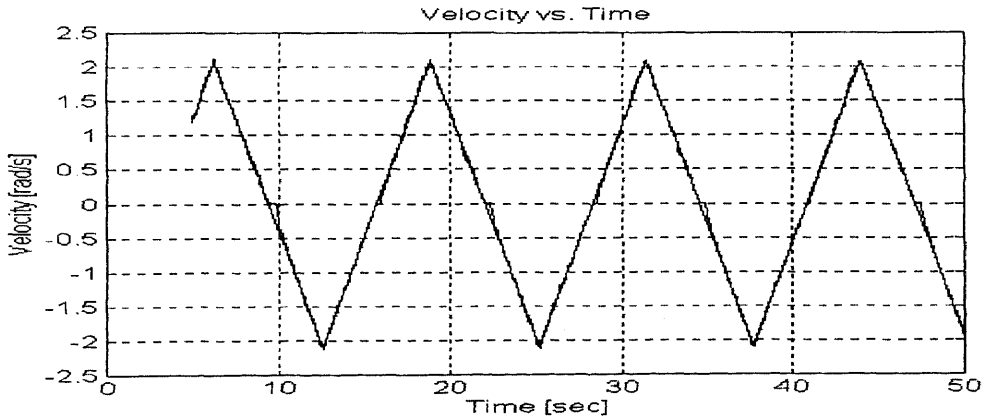


Figure 6.30 Experimental plot of velocity vs. time for triangular reference input; Coulomb friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

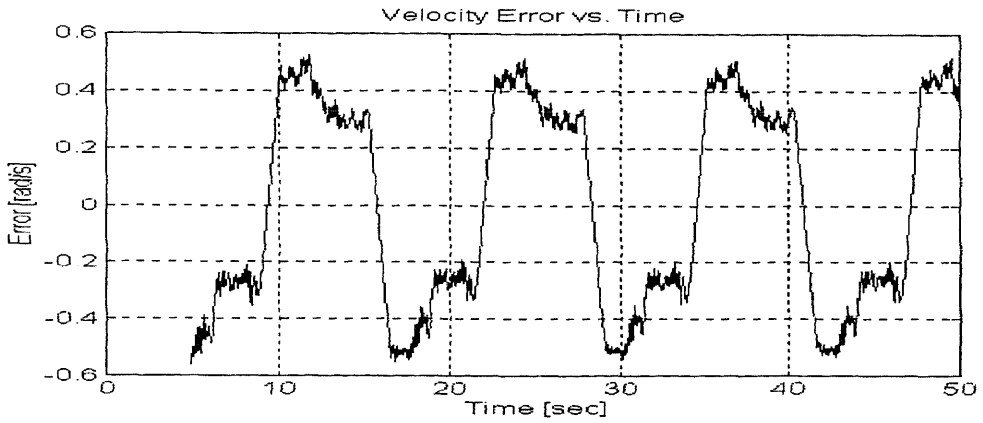


Figure 6.31 Experimental plot of velocity error vs. time for triangular reference input; No friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

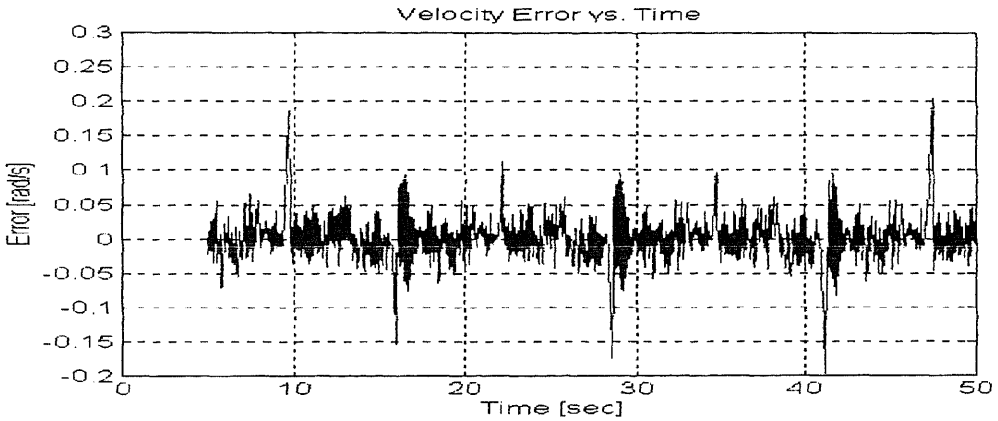


Figure 6.32 Experimental plot of velocity error vs. time for triangular reference input; Dynamic friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

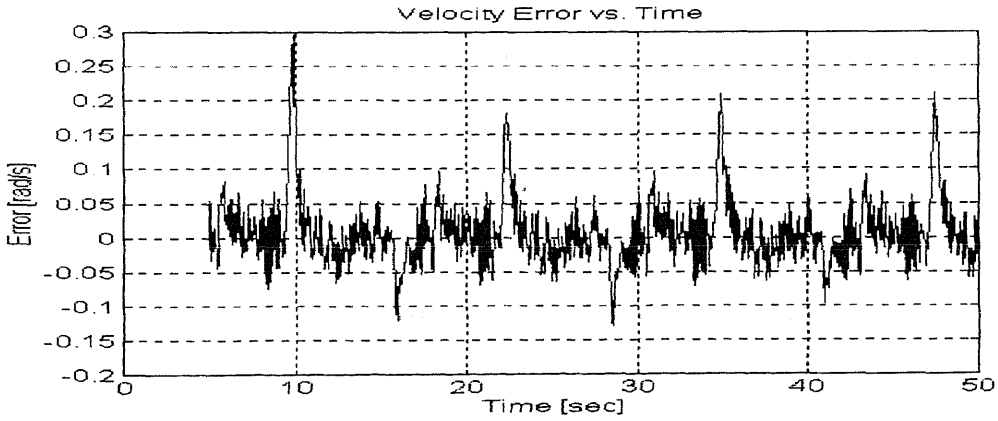


Figure 6.33 Experimental plot of velocity error vs. time for triangular reference input; Coulomb friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

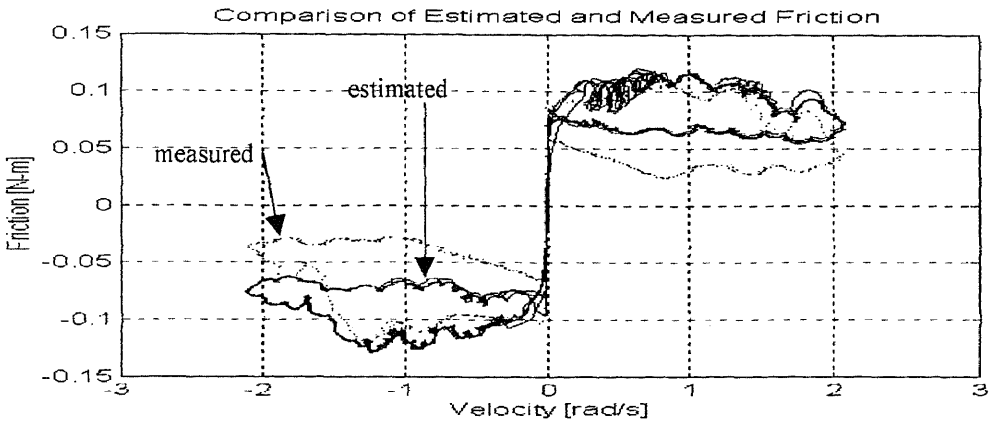


Figure 6.34 Experimental plot of estimated vs. measured friction for triangular reference input; Dynamic friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

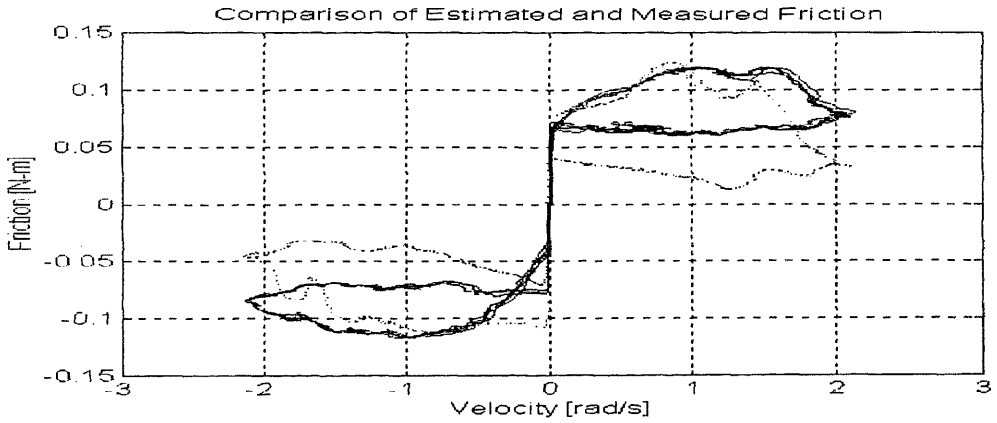


Figure 6.35 Experimental plot of estimated vs. measured friction for triangular reference input; Coulomb friction compensation; range = ±2.08 rad/s; frequency = 0.5 rad/s.

