





This is to certify that the

dissertation entitled

Modelling and Control of Mechanical Systems with Stick-Slip Friction

presented by

Stephen Charles Southward

has been accepted towards fulfillment of the requirements for

Ph.D. degree in Mechanical Engineering

Clark Kadeh Major professor

Date May 4, 1990

MSU is an Affirmative Action/Equal Opportunity Institution

0-12771

DATE DUE	DATE DUE	DATE DUE		
KS also TTT581				
JUN 1 3 1994				
APR 06 1998				
MSU Is An Affirmative Action/Equal Opportunity Institution				

- - ----

.

÷

PLACE IN RETURN BOX to remove this checkout from your record. TO AVOID FINES return on or before date due.

# Modelling and Control of Mechanical Systems with Stick-Slip Friction

Вy

Stephen Charles Southward

## **A DISSERTATION**

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

## **DOCTOR OF PHILOSOPHY**

Department of Mechanical Engineering

## Abstract

## Modelling and Control of Mechanical Systems with Stick-Slip Friction

## By

## **Stephen Charles Southward**

Whenever two mechanical surfaces in a dynamical system are forced together in direct contact, a friction force is induced at the point of contact which acts to resist the relative motion of the surfaces. The result is often a stick-slip friction force that is highly nonlinear, and generally has an undesirable influence on the system dynamics. Its primary function is to remove useful energy from the system. With a representative model of this phenomena, control schemes can be developed and evaluated to compensate for the undesirable effects. In this dissertation, a model is presented which, when incorporated into a dynamic system model, will provide the stick-slip dynamics observed in practice. Using this model, classical control techniques are evaluated for their effectiveness.

A Proportional+Derivative (PD) control law is the simplest classical control law to regulate the position of a one-degree-of-freedom (1-DOF) system. A stable set of multiple equilibrium points is found to exist for the 1-DOF system under PD control, and the steady-state error is guaranteed to be bounded. Integral control (PID) is added in an attempt to remove the steady-state error, but certain types of slipping force models are shown to promote the generation of limit cycles. Neither of these classical techniques are able to effectively regulate the position of the 1-DOF system.

A nonlinear compensation force for stick-slip friction is developed to supplement a PD control law applied to the 1-DOF mechanical system. The choice of a discontinuous compensation force is motivated by the requirement that the desired reference be a unique equilibrium point of the system. The stick-slip friction force, modelled with a sticking force and a slipping force, generates discontinuous state derivatives. A Lyapunov function is introduced to prove global asymptotic stability of the desired reference using a modification of the direct method for discontinuous systems. Stability is verified numerically as well as experimentally. The nonlinear compensation force is robust with respect to the character of the slipping force which is assumed to lie within piecewise linear bounds. Exact knowledge of the static friction force levels is not required, only upper bounds for these levels.

# Copyright

.

Copyright by

Stephen Charles Southward
1990

## Certification

This is to certify that the dissertation entitled

Modelling and Control of Mechanical Systems with Stick-Slip Friction

presented by

Stephen Charles Southward DOCTOR OF PHILOSOPHY Candidate

> has been accepted towards fulfillment of the requirements for a DOCTOR OF PHILOSOPHY DEGREE in Mechanical Engineering at Michigan State University

by

Clark J. Radcliffe Associate Professor, Dissertation Advisor

· NunC

Charles R. MacCluer Professor, Dept. of Mathematics

Ron Rosenberg Professor, Dept. of Mechanical Engineering

Fathi Salam Associate, Professor, Dept. of Electrical Engineering

Charles Seebeck Professor, Dept. of Mathematics

Int

Steven W. Shaw Associate Professor, Dept. of Mechanical Engineering

pril 27,

<u>30, 19</u>90

Hpri) <u>199</u>0 Date

m/30 1990

April 27,

## Dedication

To my parents, Charles and Mildred Southward.

## Acknowledgements

I would like to extend my most sincere gratitude to my academic advisor Dr. Clark J. Radcliffe. He has taught me much more than I had anticipated learning through this experience. Special thanks go to Dr. Chuck MacCluer for his advice and assistance through our brief association. I doubt that I could have finished this dissertation without his guidance and expertise. Both of these men shall remain a source of positive inspiration to me in my future pursuits, academic or otherwise. I am very grateful to claim them as friends and mentors.

Each member of my guidance committee played an integral part in the development of this research effort, and I would like to extend my gratitude to each of them. To Dr. Steven Shaw for his support and guidance with the geometrical analysis. To Dr. Ronald Rosenberg for his assistance with interpretation of the nonlinear simulation results. And finally to Dr. Fathi Salam and Dr. Charles Seebeck for providing me with the fundamental expertise in each of their respective disciplines. I am proud to have been able to work with such an outstanding group of individuals.

Special thanks are extended to Bob Rose for his mechanical design expertise. I would also like to thank the General Motors Corporation for providing funds to equip the Dynamic Systems Laboratory, and the CASE Center for CAEM for providing computational services.

Finally, I would like to thank all my friends at Michigan State, especially Andy (Cap'n DPH Albano) Hull, who has truly shared this experience with me.

vii

## **Table of Contents**

List of Tables	
List of Figures	
Chapter 1 - Introduction	
1.1 The Study of Stick-Slip Friction	1
1.2 The Control of Systems with Stick-Slip Friction	3
1.3 Chapter Overview	5
Chapter 2 - Mathematical System Models	7
2.1 The Physics of Stick-Slip Friction	7
2.2 The Stick-Slip Friction Model	9
2.3 The Reaction Force	12
2.4 Common Slipping Force Models	15
2.5 The One-Degree-Of-Freedom System Model	19
Chapter 3 - PD Control Response	21
3.1 Theoretical Investigation	21
3.2 Experimental Investigation	23
Chapter 4 - PID Control Response	
4.1 Theoretical Investigation	29
4.2 Simulated Time Responses	34
4.3 Limit Cycle Simulations	41
4.4 Experimental Investigation	47
Chapter 5 - Robust Nonlinear Compensation	
5.1 Derivation of the Control Law	52

5.2 Surface of Discontinuity and State Trajectories	55
5.3 Stability Theorem and Proof	57
5.4 Simulation Results	61
5.5 Experimental Results	68
Chapter 6 - Conclusions	72
6.1 Summary	72
6.2 Recommendations for future work	75
Appendix A - Simulation Models	76
A.1 Numerical Simulation	76
A.2 Stick-Slip Friction Model	76
A.3 One-Degree-Of-Freedom Model	78
Appendix B - Experimental Controller	82
B.1 Controller Implementation	82
B.2 PID Controller	82
B.3 PD Control with Nonlinear Stick-Slip Compensation	94
B.4 Common Control Modules	106
References	110

# List of Tables

Table	J	Page
3.1	Static friction bound results from PD control experiment.	26
3.2	Experimental parameters for PD control.	27
4.1	Common parameters for all three PID simulations.	36
4.2	PID Simulation parameters for the Coulomb model.	37
4.3	PID Simulation parameters for the stiction model.	38
4.4	PID Simulation parameters for the exponential model.	40
4.5	Classifications for each of the three stick-slip models.	42
4.6	Common parameters for PID limit cycle simulations.	43
4.7	Numerical results of fitted iteration functions for the Coulomb model.	44
4.8	Numerical results of fitted iteration functions for the stiction model.	45
4.9	Numerical results of fitted iteration functions for the exponential model.	46
4.10	Experimental parameters for PID control.	48
4.11	Slipping force parameters fitted to experimental limit cycle response.	50
5.1	Common parameters for all nonlinear compensation simulations.	62
5.2	Simulation parameters for nonlinear compensation with the Coulomb model.	63
5.3	Simulation parameters for nonlinear compensation with the stiction model.	65
5.4	Simulation parameters for nonlinear compensation with the exponential model.	66
5.5	Experimental parameters for nonlinear compensation.	69

# List of Figures

Figure		Page
2.1	The sticking force model with unequal static friction levels.	11
2.2	Example nonlinear slipping force model bounded by linear bounds.	11
2.3	Coulomb plus viscous friction model.	16
2.4	Stiction plus viscous friction model.	17
2.5	Exponential plus viscous friction model.	18
2.6	The 1-DOF conceptual stick-slip mass system.	19
3.1	Typical PD phase portrait and the set of multiple equilibria $E_{PD}$ .	22
3.2	Experimental 1-DOF system schematic diagram.	24
3.3	Experimental brake to alter the friction characteristics.	24
3.4	Experimental system time response for PD control.	27
4.1	Parallel trajectories inside the sticking band S.	32
4.2	Typical PID trajectory with sticking on both sides of the origin.	34
4.3	PID Control simulated time response with the Coulomb model.	37
4.4	PID Control simulated time response with the stiction model.	39
4.5	PID Control simulated time response with the exponential model.	40
4.6	Iteration functions for PID control with the Coulomb model.	44
4.7	Iteration functions for PID control with the stiction model.	45
4.8	Iteration functions for PID control with the exponential model.	46
4.9	Experimental system time response for PID control.	49
4.10	Simulated limit cycle response fitted to experiment.	51
5.1	The nonlinear proportional feedback control force.	54

5.2	Nonlinear addendum, $g(x)$ , to the Lyapunov function candidate.	58
5.3	Simulated time response for nonlinear compensation with Coulomb model.	64
5.4	Simulated time response for nonlinear compensation with stiction model.	65
5.5	Simulated time response for nonlinear compensation with exponential model.	67
5.6	Experimental time response for nonlinear compensation with braking.	70
5.7	Experimental time response for nonlinear compensation without braking.	71

# Chapter 1

## Introduction

## 1.1 The Study of Stick-Slip Friction

Ever since man began using machines to improve his control over nature, stick-slip friction has been present to hinder and sometimes even help in this process. Stick-slip friction, sometimes known as dry friction, is a natural resistance to relative motion between two contacting bodies. This resistance can be reduced to effect the prevention of surface wear with the use of lubricants, as the Egyptians, Greeks, and Romans were well aware (Davison, 1957). Although Aristotle recognized and wrote of the reality of friction during his time, it was nearly 2000 years before a quantitative study of friction took place (Bowden and Tabor, 1964).

In the mid-fifteenth century, Leonardo da Vinci enunciated the first definitive laws of friction based on direct experiments illustrated in his notebooks. The concept of a coefficient of friction is clearly expressed in his statement, "The friction of a polished smooth surface resists the engine with a force equal to one quarter of its weight" (Vinci, circa 1500c). Although the details were not quite correct, the fundamental concepts behind his laws of friction still hold true today. It is remarkable to note that these laws were developed 200 years before Newton presented a clear definition of force.

The rediscovery of these laws by the French engineer Guillaume Amontons was presented in 1699 to the Royal Academy of Sciences in France, but was met with some skepticism. Amontons, working with dirty surfaces, determined the coefficient of friction to be one third for all surfaces. These observations were verified by Charles Augustine Coulomb in 1781 working with clean surfaces, and his results were also presented to the Royal Academy. Through meticulous experimental work, Coulomb observed that static friction is usually higher than kinetic friction, but for metals, the static friction is almost identical to the kinetic friction. With strong encouragement from the Royal Academy, Coulomb pioneered the research effort to determine the fundamental causes of friction.

Much of the friction research through the nineteenth century was concerned with the mode of energy dissipation in friction. It is now accepted that friction is largely due to interfacial adhesion and to the energy expended in deforming the surfaces (Bowden and Tabor, 1964). Although this research continues to some degree today, friction research has branched off into many new disciplines. The areas of primary interest to this research are modelling stick-slip friction, and the associated effects of these models on the control of dynamic systems.

Many models have been formulated for this type of friction, but only a few are regularly used in practice today. The analytical simplicity of these common models are based on the fundamental research of Coulomb. Through experimental evidence, Coulomb observed that kinetic friction was nearly independent of the relative velocity of the surfaces, thus the coulomb friction model was postulated. For some surfaces, the kinetic friction was found to be lower than the static levels. This model is currently accepted as the stiction model. A linear viscous model, where the friction force is proportional to the velocity, emerged from investigations with various lubricants. Linear combinations of these models are usually able to predict observed system dynamics, and because of their simplicity, they are the most common models found in practice.

Extensive research efforts have been directed toward characterizing the actual friction force in experimental systems. Rabinowicz (1951) investigated the nature of the transition region, from static to kinetic friction, when stationary metal surfaces were set into

motion. He postulated that the transition might be a continuous drop in contrast to the discontinuous transition of the stiction model. Walrath (1984) introduced a first-order dynamic friction model which effectively produced a continuous transition from static to kinetic friction. In order to more accurately represent the stick-slip dynamics observed in practice, Karnopp (1985) introduced a model with explicit simulation considerations. Kolston (1988) utilized electrical circuit analysis equivalences to study a mechanical system with stick-slip friction.

Research efforts have also been directed toward identifying the associated model parameters for an experimental system. Based on the in-phase and quadrature power dissipated when exciting a normal mode, Tomlinson and Hibbert (1979) developed a technique to evaluate the magnitude of the nonlinear friction force and the hysteretic damping constant. Parameter identification has been posed as an optimization problem, and Cheok et al. (1988) employed a modified simplex algorithm to identify the system parameters for servomechanisms with stick-slip friction.

Motivated by the desire to make our machines more efficient and/or more effective, the direct effect of stick-slip friction on dynamical systems has also been studied extensively. Den Hartog (1930) presented the exact solution to the forced response of a one-degree-of-freedom (1-DOF) system using a linear combination of coulomb and viscous friction. Shaw (1986) extended this work to include a stiction model using bifurcation theory to study the nonlinear dynamics. Gogoussis and Donath (1987) examined the fundamental issues associated with stick-slip friction, and their consequences on the design and performance of robots.

#### 1.2 The Control of Systems with Stick-Slip Friction

Feedback control techniques implemented for precise positioning of mechanical systems is very difficult because stick-slip friction is discontinuous when the relative

velocity is zero. In order to use the well-developed design tools from linear control theory, dynamic system models are typically linearized about some operating point. Even with an accurate model, attempts to linearize or neglect stick-slip friction at zero velocity are either impossible or produce invalid predictions of system response. For this reason, regulation of position is much more difficult than tracking a desired trajectory. This observation is implicit in the literature where the regulator problem is investigated almost exclusively.

The effects of nonlinear stick-slip friction on feedback control were examined using a vector graphic technique (Tustin, 1947), which is similar to the modern describing function analysis technique. Tou and Schultheiss (1953) thoroughly studied the impact on control with static and kinetic friction using the describing function technique. More recently, Townsend and Salisbury (1987) investigated the effect of coulomb friction and stiction on force control through a compliant transmission with integral feedback.

The degree of difficulty of the control problem is clearly indicated with the types of problems that are attacked in the literature. The regulation of 1-DOF systems with stickslip friction is studied almost exclusively. This may be due in part to the large number of 1-DOF regulator applications requiring stick-slip friction compensation. It may also reflect the problems associated with the control of multi-DOF systems which have stick-slip friction.

A broad range of compensation techniques have been developed for an even broader range of 1-DOF applications. Gilbart and Winston (1974) developed a model reference adaptive control system to compensate for bearing friction in an optical tracking telescope. Walrath (1984) used a new dynamic friction model as the basis for a predictive, adaptive friction compensator, and demonstrated a significant improvement in the performance of airborne servo mechanisms. Kubo et al. (1986) presented a feedback compensation controller for a robot manipulator which attempts to eliminate the effect of the

static friction force. Canudas et al. (1987) developed an adaptive control scheme to feedback linearize a DC motor system with nonlinear friction. Yang and Tomizuka (1987) developed and verified an adaptive pulse-width control scheme which adaptively updates the width and height of the applied input control force pulse to regulate each of the X-Y table coordinates. Armstrong (1988) utilized experimental results of friction behavior to precompute motion torques for open loop compensation of a brush type DC servo motor with gearing.

In this dissertation, we develop a simple nonlinear compensation technique for the regulation of a 1-DOF system with stick-slip friction. The choice for the compensation force is motivated by the requirement that the desired reference be a unique equilibrium point of the system. This leads to a discontinuous nonlinear proportional feedback force which appears to be a bang-bang force in a region near the desired reference. This compensation force is robust to the character of the slipping friction force which is assumed to lie within piecewise linear bounds. Only upper bounds on the static friction levels are required to obtain stability with this control law. A modified form of Lyapunov's direct method is used to prove global asymptotic stability of the equilibrium point. Stability is illustrated with a numerical simulation, and experimental test results verify the stability of the reference position.

#### **1.3 Chapter Overview**

The physical phenomena of stick-slip motion is examined in Chapter 2. A general mathematical model is presented which produces the desired stick-slip dynamics when incorporated into a dynamic system model. This model is compared with the classical friction models found in practice today. The state equations for the one-degree-of-freedom system are also presented in this chapter.

Chapter 3 investigates the theoretical time response of the 1-DOF system with stickslip friction under PD control. A procedure for estimating the static friction levels is derived from the theoretical analysis of this system. Experimental static friction levels are determined with this procedure, and the theoretical time response is verified with an experimental system.

The response of the 1-DOF system with stick-slip friction under PID control is explored in Chapter 4. A theoretical analysis of the PID time response indicates that trajectories may have a limit cycle behavior. Further numerical analysis verifies the existence of limit cycle behavior for some slipping force models. An experimental time response verifies the predicted theoretical limit cycle response.

A robust nonlinear compensation force is derived in Chapter 5. This compensation supplements a standard PD control law to regulate the position of the 1-DOF system with stick-slip friction. Global asymptotic stability is proven with Lyapunov stability methods. Simulation and experimental time responses verify the global stability of the origin.

Chapter 6 concludes and summarizes this dissertation with recommendations for future work.