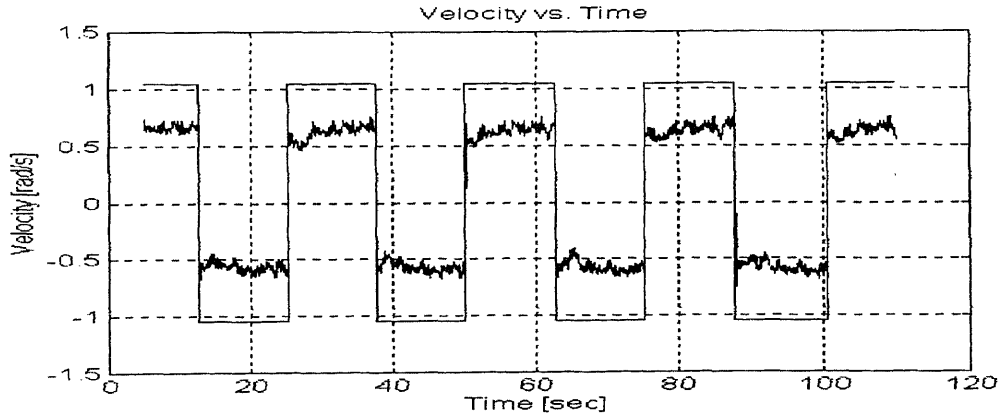


Figure 6.36 Experimental plot of velocity vs. time for square reference input; No friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

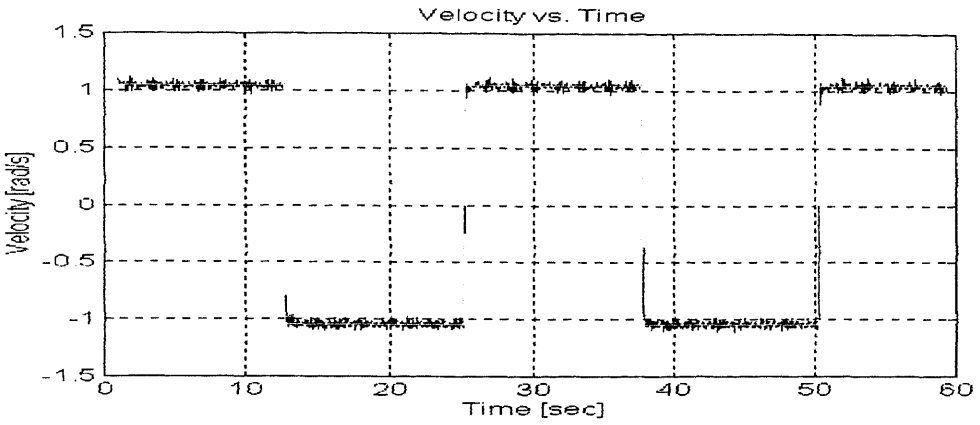


Figure 6.37 Experimental plot of velocity vs. time for square reference input; Dynamic friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

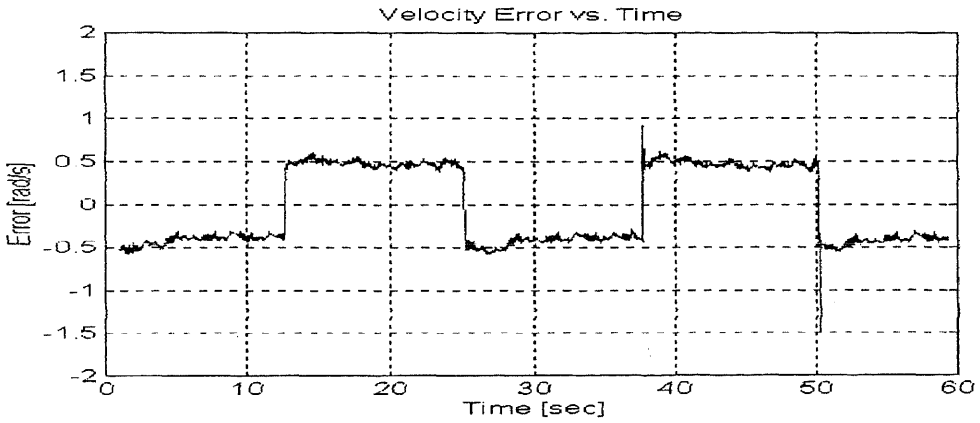


Figure 6.38 Experimental plot of velocity error vs. time for square reference input; No friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

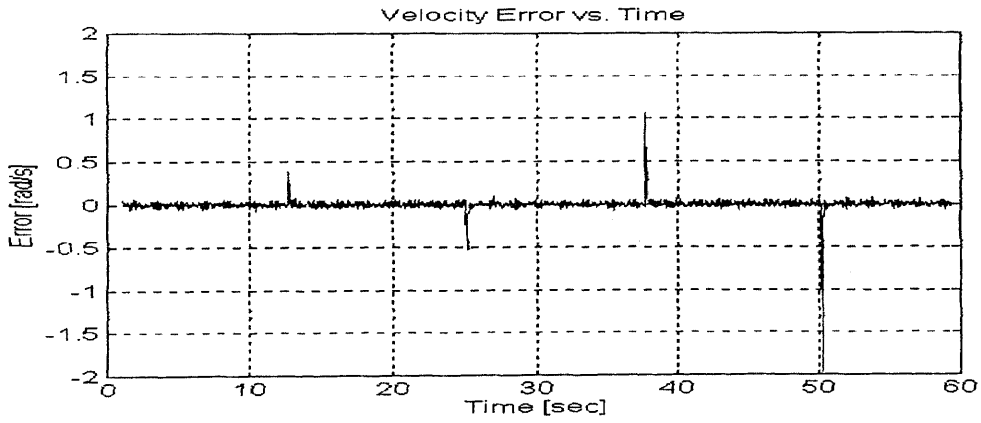


Figure 6.39 Experimental plot of velocity error vs. time for square reference input; Dynamic friction compensation; range = ±1.04 rad/s; frequency = 0.25 rad/s.

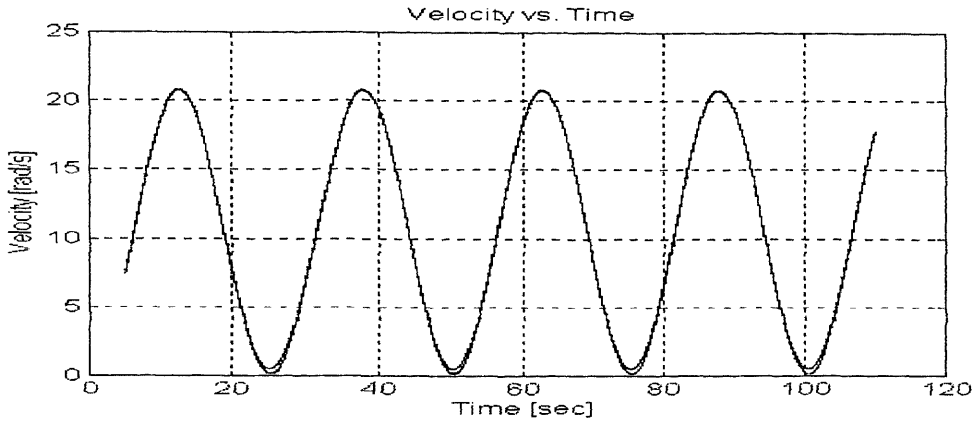


Figure 6.40 Experimental plot of velocity vs. time for sinusoidal reference input; No friction compensation; range = 0 to 20.8 rad/s; frequency = 0.25 rad/s.

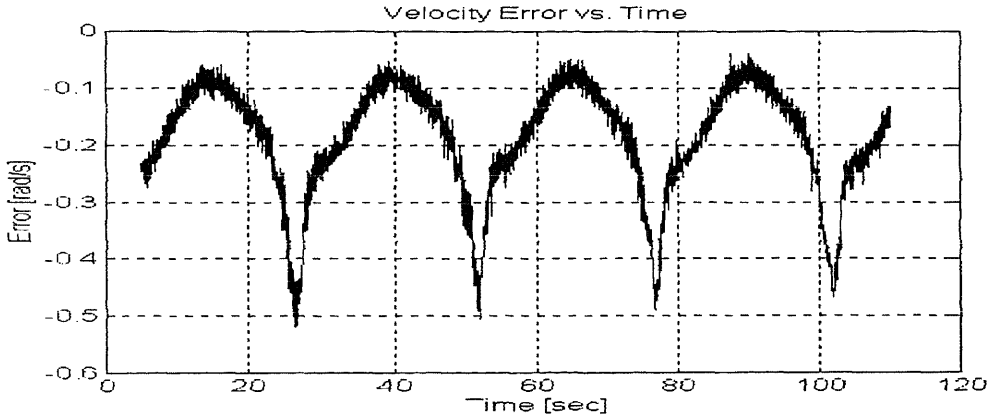


Figure 6.41 Experimental plot of velocity error vs. time for sinusoidal reference input; No friction compensation; range = 0 to 20.8 rad/s; frequency = 0.25 rad/s.

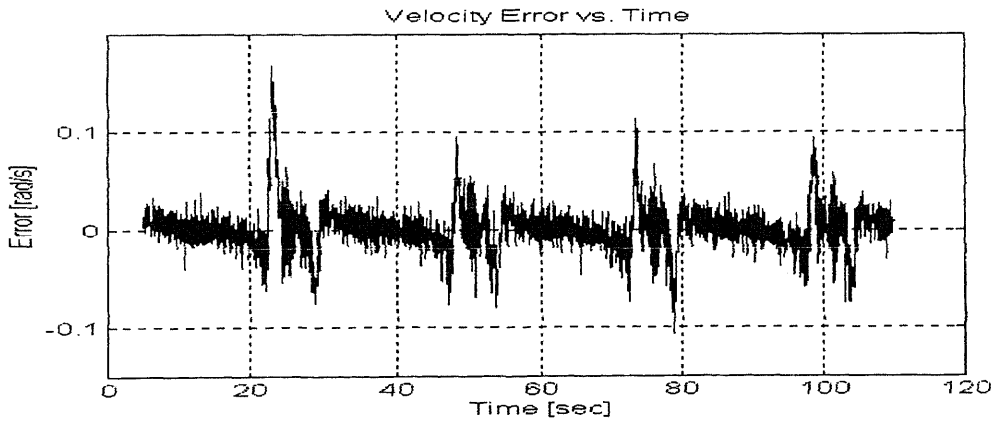


Figure 6.42 Experimental plot of velocity error vs. time for sinusoidal reference input; Dynamic friction compensation; range = 0 to 20.8 rad/s; frequency = 0.25 rad/s.