## Chapter 2

## Mathematical System Models

... it is necessary to know the nature of the contact which this weight has with the smooth surface where it produces friction by its movement, because different bodies have different kinds of friction; because if there shall be two bodies with different surfaces, that is that one is soft and polished and well greased or soaped, and it is moved upon a smooth surface of a similar kind, it will move much more easily than that which has been made rough by the use of lime or a rasping-file.

> Leonardo da Vinci Forster Manuscript S.K.M. II, 86v. and 87r.

A model is required to represent the physical characteristics of stick-slip friction. In this chapter, the qualitative character of this type of friction is investigated, and a general mathematical model is presented. This mathematical stick-slip friction model, when incorporated into a dynamical system model, will exhibit the same behavior as that observed in the corresponding physical system. Common qualitative slipping force models, found in practice today, and their mathematical representations are reviewed here. The dynamical system equations for a one-degree-of-freedom system with stick-slip friction are also presented in this chapter.

#### 2.1 The Physics of Stick-Slip Friction

When two mechanical surfaces are in contact with each other, a stick-slip friction force is generated at the contacting surface. This stick-slip friction force has two defining characteristics which can effectively be separated into two force components. A slipping, or kinetic friction force dissipates energy when the relative velocity of the surfaces is nonzero. The slipping force thus impedes the relative motion of the surfaces. A sticking, or static friction force acts when the relative velocity is zero, therefore it does not dissipate energy. The sticking force has the ability to keep the relative velocity zero under certain limiting conditions. These two forces do not act simultaneously though they are dependent on each other as will be shown.

The static friction force will take on whatever value is necessary to keep the relative velocity zero, limited by positive and negative extremes. These are the smallest forces necessary to start a relative motion in either direction. Leonardo da Vinci first discovered that these maximum static friction forces generally followed two empirical laws. The first law; "Friction produces double the amount of effort if the weight be doubled" (Vinci, circa 1500a), expresses the proportionality of the static friction force to the normal force between the surfaces (Halliday and Resnick, 1978). The second law is; "The friction made by the same weight will be of equal resistance at the beginning of the movement although the contact may be of different breadths or lengths" (Vinci, circa 1500b). This is a statement of the observation that the static friction forces are approximately independent of the surface area of contact. The static coefficient of friction is defined to be the ratio of the maximum static friction force to the normal force. It is difficult in practice to obtain a globally constant coefficient of friction due to the strong dependence on local surface properties (Bowden and Tabor, 1954).

The slipping force is generally a function of the relative velocity between the surfaces when that velocity is non-zero. As the static friction force bounds are dependent on the normal force between the surfaces, so too is the slipping force. In practice, a kinetic coefficient of friction is determined as a function of velocity. The slipping force is the product of the normal force and the kinetic coefficient of friction. For many dry polished surfaces of dissimilar contacting materials, the kinetic coefficient of friction is approximately constant (Amontons, 1699; Euler, 1750; Coulomb, 1785). Experimental determination of the kinetic coefficient of friction is even more difficult than the static coefficient (Bowden and Tabor, 1954).

The analysis in this dissertation will be restricted to systems where the normal force between the surfaces is constant. With this simplification, the static friction force bounds are constant, and the slipping force can be represented by a function of velocity only. Even with this simplification, the stick-slip friction force remains highly nonlinear. The overall stick-slip friction force can be expressed as a function of relative velocity when that velocity is non-zero, but must be determined by an algebraic equilibrium constraint when zero. The apparent switching of input-output causality on the stick-slip friction force causes a real modelling problem. One set of dynamic equations are required for the slipping motion, and another set for the sticking motion. A model for representing the stick-slip phenomena with constant causality was introduced by Karnopp (1985). With this method, only one set of state equations are required.

#### 2.2 The Stick-Slip Friction Model

The model for the stick-slip friction force  $F_d$ , used throughout this analysis is a modification of the model presented by Karnopp (1985). This model allows for the evaluation of the friction force during sticking and slipping motions. It also eliminates the numerical problems associated with near-zero-velocity force computation. True zero velocity will rarely be numerically attained, therefore an artificial zero region is defined around zero velocity  $[-\alpha,\alpha]$ . The conceptual stick-slip friction model is achieved with equation (2.1) only in the limit as  $\alpha \rightarrow 0$ .

$$F_{d} = F_{slip}(\nu)[\lambda(\nu)] + F_{stick}(F)[1 - \lambda(\nu)]$$
(2.1a)

$$\lambda(\nu) = \begin{cases} 1 & |\nu| > \alpha \\ 0 & |\nu| \le \alpha \end{cases}, \alpha > 0, \tag{2.1b}$$

where v is the relative velocity between the surfaces, and F is a *reaction* force which depends on the particular system configuration. The reaction force is, briefly, that force which is required to keep the relative velocity zero and is the solution of an algebraic constraint which depends on the overall system equations. A thorough description of this reaction force can be found in the following section, and in Karnopp (1985). Throughout the following analytic investigation, the artificial zero parameter  $\alpha$  is taken to be identically zero. Non-zero values of  $\alpha$  are only used in simulations to insure that the numerical integration algorithms remain stable. Any velocity which is numerically within  $\alpha$  of zero is taken to be zero velocity.

The sticking function,  $F_{sick}$ , provides values of the friction force at zero velocity. This term is used to determine whether the contacting surfaces will stick, or break free from the static friction forces. The positive and negative limits on the static friction forces are given by  $F_s^+$  and  $F_s^-$  respectively. As mentioned above, the static friction force bounds are assumed to be constant throughout this analysis, but they are not presumed to be equal in magnitude.

$$F_{stick}(F) = \begin{cases} F_s^+ & F \ge F_s^+ > 0\\ F & F_s^- < F < F_s^+\\ F_s^- & F \le F_s^- < 0 \end{cases}$$
(2.2)

When the relative velocity is zero, v=0, and the reaction force F is between the static friction levels, the damping mechanism provides a force which keeps the surfaces from slipping. The surfaces remain stuck until F is greater in magnitude than the respective static friction force (Figure 2.1).



Figure 2.1. The sticking force model with unequal static friction levels.

The slipping function,  $F_{slip}$ , provides values of the friction force at non-zero velocity, and is represented by two functions of the relative velocity,

$$F_{alio}(v) = F_{d}^{+}(v-\alpha)\mu(v) + F_{d}^{-}(v+\alpha)\mu(-v), \qquad (2.3)$$

where  $\mu(.)$  is the right-continuous Heavyside step function. The function  $F_d^+(.)$  defines the slipping force for positive velocity only, and the function  $F_d^-(.)$  defines the slipping force for negative velocity only. Two separate functions are required for this representation since the slipping force is not assumed symmetric. An example slipping force with  $\alpha=0$  is shown in Figure 2.2.



Figure 2.2. Example nonlinear slipping force model bounded by linear bounds.

Throughout the following analysis,  $F_d^+(.)$  and  $F_d^-(.)$  are assumed to be arbitrary continuous functions which satisfy a Lipschitz condition (Vidyasagar, 1978, pp. 79-80) over their respective domains. The slipping force is assumed to dissipate energy at all non-zero velocities, and therefore is bounded within the first and third quadrants as indicated in Figure 2.2. We assume there exist constants  $b_1 \ge b_0 > 0$ ,  $F_s^- \le F_o^- < 0$ , and  $F_s^+ \ge F_o^+ > 0$ , which define piecewise linear bounds for the slipping functions (Figure 2.2).

$$b_o v \le F_d^+(v) \le F_o^+ + b_1 v \quad \forall v > 0$$
 (2.4a)

$$F_{o}^{-} + b_{1}v \le F_{d}^{-}(v) \le b_{0}v \quad \forall v < 0$$
(2.4b)

As the velocity approaches zero from the right or left, the limiting values of the slipping force do not, in general, exceed the respective static friction force levels,  $F_{o}^{+}$  and  $F_{o}^{-}$ . (Rabinowicz, 1951), thus we have natural lower limits on the magnitudes of  $F_{o}^{+}$  and  $F_{o}^{-}$ . This property will be required in the following analysis, but it should not be taken as a restriction since this property is commonly observed in practice.

### 2.3 The Reaction Force

The reaction force F is the sticking force which is required to keep the relative velocity zero, limited by the static friction bounds. This force is the solution of an algebraic equilibrium relation which depends on the particular system equations. Presented here is an algorithm for computing the reaction force(s) for a general N-DOF mechanical system. The overall mechanical system can contain rotational and/or translational degrees of freedom. This restriction is made since these types of motions have relative surface motions and thus stick-slip friction exists at the surface contact.

A general notation is required to set up the framework for this algorithm. For the N-DOF system, there are N generalized coordinates given by the vector  $\mathbf{q}$ , N associated

generalized velocities given by the vector  $\dot{\mathbf{q}}$ , and N associated stick-slip friction forces given by the vector  $\mathbf{F}_{d}$ . There are also M generalized inputs given by the vector  $\mathbf{u}$ .

$$\mathbf{q} = [q_1, \cdots, q_N]^T \tag{2.5a}$$

$$\dot{\mathbf{q}} = [\dot{q}_1, \cdots, \dot{q}_N]^T \tag{2.5b}$$

$$\mathbf{F}_{\mathbf{d}} = [F_{d1}, \cdots, F_{dN}]^{\prime} \tag{2.5c}$$

$$\mathbf{u} = \left[u_{1}, \cdots, u_{M}\right]^{T} \tag{2.5d}$$

Note that only N stick-slip friction forces are allowed in this system representation. This should not be taken as a restriction since multiple stick-slip friction forces associated with a single degree of freedom can always be combined, or lumped together, to form a single stick-slip friction force.

The individual components of the stick-slip friction vector are expressed similar to the model in equation (2.1). Since this is a multi-degree of freedom representation, the relative velocity between two surfaces associated with the generalized coordinates is expressed as the difference between two generalized velocities.

$$F_{di} = F_{slip}^{(i)}(v_{jk}^{(i)})[\lambda(v_{jk}^{(i)})] + F_{stick}^{(i)}(F_i)[1 - \lambda(v_{jk}^{(i)})]$$
(2.6a)

where the relative velocities are

$$v_{jk}^{(i)} = (\dot{q}_j - \dot{q}_k)_i \quad j,k \in \{0,\dots,N\}$$
(2.6b)

The relative velocity associated with the i<sup>th</sup> generalized coordinate is the difference between the j<sup>th</sup> and k<sup>th</sup> generalized velocities. Since it is possible for the slipping velocity to be relative to ground, or zero velocity, the 0 index is used to represent this situation (i.e.  $\dot{q}_0 \equiv 0$ ). The particular indices (j,k) associated with the i<sup>th</sup> generalized coordinate are determined from the system model. Associated with each component of the stick-slip friction force then is a slipping force which is a function of a particular relative velocity, and a sticking force which is a function of its associated reaction force. These reaction forces only need to be computed when the associated relative velocity becomes zero. With the following general state equations,

$$\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}_{\mathbf{d}}, \mathbf{u}) \tag{2.7}$$

the algorithm for the computation of the reaction forces can explicitly be stated. The first step is to compute the N relative velocities associated with each of the stick-slip friction forces. For every relative velocity that is non-zero, evaluation of the associated stick-slip friction force is explicit. The slipping force can be evaluated at this velocity to generate the proper friction force.

$$v_{jk}^{(i)} \neq 0 \quad \Rightarrow \quad F_{di} = F_{slip}^{(i)}(v_{jk}^{(i)}) \tag{2.8}$$

When the  $i^{th}$  relative velocity is zero, the sticking force will take on values to keep that velocity zero, limited by the associated static friction bounds for that generalized coordinate. Initially, we will assume that the sticking force is not limited, therefore from equation (2.2)

$$v_{jk}^{(i)} = 0 \implies F_{di} = F_i \tag{2.9}$$

In order to keep the relative velocity zero, there must be no relative acceleration, thus

$$(\ddot{q}_j - \ddot{q}_k)_i = 0 \implies f_j(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F_d}, \mathbf{u}) = f_k(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F_d}, \mathbf{u})$$
 (2.10)

The resulting algebraic equation (2.10) has the unknown reaction force  $F_i$  appearing explicitly in the stick-slip friction force vector. For every relative velocity which is zero, there will exist an associated algebraic equation. There will always be an equivalent number of equations and unknowns which can be solved simultaneously for the reaction forces.

Repeating the above process, the unlimited reaction forces are obtained through the simultaneous solution of the algebraic equilibrium constraints, and finally they must be limited by their respective sticking force functions.

$$F_{di} = F_{rick}^{(i)}(F_i) \tag{2.11}$$

Under the premise that the sticking forces are not limited, when a particular relative velocity becomes zero, it must remain zero for all time thereafter. The physical limitations on the sticking forces indicate that the relative velocity may not remain zero.

#### 2.4 Common Slipping Force Models

Many slipping force models are available for use in dynamic simulations or analysis (Kolston, 1988; Craig, 1988; Canudas, et al., 1987; Gogoussis and Donath, 1987; Shaw, 1986). Most of the standard slipping force models found in the literature can be classified into three categories. In each of these categories, a linear viscous damping term is included since it is the most common damping model, and the simplest model for analysis.

The first category is a Coulomb plus viscous damping model. The slipping force approaches the respective static friction bound in a linear fashion from the left or right (Figure 2.3).



Figure 2.3. Coulomb plus viscous friction model.

The slipping force equations for this model are

$$F_{d}^{+}(v) = F_{s}^{+} + b^{+}v \tag{2.12a}$$

$$F_{d}^{-}(v) = F_{s}^{-} + b^{-}v,$$
 (2.12b)

where  $b^+>0$  and  $b^->0$  are the viscous damping coefficients for positive and negative velocity respectively. The slipping force can thus have different viscous damping characteristics for positive and negative velocity.

The next category is a stiction plus viscous damping model. The slipping force approaches a kinetic friction level which is at a lower level than the respective static friction bound (Figure 2.4). For many materials the kinetic friction level is approximately half of the static friction level (Miller, 1977).



Figure 2.4. Stiction plus viscous friction model.

The slipping force equations for this model are

$$F_d^+(v) = F_k^+ + b^+ v \tag{2.13a}$$

$$F_d^-(v) = F_k^- + b^- v,$$
 (2.13b)

where  $b^+>0$  and  $b^->0$  are the viscous damping coefficients for positive and negative velocity respectively, and  $F_s^+>F_k^+>0$ , and  $F_s^-<F_k^-<0$ .

The last category is an exponential model with viscous damping. This slipping friction force exponentially decays from the static level to a kinetic friction force with increasing velocity (Tustin, 1947; Armstrong, 1988). After a certain velocity, the viscous term dominates and the slipping force increases linearly with velocity (Figure 2.5).



Figure 2.5. Exponential plus viscous friction model.

The slipping force equations for this model are

$$F_{d}^{+}(v) = F_{s}^{+} + (F_{k}^{+} - F_{s}^{+}) \left[ 1 - e^{-(v/v_{s}^{+})} \right] + b^{+}v$$
(2.14a)

$$F_{d}^{-}(v) = F_{s}^{-} + (F_{k}^{-} - F_{s}^{-}) \left[ 1 - e^{-(v/v_{s}^{-})} \right] + b^{-}v, \qquad (2.14b)$$

where  $b^+>0$  and  $b^->0$  are the viscous damping coefficients for positive and negative velocity respectively, and  $F_s^+>F_k^+>0$ , and  $F_s^-<F_k^-<0$ . The two velocity constants,  $v_o^+$  and  $v_o^-$ , are the characteristic velocities where 63% of the drop from the static to the kinetic force level occurs. Though the limiting values as velocity approaches zero in (2.14) are the static friction bounds, a more general exponential model can be obtained by lowering these limits below the static levels.

Any pair of contacting surfaces will have a slipping force relation which may be approximately represented by one of the three models presented above. In a purely analytic investigation, the primary motivation for the choice of slipping force model is simplicity. If an experimental system is to be modelled, a complex model may be required for a more accurate response prediction. It is also reasonable to expect that some physical mechanical systems will exhibit a slipping force characteristic which can be represented by the simplest slipping force model.

#### 2.5 The One-Degree-Of-Freedom System Model

The 1-DOF system under investigation is a mass constrained to move in one dimension along a horizontal ground surface. Displacement and velocity of the mass are measured relative to the ground. Stick-slip friction is present between the mass and the supporting surface (Figure 2.6).



Figure 2.6. The 1-DOF conceptual stick-slip mass system.

The two first-order state equations for this system are given by

$$\dot{x} = v \tag{2.15a}$$

$$\dot{\nu} = (\mathcal{Y}_m)(F - F_d). \tag{2.15b}$$

These equations can also be used to represent a rotational 1-DOF system with the proper change of units. Linear displacements and velocities must be replaced with rotational displacements and velocities, forces must be replaced with torques, and the mass must be replaced with a rotational inertia. However, if the stick-slip friction model presented in section 2.2 above is to be used for a rotational 1-DOF system, the assumption of a constant normal force between the contacting surfaces must remain valid.

The differential equations of motion (2.15) are linear except for the stick-slip friction force,  $F_d$ . The stick-slip friction force will be completely defined when a particular slipping force model is chosen and an expression for the reaction force is obtained from the system equations. In this example, the relative velocity is just the absolute velocity of the mass (with respect to ground). In order to keep the mass at rest when the velocity is zero, the acceleration must remain zero. Setting the acceleration term in (2.15b) to zero yields the

algebraic relation whereby the reaction force is clearly equal to the input control force F. As long as the applied force F remains between the static friction bounds, the mass will remain stuck. When the applied force is greater than the respective static friction level, there will be a non-zero acceleration, and the mass will begin to slip.

.

## Chapter 3

## **PD** Control Response

The very rapid friction of two thick bodies produces fire. Leonardo da Vinci Codex Atlanticus

We would like to find a simple control law to regulate the position of the mass under the influence of stick-slip friction. Classical control theory offers several attractive possibilities. The simplest classical control technique is a Proportional+Derivative (PD) law, where the control force is proportional to the position error and also to its derivative. A substantial amount of useful information can be obtained through the study of this control law applied to the mass system. A bounded steady-state error is found to exist for the 1-DOF system with a PD control law. The theoretical 1-DOF system response to a PD control law is analyzed and the results are experimentally verified in this chapter.

#### 3.1 Theoretical Investigation

A PD control law is simple and easy to implement. When applied to the 1-DOF mass system (Figure 2.6) with a type of stick-slip friction modelled by (2.1-2.4), state trajectories end up at or near the desired reference with a bounded steady-state error (Hahn, 1967; Shaw, 1986). The steady-state error bounds can be explicitly determined by solving for the equilibrium points of the system. Without loss of generality, the origin of the phase space is taken as the desired reference position. The PD control force then simplifies to

$$F = -K_{p}x - K_{d}v, \qquad (3.1)$$

where  $K_p > 0$  is the proportional gain, and  $K_d > 0$  is the derivative gain. Solving the 1-DOF system equations (2.15) with (3.1) for the equilibrium points
$$\dot{x} = 0 \implies v = 0$$

$$\dot{v} = 0 \implies [-K_{p}x - K_{d}v - F_{slip}\lambda(v) - F_{slick}(1 - \lambda(v))] = 0$$
(3.2)

$$\Rightarrow -K_{p}x - F_{stick} = 0. \tag{3.3}$$

Note that since the velocity is zero at equilibrium,  $F_{stick}$  will take on any value of the applied force limited by the static friction levels. In the equilibrium constraint (3.3), the applied force is just  $-K_{p}x$ . There are two maximum displacements from the origin,  $x_{L}$  and  $x_{H}$ , whereby the proportional control forces are equal to the respective static friction forces. These maximum displacements are the steady-state error bounds, and are given by

$$x_L = -\left(\frac{F_s^+}{K_\rho}\right), \quad x_H = -\left(\frac{F_s^-}{K_\rho}\right). \tag{3.4}$$

From (3.2-3.4), we see that there exists a set of multiple equilibrium points which can be written as  $E_{PD} = \{(x,v)|v = 0, x_L \le x \le x_H\}$ . The existence of this set of multiple equilibrium points,  $E_{PD}$ , (Figure 3.1) is undesirable.



Figure 3.1. Typical PD phase portrait and the set of multiple equilibria  $E_{PD}$ .

The PD-space equilibrium set is an invariant set and Lyapunov's direct method for stability of invariant sets can be employed to verify the stability of  $E_{PD}$  (Hahn, 1963, and

1967). If the PD control gains are chosen to stabilize the undamped system, then with the proper choice of a Lyapunov function, it can be shown that the equilibrium set  $E_{PD}$  is globally asymptotically stable. Every trajectory will terminate with zero velocity and some bounded steady-state position error. More importantly, no limit cycles can exist in the PD control space for damping functions of the form presented in Chapter 2 with the given assumptions.

There are only two trajectories in the PD space that take the mass position directly to the origin. Unless initial conditions are specified on one of these two trajectories, there will be some non-zero bounded steady-state error. The steady-state error bounds (3.4) can be reduced by increasing the proportional gain  $K_p$ , but at the expense of large control forces for states far from the origin. For any finite proportional gain, there will exist a non-zero steady-state error. This suggests that in order to get zero steady-state error, the proportional gain must be infinite at the origin, or the control force may need to be discontinuous.

#### 3.2 Experimental Investigation

An experimental 1-DOF system was available to validate the predicted theoretical system responses. A cart, constrained to move in one dimension along a track, was cable driven by a DC servo-motor. A control voltage to the DC servomotor was transformed to a force which could push the cart in either direction along a one meter track. A rotational potentiometer, rigidly coupled to the drive-train, admits electrical measurement of cart positions. The general experimental system schematic is shown in Figure 3.2.



Figure 3.2. Experimental 1-DOF system schematic diagram.

The friction present in the 1-DOF experimental system was actually distributed throughout the various components in Figure 3.2. For example, bearing friction existed in the servomotor, cart wheels, position sensor, and drive cable shafts. Because it is assumed to be a 1-DOF system, the various friction components could be lumped into one global stick-slip friction force. In order to increase the number of types of friction possible for investigation, a friction brake was installed. With this brake, the overall friction force could be altered to provide a range of static friction levels, and also vary the slipping force characteristics.



Figure 3.3. Experimental brake to alter the friction characteristics.

A full-scale drawing of the experimental friction brake is shown in Figure 3.3. The spring loaded brake pressed down on the drive shaft of the cable drive system. By changing the compression in the springs and using different brake materials, different desirable or undesirable friction characteristics could be obtained.

All control algorithms investigated here were implemented digitally using 12-bit A/D and D/A converters to interface with the position sensor and DC motor respectively. The static calibration constants for the actuator and sensor were determined to be 0.75936 N/V, and 19.895 V/m respectively (Southward, 1986). Based on these calibration constants and the digital hardware configuration, the resolution of position measurements was 0.000245 m, and the resolution of force output was 0.0037 N. These resolutions allow ranges of approximately  $\pm 0.5$  m. for position, and  $\pm 7.6$  N. for force. The velocity of the cart, which is needed to compute the derivative control force, was estimated with a finite difference algorithm using position data. First order digital filters were available for both the position data and the estimated velocity data. The controller algorithm was implemented on a DEC LSI-11/23 computer in Fortran IV. Sampling rates up to 200 Hz. were possible with the floating-point implementation (Appendix B).

Equation (3.4) suggests a method of obtaining estimates for the upper bounds on the static friction forces. Assuming the static friction bounds are constant (i.e. independent of position), the PD control law can be applied to the experimental system to determine the sticking limits  $x_L$  and  $x_H$  for a given proportional gain  $K_p$ . Theoretically, the cart must stick at all points between the sticking limits. These limits were experimentally determined by starting the PD control law with the mass inside the sticking zone, and manually pushing it toward either bound. Beyond the sticking limit, the proportional force is greater than the static friction level, and the mass will be forced back into the sticking zone. This test was performed on the experimental 1-DOF system with PD control and the friction brake applied to the drive shaft. Over the indicated range of proportional gains, with the derivative gain set equal to zero, the static friction bounds were determined from the measured sticking limits (Table 3.1).

$K_{p}(N/m)$	$F_s^+(\mathbf{N})$	$F_{s}^{-}(\mathbf{N})$
8	1.768	-1.483
10	1.791	-1.479
12	1.705	-1.431
14	1.761	-1.372
16	1.750	-1.449
20	1.765	-1.442
25	1.710	-1.416
30	1.713	-1.435
35	1.776	-1.468
40	1.705	-1.420

Table 3.1. Static friction bound results from PD control experiment.

As seen from the above results, the static friction bounds appear to vary slightly with position. The mean values and respective standard deviations from the above data are:  $F_s^+ = 1.744 \pm 0.033$  N. and  $F_s^- = -1.439 \pm 0.033$  N. From the relatively small standard deviations, it is reasonable to assume that the static friction bounds are not strongly position dependent, thus we are justified in assuming it to be constant for the 'braked' system.

The existence of the sticking zone can now be verified with an experimental time response. Using the estimates for the static friction forces, the sticking zone limits are estimated for a given proportional gain. These computed sticking zone bounds are shown in Table 3.2 with all the pertinent control parameters for the experimental setup.

Mass	m	1.375 kg.
PD Control	K <sub>p</sub>	20.0 N/m.
Gains	K <sub>d</sub>	0.5 Ns/m.
Static Friction	$F_s^+$	1.744 N.
Bounds	$F_s^-$	-1.439 N.
Sticking	x <sub>L</sub>	-0.0872 m.
Limits	x <sub>H</sub>	0.0719 m.
Initial	$x(t_o)$	0.3 m.
Conditions	$v(t_o)$	0.0 m/s.

Table 3.2. Experimental parameters for PD control.

As indicated above, the mass is initially at rest outside of the sticking zone. The time responses of the mass position and the applied control force are shown in Figure 3.4, thus demonstrating that a non-zero equilibrium can exist for PD control.



Figure 3.4. Experimental system time response for PD control.

Voltages from the sensor and actuator were sampled at 2000 Hz. per channel and 1000 samples were taken from each to generate the experimental time response. From Figure

3.4, the mass gets inside the sticking zone but has enough momentum to carry it a bit further (see Figure 3.1). After one second the mass is stuck at a non-zero position, and a non-zero control force is still being applied. The control force is approximately 1.0 N (Figure 3.4), which is less than the estimated static friction level therefore we would expect the mass to remain stuck.

,

# Chapter 4

## **PID Control Response**

Define to me why one who slides on the ice does not fall. Leonardo da Vinci Codex Atlanticus

The classical method for removing steady-state error from a system is to include integral action with the PD control. Thus a Proportional+Integral+Derivative (PID) control force is proportional to the position error, its derivative, and its integral. Though the PID control law is not the simplest, it is a logical choice for attempting to remove the undesirable steady-state error observed with the PD law. The effects of PID control on the 1-DOF system response are investigated in this chapter. A theoretical analysis demonstrates that state trajectories appear to have a limit cycle character. Further investigation with numerical simulations provides evidence that limit cycles can exist under PID control. A property of slipping force models is shown to promote limit cycle generation. This limit cycle behavior is experimentally verified. Based on the time responses of the 1-DOF system under PID control for the three classes of slipping force models presented in Chapter 2, a qualitative characterization of the slipping force present in the experimental system is determined.

### 4.1 Theoretical Investigation

The inclusion of an integral force term to the control law introduces an added dimension to the state space. To simplify the following PID analysis, the static friction bounds are assumed to be constant and equal in magnitude, therefore  $F_s^+ = -F_s^- \equiv F_s$ . The stick-slip friction force is also assumed to be symmetric for analytical simplicity. The general system equation for PID control is

$$m\ddot{x} + F_{d} = K_{p}e + K_{d}\dot{e} + K_{i}\int_{t_{o}}^{t} e\,dt,$$
(4.1)

where the actuating error,  $e = x_d - x$ . Again, for the regulator problem, and without loss of generality the desired states are chosen to be at the origin (i.e.  $x_d(t) = 0$  and  $v_d(t) = 0$  $\forall t \ge t_o$ ). This system is also linear with the exception of the stick-slip friction force  $F_d$ . The system equations can be written in state-space canonical form for  $x_d(t) = 0$  as

$$\begin{bmatrix} \dot{y} \\ \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\left(\frac{K_i}{m}\right) & -\left(\frac{K_p}{m}\right) & -\left(\frac{K_d}{m}\right) \end{bmatrix} \begin{bmatrix} y \\ x \\ v \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \left(\frac{F_d}{m}\right) \end{bmatrix}$$
(4.2)

with the identities  $v = \dot{x}$  and  $y = -\int_{t_0}^{t} e dt$ .

The stick-slip friction force is the same as that presented in Chapter 2, (2.1-2.4), except for the evaluation of the reaction force. For the PID controller, the reaction force is again equal to the control force which is now given by

$$F = -K_{d}v - K_{p}x - K_{i}y.$$
(4.3)

This PID control force is used in the evaluation of the sticking force (2.2).

The PID control solution space for (4.2) is three dimensional with state coordinates (y,x,v). There are several subspaces in this phase space which have interesting features. The first subspace of interest is the equilibrium set, which is the set of all points in the space where the derivatives are zero. The solution of (4.2) with zero derivatives yields

$$\dot{y} = 0 \implies x = 0$$
 (4.4)

$$\dot{x} = 0 \implies v = 0 \tag{4.5}$$

$$\dot{v} = 0 \implies [-K_i y - K_p x - K_d v - F_{slip} \lambda(v) - F_{stick} (1 - \lambda(v))] = 0$$
  
$$\implies -K_i y - F_{stick} = 0.$$
(4.6)

From the definition of the sticking force (2.2), the system is in equilibrium whenever  $|K_i y| \le F_s$ . Two new constants,  $y_L$  and  $y_H$ , can be defined similar to those found in Chapter 3 for the steady-state error bounds on displacements. The set of equilibrium points for the PID phase space can then be written as:  $E_{PID} = \{(y, x, v) | x = 0, v = 0, y_L \le y \le y_H\}$ , where

$$y_L = -\left(\frac{F_s}{K_i}\right), \quad y_H = \left(\frac{F_s}{K_i}\right)$$
(4.7)

Notice the difference between this set and the set of equilibrium points for the PD control space,  $E_{PD}$ . Even though it is not a single point set, any trajectories which end up in  $E_{PID}$  reach both the desired states  $x_d = 0$  and  $v_d = 0$ . Unlike the stable PD equilibrium set. this equilibrium set is unstable in the sense that a small perturbation from an equilibrium point may result in convergence to a new equilibrium or divergence from the set entirely. As will be shown below, for certain slipping force models, trajectories near the set  $E_{PID}$  diverge away from the set  $E_{PID}$  into a stable limit cycle.

There exists a subspace where the system dynamics are reduced to first order independent of the type of slipping force. For trajectories to exist in the x-y plane, the velocity must be zero, v = 0, and there must be zero acceleration. From (4.2) and the definition of the stick-slip friction force, zero acceleration is possible whenever  $|K_{y} + K_{p}x| \le F_{s}$ . Using the previous definitions for the coordinate bounds (3.4) and (4.7), one can write the following equivalent expressions

$$\dot{v} = 0 \quad \Leftrightarrow \quad \begin{cases} y_L \le (y + (K_p / K_i)x) \le y_H \\ x_L \le (x + (K_i / K_p)y) \le x_H \end{cases}$$
(4.8)

These inequalities define a band in the x-y plane shown in Figure 4.1. This "sticking" band can be defined in a way that is similar to the definition of the equilibrium set as the union of two sets;  $S = S_1 \cup S_2$ , where  $S_1 = \{(y, v, x) | y_L \le (y + (K_p / K_i)x) \le y_H, v = 0, x > 0\}$ , and  $S_2 = \{(y, v, x) | y_L \le (y + (K_p / K_i)x) \le y_H, v = 0, x < 0\}$ . Inside this band, the system

dynamics (4.2) are reduced to first order in the state y. Trajectories in S represent "sticking" trajectories where the mass remains stationary until the force due to the integral of the error builds up to a level greater than the maximum static friction force. Given an initial condition  $(y_{\omega}, x_{\omega}, v_{\omega}) \in S$  at time  $t_{\omega}$ , state solutions are given by

$$\begin{bmatrix} y_{s}(t) \\ x_{s}(t) \\ v_{s}(t) \end{bmatrix} = \begin{bmatrix} x_{so}(t-t_{so}) + y_{so} \\ x_{so} \\ 0 \end{bmatrix} \quad t_{so} \le t \le t_{sf}$$

The mass will break free at time  $t_{sf}$  given by

$$t_{sf} = t_{so} + (1 / x_{so}) \begin{cases} y_H - (y_{so} + (K_p / K_i)x_{so}) & x_{so} > 0\\ y_L - (y_{so} + (K_p / K_i)x_{so}) & x_{so} < 0 \end{cases}$$

Trajectories in S are parallel to the y-axis, and their direction is dependent on the sign of x, (Figure 4.1). These trajectories show the equilibrium,  $E_{ptD}$ , to be unstable for small perturbations in x and y.



Figure 4.1. Parallel trajectories inside the sticking band S.

The remainder of the three-dimensional PID control space is divided up into two regions N and P, characterized by the sign of the velocity component of state trajectories inside each region. These regions are:  $P = P_1 \cup P_2$ , where  $P_1 = \{(y,v,x) | y_L \ge (y + (K_p / K_i)x), v = 0\}$ , and  $P_2 = \{(y,v,x) | v > 0\}$ . The other half of the space is similarly defined as:  $N = N_1 \cup N_2$ , where  $N_1 = \{(y,v,x) | y_H \le (y + (K_p / K_i)x), v = 0\}$ , and  $N_2 = \{(y,v,x) | v < 0\}$ . From (2.2) and (4.3), we have:  $(y,x,v) \in N_1 \Rightarrow (dv/dt) \le 0$  and  $(y,x,v) \in P_1 \Rightarrow (dv/dt) \ge 0$ . All initial conditions  $(y_{po}, v_{po}, x_{po}) \in P$  at time  $t_{po}$  have trajectories with non-negative velocity component until time  $t_{pf}$ , and all initial conditions  $(y_{no}, v_{no}, x_{no}) \in N$  at time  $t_{no}$  have trajectories with non-positive velocity is zero, i.e. in  $P: v_p(t_{pf}) = 0$  and  $v_p(t) \ne 0$   $\forall t \in (t_{po}, t_{pf})$ , and in  $N: v_n(t_{nf}) = 0$  and  $v_n(t) \ne 0$ 

This sticking band S plays an important role in the possible generation of limit cycles in the PID space. The x-coordinate of a point in S physically represents the displacement of the mass away from the origin (the desired position) where the mass remains stuck until the integral force term breaks it free. Near the origin, all trajectories in  $S_1$  enter N and then enter  $S_2$ . Also near the origin, trajectories in  $S_2$  enter P and similarly reenter  $S_1$ . Physically, the mass is stuck on one side of the origin until the force due to the integral term builds up enough to break free from the maximum static friction force. It then overshoots the origin and sticks on the other side. A similar process is repeated on the return motion. A typical trajectory is shown in Figure 4.2.



Figure 4.2. Typical PID trajectory with sticking on both sides of the origin.