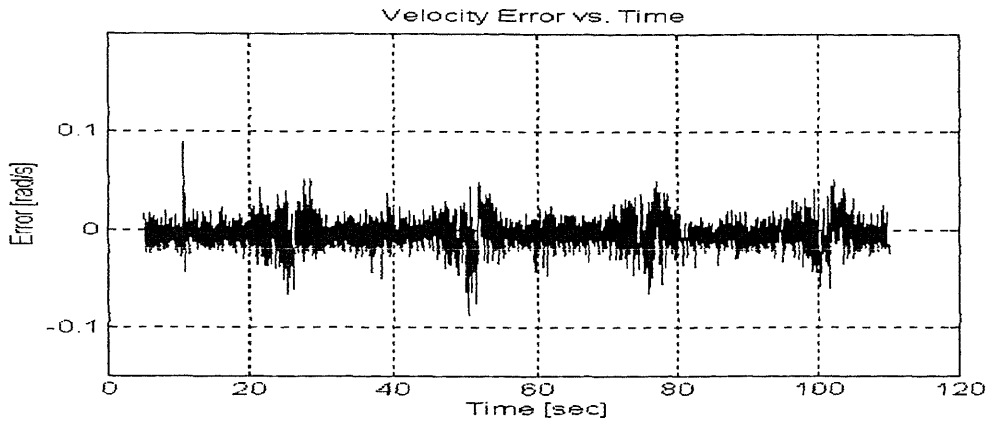


Figure 6.43 Experimental plot of velocity error vs. time for sinusoidal reference input; Coulomb friction compensation; range = 0 to 20.8 rad/s; frequency = 0.25 rad/s.

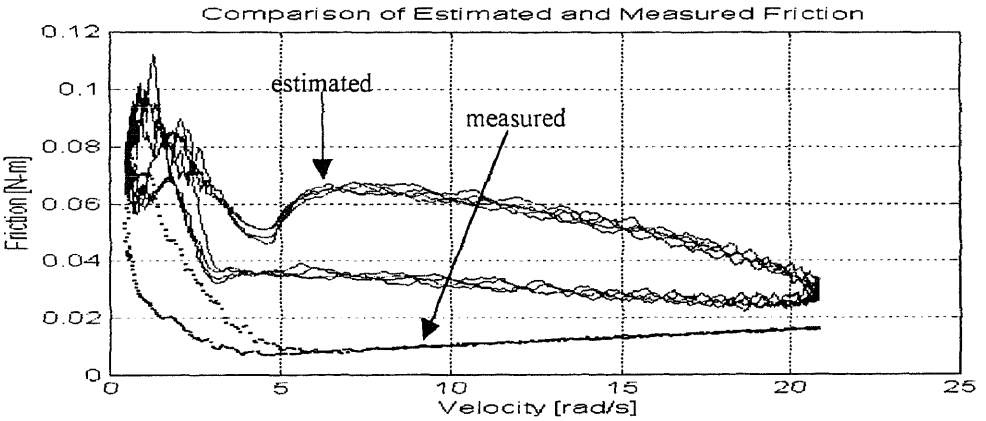


Figure 6.44 Experimental plot of estimated vs. measured friction for sinusoidal reference input; Dynamic friction compensation; range = 0 to 20.8 rad/s; frequency = 0.25 rad/s.

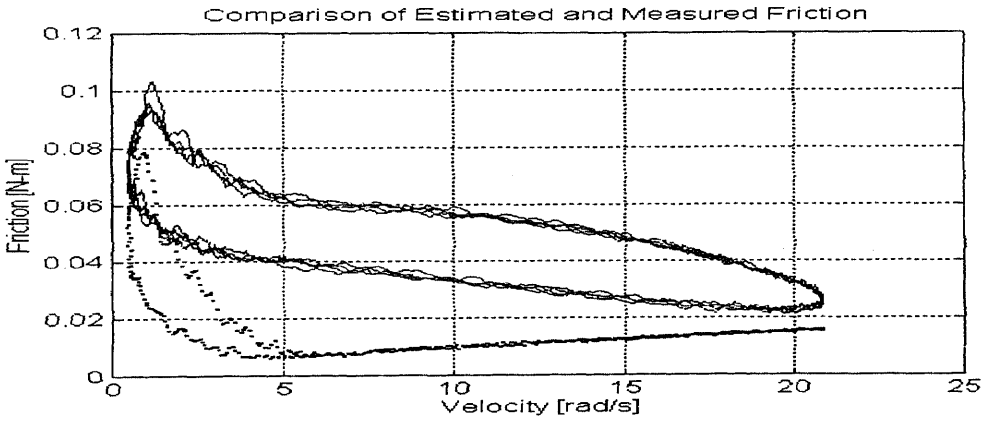


Figure 6.45 Experimental plot of estimated vs. measured friction for sinusoidal reference input; Coulomb friction compensation; range = 0 to 20.8 rad/s; frequency = 0.25 rad/s.

CHAPTER 7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE STUDY

In this thesis, the nature of friction in hydrodynamic journal bearings is experimentally measured for various loads and lubricants and no lubricant. Results agree with previous studies in the dynamic behavior of friction in lubricated contacts. A dynamic friction model is simulated and compared to experimental results. The model is shown to correlate very well with experimental measurements.

Next, the dynamic friction model is reduced from a fourth to a second order equation by neglecting inertia terms in the model. Comparison of the second and fourth order model shows negligible difference in predicted friction. Also, the model is slightly simplified by combining several terms in order to ease the selection of model parameters.

The simplified, reduced-order dynamic friction model is then incorporated into a control program. The program runs an apparatus with a large amount of friction generated in four hydrodynamic journal bearings. Two of the parameters in the model were learned during the experiment to better match the estimated friction to the total friction in the system. The adaptive dynamic friction model is shown to greatly improve the accuracy of both position and velocity control.

The adaptive dynamic friction model is also compared to a control program incorporating a Coulomb friction observer. Comparison shows that the two methods provide similar reductions in tracking errors compared to a baseline control algorithm with no friction compensation.

The method used to learn parameters for the dynamic friction model was shown to perform adequately in this experimental study. However, more theoretical study needs to be performed to justify this method. Also, the accuracy of friction estimation by the dynamic friction model should be studied with more advanced parameter estimation techniques, such as the one used in the Coulomb friction observer.

Finally, more research needs to be performed to study the behavior of friction in other contacts – line contacts, surface contacts, etc. - such as is found in machines with ball screw drives and radial seals.