Since the general stick-slip friction force is nonlinear, and only piecewise linear for a few special cases, an explicit solution is not possible. It is not possible to determine analytically whether a limit cycle exists for a general slipping force function.

#### 4.2 Simulated Time Responses

Numerical simulations of the time responses of the 1-DOF system under PID control with the various slipping force models introduced in Chapter 2 provides a means of establishing the existence or non-existence of limit cycles. Using the stick-slip friction model with the three common classes of slipping force models, the simulated time responses are evaluated for comparison.

All simulations were performed in double precision using the DIFFEQ program (Southward, 1989) on a Digital Equipment Corporation Micro-VAX computer (Appendix A). Several integration algorithms were utilized to verify that each would yield the same prediction. The algorithms tested were an Euler method, a fourth order fixed step size Runge-Kutta method, and a fourth order Adams variable step size predictor-corrector method. The final results were produced from a Hamming predictor-corrector general (HPCG) method, with an initial step size of 0.001 s.

The differential equations of motion from Chapter 2 (2.15) are expressed in continuous time and were integrated as such. However, in a digital controller implementation, the sensor measurements and the actuator output are quantized into discrete levels. In order to obtain a more realistic simulation, quantized states (simulating the measurement process) are used to compute the control force, which is in turn quantized (simulating the output process). Signals from the position and velocity sensors are assumed to be quantized by 12-bit A/D and D/A converters. Based on the experimental parameters from Chapter 3, it is reasonable to assume resolutions of 0.00025 m. for positions, and 0.005 N. for the force. These resolutions will allow ranges of  $\pm 0.5$  m., and  $\pm 10.0$  N. It is also reasonable to assume a resolution of 0.01 m/s. for velocities, providing a range of  $\pm 20.0$  m/s.

Using the above assumptions, the following parameters were chosen to approximately model an experimental system (Table 4.1). The proportional control gain is chosen so as not to saturate the actuator when the mass is at the extreme measurable limits of position. The static friction force bounds were based on the experimental estimates from Chapter 3. The sticking zone bounds are determined from these static friction levels and the proportional gain (3.4). The derivative and integral gain were chosen somewhat arbitrarily. The mass is assumed to be initially at rest outside of the sticking zone.

Mass	m	1.0 kg.
PID	K <sub>p</sub>	20.0 N/m.
Control	K <sub>d</sub>	0.5 Ns/m.
Gains	K <sub>i</sub>	40.0 N/ms.
Static Friction	$F_s^+$	1.744 N.
Bounds	$F_s^-$	-1.439 N.
Sticking	x <sub>L</sub>	-0.0872 m.
Limits	x <sub>H</sub>	0.0719 m.
Initial	$x(t_o)$	0.15 m.
Conditions	$v(t_o)$	0.0 m/s.

Table 4.1. Common parameters for all three PID simulations.

A single pair of equations can be used to define each of the three classes of slipping force models. These slipping force functions are represented as

$$F_d^+(v) = F_o^+ - \Delta F^+[1 - e^{-(v/v_o^+)}] + b^+ v$$
(4.9a)

$$F_{d}^{-}(v) = F_{o}^{-} - \Delta F^{-}[1 - e^{-(v/v_{o}^{-})}] + b^{-}v, \qquad (4.9b)$$

where  $F_o^+$  and  $F_o^-$  are the limiting values of the slipping force at zero velocity,  $\Delta F^+$  and  $\Delta F^-$  are the respective drops from this level to the kinetic force level, and  $v_o^+$  and  $v_o^-$  are velocity constants defining the characteristic velocity at which 63% of the drop occurs. Where applicable, numerical values for the drop to kinetic friction are based on a kinetic coefficient of friction of 0.5, which roughly corresponds to a steel mass sliding on a wooden surface (Miller, 1977).

The first PID time response uses the Coulomb plus viscous friction slipping force model (2.12). As seen from the parameters in Table 4.2, the slipping force linearly approaches the static friction bounds as velocity approaches zero from the left or right respectively. The artificial zero parameter is naturally chosen to be the assumed resolution of velocity measurement due to sensor quantization.

	$F_o^+$	1.744 N.
$F_d^+(v)$	$\Delta F^+$	0.0 N.
	$v_o^+$	NA
	<i>b</i> <sup>+</sup>	0.5 Ns/m.
	F_o	-1.439 N.
$F_d^-(v)$	∆F⁻	0.0 N.
	vo	NA
	b <sup>-</sup>	0.5 Ns/m.
Artificial Zero	α	0.01 m/s

 Table 4.2. PID Simulation parameters for the Coulomb model.

The simulated time response for PID control with the Coulomb plus viscous slipping force model is shown in Figure 4.3. The time scale is extended to 120 seconds since the mass is stuck most of the time. While stuck, the integral force term builds up until it reaches the respective static friction bound. At this point, the mass begins to move and gets stuck on the other side of the origin, as predicted by the theoretical analysis.



Figure 4.3. PID Control simulated time response with the Coulomb model.

From the displacement response and shape of the force response, the mass position appears to converge to the origin, though it looks like it will take a long time. As the error becomes smaller, the force due to the integral of the error requires more time to build up to a static friction level. From Figure 4.3, the time between successive "jumps" of the mass increases with each jump. This result indicates that PID control can be used to effectively regulate the position of the mass with a Coulomb plus viscous friction model, and will be investigated with additional detail in Section 4.3.

The second PID time response uses the stiction plus viscous slipping force model. Similar to the previous slipping force model, and as indicated in (2.13), this model linearly approaches kinetic friction levels as the velocity approaches zero from the left or right respectively. The kinetic friction levels are arbitrarily chosen for the simulation to be half of the static friction levels. The artificial zero parameter is again chosen to be the assumed resolution of velocity measurement due to sensor quantization.

	$F_o^+$	0.872 N.
$F_d^+(v)$	$\Delta F^+$	0.0 N.
	$v_o^+$	NA
	<b>b</b> <sup>+</sup>	0.5 Ns/m.
	F_o	-0.719 N.
$F_d^-(v)$	$\Delta F^{-}$	0.0 N.
	v_	NA
	b <sup>-</sup>	0.5 Ns/m.
Artificial Zero	α	0.01 m/s

Table 4.3. PID Simulation parameters for the stiction model.

The simulated time response for PID control with the stiction plus viscous friction model is shown in Figure 4.4. The time scale is one tenth that of the previous plot (Figure 4.3), since the mass is no longer stuck most of the time. The mass still gets stuck twice per period of the apparent oscillation.



Figure 4.4. PID Control simulated time response with the stiction model.

The overall friction force and the applied force is plotted in Figure 4.4. When the mass is stuck, the friction force is identical to the applied force until the static friction bound is reached. At this point, the mass begins to move and the slipping force drops to the kinetic level. The applied force overshoots the static friction bound thus increasing the acceleration of the mass (2.15). Unlike the previous case, the position of the mass appears to settle into a stable limit cycle. This result indicates that PID control can not be used to effectively regulate the position of the mass with a stiction plus viscous friction model, and will also be further investigated in Section 4.3.

The final PID time response uses an exponential plus viscous slipping force model (2.14). This model exponentially approaches the kinetic friction levels from the static level as the velocity increases away from zero from the left or right respectively. As in the

stiction model, the kinetic friction forces are chosen to be half the level of the respective static friction forces.

	$F_o^+$	1.744 N.
$F_d^+(v)$	$\Delta F^+$	0.872 N.
	<i>v</i> <sub>o</sub> <sup>+</sup>	0.1 m/s.
	b <sup>+</sup>	0.5 Ns/m.
	F_	-1.439 N.
$F_d^-(v)$	· \$\$F^-	-0.719 N.
	v_	-0.1 m/s.
	<i>b</i> <sup>-</sup>	0.5 Ns/m.
Artificial Zero	α	0.01 m/s

 Table 4.4. PID Simulation parameters for the exponential model.

The simulated time response for PID control with the exponential plus viscous friction model is shown in Figure 4.5. As in the previous case, the mass appears to oscillate in a stable limit cycle motion and also gets stuck twice per period.



Figure 4.5. PID Control simulated time response with the exponential model.

The overall friction force is again plotted with the applied force on this plot. Though the applied force overshoots the static friction bound when the mass breaks free, the friction force does not drop as rapidly as in the previous case, thus the resulting acceleration of the mass is not as large. These results also indicate that PID control can not be used to regulate the position of the mass with friction modelled by exponential plus viscous friction.

#### 4.3 Limit Cycle Simulations

Existence and stability properties of a limit cycle for an explicit slipping force function can be determined through the construction and examination of mappings of trajectories from points in  $S_1$  to  $S_2$  and then back to  $S_1$  (Figure 4.2). In fact, existence and stability of limit cycle trajectories can be determined from mappings of the x-coordinate of a trajectory in  $S_1$  into the subsequent x-coordinate in  $S_1$ . Simulation data of x-coordinates in  $S_1$  mapped back into  $S_1$  can be used to numerically construct a continuous iteration function for a given set of system parameters. Limit cycles will appear as fixed points of these iteration map functions, where the fixed point is the x amplitude of the cycle. The contraction mapping theorem can then be applied to prove the existence of fixed points (Vidyasagar, 1978, pp. 73-78).

The equations for the stick-slip friction force (2.1-2.4), along with the three classes of models for  $F_{slip}(v)$  (2.12-2.14) completely define the friction force. Motivated by the time response simulations above it is useful to classify symmetric slipping force models with the following property (4.10).

$$U = \left\{ F_{slip}(\nu) \middle| \exists \nu^* > 0 \ni F_{slip}(\nu) < F_s \forall \nu \in (0, \nu^*) \right\}$$
(4.10)

If a slipping force model belongs to the set U, it will take on values which are less than the static friction level in a region near zero velocity. Note that since the slipping force is assumed symmetric, we only need to look at positive velocities. It will be demonstrated

that models belonging to the set U will promote limit cycles in the 1-DOF mass system under PID control. Using the models defined in Chapter 2, the following classifications are obtained (Table 4.5).

Coulomb Model plus Viscous Damping $F_{slip}(v) \notin U$ Stiction Model plus Viscous Damping $F_{slip}(v) \in U$ Exponential Model plus Viscous damping $F_{slip}(v) \in U$ 

Table 4.5. Classifications for each of the three stick-slip models.

The slipping friction forces from the Coulomb model are always greater than or equal to the static friction levels therefore it is not in the set U. The stiction model has a discontinuous drop from the static level to a non-decreasing slipping force as the velocity increases and the exponential model undergoes a continuous drop from the static level to a lower kinetic level then increases, therefore both belong to the set U.

Using the any of the three slipping force models, the system equations (4.2) are nonlinear. They are only piecewise linear for the Coulomb and stiction models. Since no general closed form solution exists for all three classes, a numerical simulation was performed to determine the contraction mappings. Three representative examples from each of the slipping force model classes were numerically simulated. All cases used the following set of common parameters (Table 4.6).

Mass	m	1.0 kg.
PID	K <sub>p</sub>	10.0 N/m.
Control	K <sub>d</sub>	10.0 Ns/m.
Gains	K <sub>i</sub>	100.0 N/ms.
Static and Kinetic	F,	4.0 N.
Friction Bounds	F <sub>k</sub>	2.0 N.
Artificial Zero	α	0.001 m/s.

Table 4.6. Common parameters for PID limit cycle simulations.

Simulations were performed in double-precision on a Digital Equipment Corporation Micro-VAX, using a Hamming predictor-corrector general integration method with an initial step size of 0.001 s. For each run, a number of initial conditions inside  $S_1$ were chosen to provide an evenly distributed data set for generating the iteration functions. Each  $x_i$  is an x-coordinate in  $S_1$  and  $x_{i+1}$  is the x-coordinate for the next time the trajectory enters  $S_1$ . A second order polynomial of the form:  $x_{i+1} = (a_0 + a_1x_i + a_2x_i^2)$ , was fit through each set, and then plotted. There is one iteration function fit associated with each set of parameters.

The numerical fixed points were obtained by guessing an initial value and then iterating the fitted function until the error between successive iterations was less than 1.0E-9. Convergence is guaranteed since the slopes of the iteration function over the fitted domain are less than unity. A conservative error estimate,  $\delta$ , for each fixed point is obtained by using the largest absolute residual error from the curve fit to form a band around the fitted function. All data from the respective curve fit is therefore guaranteed to be inside this band. The fixed points of the two functions defining the edges of this band are then used to determine conservative error estimates in the original fixed point. These error estimates are presented with each of the fixed point estimates.

For the Coulomb plus viscous friction models, the three representative cases were obtained by varying the viscous damping coefficient, b (Table 4.7).

b	$a_0(\times 10^{-4})$	<i>a</i> <sub>1</sub>	$a_2(\times 10^{-3})$	$(x \pm \delta)(\times 10^{-3})$
2.0	1.87	0.814	6.37	$1.01 \pm 1.51$
1.0	2.05	0.853	7.39	$1.40 \pm 2.10$
0.0	1.99	0.906	7.00	$2.12 \pm 4.65$





Figure 4.6. Iteration functions for PID control with the Coulomb model.

The simulation results for this class are plotted in Figure 4.6. Shown on this plot is the line:  $x_{i+1} = x_i$ , along with the fitted iteration functions for the three test cases. A fixed point occurs wherever an iteration function crosses the  $x_{i+1} = x_i$  line. All three test cases (Figure 4.6) have a fixed point at the origin, i.e.  $x_{i+1} = x_i \Leftrightarrow x_i = 0$ . The magnitudes of the intercepts  $(a_0)$  of the polynomials from the curve fit, as seen from Table 4.7, indicate that no limit cycles exist for this class of models. Within the accuracy of the simulation, all trajectories eventually converge to the origin (Table 4.7), and no limit cycle exists for the Coulomb friction model. For the stiction plus viscous friction models, the three representative test cases were similarly obtained by varying the viscous damping coefficient, b (Table 4.8).

b	$a_0$	$a_1$	$a_2$	x±δ
2.0	0.0452	0.635	0.236	$0.136 \pm 0.003$
1.0	0.0489	0.663	0.246	$0.165 \pm 0.004$
0.0	0.0565	0.669	0.334	$0.219 \pm 0.006$
0.5				
0.5	T		·····	
	-			$x_{i+1} = x_i$
0.4	_			/
¥				<i>i</i>
A <sub>i+1</sub>	-		/	
0.3	4			
	_			
0.2				
0.2				
	-			
0.1	<b>!</b>	<i>,</i>		<b>→</b> <i>b</i> =0.0
				b=1.0
	1 /			×-× <i>b</i> =2.0
0.0	· <del>K</del> r		·····	
	0.0 0.	1 0.2	$x_i = 0.3$	0.4 0.5

Table 4.8. Numerical results of fitted iteration functions for the stiction model.

Figure 4.7. Iteration functions for PID control with the stiction model.

The simulation results for the stiction plus viscous friction models are plotted in Figure 4.7. Each of these three test cases have distinct non-zero fixed points which are presented in Table 4.8 for each of the cases. From Figure 4.7, and as verified with the polynomial curve fit, the local slopes of these iteration functions near the fixed points are always less than unity. The contraction mapping theorem can be used to prove that these fixed points are all stable (Vidyasagar, 1978, pp. 73-78), thus stable limit cycles exist for the stiction plus viscous friction model. These limit cycles are reduced in magnitude with increasing viscous friction. For the exponential plus viscous friction models, the three representative cases were obtained by varying both the viscous damping coefficient b, and the velocity constant  $v_o$  (Table 4.9).

b	vo	<i>a</i> <sub>0</sub>	<i>a</i> <sub>1</sub>	<i>a</i> <sub>2</sub>	$x \pm \delta$	
0.0	0.05	0.0628	0.732	0.190	$0.297 \pm 0.008$	
0.0	0.10	0.0638	0.767	0.138	$0.343 \pm 0.004$	
1.0	0.10	0.0553	0.735	0.120	$0.233 \pm 0.002$	
0	5					
U	.5				$\mathbf{r} = \mathbf{r}$	
	-				x <sub>i+1</sub> - x <sub>i</sub>	
0	.4-					
<b>x</b> .				//		
	F1				-	
0	.3-					
	4					
•						
0	.2-		<i>1</i> /1			
	-					
0	1_			$\sim b=$	=10 v = 01	
0	.1	/		с-е <i>b</i> =	$=0.0, v_{a}=0.1$	
	-	/		×-× b=	=0.0, v <sub>o</sub> =0.05	
Λ						
U	0.0	0.1	$0.2 x_{i}$	0.3	0.4 0.5	

Table 4.9. Numerical results of fitted iteration functions for the exponential model.

Figure 4.8. Iteration functions for PID control with the exponential model.

The exponential plus viscous friction model results (Figure 4.8) also indicate the existence and stability of limit cycles but at larger amplitudes than those for stiction (Table 4.9). The local slopes near the corresponding fixed points are less than unity, so these fixed points are also stable, thus stable limit cycles will occur for the exponential friction model.

The Coulomb iteration function results indicate that as viscous damping is decreased, the slope of the iteration function increases and the fixed point remains near

zero. Similarly, the stiction iteration function results indicate that as the viscous damping is decreased, the fixed point of the iteration function increases. This observation is important since fast convergence to the fixed point (stable limit cycle) is obtained when the slope of the iteration function near a fixed point is close to zero. Since the cases investigated here all have slopes near unity, the convergence is slow. Raising the iteration functions further away from the x-axis (i.e. increasing  $a_0$ ) has the effect of increasing the amplitude of the limit cycle. The presence of viscous damping can therefore help stabilize by either decreasing convergence times as in the case of Coulomb models, or by reducing the limit cycle amplitude as in the case of stiction and exponential models.

#### 4.4 Experimental Investigation

Using the experimental 1-DOF mass system described in Chapter 3, the theoretical existence of limit cycle behavior was validated. As observed above, slipping force functions which do not have a definite drop from the static level will not admit a limit cycle response. A suitable braking material was found which produced a limit cycle behavior in the experimental system. This same brake was used in all the experimental investigations studied in this dissertation which required a brake.

The original PD control algorithm with integral control added (Appendix B), was implemented on the same DEC LSI-11/23 computer, and ran at 200 Hz. For comparison with the PD response, all control parameters remained the same for the PID implementation except for the addition of integral gain. The pertinent control parameters for the experimental PID investigation are presented in Table 4.10.

Mass	m	1.375 kg.
PID	K <sub>p</sub>	20.0 N/m.
Control	K <sub>d</sub>	0.5 Ns/m.
Gains	K <sub>i</sub>	40.0 N/ms.
Static Friction	$F_s^+$	1.744 N.
Bounds	$F_{s}^{-}$	-1.439 N.
Sticking	x <sub>L</sub>	-0.0872 m.
Limits	x <sub>H</sub>	0.0719 m.
Initial	$x(t_o)$	0.3 m.
Conditions	$v(t_o)$	0.0 m/s.

Table 4.10. Experimental parameters for PID control.

The experimental system started up with the initial conditions given in Table 4.10. Under PID control, the mass moved toward the origin, and ended in a stable limit cycle. Due to the excessive amount of time required to reach this steady limit cycle, the time response for mass position and control force were not sampled from the initial startup conditions listed above. Voltage data was acquired from two channels with the use of An Apple Macintosh IIx computer and National Instruments LabView hardware and software at 1920 samples per channel and 8 Hz. per channel. The data was then scaled and plotted (Figure 4.9).



Figure 4.9. Experimental system time response for PID control.

Several interesting observations can be made about the experimental time response of Figure 4.9. When the mass breaks free, the control force appears to be at the same level of the static friction forces which were estimated from the PD control experiment (Chapter 3). This agrees with the simulated predictions, and indicates that the estimates for static friction bounds are reasonable. As the static friction bounds are at different levels in the experiment, so too are the slipping forces. This can be seen by the difference in the amount of time the mass sticks on each side of the origin. For positive displacement error, the mass sticks for a longer time than for negative displacement error. The simulations were performed with symmetric slipping force functions except for different offsets, therefore the amount of time spent on either side of the origin was roughly equivalent.

Meticulous observation of the experimental PID time response will prove to yield qualitative if not quantitative information about the character of the friction present. In this example (Figure 4.9), the static friction bounds can accurately be estimated, and we know that the slipping force is not of the Coulomb type. The slipping force must drop from the

49

static level to a lower kinetic level, and from the small amplitude of the limit cycle (approximately 1 cm.), we know this drop must be very small. Although this qualitative information does not provide a direct method for obtaining explicit parameters, it is still useful. Trial and error parameter tuning was used to produce a simulation in an attempt to match the above experimental result. The following slipping force parameters were found to produce a reasonable match (Table 4.11).

	$F_o^+$	1.69 N.
$F_d^+(v)$	$\Delta F^+$	0.0 N.
	$v_o^+$	NA
	<i>b</i> <sup>+</sup>	20.0 Ns/m.
	F_	-1.39 N.
$F_d^-(v)$	$\Delta F^{-}$	0.0 N.
	v_	NA
	b <sup>-</sup>	3.0 Ns/m.
Artificial Zero	α	0.01 m/s

Table 4.11. Slipping force parameters fitted to experimental limit cycle response.



Figure 4.10. Simulated limit cycle response fitted to experiment.

Initial conditions were chosen to match those of the experimental data. The simulated time response does not match up exactly with the experimental results, but the same character of the response is certainly present. The simple observations made above regarding the character of the slipping force model can be utilized to guide an engineer as to which models should be used. It is certainly important to use models which will actually admit the responses observed in an experimental situation.

## Chapter 5

## **Robust Nonlinear Compensation**

Putting two solids together is rather like turning Switzerland upside down and standing it on Austria – the area of intimate contact will be small. F.P. Bowden, B.B.C. Broadcast 1950.

A simple but robust control law is still needed for the regulation of the 1-DOF system. PID control can only be used for some types of slipping force models, and therefore is not robust to small variations in stick-slip behavior. With PD control, there exists an undesirable set of multiple equilibrium points. This set can be reduced to a single point set by allowing the proportional gain to approach infinity, but this is not possible in practice. A discontinuous nonlinear compensation force is developed to supplement the PD control law. The resulting robust control law is proven to stabilize the 1-DOF system for any slipping force model satisfying the assumptions made in Chapter 2. This result is verified through numerical and experimental time responses.

### 5.1 Derivation of the Control Law

The existence of a set of multiple equilibrium points,  $E_{PD}$ , (Figure 3.1) is undesirable, but the origin can be made into a unique equilibrium point of the system with the inclusion of a nonlinear compensation force  $F_c(x)$  in the PD control law

$$F = -K_{\mu}x - K_{d}v - F_{c}(x), \tag{5.1}$$

where  $K_p > 0$  and  $K_d > 0$ . Solving the 1-DOF system equations (2.15) with the new control law (5.1) for the equilibrium points

$$\dot{x} = 0 \implies v = 0$$

$$\dot{v} = 0 \implies [-K_{p}x - K_{d}v - F_{c}(x) - F_{slip}\lambda(v) - F_{slick}(1 - \lambda(v))] = 0$$

$$\implies -K_{p}x - F_{c}(x) - F_{stick} = 0.$$
(5.2)
(5.2)
(5.2)
(5.2)

In order to have a unique equilibrium at the origin, we must choose a compensation force such that (5.3) is satisfied for x=0 only. Choose the nonlinear compensation force to have the following form

$$F_c = K_p x_c. \tag{5.4}$$

Using the previous sticking limit definitions (3.4) with this compensation force and the equilibrium constraint (5.3), an equivalent requirement for uniqueness of the equilibrium is obtained by choosing an  $x_c(x)$  such that the inequality

$$x_L \le (x + x_C) \le x_H,\tag{5.5}$$

is satisfied for x=0 only. This can be achieved with the following nonlinear function  $x_c(x)$ 

$$x_{C} = \begin{cases} 0 & x > \tilde{x}_{H} \\ (\tilde{x}_{H} - x) & 0 < x \le \tilde{x}_{H} \\ 0 & x = 0 \\ (\tilde{x}_{L} - x) & \tilde{x}_{L} \le x < 0 \\ 0 & x < \tilde{x}_{L} \end{cases}$$
(5.6)

where

$$\tilde{x}_{H} = x_{H} + \varepsilon 
\tilde{x}_{L} = x_{L} - \varepsilon , \varepsilon > 0.$$
(5.7)

The addition of the nonlinear compensation force, defined by (5.4), (5.6), and (5.7), to the PD control law insures that the origin is a unique equilibrium point. The compensation force (5.4) is only active when the mass is between the extended sticking

limits  $\tilde{x}_L \leq x \leq \tilde{x}_H$ . The combination of the compensation force with the proportional force,  $K_p(x + x_c)$ , can be viewed as a nonlinear proportional feedback control force (Figure 5.1). To simplify the implementation of this control force, the following two parameters are defined

$$\tilde{F}_s^+ = F_s^+ + K_p \varepsilon \tag{5.8a}$$
$$\tilde{F}^- = F^- - K \varepsilon. \tag{5.8b}$$



Figure 5.1. The nonlinear proportional feedback control force.

When the mass is between the original sticking limits  $x_L \le x \le x_H$ , any positive value of  $\varepsilon$  is enough to theoretically guarantee that the feedback force will always exceed the static friction force levels. This result coupled with equation (2.15) imply that there must be a non-zero acceleration of the mass in the original PD sticking zone, thus the mass continues moving. In this region, the nonlinear proportional feedback force is essentially a bang-bang force.

#### 5.2 Surface of Discontinuity and State Trajectories

The right-hand side of the system of differential equations (2.15) is discontinuous. The slipping force,  $F_{slip}$ , is discontinuous at v=0, and the applied feedback control force F, given by equations (5.1), (5.4), (5.6), and (5.7) is discontinuous at x=0 because of the nonlinear compensation term. The set of points in the state space where the right hand side of (2.15) is discontinuous is called a *surface of discontinuity* similar to what is found in the theory of variable structure systems (Utkin, 1977). The surface of discontinuity S, can be represented by

$$S = \{(x,v) \in \Re^2 | s(x,v) = 0\},$$
(5.9a)

where, for this 1-DOF system

$$s(x,v) \equiv xv. \tag{5.9b}$$

This surface divides the phase space into two open regions,  $R^+ = \{(x,v) \in \Re^2 | s > 0\}$  and  $R^- = \{(x,v) \in \Re^2 | s < 0\}$  (Siljak, 1969). Inside these regions, the right-hand side of (2.15) satisfies a local Lipschitz condition, therefore we are guaranteed existence, uniqueness, and continuous dependence on initial conditions for solutions there (Vidyasagar, 1978, pp. 78-88). Trajectories are well defined and continuous until they hit the surface of discontinuity S where state derivatives are discontinuous. The notion of a "solution" of (2.15) must be generalized for trajectories which have discontinuous derivatives (Filippov, 1964).

This 1-DOF system (2.15) and (2.1-2.4) has discontinuities of the first kind, or simple discontinuities. This means that there exists finite limits for the state derivatives as  $s \rightarrow 0^+$  and  $s \rightarrow 0^-$  which do not coincide (Barbashin, 1970). It is easy to check from the limiting values of the state derivatives that at every point of S, the vector gradients of state

trajectories are directed with the same sense through the surface. Trajectories are never tangent to S and thus will always pass from one open region  $R^+(R^-)$  through S into  $R^-(R^+)$ . Trajectories are also never directed toward points in S from both open regions, therefore there cannot exist a sliding mode as found in the theory of variable structure systems. Over the entire phase space, trajectories are absolutely continuous.

The qualitative behavior of trajectories in the phase space can be determined from the differential equations (2.15) using the bounds on the slipping force (2.4) and the nonlinear compensation force (5.4).

*Lemma*: Every non-trivial trajectory will exit the quadrant it occupies in a clockwise fashion. The time spent in any one quadrant is bounded above by a constant determined by the initial condition.

*Proof:* Consider solutions of the differential equations in each of the four quadrants of the phase space. In the first quadrant, x>0, v>0, and from (2.15), (2.4), (5.1), (5.6), Figure 2.2, and Figure 5.1

$$m\dot{v} + K_{d}v = -K_{p}(x + x_{c}) - F_{d}^{+}(v) \le \tilde{F}_{s}^{-} - b_{0}v, \qquad (5.10a)$$

and thus

$$m\dot{v} + (K_d + b_0)v \le \tilde{F}_s^- < 0.$$
 (5.10b)

From (5.10b) we know that v(t) is bounded above by a strictly decreasing function which approaches a negative value

$$v(t) \leq \left\{ \left[ v_o - (\tilde{F}_s^-) / (K_d + b_0) \right] e^{-(K_d + b_0)t/m} + (\tilde{F}_s^-) / (K_d + b_0) \right\},$$
(5.11)

where  $v_o$  is the initial velocity in the quadrant. Trajectories must leave the first quadrant in a clockwise fashion before the right-hand side of (5.11) becomes zero. In the second quadrant, x<0, v>0, and again from (2.15), (2.4), (5.1), (5.6), Figure 2.2, and Figure 5.1

$$m\dot{v} + K_{d}v = -K_{p}(x + x_{c}) - F_{d}^{+}(v) \ge \tilde{F}_{s}^{+} - F_{o}^{+} - b_{1}v, \qquad (5.12a)$$

and thus

$$m\dot{v} + (K_d + b_1)v \ge (\tilde{F}_s^+ - F_o^+) \ge (\tilde{F}_s^+ - F_s^+) = K_p \varepsilon > 0.$$
(5.12b)

Similarly, from (5.12b) we know that v(t) is bounded below by a monotonic function which approaches a positive value

$$v(t) \ge \left\{ \left[ v_o - (K_p \varepsilon) / (K_d + b_1) \right] e^{-(K_d + b_1)t/m} + (K_p \varepsilon) / (K_d + b_1) \right\}.$$
(5.13)

Trajectories must also leave the second quadrant in a clockwise fashion. Proof of this lemma in quadrants III and IV proceeds as in quadrants I and II respectively. *Q.E.D.* 

#### 5.3 Stability Theorem and Proof

Now that a control force has been determined such that the 1-DOF system (2.15) has a single equilibrium point at the origin of the state space, we need to show that it is stable.

Theorem: The origin of the state space for the system given by equations (2.15) and 2.1-2.4), with the control force given by equations (5.1), (5.4), (5.6), and (5.7), is a globally asymptotically stable equilibrium point.

*Proof:* Choose the following Lyapunov function candidate, which closely resembles the actual energy function of the system
$$V(x,v) = \frac{1}{2}K_{x}x^{2} + \frac{1}{2}mv^{2} + g(x), \qquad (5.14)$$

where



Figure 5.2. Nonlinear addendum, g(x), to the Lyapunov function candidate.

It is easy to check that this Lyapunov function candidate is positive definite, decrescent, and satisfies a global Lipschitz condition. This function is also continuous with respect to x and v.

We must now evaluate the time derivative of the Lyapunov function candidate,  $\dot{V}(x,v)$ , along solution trajectories of (2.15). The time derivative of V is the dot product of the gradient of V with the vector of state derivatives. As mentioned above, the state derivatives are discontinuous at all points in S, and from (5.15) the gradient of V is discontinuous at x=0. The derivative  $\dot{V}(x,v)$  is discontinuous or does not exist for all points of S, however it exists and is well-defined for all points in the open regions  $R^+$  and  $R^-$ . A modification of Lyapunov's direct method must be used for non-differentiable Lyapunov function candidates, such as the one presented above (Solncev, 1951; Hahn, 1963). The global asymptotic stability theorem of Lyapunov for continuous systems is modified by replacing the derivative  $\dot{V}(.)$  with the Dini-derivate  $D^*V(.)$ , where the \* represents any of the four possible Dini-derivates (Rouche, et al., 1977). At any point where the derivative  $\dot{V}(.)$  exists, all four Dini-derivates will have a common value equal to the derivative at that point (McShane, 1947).

$$D^{\bullet}V(.) = V(.) \quad \forall (x,v) \notin S.$$
 (5.16)

Along solution trajectories, excluding points in S, we can compute the time derivative of the Lyapunov function candidate as

$$\dot{V}(x,v) = K_{p}xv + v(F - F_{d}) + g'(x)v$$
  
=  $K_{p}xv + v(-K_{p}x - K_{d}v - K_{p}x_{c} - F_{slip}\lambda(v) - F_{stick}(1 - \lambda(v))) + g'(x)v$   
=  $-K_{d}v^{2} - vF_{slip}(v) + v(g'(x) - K_{p}x_{c}),$  (5.17)

where

$$g'(x) = \begin{cases} 0 & x > \tilde{x}_{H} \\ K_{p}(\tilde{x}_{H} - x) & 0 < x \le \tilde{x}_{H} \\ \text{undefined} & x = 0 \\ K_{p}(\tilde{x}_{L} - x) & \tilde{x}_{L} \le x < 0 \\ 0 & x < \tilde{x}_{L} \end{cases}$$
(5.18)

From equation (5.6) we see that the last term  $(g'(x) - K_{p}x_{c})$  in (5.17) is identically zero everywhere except at x=0 where g'(x) is undefined, therefore we conclude that

$$\dot{V}(x,v) = -\left\{K_d v^2 + v F_{slip}(v)\right\} \le 0 \quad \forall (x,v) \notin S.$$
(5.19)

For the points in S, we must look at the Dini-derivates. All the discontinuities are simple, therefore both the left and right limiting values of the derivatives of V(.) exist outside of S. The Dini-derivates are simply these limiting values. Since V(x(t),v(t)) is continuous, and  $\dot{V}(x,v)$  is negative semi-definite for all points outside of S, the Diniderivates are negative semi-definite for points in S (Yoshizawa, 1966). Using the equivalence relation (5.16), the Dini-derivates are negative semi-definite over the entire phase space. From (5.19), the set of points where  $\dot{V}(x,v)=0$  is just the x-axis, and we have already shown that no trajectories are completely contained in this set, therefore the origin is a globally asymptotically stable equilibrium point (Vidyasagar, 1978, Theorem 5.2(87), pg. 158). Q.E.D.

Alternative Proof: For the Lyapunov function candidate given above (5.14) and (5.15), assume there exists some  $V_o > 0$  where, along some trajectory (x,v),

$$\lim_{t \to \infty} V(t) = V_o. \tag{5.20}$$

The level set  $V(x, v) = V_o$  is a closed curve in the phase space. From the previous lemma, trajectories must spiral clockwise around this curve, and must traverse the cycle within a fixed period. Along solution trajectories, going from  $t_0$  to  $t_1$ ,

$$V\Big|_{t_0}^{t_1} = \int_{t_0}^{t_1} \dot{V} \, \mathrm{d}\tau = -\int_{t_0}^{t_1} (K_d v^2 + v F_{slip}) \, \mathrm{d}\tau, \tag{5.21}$$

and from (2.4) we have,  $vF_{slip} \ge b_0 v^2 \forall v$ , therefore

$$V\Big|_{t_0}^{t_1} \le -(K_d + b_0) \int_{t_0}^{t_1} v^2 \, \mathrm{d}\tau = -(K_d + b_0) \int_{x_0}^{x_1} v \, \mathrm{d}x.$$
(5.22)

The last integral in (5.22) is just the area contained inside the path from  $x_0$  to  $x_1$  in the phase space. Each limit cycle will decrease V(t) by at least an amount which is proportional to the area contained within the limit cycle, therefore no such  $V_o$  can exist. Q.E.D.