APPENDIX A

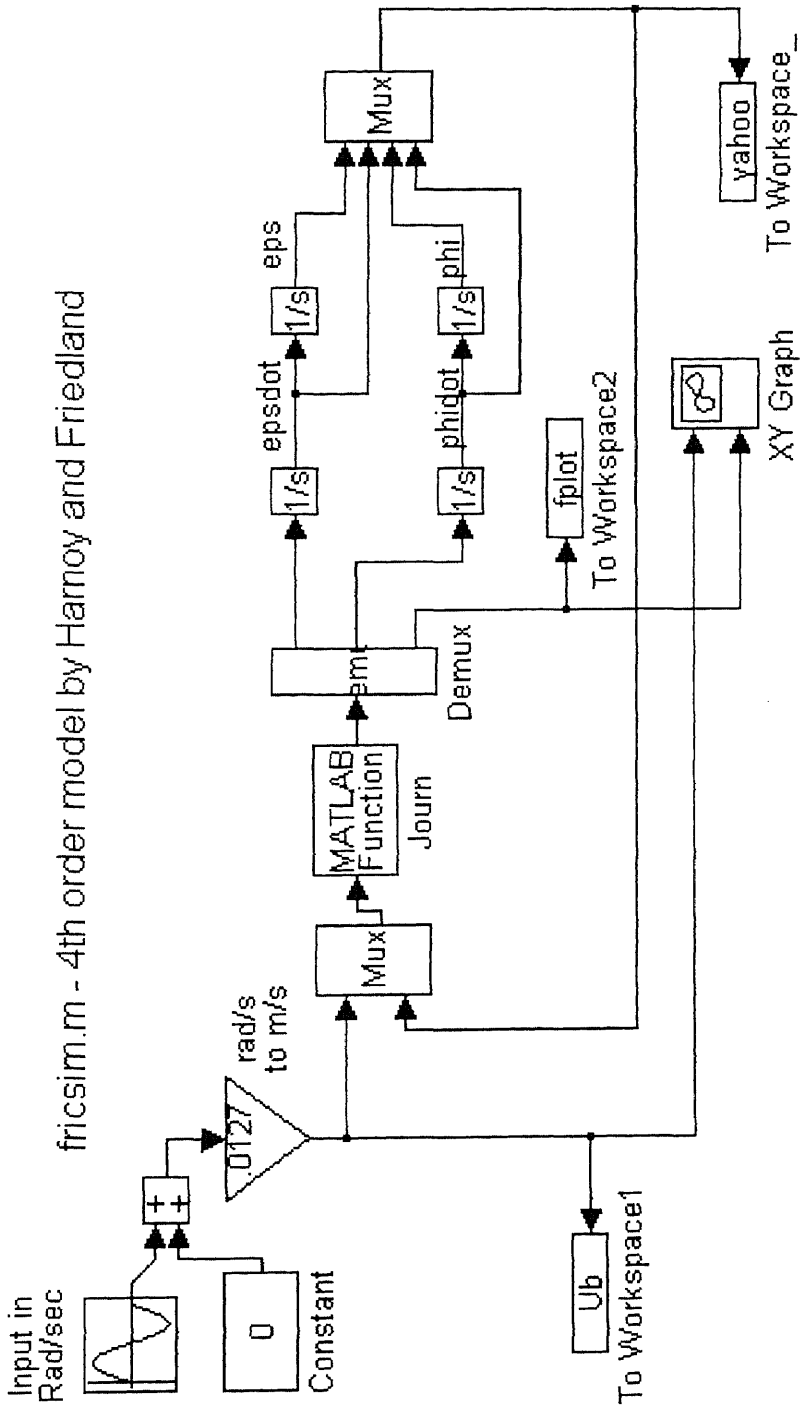
LISTING OF MATLAB ROUTINE JORDAT.M

```
% jordat3.m - data file for fricsim.m: journal bearing dynamic friction model
%  Includes all modifications necessary to comply with the Harnoy/Friedland paper
%  Created October 7, 1996
%
% jordat3.m  1) Lets the user change the simulation variables  2) Calculates the requisite
%  dimensionless coefficients  3) Calculates epsilon transition by iteration

global FF EPS_CR EPS_B RCL MM FM KO Utr eps epsdot phi phidot;
global slope_fudge KOC CKO SKO slope_fudgec;

if (~exist('switch2'))
 % *** DIMENSIONED CONSTANTS ***

 Utr= 0.05;                    % Transition velocity [m/s]
 C=  0.002 * 2.54 / 100;       % Average journal clearance.
                               % Converted from [inches] to [meters]
 R=  0.5 * 2.54 / 100;         % Journal diameter converted from [in] to [meters]
 M=  5 / 2.2;                  % Rotating mass converted from [lbs] to [kg]
 MIU= .002;                    % Viscocity of oil in [kg/(m-sec)]
 L= 0.75 * 2.54 /100;          % Length of journal converted from [in] to [m]
 F= 83.5;                      % Average Normal load PER JOURNAL [Newtons]
 EPS_B = 0.99;                 % Eccentricity at boundary to mixed transition
 Kv=15;                        % Tafazoli constant


 % *** DIMENSIONLESS CONSTANTS ***

 slope_fudge=32;               % Factor for hydrodynamic slope
 global slope_fudge;
 FM= 0.27;                     % Maximum Stribek Friction Coefficient
 KO =625000;                   % Stiffness constant for the asperities
 KOC=625000;                   % Initial guess at KO for compensation
 CKO=1.5e5;                    % Learning constant for KOC
 SKO=1;                        % Learning constant for slope_fudgec
 slope_fudgec=30;              % Initial guess at slope_fudge for compensation
end % end if ~exist
switch2=1;

switch=1;
while (switch~=0)              % Menu of current coefficient values
 clc;
 disp('The following are the current values of the coefficients:')
 disp(' ');
 disp(['1.  FM=   ', num2str(FM)]);
 disp(['2.  Utr=  ', num2str(Utr), ' m/s']);
 disp(['3.  KO=   ', num2str(KO), ' ']);
 disp(['4.  M=    ', num2str(M*2.2), ' lbs']);
```

```
disp(['5.  MIU=    ', num2str(MIU), ' kg/(m-sec)']);
disp(['6.  F=      ', num2str(F), ' Newtons']);
disp(['7.  EPS_B= ', num2str(EPS_B)]);
disp(['8.  C=      ', num2str(C*100/2.54), ' inches']);
disp(['9.  Fudge= ', num2str(slope_fudge)]);
disp(['10. KOC=   ', num2str(KOC)]);
disp(['11. CKO=   ', num2str(CKO)]);
disp(['12. SKO=   ', num2str(SKO)]);
disp(['X.  R=     ', num2str(R*100/2.54), ' inches']);
disp(['X.  L=     ', num2str(L*100/2.54), ' inch']);
disp(' ');
switch=input('Which would you like to modify?');
disp(' ');
if isempty(switch); switch=0; end
if switch==1; FM=input('Enter new FM: '); end
if switch==2; Utr=input('Enter new Utr [m/s]: '); end
if switch==3; KO=input('Enter new KO: '); end
if switch==4; M=input('Enter new M [lbs]: ')/2.2; end
if switch==5; MIU=input('Enter new MIU [kg/(m-sec)]: '); end
if switch==6; F=input('Enter new F [N]: '); end
if switch==7; EPS_B=input('Enter new EPS_B: '); end
if switch==8; C=input('Enter new C [inches]: ')*2.54/100; end
if switch==9; slope_fudge=input('Enter new slope fudge factor: '); end
if switch==10; KOC=input('Enter new KOC: '); end
if switch==11; CKO=input('Enter new CKO: '); end
if switch==12; SKO=input('Enter new SKO: '); end
end % end while loop


FF= (C^2/ (MIU* Utr* L^3)) * F;            % Dimensionless Load calculated from eq. [10]
MM= (C^3* Utr/ (MIU* L^3* R^2))* M;        % Dimensionless rotating mass from eq. [25]
RCL= R*C/L^2;                              % Dimensionless ratio R*C/L^2

Nave_lbs= F*2.2/9.81;                      % Normal load per journal [lbs]

x0=[0.96];                                 % Initial guess at eps_cr
% options(2)=1e-5;                         % Sets tolerance on x
% options(3)=1e-5;                         % Sets tolerance on f(x)

P1=FF;                                     % Determines the value of EPS_CR
x= fsolve('4*P1^2 - x(1)^2*(pi^2/4/(1-x(1)^2)^3 + 4*(x(1)^2/(1-x(1)^2)^4))', x0, [] , [], P1);

EPS_CR=x(1)
Kactual=KO/(EPS_B - EPS_CR)

% I'm not sure if this next section works, but it can't hurt
eps= EPS_CR;              % Initializes eps to EPS_CR.  Helps it to converge faster
epsdot= 0;                % Initializes epsdot
phi=0.5;                  % Initializes phi
phidot=0;                 % Initializes phidot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B

BLOCK DIAGRAM USED TO SIMULATE THE MODEL



APPENDIX C

LISTING OF MATLAB FUNCTION JOURN.M

```
% journ.m - function defining the friction model differential equations in
% Harnoy & Friedland

% 4:34PM 9/23/93
% Corrected 3:22PM 10/20/93 for wrong phi_dd equation
% Corrected 8:17PM 2/24/94: changed sign in eps_dd equation on +eps*thdot^2
% modified 8:16PM 7/9/94 to make all global variables caps
% Modiifed on 7/17/95 by Hanuman Rachoor with equations having dimesions
% Modified 10/96 by Simon Cohn to correct U to dimensionless form and to change the
%  Calculation of k
function x = journal(u)

global FF EPS_CR EPS_B RCL MM FM KO Utr slope_fudge
% global eps epsdot phi phidot;
% global U d J11 J12 J22 k delta eps_dd phi_dd f Ff

U = u(1)/ Utr;              % U= dimensionless velocity  Utr MUST be in m/s
eps = u(2);
epsdot = u(3);
phi = u(4);
phidot = u(5);

d = 1 - eps^2;
J11 = pi/2/d^1.5;
J12 = -2*eps/d^2;
J22 = pi*(1+2*eps^2)/2/d^2.5;

k=KO * (eps-EPS_CR);

delta = 1;
if (eps<EPS_CR) delta = 0;
end

d = 0.5* eps* abs(U) - eps* phidot;
eps_dd = (FF*cos(phi) - k*(eps-EPS_CR)*delta + d*J12 - epsdot*J22)/MM ...
        + eps*phidot^2;
phi_dd = (-FF*sin(phi) + d*J11 - epsdot* J12)/ (eps*MM) - 2*epsdot* phidot/ eps;
Ff = FM * k* (eps-EPS_CR) *delta* sign(U)  +  slope_fudge*RCL* 2* pi* U/ (1-eps^2)^0.5;
f = Ff/FF;

x = [eps_dd  phi_dd f]';
```

APPENDIX D

LISTING OF SOURCE CODE FOR VELOCITY CONTROL PROGRAM

```
/* CONTRLV2.C
   Written by: Jayesh Amin
   Last modified: Nov. 23 1995

   Modified by Simon Cohn (last modification on 12/19/96) to change
   method of friction estimation from Coulomb Friction Observer to
   estimation by Harnoy/Friedland model (simplified to 2nd order) with
   variable K0 and slope

   Source code for velocity control of the friction apparatus.
   Uses DACA board for I/O and requires to be linked to the modified
   version of the DACA library (modified by Jayesh on April 20th -
   available in Dynamic Systems Lab ).
   Uses LabWindows User interface Library for GUI. vcontrol.uir contains
   the LabWindows resources and should be present in the same directory
   as this executable at run-time.
*/


#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <dacamu.h>
// Header file created by LabWindows
#include "control.h"

int hpanel,signal;                  // Handle for panel and signal pointer
float low,high,freq,period;         // Parameters for the signal generator
float (*sigfun)(void);              // Pointer for the ref. signal generator

float time=0.0,TS=0.002,totime=20.00;   // Running time, Sampling Period and max
int n=0,i,compornot=1,nsamp=4;          // Sample Number,flag for indicating
                                        //  whether compensating or not.

float z=0,zd,prad=0,padd=0,zf=0,zfd,a;          // Observer states and derivatives
float *u,*x1,*x2,*ref,*error,*tptr,*fric,*mfric; // Important sampled variable storage
float l=15.0,kf=.01;                            // Velocity and friction observer gains

// Variables important for dynamic friction model
float R=0.0127,C=0.0000508,M=2.2727,MIU=0.002;
float L=0.01905,F=104.0,CK0=15000.0,SK0=2.0,FM=0.26,SLOPE_FUDGE=33.0,N=104.0;
float UTR=0.05,K0=650000.0,EPS_CR=0.9747;
float EPS,EPSDOT,PHI,PHIDOT;
float FF,MM,RCL;

int getcount();             // Returns the current count from the Encoder
float triagen();            // Reference signal generators
float squaregen();
```

```
float sinegen();

float sinphase=1.5708;     // initial phase for sine generator (for smooth start)

FILE *fp;                  // File pointer for storing data

void timerISR();           // Sampling and Control Routine (the main engine!)