## 5.4 Simulation Results

A numerical simulation of the nonlinear stick-slip friction compensation force applied to the mass system will verify the stability of the origin. As in the previous chapter, in order to obtain a more realistic simulation, quantized states (simulating the measurement process) are used to compute the control force, which is in turn quantized (simulating the output process). In order to numerically verify robustness of the control law, a representative slipping force model from each of the three classes will be tested with the same control law.

All simulations were performed in double-precision on a Digital Equipment Corporation Micro-VAX, using a Hamming predictor-corrector general integration method with an initial step size of 0.001 s. (Appendix A). Using similar assumptions as those in Chapter 4, the following parameters were chosen to approximately model an experimental system (Table 5.1), and were common to each of the three numerical simulations.

Mass	m	1.0 kg.
PD Control	K <sub>p</sub>	20.0 N/m.
Gains	K <sub>d</sub>	0.5 Ns/m.
Static Friction	$F_s^+$	4.2 N.
Bounds	$F_s^-$	-4.0 N.
Sticking	x <sub>L</sub>	-0.21 m.
Limits	x <sub>H</sub>	0.20 m.
Nonlinear	Э	0.005 m.
Compensation	$ ilde{F}_{s}^{+}$	4.3 N.
Force	$\tilde{F}_{s}^{-}$	-4.1 N.
Extended	<i>x</i> <sub>L</sub>	-0.215 m.
Sticking Limits	<i>x</i> <sub>H</sub>	0.205 m.
Initial	$x(t_o)$	0.3 m.
Conditions	$v(t_o)$	0.0 m/s.

Table 5.1. Common parameters for all nonlinear compensation simulations.

Theoretically, any positive value for  $\varepsilon$  can be used for the nonlinear compensation force. In a real implementation,  $\varepsilon$  must be chosen carefully. This control law assumes that the discontinuous jump in the input force can actually be produced instantaneously. In reality, the bandwidth of the actuator will limit the bang-bang nature of the control. In (5.8), we have seen that  $K_{\rho}\varepsilon$  is a force which is added to the static friction levels to insure that the net force on the mass is non-zero in the sticking zone (Figure 5.1). In this example, the static friction force bounds are known exactly. In actuality, there will only be estimates for these forces. Values for  $\tilde{F}_{s}^{+}$  and  $\tilde{F}_{s}^{-}$  should be chosen to exceed the upper bounds of the estimates for the static friction force levels. The value of  $\varepsilon$  in Table 5.1 corresponds to assumed measurements of the static friction levels with less than a 2.5% error, thus the control force levels,  $\tilde{F}_{s}^{+}$  and  $\tilde{F}_{s}^{-}$ , are chosen to be 0.1 N above the static friction levels. Again, the mass is assumed to initially be at rest outside the sticking zone.

The slipping friction force function used in this numerical study is the same general representation as used in Chapter 4 (4.9) since each of the three classes of slipping force

models are easily represented by these functions. The first model is the Coulomb plus viscous friction model. The defining parameters for this model are given in Table 5.2.

	$F_o^+$	4.2 N.
$F_d^+(v)$	$\Delta F^+$	0.0 N.
	<i>v</i> <sub>o</sub> <sup>+</sup>	NA
	b <sup>+</sup>	0.5 Ns/m.
	F <sub>o</sub> <sup>-</sup>	-4.0 N.
$F_{d}^{-}(v)$	$\Delta F^{-1}$	0.0 N.
	<i>v_</i>	NA
	b <sup>-</sup>	0.5 Ns/m.
Artificial Zero	α	0.01 m/s

Table 5.2. Simulation parameters for nonlinear compensation with the Coulomb model.

A simulated time response verifies the stability of the nonlinear compensator for the Coulomb plus viscous slipping force model (Figure 5.3). Several important observations can be made regarding this time response. As soon as the mass enters the extended sticking zone, the control force takes on its bang-bang character. Note that since the velocity is small, the derivative term in the control force is small also.



Figure 5.3. Simulated time response for nonlinear compensation with Coulomb model.

Whenever the velocity becomes zero, there exists a possibility of becoming stuck. From Figure 5.3, we see that the control force always exceeds the friction force when the velocity becomes zero, thus the mass continues moving. Finally, notice that the mass displacement does not overshoot the desired reference by very much. This is a characteristic of systems with a Coulomb type of slipping force.

The second model is the stiction plus viscous friction force. The defining parameters for this model are given in Table 5.3.

	$F_o^+$	2.1 N.
$F_d^+(v)$	$\Delta F^+$	0.0 N.
	<i>v</i> _{o}^{+}	NA
	<b>b</b> <sup>+</sup>	0.5 Ns/m.
$F_d^-(v)$	F_o^	-2.0 N.
	ΔF <sup>-</sup>	0.0 N.
	v	NA
	b <sup>-</sup>	0.5 Ns/m.
Artificial Zero	α	0.01 m/s

Table 5.3. Simulation parameters for nonlinear compensation with the stiction model.

The time response for the 1-DOF system with a stiction plus viscous slipping force model and the nonlinear compensator is shown in Figure 5.4. Again, when the mass enters the extended sticking zone, the control force takes on its bang-bang character. For this slipping force model, both the displacement and force oscillate more than in the previous example before stabilizing, thus more control energy is required and it takes slightly longer to reach the desired equilibrium.



Figure 5.4. Simulated time response for nonlinear compensation with stiction model.

Again, the control force always exceeds the friction force when the velocity becomes zero, thus the mass continues moving. For this model, the mass displacement does overshoot the desired reference and oscillates about the reference until stabilizing.

The final model is an exponential plus viscous friction force, and is more similar in form to the stiction model than the Coulomb model. The defining parameters for this model are given in Table 5.4.

	$F_o^+$	4.2 N.
$F_d^+(v)$	$\Delta F^+$	2.1 N.
	<i>v</i> _{o}^{+}	0.1 m/s
	<i>b</i> <sup>+</sup>	0.5 Ns/m.
	F_o	-4.0 N.
$F_d^-(v)$	∆F⁻	-2.0 N.
	v	-0.1 m/s
	b <sup>-</sup>	0.5 Ns/m.
Artificial Zero	α	0.01 m/s

Table 5.4. Simulation parameters for nonlinear compensation with the exponential model.

The time response for the 1-DOF system with an exponential plus viscous slipping force model and the nonlinear compensator is shown in Figure 5.5. This response, as expected, is more similar in character to the stiction model response than the Coulomb model response.



Figure 5.5. Simulated time response for nonlinear compensation with exponential model.

Again, when the mass enters the extended sticking zone, the control force takes on its bang-bang character. The displacement and force oscillate more than in the Coulomb example, but not as much as in the stiction example. Less control energy is required for stabilization, and it takes about the same amount of time to reach the desired equilibrium as in the case of stiction.

Each of the simulation plots have a common characteristic. The bang-bang effect of the control force is one where the force switches sign whenever the mass passes through the desired reference. For example, the force may be positive for a period of time until the mass overshoots the origin, then it becomes negative for a shorter period of time. This decrease in the time width of the pulse is observed in each successive pulse for each of the three examples.

From the simulation results, the desired reference position of the 1-DOF system is a stable and unique equilibrium point with the nonlinear compensation force supplementing PD control. As long as reasonable estimates for the upper bounds on the maximum static

67

friction forces can be obtained, this control technique is robust to the exact character of the slipping force. No limit cycle behavior is observed as in the case of PID control with the stiction and exponential models.

#### 5.5 Experimental Results

The nonlinear proportional feedback control law (5.4), (5.6), and (5.7), has also been tested on the experimental system to verify the stability of the origin. The estimates for the static friction bounds from Chapter 3 (Table 3.1) are used to determine the compensation force levels. In an experimental implementation of this compensation force, since we only have estimates for the static friction bounds, we do not need to choose an  $\varepsilon$ explicitly. In order to insure that no sticking would occur in the sticking zone, the applied nonlinear control forces,  $\tilde{F}_s^+$  and  $\tilde{F}_s^-$ , only need to be chosen to exceed the largest estimate. The proportional and derivative gains were chosen identical to the gains in the simulation section for direct comparison. The initial conditions were also chosen identical to those from the simulations.

Mass	m	1.375 kg.
PD Control	K <sub>p</sub>	20.0 N/m.
Gains	K	0.5 Ns/m.
Static Friction	$F_s^+$	1.744 N.
Bounds	$F_s^-$	-1.439 N.
Sticking	x <sub>L</sub>	-0.0872 m.
Limits	x <sub>H</sub>	0.0719 m.
Nonlinear	$ ilde{F}_{s}^{+}$	2.0 N.
Compensation	$\tilde{F}_{s}^{-}$	-2.0 N.
Extended	$ ilde{x}_L$	-0.1 m.
Sticking Limits	$ ilde{x}_{\scriptscriptstyle H}$	0.1 m.
Initial	$x(t_o)$	0.3 m.
Conditions	$v(t_o)$	0.0 m/s.

Table 5.5. Experimental parameters for nonlinear compensation.

The velocity state, which is needed to compute the derivative control force, was estimated with a finite difference algorithm using position data. The floating-point controller algorithm was implemented on a DEC LSI-11/23 computer in Fortran IV, and run at 200 Hz. In the implementation, a first order digital filter was used in the estimation of the velocity state, but no filtering was used on the position data. The experimental data was acquired with the use of an Apple Macintosh IIx computer and National Instruments LabView hardware and software. From each channel, 2000 samples were taken at 1000 Hz. per channel.

To experimentally verify stability and robustness of the nonlinear compensation technique, two tests were performed. In the first test, the friction brake was applied, since the estimates for static friction were obtained under this condition. It was observed in Chapter 3 that PD control led to a non-zero steady-state error, and in Chapter 4, PID control led to a limit cycle response when the friction brake was applied. The resulting time response for this experiment is shown in Figure 5.6. Note the decrease in the width of the control force pulses as observed in the simulations.



Figure 5.6. Experimental time response for nonlinear compensation with braking.

The time response of the experimental system with braking (Figure 5.6) is similar in character to the simulation results for a Coulomb plus viscous friction model (Figure 5.3). This result agrees well with the results in Chapter 4 obtained through the PID simulations and experiment. It was observed that the friction present in the braked system was stiction with a very small but definite drop from the static levels. The drop was estimated to be so small as to be almost Coulomb friction, but the definite drop was required to give a limit cycle response for PID control.

The friction brake was completely removed for the second experiment. Because of this, the static friction bounds obviously decreased, but the remaining stick-slip friction was observed to be much less uniform with position. Theoretically the controller is robust enough to handle this new situation with no change. The resulting time response for this experiment is shown in Figure 5.7.



Figure 5.7. Experimental time response for nonlinear compensation without braking.

The resulting unbraked time response verifies the stability of the origin. Due to the overall decrease in the friction present, the mass reaches the origin much faster than in the braked situation.

# Chapter 6 Conclusions

#### 6.1 Summary

The physical phenomena of the stick-slip motion of one surface over another has been investigated. The friction force induced by the surface contact has been identified as the causal agent, thus it is termed a stick-slip friction force. A good mathematical model for this stick-slip friction force is presented which, when incorporated into a dynamic system model, will provide the stick-slip dynamics observed in practice. This model is composed of a sticking force and a slipping force which do not act simultaneously. The sticking force model provides values of the friction force when the relative velocity of the surfaces is zero, and the slipping force model is assumed to be a function of the relative velocity.

The common models for friction exclusively neglect the sticking force altogether for reasons of analytical simplicity. Omission of the sticking force from the overall friction model will certainly reduce the richness of the set of possible dynamics obtainable from the model. Three of the common slipping force models and their characteristics are investigated throughout this analysis to determine and compare their effects with different control strategies on the regulation of a single degree-of-freedom system. These three models were Coulomb, stiction, and exponential. All three included a linear viscous damping term since it is the most common friction model.

Using a Proportional+Derivative (PD) control law, there exists a set of multiple equilibrium points around the desired equilibrium. This result is verified theoretically as well as experimentally. The steady-state error bounds are inversely related to the

72

proportional gain. Since this gain must be finite, there will always exist non-zero steadystate error bounds for PD control. The static friction bounds for the stick-slip friction force in a given 1-DOF system can be experimentally estimated using the PD control law. With each new proportional gain, there are two new steady-state error bounds which can easily be measured. From these measured bounds, the static friction bounds can be computed. As long as the static friction forces are not dependent on position, the estimates obtained through this procedure will be very good. Each of the estimated static friction bounds for the experimental system with the friction brake applied were no more than 3% away from the mean values of the respective samples. This indicates that there is only a minimal dependence on position, and the estimates are very reasonable.

A Proportional+Integral+Derivative (PID) control law was investigated in an attempt to remove the steady-state error. The initial theoretical analysis demonstrated the existence of trajectories which appeared to have a limit cycle behavior. The nonlinearity of the stick-slip friction force did not allow a closed form solution, therefore numerical simulations were utilized to provide the limit cycle existence results. Simulations with the stiction and exponential models appeared to exhibit limit cycle behavior over the finite time interval of the simulations. The Coulomb model did not appear to exhibit a limit cycle behavior. These results were verified through the numerical construction of a mapping of points in the sticking zone, where the fixed point of this mapping is the limit cycle amplitude. Several representative cases from each class of slipping force model were tested under the same conditions.

A Coulomb friction model in the 1-DOF system with PID control does not exhibit a limit cycle response, whereas the stiction and exponential models promote definite limit cycle dynamics. This fact can be used advantageously to determine the qualitative character of the slipping force present in a real experimental 1-DOF system. If a limit cycle exists

under PID control, the slipping force cannot have the characteristics of the Coulomb model. The amplitude of the observed limit cycle depends somewhat on the amount of the drop from the static friction level to the kinetic friction level. As the drop decreases to zero, the limit cycle amplitude also decreases to zero.

A discontinuous nonlinear compensation force is developed to supplement the PD control law. This compensation force is constructed to make the origin a unique equilibrium point of the system. The discontinuous compensation, combined with the natural discontinuity of the stick-slip friction force create a surface of discontinuity in the phase space. A sliding mode does not exist, and since the discontinuities are simple, all trajectories are absolutely continuous. The origin of the 1-DOF state space is proven to be globally asymptotically stable by two Lyapunov methods. The first uses a modification of the direct method for discontinuous systems. Dini-derivates must be used instead of derivatives for non-differentiable Lyapunov function candidates. The second method shows that the energy of the system (given by the Lyapunov function) must decrease a finite amount with each cycle about the origin, therefore no sustained limit cycle can exist.

Stability of the origin is verified through numerical simulations and an experimental demonstration. Each of the three classes of slipping force models were used in a simulation study with the same nonlinear control law, and the same operating conditions. All three test cases satisfied the assumptions that the slipping force functions be Lipschitz in their respective domains, and they lie within prescribed bounds. Stability was verified with each of the three models, thus the nonlinear compensation is robust to the character of the slipping force. Similar to the results with PID control response, the qualitative character of an experimental slipping force can be determined by comparison with the simulated responses for each of the three classes. The braked and un-braked experimental responses verify the stability of the nonlinear compensation force as well as its robustness.

#### 6.2 Recommendations for future work

The robust nonlinear compensation technique works well for the 1-DOF system, but its effectiveness on a multi-DOF system is questionable. Preliminary investigations of the extension of this control law applied to a 2-DOF system (the inverted pendulum) with stick-slip friction present indicate that it cannot be simply extended. This can be explained by looking at the sticking force. When one of the degrees of freedom is "stuck" the nonlinear control law injects enough energy into the system to get it un-stuck. This can adversely affect the resulting motion as indicated by the simulation study with the inverted pendulum. It is probable that a greater degree of success is attainable with this compensation technique for systems where an actuator and sensor pair are directly associated with each degree of freedom. These types of systems are commonly found in the robotics industry.

A natural extension of this work is to develop online or offline observation algorithms for identification of the slipping forces and static friction levels in an experimental system. The qualitative characterizations which can be obtained through the PID time responses or the nonlinear compensation time responses will provide a better "first guess" to the exact character of the slipping force present.

The nonlinearity associated with real stick-slip friction tends to be very complex and dependent on many quantities. The static friction forces can be strongly dependent on position, and even the slipping forces may vary somewhat with position. With this in mind, successful observation and identification may only be attainable with the "nicest" experimental systems where the slipping forces and static friction bounds are uniform. In dealing with real systems, it is better to investigate the control and observation techniques which do not attempt to identify the exact friction present, but are robust enough to successfully stabilize the system.

Appendices

.

# Appendix A Simulation Models

#### A.1 Numerical Simulation

Numerical solutions to the non-linear differential system equations were computed on a MicroVAX-II running VMS V5.2 using the simulation package DIFFEQ (CCUG, 1989). All computations were performed in Double Precision. The following FORTRAN-77 Functions/Subroutines were linked into the DIFFEQ program for the actual simulation.