int main()
{
int done=0,sw=0,csw;              // some internal variables
int hp,hc;                        // Event Handles
void StartRun();                  // Initializes everything at start of run
void StopRun();                   // Cleans up the house after the run
void LatchParams();               // Latches critical parameters at start

hpanel=LoadPanel("control.uir",CONTROL);        // GUI Initialization
DisplayPanel(hpanel);
MessagePopup("Simon's Velocity Control '96");   // Fancy stuff !

u=(float*)malloc(sizeof(float)*6500);           // Allocate RAM for storage of variables
x1=(float*)malloc(sizeof(float)*6500);
x2=(float*)malloc(sizeof(float)*6500);
ref=(float*)malloc(sizeof(float)*6500);
error=(float*)malloc(sizeof(float)*6500);
fric=(float*)malloc(sizeof(float)*6500);
mfric=(float*)malloc(sizeof(float)*6500);

if ((tptr=(float*)malloc(sizeof(float)*6500))==NULL)
{
 MessagePopup("Memory Allocation Problem - Not Enough memory !!");
 return 1;
}

BinaryWrite(0x0018);              // reset the encoder count to 0
AnalogWrite(0,2048);              // Reset D/A output to 0 V
LatchParams();                    // Latch critical parameters

while(!done)                      // endless loop till it's all done
{
  if(GetUserEvent(0,&hp,&hc));    // Check for user actions
  switch(hc)
  {
        case(CONTROL_DONE):             // Its all done
                done=1;break;
        case(CONTROL_RUN):              // User toggled RUN switch
                GetCtrlVal(hpanel,CONTROL_RUN,&csw);
                if(sw==csw) break;
                sw=csw;
                if (sw)
                  StartRun();
                else
                  StopRun();
                break;
        case(CONTROL_TOTALTIME):  // User changed total run-time
```

```
                GetCtrlVal(hpanel,CONTROL_TOTALTIME,&totime);
                nsamp=ceil(totime/5.0)+1;
                break;
        case(CONTROL_LOW):              // User changed low bound of signal
                GetCtrlVal(hpanel,CONTROL_LOW,&low);
                if (high*low<0.0) sinphase=asin((high+low)/(high-low));
                else sinphase=1.5708;
                break;
        case(CONTROL_HIGH):             // user changed high bound
                GetCtrlVal(hpanel,CONTROL_HIGH,&high);
                if (high*low<0.0) sinphase=asin((high+low)/(high-low));
                else sinphase=1.5708;
                break;
        case(CONTROL_FREQ):             // frequency changed
                GetCtrlVal(hpanel,CONTROL_FREQ,&freq);
                period=6.28/(freq?freq:1);
                break;
        case(CONTROL_SIGNAL):           // Type of reference signal changed
                GetCtrlVal(hpanel,CONTROL_SIGNAL,&signal);
                switch(signal)          // Set appropriate signal generator
                {
                case(1): sigfun=squaregen; break;
                case(2): sigfun=triagen;  break;
                case(3): sigfun=sinegen;  break;
                default: break;
                }
                break;
        case(CONTROL_INPUT):            // Show graph for control input
                YGraphPopup(u,n-1,3);
                break;
        case(CONTROL_VELOCITY):         // Plot sampled velocity
                YGraphPopup(x2,n-1,3);
                break;
        case(CONTROL_ERROR):            // plot error variable
                YGraphPopup(error,n-1,3);
                break;
        case(CONTROL_FRICTION):         // plot estimated friction
                YGraphPopup(fric,n-1,3);
                break;
        case(CONTROL_FRICVEL):          // plot friction v/s velocity
                XYGraphPopup(x2,fric,n-1,3,3);
                break;
        case(CONTROL_POSPRINT):         //print the main graph
                OutputGraph(0,"",ConfirmPopup("Resize to fit page
?"),hpanel,CONTROL_POSITION);
                break;
        case(CONTROL_COMPORNOT):                // toggle compensation/no-compensation
                GetCtrlVal(hpanel,CONTROL_COMPORNOT,&compornot);
                break;
        default:
                break;
   } //endswitch(hc)
   if (sw)                      // If the motor is running
    if(time<=totime)            // and time < total time required
        {
          SetCtrlVal(hpanel,CONTROL_TIME,time) ;        // Update runtime box
```

```
        }
    else
     {
          SetCtrlVal(hpanel,CONTROL_RUN,sw=0);        // Reset the run switch !!
          StopRun();                                  // Max. seconds over !! stop
     }

}                                                     // endwhile(!done)
free(u);free(x1);free(x2);                            // release all the allocated memory
free(ref);free(tptr);free(error); free(fric);
return 0;
}

// Function StartRun:  Initialization function before the run begins.
// Its disbales certain controls which are not usable while the
//  apparatus is running. It also initializes control states.
void StartRun()
{
          fp=fopen("data.out","wt");
          time=n=z=zf=prad=padd=0.0;                  // Initialization
          EPSDOT=0.00; PHI= 0.0; PHIDOT=0.0;          // of various
          EPS=0.9747;                                 // parameters
          SLOPE_FUDGE=33.0; K0=700000.0;
          SetCtrlVal(hpanel,CONTROL_RUNLED,1);        // Put on the LED

     // Disable unwanted controls !!
          SetInputMode(hpanel,CONTROL_POSPRINT,0);
          SetInputMode(hpanel,CONTROL_VELOCITY,0);
          SetInputMode(hpanel,CONTROL_FRICTION,0);
          SetInputMode(hpanel,CONTROL_ERROR,0);
          SetInputMode(hpanel,CONTROL_INPUT,0);
          SetInputMode(hpanel,CONTROL_TOTALTIME,0);
          SetInputMode(hpanel,CONTROL_FRICVEL,0);

          EnableISR(timerISR,TIMER,TS);               // Start the timer
}


// Function invoked when the run finishes. It stops the timer, reenables
//  the controls, plots new data and writes new data to the file.
void StopRun()
{
          DisableISR() ;          // Stop the Experiment

          // Reenable the controls
          SetInputMode(hpanel,CONTROL_VELOCITY,1);
          SetInputMode(hpanel,CONTROL_ERROR,1);
          SetInputMode(hpanel,CONTROL_INPUT,1);
          SetInputMode(hpanel,CONTROL_POSPRINT,1);
          SetInputMode(hpanel,CONTROL_FRICTION,1);
          SetInputMode(hpanel,CONTROL_TOTALTIME,1);
          SetInputMode(hpanel,CONTROL_FRICVEL,1);

          SetCtrlVal(hpanel,CONTROL_RUNLED,0);        // Put off the LED
          BinaryWrite(0x0018);                        // reset the encoder count to 0
```

```
                AnalogWrite(0,2048);                    // Reset D/A output to 0 V

                // Clear the main graph and plot the new data
                DeleteGraphPlot(hpanel,CONTROL_POSITION,-1,0);
                PlotXY(hpanel,CONTROL_POSITION,tptr,ref,n-1,3,3,0,0,0,0);
                PlotXY(hpanel,CONTROL_POSITION,tptr,x2,n-1,3,3,0,0,0,0);

                // store the data in the file
                for(i=0;i<=n-1;i++)
                        fprintf(fp,"%f6.3 %f5.2 %f5.2 %f6.2 %f5.2 %f7.4 %f9.4 %f5.2 \n",
                          *(tptr+i),*(ref+i),*(x1+i),*(x2+i),
                          *(u+i),*(error+i),*(fric+i),*(mfric+i));

                fclose(fp);
}


// Function LatchParams:  Function used for latching up the signal generator parameters from the
//  GUI controls to internal variables.
void LatchParams()
{
        GetCtrlVal(hpanel,CONTROL_LOW,&low);        // Get the default signal
        GetCtrlVal(hpanel,CONTROL_HIGH,&high);      //  generator parameters
        GetCtrlVal(hpanel,CONTROL_FREQ,&freq);
        period=6.2832/freq;
        FF=(C*C/(MIU*UTR*L*L*L))*F;                 // Calculation of dimensionless
        MM=(C*C*C*UTR/(MIU*L*L*L*R*R))*M;           // normal force, dimensionless
        RCL=R*C/(L*L);                              // mass, and the ratio RC/L^2
        GetCtrlVal(hpanel,CONTROL_SIGNAL,&signal);
        switch(signal)
        {
          case(1): sigfun=squaregen; break;
          case(2): sigfun=triagen;   break;
          case(3): sigfun=sinegen;   break;
          default: break;
        }
}



/*
 void timerISR()
*** This is the main 'engine' for the control. Its a timer service routine.
        It is invoked every TS seconds when enabled. This routine samples the
        data and performs all the necessary calculations for the controller
        and the observers.
*/
void timerISR()
{
static float tx1,tx2,tref,F,uf,ii=0;            // some internal variables
static float Ubar,d,deps,depsm,rt_deps,j11,j12,j22,k,sinPHI,cosPHI;
static int delta;

tref=*(ref+n)=sigfun();                         // Calculate the reference signal
tx1=getcount()/2387.3;                          // Read counts and convert to radians
```

```
*(mfric+n)=AnalogRead(1);                    // Read the measured friction
                                             // (the # in brackets is the channel input # on
                                             // DACA board)

tx2=tx1-prad;                                // This is a mechanism to detect and correct
if (tx2<-10) {padd+=27.45; tx2+=27.45;}      // the roll-over occuring in
else if (tx2>10) {padd-=27.45; tx2-=27.45; }// the encoder-count
prad=tx1;                                    // (by checking for sudden large
*(x1+n)=(tx1+=padd);                         // change in the value

// Velocity observer
*(x2+n)=tx2=l*tx1+z;

*(error+n)=tx2-tref;                         // error from the reference velocity

// Now the friction estimate - see Simon's thesis to understand the
//  various constants
Ubar=tx2/(UTR/0.0127);                       // Calculation of dimensionless velocity
deps=1.0-EPS*EPS;

rt_deps=(float)sqrt((double)deps);

j11=3.14159/2/(deps*rt_deps);
j12=-2.0*EPS/(deps*deps);
j22=3.14159*(1+2*EPS*EPS)/2/(deps*deps*rt_deps);

k=K0*(EPS-EPS_CR);                           // Force due to asperities

if (EPS < EPS_CR) {delta=0;}
else {delta=1;}

// Calculation of sin(PHI) and cos(PHI) by finite series.
// Calculates faster than internal C function for sin and cos
//  and is accurate enough for this application
sinPHI= PHI - PHI*PHI*PHI/6 + PHI*PHI*PHI*PHI*PHI/120 -
        PHI*PHI*PHI*PHI*PHI*PHI*PHI/5040;
cosPHI= 1 - PHI*PHI/2 + PHI*PHI*PHI*PHI/24 - PHI*PHI*PHI*PHI*PHI*PHI/720;

EPSDOT= (FF*cosPHI - k*(EPS-EPS_CR)*delta + FF*j12*sinPHI/j11)/
          (j22 - j12*j12/j11);
PHIDOT= (0.5*j11*EPS*(Ubar<0?(-Ubar):(Ubar>0?Ubar:0)) -
          j12*EPSDOT - FF*sinPHI)/(j11*EPS);

// Calculation of estimated friction using Harnoy/Friedland model
// with modifications described in Simon's Thesis
*(fric+n)=F=compornot*(FM*k*(EPS-EPS_CR)*delta*(tx2<0?(-1):(tx2>0?1:0)) +
          (SLOPE_FUDGE*RCL*2*3.14159*Ubar)/rt_deps)/FF *0.5*N*0.2245*4*0.547;

// 'Learning' of K0 and SLOPE
// Limits learning to velocities away from zero
if (tx2>0.05 || tx2<0.05)
{
 if ((EPS-EPS_CR) > 0.0)          // Limits learining for K0 to when it can make a difference
   {
   K0+=CK0*(-*(error+n))*(tx2<0?(-1):(tx2>0?1:0));          // CK0 is the learning gain for K0
   }
```

```
 else
   {                                                    // Learning for SLOPE_FUDGE
   SLOPE_FUDGE+=SK0*(-*(error+n))*(tx2<0?(-1):(tx2>0?1:0));  // takes place only when it can
   }                                                    // make a difference.  SK0 is the
}                                                       // learning gain for SLOPE_FUDGE

// Control signal - proportional control and friction compensation
//    saturated at maximum of 10 volts (D/A limit)
*(u+n)=min(max((uf=-1.0*(tx2-tref)+0.295*tref)+F,-10),10);

// Scale the control signal for D/A and send it out.
AnalogWrite(0,(*(u+n))*204.7+2048);

// Velocity observer differential equation (integrated by first order Euler)
zd=-15.0*tx2;
z+=TS*zd;

// Calculation of EPS, PHI -- Integration by 1st order Euler
EPS+=TS*EPSDOT;
PHI+=TS*PHIDOT;

// Update the current time
*(tptr+n)=time;
time+=TS;
if (++ii>=nsamp) {n++;ii=0;}

}

// This function gets the count from the encoder pulse counter
int  getcount()
{
unsigned int lowb,highb;
BinaryWrite(0x0020);
highb=BinaryRead();
BinaryWrite(0x0028);
lowb=BinaryRead();
BinaryWrite(0x0030);
return ((highb&0xff00)+(lowb&0xff00)/256.0);
}

// The following functions generate the desired reference signals

/* Sine Wave genereter */
float  sinegen()
{
return((high+low+(high-low)*sin(freq*time-sinphase))/2);
}


/* Triangle Wave Generator */
float  triagen()
{
 float dtime;
 dtime=time-(floor(time/period)*period);
 if(dtime>period/2)
   {
```

```
        dtime-=period/2;
        return(high-2*(high-low)*dtime/period);
   }
  else
   {
    return(low+2*(high-low)*dtime/period);
   }
}

/* Square Wave generator */
float  squaregen()
{
 float dtime;
 dtime=time-(floor(time/period)*period);
 if(dtime>period/2)
        return (low);
 else
        return (high);
}
```

APPENDIX E

LISTING OF SOURCE CODE FOR POSITION CONTROL PROGRAM

```
/* CONTRLP2.C
   Written by: Jayesh Amin -- Last modified: Nov. 23 1995

   Modified by Simon Cohn (last modification on 12/19/96) to change
   method of friction estimation from Coulomb Friction Observer to
   estimation by Harnoy/Friedland model (simplified to 2nd order) with
   variable K0 and slope

   Source code for velocity control of the friction apparatus.
   Uses DACA board for I/O and requires to be linked to the modified
   version of the DACA library (modified by Jayesh on April 20th -
   available in Dynamic Systems Lab ).
   Uses LabWindows User interface Library for GUI. vcontrol.uir contains
   the LabWindows resources and should be present in the same directory
   as this executable at run-time.
*/


#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <dacamu.h>
// Header file created by LabWindows
#include "control.h"

int hpanel,signal;              // Handle for panel and signal pointer
float low,high,freq,period;     // Parameters for the signal generator
float (*sigfun)(void);          // Pointer for the ref. signal generator

float time=0.0,TS=0.002,totime=20.00;   // Running time, Sampling Period and max
int n=0,i,compornot=1,nsamp=4;          // Sample Number,flag for indicating
                                        //  whether compensating or not.

float z=0,zd,prad=0,padd=0,zf=0,zfd,a;          // Observer states and derivatives
float *u,*x1,*x2,*ref,*error,*tptr,*fric,*mfric;  // Important sampled variable storage
float l=15.0,kf=.01;                            // Velocity and friction observer gains

// Variables important for dynamic friction model
float R=0.0127,C=0.0000508,M=2.2727,MIU=0.002;
float L=0.01905,F=104.0,CK0=12500.0,SK0=0.5,FM=0.26,SLOPE_FUDGE=33.0,N=104.0;
float UTR=0.05,K0=650000.0,EPS_CR=0.9747;
float EPS,EPSDOT,PHI,PHIDOT;
float FF,MM,RCL;

int getcount();     // Returns the current count from the Encoder
float triagen();    // Reference signal generators
float squaregen();
float sinegen();
```

```
float sinphase=1.5708;     // initial phase for sine generator (for smooth start)

FILE *fp;                  // File pointer for storing data

void timerISR();           // Sampling and Control Routine (the main engine!)


int main()
{
int done=0,sw=0,csw;       // some internal variables
int hp,hc;                 // Event Handles
void StartRun();           // Initializes everything at start of run
void StopRun();            // Cleans up the house after the run
void LatchParams();        // Latches critical parameters at start

hpanel=LoadPanel("control.uir",CONTROL);          // GUI Initialization
DisplayPanel(hpanel);
MessagePopup("Simon's Position Control, '96");    // Fancy stuff !

u=(float*)malloc(sizeof(float)*6500);             // Allocate RAM for storage of variables
x1=(float*)malloc(sizeof(float)*6500);
x2=(float*)malloc(sizeof(float)*6500);
ref=(float*)malloc(sizeof(float)*6500);
error=(float*)malloc(sizeof(float)*6500);
fric=(float*)malloc(sizeof(float)*6500);
mfric=(float*)malloc(sizeof(float)*6500);

if ((tptr=(float*)malloc(sizeof(float)*6500))==NULL)
{
 MessagePopup("Memory Allocation Problem - Not Enough memory !!");
 return 1;
}

BinaryWrite(0x0018);       // reset the encoder count to 0
AnalogWrite(0,2048);       // Reset D/A output to 0 V
LatchParams();             // Latch critical parameters

while(!done)               // endless loop till it's all done
{
  if(GetUserEvent(0,&hp,&hc));          // Check for user actions
  switch(hc)
  {
        case(CONTROL_DONE):             // Its all done
                done=1;break;
        case(CONTROL_RUN):              // User toggled RUN switch
                GetCtrlVal(hpanel,CONTROL_RUN,&csw);
                if(sw==csw) break;
                sw=csw;
                if (sw)
                  StartRun();
                else
                  StopRun();
                break;
        case(CONTROL_TOTALTIME):  // User changed total run-time
                GetCtrlVal(hpanel,CONTROL_TOTALTIME,&totime);
```

```
                nsamp=ceil(totime/5.0)+1;
                break;
        case(CONTROL_LOW):              // User changed low bound of signal
                GetCtrlVal(hpanel,CONTROL_LOW,&low);
                if (high*low<0.0) sinphase=asin((high+low)/(high-low));
                else sinphase=1.5708;
                break;
        case(CONTROL_HIGH):             // user changed high bound
                GetCtrlVal(hpanel,CONTROL_HIGH,&high);
                if (high*low<0.0) sinphase=asin((high+low)/(high-low));
                else sinphase=1.5708;
                break;
        case(CONTROL_FREQ):             // frequency changed
                GetCtrlVal(hpanel,CONTROL_FREQ,&freq);
                period=6.28/(freq?freq:1);
                break;
        case(CONTROL_SIGNAL):           // Type of reference signal changed
                GetCtrlVal(hpanel,CONTROL_SIGNAL,&signal);
                switch(signal)          // Set appropriate signal generator
                {
                case(1): sigfun=squaregen; break;
                case(2): sigfun=triagen;  break;
                case(3): sigfun=sinegen;  break;
                default: break;
                }
                break;
        case(CONTROL_INPUT):            // Show graph for control input
                YGraphPopup(u,n-1,3);
                break;
        case(CONTROL_VELOCITY):         // Plot sampled velocity
                YGraphPopup(x2,n-1,3);
                break;
        case(CONTROL_ERROR):            // plot error variable
                YGraphPopup(error,n-1,3);
                break;
        case(CONTROL_FRICTION):         // plot estimated friction
                YGraphPopup(fric,n-1,3);
                break;
        case(CONTROL_FRICVEL):          // plot friction v/s velocity
                XYGraphPopup(x2,fric,n-1,3,3);
                break;
        case(CONTROL_POSPRINT):         //print the main graph
                OutputGraph(0,"",ConfirmPopup("Resize to fit page
?"),hpanel,CONTROL_POSITION);
                break;
        case(CONTROL_COMPORNOT): // toggle compensation/no-compensation
                GetCtrlVal(hpanel,CONTROL_COMPORNOT,&compornot);
                break;
        default:
                break;
   }                    //endswitch(hc)
   if (sw)              // If the motor is running
    if(time<=totime)    // and time < total time required
        {
          SetCtrlVal(hpanel,CONTROL_TIME,time) ;        // Update runtime box
        }
```