#### A.2 Stick-Slip Friction Model.

The following function computes values of the sticking force when the velocity is within the artificial zero range as defined in Chapter 2, Section 2.2. When the velocity is outside of the zero range, the slipping force is evaluated.

```
С
C-----
                                 -----
C----- Fd(V) -----
С
     FUNCTION Fd (V, Eps, Fsp, Fsn, Fr, Cp, Cn)
С
С
  This function computes the value of the damping force.
C
  ---Variable Identification
С
С
     v
С
       the sliding velocity between surfaces
    Eps artificial zero ( > 0 ) for computation purposes
С
     Fsp Positive ( > 0 ) static friction force
с
    Fsn Negative ( < 0 ) static friction force
С
    Fr Reaction force (see notes for definition)
С
С
    Cp coefficient array for pos. non-zero function
    Cn coefficient array for neg. non-zero function
С
С
  ---Declare variables
С
С
     REAL*8 Fd, V, Eps, Fsp, Fsn, Fr, Cp(4), Cn(4)
С
C ---Compute the damping force
С
    IF (V .GT. Eps) THEN
С
    ---Velocity is positive
С
```

С Fd = F(V-Eps, Cp)с ELSE IF (V .LT. -Eps) THEN С ---Velocity is negative С С Fd = F(V+Eps,Cn)с ELSE С ---Velocity is zero С С IF (Fr .GE. Fsp) THEN С Fd = Fsp С ELSE IF (Fr .LE. Fsn) THEN С Fd = FsnС ELSE С Fd = FrС ENDIF С ENDIF С RETURN END C \_\_\_\_\_ C-C----- Fd(V) ------С

The slipping force is computed with the following function whenever the velocity is non-zero, or outside of the artificial zero range. The value of the slipping force is a function of the velocity, and a set of four parameters stored in a parameter vector. With a proper choice of the parameters, any of the three general stick-slip friction models presented in Chapter 2, Section 2.3, can be represented. This function can produce Coulomb, Stiction, or Exponential models with viscous friction added in.

```
с
c--
C----- F(V,C) -----
С
    FUNCTION F (V, C)
С
  This function computes the value of the damping force for
С
С
  nonzero velocity.
С
С
 ---Declare variables
С
    REAL*8 F,V,C(4)
С
```

### A.3 One-Degree-Of-Freedom Model

The 1-DOF system equations were written into the following subroutine for evaluation by the DIFFEQ package. Through this package, a set of parameters could be interactively changed to obtain different modes of operation. For example, the PD and PID controllers were both included in the following code.

```
С
C-
        C-
       ----- P2SIM ------
С
С
     SUBROUTINE P2SIM
С
C This subroutine has the simple forced mass problem with
C the stick-slip damping model, controlled by PID feedback.
С
С
  ---DIFFEQ subroutine name
С
     SUBROUTINE FCT ( T, S, DerS )
с
С
  ---Variable Identification
С
С
     Т
           The real time variable
С
     s
           State vector
С
     DerS Derivative of state vector
С
     PA
           Parameter vector
С
С
     Kp
           Position control gain
С
     Kd
           Velocity control gain
С
     Ki
           Integral control gain
С
     М
           Actual mass
С
С
    Fsp
          Positive static friction force
С
    Fsn
           Negative static friction force
С
           Coefficient array for damping force (+)
     Ср
с
    Cn
           Coefficient array for damping force (-)
С
    dF
          Damping force from the model
С
           Name of function generating damping force
    Fd
С
с
     Fr
           Reaction force (or control force)
С
     Fc
           Compensation force
С
     Fpid PID control force
с
С
     EpsX Quantization precision for position
С
     EpsV
          Quantization precision for velocity
С
     EpsF Quantization precision for force
```

```
С
      EpsT
             Quantization precision for time
С
С
      Τk
              Quantized time variable
С
      Х
              Quantized mass position
              Quantized mass velocity
С
      v
С
      Y
              Quantized integral of the error
С
      Е
              Position error
              Velocity error
С
      DE
С
      ΙE
              Integral of the error
С
С
   ---Declare variables
С
      REAL*8 DerS(20), S(20), PA(20), T
С
      REAL*8 Kp, Kd, Ki, M, Fsp, Fsn, Cp(4), Cn(4),
              Fd, dF, Fr, X, V, Y, E, DE, IE, Xd, Vd,
     +
              EpsX, EpsV, EpsF, EpsT
С
      REAL*8 Fpid, Fc, Xc, Xh, Xl, Eps, G, dG
С
      REAL*8 Tk, Told, Xold, Vold
С
С
   ---Common for fct parameters
С
      COMMON /FCTCOM/PA
С
С
   ---Set up all local parameters
С
      EQUIVALENCE (Kp,
                           PA(1)),
                   (Kd,
                           PA(2)),
     +
     +
                   (Ki,
                           PA(3)),
                   (EpsX, PA(4)),
     +
                   (EpsV, PA(5)),
(EpsF, PA(6)),
     ŧ
     +
     ŧ
                   (EpsT, PA(7)),
                   (Eps, PA(8)),
     +
                   (Fsp, PA(11)),
     +
                   (Fsn, PA(12)),
     +
                   (Cp(1),PA(13)),
     +
                   (Cn(1), PA(17))
С
  ---Initialize local constants
С
С
      DATA M, Xd, Vd /1.0D0,0.0D0,0.0D0/
С
С
   ---Simulate measurement by quantizing the states
С
      Tk = Q ( EpsT, T )
      X = Q (EpsX, S(1))
      V = Q ( EpsV, S(2) )
      Y = S(3)
С
С
  ---Compute the errors
С
      E = Xd - X
      DE = Vd - V
      IE = - Y
С
  ---Compute the PID control force
С
С
      Fpid = Kp*E + Kd*DE + Ki*IE
с
С
   ---Compute the compensation force
С
```

```
79
```

```
Xh = - (Fsn / Kp) + Eps
     Xl = - (Fsp / Kp) - Eps
С
      IF (X .LT. X1) THEN
       Xc = 0.0D0
      ELSE IF (X .GT. Xh) THEN
       Xc = 0.0D0
      ELSE IF (X .GT. EpsX/2.0D0) THEN
       Xc = Xh - X
      ELSE IF (X .LT. -EpsX/2.0D0) THEN
       Xc = Xl - X
      ELSE
       Xc = 0.0D0
     ENDIF
С
     Fc = - Kp * Xc
с
     Fr = Q ( EpsF, (Fpid + Fc) )
С
С
  ---Compute the actual damping force
С
     dF = Fd (S(2), EpsV/2.0D0, Fsp, Fsn, Fr, Cp, Cn)
С
C ---Evaluate system derivatives
С
      IF ( ABS(S(2)) .LE. EpsV/2.0D0 ) THEN
        DerS(1) = 0.0D0
      ELSE
        DerS(1) = S(2)
      ENDIF
С
      DerS(2) = (Fr - dF) / M
С
     DerS(3) = - E
С
  ---Compute auxiliary outputs
С
С
      S(4) = 10.0 * S(1)
     S(5) = 10.0 * S(2)
      S(6) = Y
      S(7) = Fr
      S(8) = dF
      S(10) = Fc
С
C ---Compute the Lyapunov function
С
      IF (S(1) .GT. Xh) THEN
        G = Kp * Xh * Xh / 2.0D0
        dG = 0.0D0
     ELSE IF (S(1) .LT. X1) THEN
        G = Kp * X1 * X1 / 2.0D0
        dG = 0.0D0
      ELSE IF (S(1) .GE. 0.0D0) THEN
        G = Kp * S(1) * (Xh - S(1)/2.0D0)
        dG = Kp + (Xh - S(1))
      ELSE
        G = Kp + S(1) + (X1 - S(1)/2.0D0)
        dG = Kp * (X1 - S(1))
      ENDIF
С
      S(11) = G + (Kp*S(1)*S(1) + M*S(2)*S(2)) / 2.0D0
      S(12) = Kp*S(1)*S(2) + M*S(2)*DerS(2) + dG*S(2)
      S(13) = 10.0 * Xh
     S(14) = 10.0 * X1
```

```
80
```

```
С
   RETURN
   END
С
C---
  C-
   С
С
 ---Insert all the INCLUDEd functions
с
   INCLUDE '[SOUTHWARD.RESRCH.PID.COMMON]FD.FOR'
   INCLUDE '[SOUTHWARD.RESRCH.PID.COMMON]F.FOR'
   INCLUDE '[SOUTHWARD.RESRCH.PID.COMMON]Q.FOR'
С
c-
         _____
c-
 с
```

In the FCT subroutine above, the actual position and velocity are quantized to simulate the discrete measurement process. This quantization is performed by the following function.

```
С
C-
   _____
                       -----Q(Eps,X) -----
C-
С
   FUNCTION Q ( Eps, X )
С
 This quantizer maps real X values into the Q-Eps space.
С
С
C ---Declare Variables
С
   REAL*8 Q, X, Eps, MAX
С
С
 ---Compute the maximum value quantizable
С
   MAX = Eps * (2.0D0 ** 30)
С
 ---Compute the R value
С
с
    IF (ABS(X) .LT. MAX) THEN
     Q = Eps * JIDINT((X/Eps) + DSIGN(0.5D0,X))
    ELSE
     Q = X
    ENDIF
С
    RETURN
   END
С
C-----
C----- Q(Eps, X) -----
С
```

# Appendix B Experimental Controller

### **B.1** Controller Implementation

The two control algorithms used in this investigation, PID and PD with nonlinear compensation, were implemented on a DEC PDP-11/23 running RSX-11M+. The programs were written in FORTRAN IV and MACRO-11. All control computations were carried out in Single Precision Floating Point. Four modules were common to both algorithm implementations. These four modules are IPOSN, PAUSE, SETICK, and SAMPLE, and can be found in Section B.3. The specific PID routines are in the following section.

### **B.2 PID Controller.**

The PID control algorithm is implemented in the following program. A single input voltage is read and filtered with a first order digital filter. The derivative of this filtered signal is computed and filtered as well at user specified time constants. The integral of the input error is also computed. The PID control signal is then computed and applied. This program allows the user to interactively tune all parameters.

```
С
c---
  ______
C----- PID -----
С
    PROGRAM PID
С
C This is the main program which accesses all the control
  subroutines for PID control.
С
С
C Written by: STEVE C. SOUTHWARD
С
С
   ---Program Calling Tree
С
С
    PID
      IPOSN
С
```

С SAMPLE с CGAINS С CONTRL С RUNLSF С DAUSE SETICK С С С ---Declare Variables С REAL G(10), F(10) С LOGICAL\*1 IDATA С DIMENSION IDLIST(21) С EXTERNAL SETICK С COMMON /ADDR1/ IKSR, IKDR, INCH, ITCK COMMON /ADDR2/ IPSR, IPDR, IBPR, IADVEC, IADERR COMMON /ADDR3/ ICSR, IADR, IDAR, IADBUF COMMON /MISC2/ F,G,IZREF COMMON /TIME/ I, IEND С С ---Initialize the zero reference С DATA IZREF/0/ С С ---Initial PID control gains, and other parameters С F(1) = 10.0!Kp \ F(2) = 5.0!Kd PID controller gains F(3) = 2.0!Ki / С F(4) = 1.0/(1.0 + 3.0)!position F(5) = 1.0 - F(4)!filter С F(6) = 1.0/(1.0 + 10.0)!derivative F(7) = 1.0 - F(6)!filter С F(8) = -5.982!input scale factor F(9) = 1.68!output scale factor С F(10) = 1.0!maximum allowable error С С ---Set up address registers С IPSR = "102!processor status word IKSR = "177560!keyboard status register IKDR = IKSR + 2!keyboard data register IPSR = IKSR + 4!printer status register IPDR = IKSR + 6!printer data register ICSR = "170420!real time clock status reg. IBPR = ICSR + 2!clock buffer preset reg. ICKV = "440!clock vector address IDAR = "170440!D/A base output addr.: ch.0 IADR = "170400!A/D control status register IADBUF = IADR + 2**!A/D** buffer register IADVEC = "400!A/D done interrupt vector IADERR = IADVEC + 4!A/D error interrupt vector IBELL = "7 !bell tone IZERO = "4000!zero value for D/A converter С С ---Protect System from crash С

IDLIST(1) = IKSR!keyboard status IDLIST(2) = IPEEK(IKSR)IDLIST(3) = IPSR!printer status IDLIST(4) = IPEEK(IPSR) IDLIST(5) = IDAR !zero D/A channel 0 IDLIST(6) = IZEROIDLIST(7) = IDAR + 2!zero D/A channel 1 IDLIST(8) = IZEROIDLIST(9) = IDAR + 4!zero D/A channel 2 IDLIST(10) = IZERO IDLIST(11) = IDAR + 6!zero D/A channel 3 IDLIST(12) = IZEROIDLIST(13) = IADR !A/D status IDLIST(14) = 0IDLIST(15) = IADVEC **!A/D done interrupt vector** IDLIST(16) = IPEEK(IADVEC) IDLIST(17) = ICSR!clock status IDLIST(18) = IPEEK(ICSR) IDLIST(19) = IPSW!processor status IDLIST(20) = IPEEK(IPSW) IDLIST(21) = 0!end of list marker С CALL DEVICE (IDLIST) С С ---Format Statements С 500 FORMAT(/,/,/,2X,'Discrete Time PID Control', + /,2X,'-----') 510 FORMAT(/,2X, 'Enter the INPUT A/D channel: ',\$) 530 FORMAT(/,2X,'Enter the OUTPUT D/A channel: ',\$) 540 FORMAT(I2) 550 FORMAT(/,/,10X,'----MAIN MENU-----', /,/,2X,'(I)nitialize the zero reference position', + /,/,2X,'(G)ain settings for PID control & filters', + /,/,2X,'(S)tatus of Zero position', + /,/,2X,'(R)un the active PID controller', + /,/,2X,'(Q)uit the program', /,/,2X,'Choose one of the above. . . ',\$) 560 FORMAT(/,/,2X,'STATUS of Zero Reference. . .', /,2X,'-----') 570 FORMAT(/,2X,'INPUT zero reference: ',16) 590 FORMAT(/,/,2X,'Exiting Control Routine. ..',5(/)) 600 FORMAT(/,/,2X,'Zero reference position NOT initialized') С С ---Calibrate the PAUSE subroutine С IEND = 10000!set initial value IPR = 7!highest priority interrupt С 10 I = INTSET(ICKV, IPR, 1, SETICK) !attach to RTC vector IF (I .EQ. 0) GOTO 20 С WRITE(7,\*)'INTSET error -- RTC vector, CODE = ',I GOTO 10 С 20 IRATE = "111 !set up clock to generate ICOUNT = -16667!an interrupt after 1 tick с CALL IPOKE (IBPR, ICOUNT) CALL IPOKE (ICSR, IRATE) !start the clock С DO 30 I = 1,10000!do nothing loop while 30 CONTINUE !waiting for one tick С

```
ITCK = IEND + 1
D
       WRITE(7, *) 'ITICK = ', ITCK
С
С
     ---Set zero (0) volts on all D/A outputs
С
       DO 40 I = 0, 6, 2
                                     !
          CALL IPOKE(IDAR+I, IZERO) !zero out all 4 channels
   40 CONTINUE
                                     !
С
     ---Get the INPUT A/D channels
С
С
       WRITE(7,500)
                                    !initial header
   50 WRITE(7,510)
                                    !get INPUT channel
       READ(5,540,ERR=50) INCH
       IF ((INCH.GT.15).OR.(INCH.LT.0)) GOTO 50
С
   70 WRITE(7,530)
                                    !get D/A output channel
       READ(5,540,ERR=70) IDACH
       IF ((IDACH.LT.0).OR.(IDACH.GT.3)) GOTO 70
       IDAR = IDAR + IDACH * 2
С
С
     ---Print out the options menu
с
   80 WRITE(7,550)
                                     !print the menu options
       CALL PAUSE (50, ITCK)
                                    !wait for printing to finish
С
С
     ---Disable the keyboard
С
       I = IPEEK(IKSR) .AND. "177477!clear keyboard interrupt
       CALL IPOKE (IKSR, I)
С
С
    ---Wait for keyboard input
С
   90 CALL IPOKE (IPDR, IBELL)
                                   !ring the bell
С
  100 I = IPEEK(IKSR) .AND. "200
                                     !test bit 7 of IKSR
       IF (I .EQ. 0) GOTO 100
С
       IDATA = IPEEK(IKDR)
                                    !a key has been pressed
с
С
     ---Check for a valid character
С
       IF ((IDATA.NE.'I').AND.(IDATA.NE.'G').AND.(IDATA.NE.'S').AND.
           (IDATA.NE.'R').AND.(IDATA.NE.'Q')) GOTO 90
     +
С
       CALL IPOKE (IPDR, IDATA)
                                    !print the proper key
С
С
     ---Enable the keyboard
С
       I = IPEEK(IKSR) .OR. "100 !set bit 6 of IKSR
       CALL IPOKE(IKSR, I)
       WRITE(7,*)'
с
С
     ---Go to the proper place
с
       IF (IDATA .EQ. 'I') GOTO 110
       IF (IDATA .EQ. 'G') GOTO 120
       IF (IDATA .EQ. 'S') GOTO 130
       IF (IDATA .EQ. 'R') GOTO 140
       IF (IDATA .EQ. 'Q') GOTO 150
С
С
     ---INITIALIZE the zero reference position
С
  110 CALL IPOSN
```