```
    else
     {
          SetCtrlVal(hpanel,CONTROL_RUN,sw=0);          // Reset the run switch !!
          StopRun();                                    // Max. seconds over !! stop
     }

}                              // endwhile(!done)
free(u);free(x1);free(x2);   // release all the allocated memory
free(ref);free(tptr);free(error); free(fric);
return 0;
}

// Initialization function before the run begins
// Its disbales certain controls which are not usable while the
//  apparatus is running. It also initializes control states.
void StartRun()
{
                    fp=fopen("data.out","wt");
                    time=n=z=zf=prad=padd=0.0;                  // Initialization of
                    EPSDOT=0.00; PHI= 0.0; PHIDOT=0.0;          //  various parameters
                    EPS=0.9747;
                    SLOPE_FUDGE=33.0; K0=650000.0;
                    SetCtrlVal(hpanel,CONTROL_RUNLED,1);        // Put on the LED

                    // Disable unwanted controls !!
                    SetInputMode(hpanel,CONTROL_POSPRINT,0);
                    SetInputMode(hpanel,CONTROL_VELOCITY,0);
                    SetInputMode(hpanel,CONTROL_FRICTION,0);
                    SetInputMode(hpanel,CONTROL_ERROR,0);
                    SetInputMode(hpanel,CONTROL_INPUT,0);
                    SetInputMode(hpanel,CONTROL_TOTALTIME,0);
                    SetInputMode(hpanel,CONTROL_FRICVEL,0);

                    EnableISR(timerISR,TIMER,TS);               // Start the timer
}



// Function invoked when the run finishes. It stops the timer, reenables
//  the controls, plots new data and writes new data to the file.
void StopRun()
{
                    DisableISR() ;    // Stop the Experiment

                    // Reenable the controls
                    SetInputMode(hpanel,CONTROL_VELOCITY,1);
                    SetInputMode(hpanel,CONTROL_ERROR,1);
                    SetInputMode(hpanel,CONTROL_INPUT,1);
                    SetInputMode(hpanel,CONTROL_POSPRINT,1);
                    SetInputMode(hpanel,CONTROL_FRICTION,1);
                    SetInputMode(hpanel,CONTROL_TOTALTIME,1);
                    SetInputMode(hpanel,CONTROL_FRICVEL,1);

                    SetCtrlVal(hpanel,CONTROL_RUNLED,0); // Put off the LED
                    BinaryWrite(0x0018);      // reset the encoder count to 0
                    AnalogWrite(0,2048);      // Reset D/A output to 0 V
```

```
                // Clear the main graph and plot the new data
                DeleteGraphPlot(hpanel,CONTROL_POSITION,-1,0);
                PlotXY(hpanel,CONTROL_POSITION,tptr,ref,n-1,3,3,0,0,0,0);
                PlotXY(hpanel,CONTROL_POSITION,tptr,x1,n-1,3,3,0,0,0,0);

                // store the data in the file
                for(i=0;i<=n-1;i++)
                        fprintf(fp,"%f6.3 %f5.2 %f5.2 %f6.2 %f5.2 %f7.4 %f9.4 %f5.2 \n",
                                *(tptr+i),*(ref+i),
                                *(x1+i),*(x2+i),*(u+i),*(error+i),
                                *(fric+i),*(mfric+i));

                fclose(fp);
}


// Function used for latching up the signal generator parameters from the
//  GUI controls to internal variables.
void LatchParams()
{
        GetCtrlVal(hpanel,CONTROL_LOW,&low);          // Get the default signal
        GetCtrlVal(hpanel,CONTROL_HIGH,&high);        // generator parameters
        GetCtrlVal(hpanel,CONTROL_FREQ,&freq);
        period=6.2832/freq;
        FF=(C*C/(MIU*UTR*L*L*L))*F;                   // Calculation of dimensionless
        MM=(C*C*C*UTR/(MIU*L*L*L*R*R))*M;             // normal force, dimensionless
        RCL=R*C/(L*L);                                // mass, and the ratio RC/L^2
        GetCtrlVal(hpanel,CONTROL_SIGNAL,&signal);
        switch(signal)
        {
          case(1): sigfun=squaregen; break;
          case(2): sigfun=triagen;   break;
          case(3): sigfun=sinegen;   break;
          default: break;
        }
}



/*
 void timerISR()
*** This is the main 'engine' for the control. Its a timer service routine.
   it is invoked every TS seconds when enabled. This routine samples the
   data and performs all the necessary calculations for the controller
   and the observers.
*/
void timerISR()
{
static float tx1,tx2,tref,F,uf,ii=0;     // some internal variables
static float Ubar,d,deps,depsm,rt_deps,j11,j12,j22,k,sinPHI,cosPHI;
static int delta;

tref=*(ref+n)=sigfun();                  // Calculate the reference signal
tx1=getcount()/2387.3;                   // Read counts and convert to radians
```

```
*(mfric+n)=AnalogRead(1);        // Read the measured friction (the # in brackets
                                 // is the channel input # on DACA board)

tx2=tx1-prad;                                  // This is a mechanism to detect and correct
if (tx2<-10) {padd+=27.45; tx2+=27.45;}        //  the roll-over occuring in
else if (tx2>10) {padd-=27.45; tx2-=27.45; }   //  the encoder-count
prad=tx1;                                      // (by checking for sudden large
*(x1+n)=(tx1+=padd);                           //  change in the value

// Velocity observer
*(x2+n)=tx2=l*tx1+z;

*(error+n)=tx1-tref;          // error from the reference position

// Now the friction estimate from the dynamic friction model - see
// Simon's thesis to understand the various constants

Ubar=tx2/(UTR/0.0127);  // Calculation of dimensionless velocity
deps=1.0-EPS*EPS;

rt_deps=(float)sqrt((double)deps);

j11=3.14159/2/(deps*rt_deps);
j12=-2.0*EPS/(deps*deps);
j22=3.14159*(1+2*EPS*EPS)/2/(deps*deps*rt_deps);

k=K0*(EPS-EPS_CR);     // Force due to asperities

if (EPS < EPS_CR) {delta=0;}
else {delta=1;}

// Calculation of sin(PHI) and cos(PHI) by finite series
// Calculates faster than internal C function for sin and cos
//  and is accurate enough for this application
sinPHI= PHI - PHI*PHI*PHI/6 + PHI*PHI*PHI*PHI*PHI/120 -
PHI*PHI*PHI*PHI*PHI*PHI*PHI/5040;
cosPHI= 1 - PHI*PHI/2 + PHI*PHI*PHI*PHI/24 - PHI*PHI*PHI*PHI*PHI*PHI/720;

EPSDOT= (FF*cosPHI - k*(EPS-EPS_CR)*delta + FF*j12*sinPHI/j11)/(j22 - j12*j12/j11);
PHIDOT= (0.5*j11*EPS*(Ubar<0?(-Ubar):(Ubar>0?Ubar:0)) - j12*EPSDOT - FF*sinPHI)/(j11*EPS);

// Calculation of estimated friction using Harnoy/Friedland model
// with modifications described in Simon's Thesis
*(fric+n)=F=compornot*(FM*k*(EPS-EPS_CR)*delta*(tx2<0?-1:1) +
          (SLOPE_FUDGE*RCL*2.0*3.14159*Ubar)/rt_deps)/
          FF*0.5*N*0.2245*4*0.547;

// 'Learning' of K0 and SLOPE
if (*(error+n)>-0.8 && *(error+n)<0.8)      // This line reduces overshoot during
                                            //  step function reference positions
                                            //  by limiting learning only to when
{                                           //  the error is relatively small
 if ((EPS-EPS_CR) > 0.0)
  {
  K0+=CK0*(min(max((-*(error+n)),-0.30),0.30))*
        (tx2*(*error+n)<=0?1:-1)*((*error+n)<0?1:((*error+n)>0?-1:0));
```

```
  }
 else
  {
  SLOPE_FUDGE+=SK0*(min(max((-*(error+n)),-0.30),0.30))*
                    (tx2*(*error+n)<=0?1:-1)*((*error+n)<0?1:((*error+n)>0?-1:0));
  }
}


// Control signal - proportional control and friction compensation
//  saturated at maximum of 10 volts (D/A limit)
*(u+n)=min(max((uf=-0.43764*(tx1-tref)+0.25164*tx2) +
        F*(tx2*(*error+n)>0?-1:(tx2*(*error+n)<0?1:((*error+n)>0?-1:1)))),-10),10);

// Scale the control signal for D/A and send it out.
AnalogWrite(0,(*(u+n))*204.7+2048);

// Velocity observer differential equation (integrated by first order Euler)
zd=-150.0*tx2 + 457.0*(uf);
z+=TS*zd;

// Calculation of EPS, PHI -- Integration by 1st order Euler
EPS+=TS*EPSDOT;
PHI+=TS*PHIDOT;

// Update the current time
*(tptr+n)=time;
time+=TS;
if (++ii>=nsamp) {n++;ii=0;}

}

// This function gets the count from the encoder pulse counter
int  getcount()
{
unsigned int lowb,highb;
BinaryWrite(0x0020);
highb=BinaryRead();
BinaryWrite(0x0028);
lowb=BinaryRead();
BinaryWrite(0x0030);
return ((highb&0xff00)+(lowb&0xff00)/256.0);
}

// The following functions generate the desired reference signals

/* Sine Wave generetor */
float  sinegen()
{
return((high+low+(high-low)*sin(freq*time-sinphase))/2);
}


/* Triangle Wave Generator */
float  triagen()
{
```

```
 float dtime;
 dtime=time-(floor(time/period)*period);
 if(dtime>period/2)
   {
          dtime-=period/2;
          return(high-2*(high-low)*dtime/period);
   }
  else
   {
     return(low+2*(high-low)*dtime/period);
   }
}

/* Square Wave generator */
float  squaregen()
{
  float dtime;
  dtime=time-(floor(time/period)*period);
  if(dtime>period/2)
          return (low);
  else
          return (high);
}
```

REFERENCES

Amin, J., B. Friedland, and A. Harnoy. 1997. "Implementation of a Friction Estimation and Compensation Technique." *IEEE Control Systems Magazine.* 17 (4): 71-76

Amin, Jayesh N. 1996. "Implementation of a Friction Estimation and Compensation Technique." M.S. Thesis, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ.

Armstrong-Hélouvry, B. 1991. *Control of Machines with Friction.* Boston, MA: Kluwer Academic Press.

Armstrong-Hélouvry, B., and P. Dupont. 1993. "Friction Modeling for Control." *Proceedings of the American Control Conference.* San Francisco, CA, 2: 1905-1919.

Armstrong-Hélouvry, B., P. Dupont, and C. Canudas de Wit. 1994. "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction." *Automatica.* 30 (7): 1083-1138.

Bell, R., and M. Burdekin. 1969. "A Study of the Stick-Slip Motion of Machine Tool Feed Drives." *Proceedings of the Institution of Mechanical Engineers.* Vol. 184, pt. 1, no. 30: 543-560.

Biel, C. 1920. "Die Reibung in Glietlagern bie Zusatz von Voltool zu Mineralol und bie Veranderung der Unlaufzahl und der Temperatur." *Zeitschrift des Vereines Seutscher Ingenieure*, 45: 449-483. In Armstrong-Hélouvry, B., P. Dupont, and C. Canudas de Wit. 1994. "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction." *Automatica.* 30 (7): 1083-1138.

Brandenburg, G., and U. Schäfer. 1991. "Influence and Compensation of Coulomb Friction in Industrial Pointing and Tracking Systems." *Proceedings of the IEEE Industrial Applications Society Annual Meeting.* Dearborn, MI, 1407-1413.

Canudas de Wit, C., and V. Seront. 1990. "Robust Adaptive Friction Compensation." *Proceedings of the IEEE International Conference on Robotics and Automation.* Cincinnati, OH, 1383-1388.

Canudas de Wit, C., et al. 1993. "Dynamic Friction Models and Control Design." *Proceedings of the American Control Conference.* San Francisco, CA, 1920-1926.

Cochin, Ira, and, Harold J. Plass Jr. 1990. *Analysis and Design of Dynamic Systems.* 2nd Ed. New York, NY: Harper Collins Publishers.

Craig, John, J. 1988. *Adaptive Control of Mechanical Manipulators*. New York, NY: Addison-Wesley Publishing Company.

Czichos, H. 1978. *Tribology*. Amsterdam: Elsevier. In Armstrong-Hélouvry, B., P. Dupont, and C. Canudas de Wit. 1994. "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction." *Automatica*. 30 (7): 1083-1138.

Dupont, P. E. 1994. "Avoiding Stick-Slip Through PD Control." *IEEE Transactions on Automatic Control*. 39 (5): 1094-1097.

Dupont, P. E., and E. P. Dunlap. 1993. "Friction Modeling and Control in Boundary Lubrication." *Proceedings of the American Control Conference*. San Francisco, CA, 2: 1910-1914.

Franklin, Gene F., J. David Powell, and Abbas Emami-Naeini. 1994. *Feedback Control of Dynamic Systems*. 3rd Ed. New York, NY: Addison-Wesley Publishing Company.

Friedland, B., and S. Mentzelopoulou. 1993. "On Estimation of Dynamic Friction." *Proceedings of the IEEE Conference on Decision and Control*. San Antonio, TX, 2: 1919-1924.

Friedland, B., and Y. Park. 1992. "On Adaptive Friction Compensation." *IEEE Transactions on Automatic Control*. 37 (10): 1609-1612.

------. 1992. "On Adaptive Friction Compensation Without Velocity Measurements." *Proceedings of the IEEE International Conference on Control Applications*. Brighton, England, 2899-2903.

Fuller, D.D. 1984. *Theory and Practice of Lubrication for Engineers*. New York: John Wiley and Sons. In Armstrong-Hélouvry, B., and P. Dupont. 1993. "Friction Modeling for Control." *Proceedings of the American Control Conference*. San Francisco, CA, 2: 1905-1919.

Haessig Jr., D.A. 1996. Private communication

Haessig Jr., D. A., and Friedland, B. 1991. "On the Modeling and Simulation of Friction." *Journal of Dynamic Systems, Measurement, and Control*. 113 (3): 354-362.

Hamrock, Bernard J. 1984. *Lubrication of Machine Elements*. National Aeronautics and Space Administration Reference Publication 1126.

Hamrock, B.J., and D. Dowson. *Ball Bearing Lubrication - The Elastohydrodynamics of Elliptical Contacts*. New York: Wiley. In Hess, D. P., and A. Soom. 1990.

"Friction at a Lubricated Line Contact Operating at Oscillating Sliding Velocities." *Journal of Tribology*. 112: 147-152.

Harnoy, A. 1995. "Model-Based Investigation of Friction During Start-Up of Hydrodynamic Journal Bearings." *Journal of Tribology*. 117: 1-7.

------. 1996. "Simulation of Stick-Slip Friction in Control Systems." T*ribology Transactions*. 40 (2): 360-366.

Harnoy, A., and B. Friedland. 1993. "Dynamic Friction Model of Lubricated Surfaces for Precise Motion Control." *Proceedings of the STLE/ASME Tribology Conference*. New Orleans, Louisiana, Preprint 93-TC-1D-2.

Harnoy, A., B. Friedland, and H. Rachoor. 1994. "Modeling and Simulation of Elastic and Friction Forces in Lubricated Bearings for Precise Motion Control." *Wear*. 172: 155-164.

Harnoy, A., et al. 1994. "Apparatus for Empirical Determination of Dynamic Friction." *Proceedings of the American Control Conference*. Baltimore, MD, 1: 546-550.

Hess, D. P., and A. Soom. 1990. "Friction at a Lubricated Line Contact Operating at Oscillating Sliding Velocities." *Journal of Tribology*. 112: 147-152.

Maqueira, B., and M. K. Masten. 1993. "Adaptive Friction Compensation for Line-of-Sight Pointing and Stabilization." *Proceedings of the American Control Conference*. San Francisco, CA, 1942-1946.

Mentzelopoulou, S., and B. Friedland. 1994. "Experimental Evaluation of Friction Estimation and Compensation Techniques." *Proceedings of the American Control Conference*. Baltimore, MD, 3: 3132-3136.

Olsson, Henrik. 1996. "Control Systems with Friction." Ph.D. Dissertation, Department of Automatic Control, Lund Institute of Technology, Sweden.

Polycarpou, A., and A. Soom. 1992. "Transitions Between Sticking and Slipping at Lubricated Line Contacts." *Friction-Induced Vibration, Chatter, Squeal, and Chaos. Proceedings of the ASME Winter Annual Meeting*. Anaheim, CA, 49: 139-148.

------. 1995. "A Two-Component Mixed Friction Model for a Lubricated Line Contact." *Proceedings of the Joint ASME/STLE Tribology Conference*. Orlando, FL, 1243-1249.

Rachoor, H., and Harnoy, A. 1996. "Modeling of Dynamic Friction in Lubricated Line Contacts for Precise Motion Control." *STLE Tribology Transactions*. 39 (2): 476-482.

Rachoor, Hanuman. 1996. "Investigation of Dynamic Friction in Lubricated Surfaces." Ph.D. Dissertation, Department of Mechanical Engineering, New Jersey Institute of Technology, Newark, NJ.

Ro, P. I., and P. I. Hubbel. 1993. "Model Reference Adaptive Control of Dual-Mode Micro/Macro Dynamics of Ball Screws for Nanometer Motion." *Journal of Dynamic Systems, Measurement, and Control*, ASME. 115: 103-108.

Schäfer, U., and G. Brandenburg. 1993. "Model Reference Position Control of an Elastic Two-mass System with Compensation of Coulomb Friction." *Proceedings of the American Control Conference*. San Francisco, CA, 1937-1941.

SKF. 1991. *General Catalog*. SKF Group

Stribeck, R. 1902. "Die Wesentlichen Eigenschaften der Gleit und Rollenlager - The Key Qualities of Sliding and Roller Bearings." *Zeitschrift des Vereines Seutscher Ingenieure*, 46(38): 1342-1348; 46(39): 1432-1437. In Armstrong-Hélouvry, B., P. Dupont, and C. Canudas de Wit. 1994. "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction." *Automatica*. 30 (7): 1083-1138.

Tafazoli, S., C. W. de Silva, and P. D. Lawrence. 1995. "Friction Estimation in a Planar Electrohydraulic Manipulator." *Proceedings of the American Control Conference*. Seattle, WA, 3294-3298.

Tung, E. D., M. Tomizuka, and Y. Urushisaki. 1996. "High-Speed End Milling Using a Feedforward Control Architecture." *Transactions of the ASME*. 118: 178-187.

Tung, E. D., Y. Urushisaki, and M. Tomizuka. 1993. "Low Velocity Friction Compensation for Machine Tool Feed Drives." *Proceedings of the American Control Conference*. San Francisco, CA, 1932-1936.

Yao, B., and M. Tomizuka. 1995. "Adaptive Control of Robot Manipulators in Constrained Motion - Controller Design." *Transactions of the ASME*. 117: 320-328.