с GOTO 80 !return to menu С С ---GAIN settings for PID control law С 120 CALL CGAINS С GOTO 80 !return to menu С С ---STATUS of Zero position С 130 WRITE(7,560) !initial header WRITE(7,570) IZREF С GOTO 80 !return to menu с С ---RUN the active control routine С 140 IF (IZREF .NE. 0) GOTO 145 с WRITE(7,600) !zero references GOTO 80 !not initialized С 145 CALL CONTRL С GOTO 80 !return to menu С с ---QUIT the program С 150 WRITE(7,590) !exit message CALL PAUSE (60, ITCK) CALL EXIT С END С C-----C----- CGAINS -----С SUBROUTINE CGAINS С C This subroutine allows the user to input the control C gains for the PID control law, and the first order C filter parameters. С С ---Definition of parameters F() С С F(1) = PID proportional gain (Kp) F(2) = PID derivative gain (Kd) С F(3) = PID integral gain (Ki) С F(4) = posn. filter parameter (alfa) С F(5) = posn. filter parameter (1-alfa) С С F(6) = deriv. filter parameter (alfa) С F(7) = deriv. filter parameter (1-alfa) С F(8) = Input scale factorС F(9) = Output scale factorС F(10) = Maximum allowable error С с ---Declare Variables с REAL G(10), F(10), SN С COMMON /MISC2/ F,G,IZREF

```
С
С
     ---Format Statements
C
  100 FORMAT(/,/,2X,'INPUT the PID Control Gains',/,
          2X. '-----')
     +
  110 FORMAT(/,2X,'1) Input PID Controller gains',
         /,2X,'2) Input Position Filter Parameters',
     +
         /,2X,'3) Input Derivative Filter Parameters',
/,2X,'4) Input INPUT/OUTPUT scale factors',
/,2X,'5) Input maximum allowable error EMAX',
/,2X,'6) Check Status of all gains',
/,2X,'7) EXIT to main routine. .')
  120 FORMAT(/,2X,'Your Selection: ',$)
  130 FORMAT(I1)
  140 FORMAT (/, 2X, A14, ' = ', $)
  150 FORMAT(/,2X,'Enter # of Samples/Filter Time Const.: ',$)
  160 FORMAT(/,2X, 'PID Controller Gains:',/,/,
          5X, 'Kp = ', F10.4, /,
     +
          5X, 'Kd = ', F10.4,/,
     +
         5X,'Ki = ',F10.4)
  170 FORMAT(/,2X,A10,' Filter Parameters:',/,/,
          5X, '#Samples/Time Constant = ',F5.1,/,/,
     +
          5X, ' alpha = ', F10.4,/,
          5X, '1-alpha = ', F10.4)
  180 FORMAT(/,2X,'Enter the ',A6,' scale [Volt/ ]: ',$)
  190 FORMAT(/,2X, 'Calibration Scale Factors:',/,/,
          5X, 'Input Scale = ', F10.4,/,
          5X, 'Output Scale = ',F10.4)
  200 FORMAT(/,2X,'Enter the maximum allowable error: ',$)
  210 FORMAT(/,2X, 'Maximum Allowable Error:',/,/,
        5X, 'Emax = ', F10.4)
с
с
     ---Main routine
С
        WRITE(7,100)
   10 WRITE(7,110)
                                       !display menu options
   20 WRITE(7,120)
        READ(5,130,ERR=20) IOP
        IF ((IOP.LT.1).OR.(IOP.GT.7)) GOTO 20
С
С
     ---Goto the proper place
С
        GOTO (30, 40, 50, 60, 70, 80, 90) IOP
С
С
     ---Input new PID controller gains
С
   30 WRITE(7,140) 'Kp [Volt/L] '!proportional gain
        READ(5,*,ERR=30) F(1)
С
   32 WRITE(7,140) 'Kd [Volt-S/L]'!derivative gain
        READ(5,*,ERR=32) F(2)
С
   34 WRITE(7,140) 'Ki [Volt/L-S]'!integral gain
        READ(5, *, ERR=34) F(3)
с
        GOTO 10
                                        !return to CGAINS menu
С
С
     ---Input the filter parameters
С
   40 WRITE(7,150)
                                        !get position filter
       READ(5,*,ERR=40) F(4)
С
        F(4) = 1.0 / (1.0 + F(4))
        F(5) = 1.0 - F(4)
```

```
С
      GOTO 10
                                  !return to CGAINS menu
С
     ---Input the derivative filter parameters
С
С
   50 WRITE(7,150)
                                  !get derivative filter
       READ(5,*,ERR=50) F(6)
С
       F(6) = 1.0 / (1.0 + F(6))
       F(7) = 1.0 - F(6)
С
       GOTO 10
                                   !return to CGAINS menu
С
С
     ---Input the INPUT/OUTPUT scale factors
С
   60 WRITE(7,180) 'INPUT'
                                  !get input scale factor
       READ(5,*,ERR=60) F(8)
С
   65 WRITE (7,180) 'OUTPUT'
                                  !get output scale factor
       READ(5,*,ERR=65) F(9)
С
                              .
       GOTO 10
                                   !return to CGAINS menu
С
С
    ---Input the maximum allowable errror EMAX
С
   70 WRITE(7,200)
                                  !get EMAX
       READ(5, *, ERR=70) F(10)
С
       GOTO 10
с
С
     ---Check the status of all gains
С
   80 WRITE(7,160) (F(I), I=1,3)
с
       SN = (1.0 / F(4)) - 1.0
       WRITE(7,170) 'Position ', SN, F(4), F(5)
С
       SN = (1.0 / F(6)) - 1.0
       WRITE(7,170) 'Derivative', SN, F(6), F(7)
С
       WRITE(7,190) F(8),F(9)
С
       WRITE(7,210) F(10)
С
       GOTO 10
                                  !return to CGAINS menu
С
С
    ---EXIT to the main program
С
   90 RETURN
       END
С
c-
C----- CONTRL -----
С
       SUBROUTINE CONTRL
С
C This is the control subroutine which sets up the
C clocked interrupt service control routine, and
C starts the control action.
С
       REAL G(10), F(10), FREQ, TICK, PER
       REAL EN, DEN, IEN, EO, DEO, IEO, VOLT
```

```
С
       LOGICAL*1 IDATA, TI1(8), TI2(8)
С
       EXTERNAL RUNLSF
                                    !to use as subroutine arg.
с
       COMMON /ADDR1/ IKSR, IKDR, INCH, ITCK
       COMMON /ADDR2/ IPSR, IPDR, IBPR, IADVEC, IADERR
       COMMON /ADDR3/ ICSR, IADR, IDAR, IADBUF
       COMMON /MISC1/ ICHAN, PER, IFAST
       COMMON /MISC2/ F,G,IZREF
       COMMON /STATE/ EO, DEO, IEO, EN, DEN, IEN, VOLT
С
С
     ---Format Statements
С
  100 FORMAT(/,/,2X,'RUN the Clocked Control Routine',
       /, 2X, '-----')
  110 FORMAT(/,2X,'Enter the SAMPLING FREQUENCY (Hz.): ',$)
  120 FORMAT(/,2X,'ERROR. . . [Sampling frequency is too high.]')
  130 FORMAT(/,2X,'ERROR. . .[Sampling frequency is too low.]')
  140 FORMAT(/,2X,'ERROR. . .[Sampling rate TOO HIGH for system.]')
  150 FORMAT(/,2X, 'Actual sampling freq. = ',E11.4, ' Hz.')
  160 FORMAT(/,2X, 'Actual sample period = ',E11.4, ' Sec.',/,/)
  170 FORMAT(/,2X, 'Press ANY KEY to STOP active control. . .')
  180 FORMAT(/,2X, 'Are you ready to begin control [Y,N]. . . ',$)
  190 FORMAT('+',5X,'Error: ',F10.5,5X,'dError: ',F10.5,
                 5X, 'Output: ', F10.5)
    +
  220 FORMAT('+',2X,'Turn OFF the Line Time Clock. . .')
  230 FORMAT(/,2X,'ERROR. . .[System too far from origin]')
  240 FORMAT(/,2X,'Is system near the origin [Y,N]. . . ',$)
c
С
     ---Initialize Variables
с
       IBELL = "7
       IPR = 7
                                    !highest priority interrupt
       IFAST = 0
                                    !reset too-high flag
       FMAX = 200.0
                                    !set max. sampling frequency
       VOLT = 0
                                    !initial voltage
С
С
     ---Convert real gains to controller gains
С
       G(1) = -F(1) + F(9) / F(8)!Kp 
       G(2) = -F(2) + F(9) / F(8)!Kd PID controller gains
       G(3) = -F(3) * F(9) / F(8)!Ki /
с
       G(4) = F(4)
                                    !position
       G(5) = F(5)
                                    !filter
С
       G(6) = F(6)
                                    !derivative
       G(7) = F(7)
                                    !filter
С
       G(10) = F(10) * F(8) * 204.8!EMAX
       G(10) = ABS(G(10))
С
С
     ---Set up clocked ISR
С
       WRITE(7,100)
                                    !intial header
с
   10 I = INTSET(IADVEC, IPR, 1, RUNLSF) ! attach to A/D done vector
       IF (I .EQ. 0) GOTO 20
с
       WRITE(7,*) 'INTSET error -- A/D Vector, CODE = ',I
       GOTO 10
С
   20 I = INTSET(IADERR, IPR, 2, RUNLSF)! attach to A/D error vector
```

```
IF (I .EQ. 0) GOTO 30
с
       WRITE(7,*) 'INTSET error -- A/D Error Vector, CODE = ',I
       GOTO 20
С
с
     ---Input the sampling frequency
с
   30 WRITE(7,110)
                                     !get the sampling frequency
       READ(5, *, ERR=30) FREO
С
       IF ((FREQ.LE.FMAX).AND.(FREQ.GT.0.)) GOTO 35
       WRITE(7,120)
                                    !sampling frequency too high
       GOTO 30
С
с
     ---Calculate the best base clock rate
С
                                     !start at highest clock rate
   35 IR = 1
с
   40 TICK = (10.0**(7-IR))/FREQ
с
       IF (TICK .LT. 32767.) GOTO 45!integer out of range
       IR = IR + 1
                                     !next lower base frequency
       IF (IR .LE. 7) GOTO 40
С
       WRITE(7,130)
                                     !sampling frequency too low
       GOTO 30
с
С
     ---Calculate the ticks for IBPR
С
   45 ITICK = IFIX(-1.0*(TICK+0.5))!nearest integer
С
С
     ---Calculate the actual sampling frequency and period
С
       FREQ = (10.0**(7-IR))/FLOAT(-ITICK)
С
       PER = 1.0 / FREQ
С
       WRITE(7,150) FREQ
       WRITE(7,160) PER
С
С
     ---Set up the clock status and A/D status registers
С
   50 IRATE = (IR * 8) + 3
       ICHAN = "40140 + INCH * (2**8)!set up input sample
С
С
     ---Make sure the line time clock is turned off
С
   55 CALL TIME(TI1)
                                     !get the first time
       CALL PAUSE (60, ITCK)
                                     !wait for 1 second
       CALL TIME(TI2)
                                     !get the second time
С
       IF ((TI1(8).EQ.TI2(8)).AND.(TI1(7).EQ.TI2(7))) GOTO 60
с
       CALL IPOKE (IPDR, IBELL)
       WRITE(7,220)
                                     !message to user
       GOTO 55
С
С
     ---Ready to begin active control
С
   60 WRITE(7,170)
                                     !user instructions
       WRITE(7,180)
                                     !stop message
       CALL PAUSE (40, ITCK)
                                     !wait for terminal
С
С
     ---Disable keyboard and wait for input [Y,N]
```

#### 90

```
с
       I = IPEEK(IKSR) .AND. "177677!reset bit 6 of IKSR
       CALL IPOKE (IKSR, I)
С
   65 I = IPEEK(IKSR) .AND. "200 !test bit 7 of IKSR
       IF (I .EQ. 0) GOTO 65
С
       IDATA = IPEEK(IKDR)
                                    !a key has been pressed
С
       IF ((IDATA.EQ.'Y').OR.(IDATA.EQ.'N')) CALL IPOKE(IPDR, IDATA)
       IF (IDATA .EQ. 'Y') GOTO 70
       IF (IDATA .EQ. 'N') GOTO 85
С
       CALL IPOKE (IPDR, IBELL)
       GOTO 65
С
С
     ---Disable the printer
С
   70 I = IPEEK(IPSR) .AND. "177677!reset bit 6 of IPSR
       CALL IPOKE (IPSR, I)
С
С
     ---Set up initial conditions for the system
с
   72 CALL SAMPLE(INCH, INVAL, IFLAG) !get initial position
С
       EN = (IZREF - INVAL)
                                   !initial error
       DEN = 0.0
       IEN = 0.0
С
с
     ---Check for initial state too far from origin
С
       IF (ABS(EN) .LE. G(10)) GOTO 74
С
       IDATA = IPEEK(IKDR)
                                    !reset the keyboard status
С
       I = IPEEK(IPSR) .OR. "100
                                    !set bit 6 of IPSR
       CALL IPOKE (IPSR, I)
С
       WRITE(7,*)' '
       WRITE(7,240)
       GOTO 65
С
С
     ---Start clock and ISR
С
   74 CALL IPOKE (IBPR, ITICK)
       CALL IPOKE (ICSR, IRATE)
                                     !start clock ticking
       CALL IPOKE (IADR, ICHAN)
                                    !start A/D conversion
С
С
     ---Wait for keyboard input or too-fast error
С
       I = IPEEK(IPSR) .OR. "100
                                  !set bit 6 of IPSR
       CALL IPOKE (IPSR, I)
С
       WRITE(7,*) ' '
       WRITE(7,*) ' '
       WRITE(7,*) '
                     .
       WRITE(7,*) '
                     .
   80 IF (IFAST .NE. 0) GOTO 85
       WRITE(7,190) EN, DEN, VOLT
с
       I = IPEEK(IKSR) .AND. "200 !test bit 7 of IKSR
       IF (I .EQ. 0) GOTO 80
С
С
     ---Enable the keyboard and printer
```

```
С
  85 I = IPEEK(IKSR) .OR. "100
                                  !set bit 6 of IKSR
      CALL IPOKE (IKSR, I)
с
       I = IPEEK(IPSR) .OR. "100
                                  !set bit 6 of IPSR
      CALL IPOKE (IPSR. I)
С
С
     ---Check for too-fast error
С
       IF (IFAST .EQ. 0) GOTO 95
       IF (IFAST .EQ. 2) GOTO 90
С
      WRITE(7,*)' '
      WRITE(7,140)
                                  !print an error message
      GOTO 95
С
   90 WRITE(7,*)' '
      WRITE(7,230)
                                  !print an error message
С
с
    ---Turn off clock and A/D converter
С
   95 CALL IPOKE (IADR, 0)
      CALL IPOKE(ICSR, 0)
      WRITE(7,*)' '
С
с
    ---Zero output voltage
С
       IZERO = "4000
      CALL IPOKE (IDAR, IZERO)
с
      RETURN
       END
С
C-----
C----- RUNLSF -----
С
       SUBROUTINE RUNLSF(ID)
С
С
 This is an interrupt service routine to get clocked samples
С
  of the input signal and compute the PID output control voltage.
С
С
    ---Definition of parameters G()
с
С
      G(1) = proportional gain
С
      G(2) = derivative gain
С
      G(3) = integral gain
С
      G(4) = filter parameter (alfa)
с
      G(5) = filter parameter (1-alfa)
С
      G(6) = deriv. filter parameter (alfa)
С
      G(7) = deriv. filter parameter (1-alfa)
С
       REAL G(10), F(10), VOLT, PER, VMAX
       REAL EO, DEO, IEO, EN, DEN, IEN
С
      COMMON /ADDR3/ ICSR, IADR, IDAR, IADBUF
      COMMON /MISC1/ ICHAN, PER, IFAST
       COMMON /MISC2/ F,G,IZREF
      COMMON /STATE/ EO, DEO, IEO, EN, DEN, IEN, VOLT
С
       DATA VMAX/2047.5/
с
С
    ---Select entry: ID = 1 . . . A/D Sample Done [PID control]
```
```
С
                      ID = 2 . . A/D Sample Error
С
       GOTO (10,20) ID
      WRITE(7,*) 'RUNLSF entry error, ID = ', ID
      CALL EXIT
С
       <<<<< Calculate PID Errors >>>>
С
С
С
     ---Get the new input sample
С
   10 EN = G(4) * (IZREF - IPEEK(IADBUF)) + G(5) *EO
с
      CALL IPOKE (IADR, ICHAN)
                                   !reset A/D status register
С
С
    ---Calculate derivative of the input error
С
       DEN = G(6) * (EN - EO) / PER + G(7) * DEO
с
С
    ---Calculate the integral of the input error
С
       IEN = IEO + EN * PER
с
С
    ---Shift new variables into the old variables
с
       EO = EN
      DEO = DEN
       IEO = IEN
С
с
     ---Use these approximated errors for the PID control law
С
С
       <<<<< Calculate PID Control Law >>>>
С
С
    ---Calculate the PID control law
С
       VOLT = G(1) * EN + G(2) * DEN + G(3) * IEN
С
С
    ---Check for voltage out of range
С
       IF (VOLT .LT. -VMAX) VOLT = -VMAX
       IF (VOLT .GT. VMAX) VOLT = VMAX
С
      IVOLT = IFIX(VOLT + 2048.0)
С
      CALL IPOKE(IDAR, IVOLT) !apply the voltage
С
      RETURN
С
С
    ---A/D Sample error
С
   20 CALL IPOKE (IADR, 0)
                                   !turn off A/D
      CALL IPOKE(ICSR, 0)
                                   !turn off RTC
С
       IFAST = 1
                                   !set the too-fast flag
С
       IF (ABS(EN) .GT. G(10)) IFAST = 2 !out of range
С
       RETURN
       END
С
C-
c-
   ----- PID -----
С
```

```
93
```

# **B.3 PD Control with Nonlinear Stick-Slip Compensation**

The PD control algorithm is implemented in the following program. A single input voltage is read and filtered with a first order digital filter. The derivative of this filtered signal is computed and filtered as well at user specified time constants. The nonlinear compensation force is also calculated. The compensated PD control signal is then computed and applied. This program allows the user to interactively tune all parameters.

```
С
C--
              C----- PD ------
С
      PROGRAM PD
С
C This is the main program which accesses all the control
С
  subroutines for PD control.
С
C Written by: STEVE C. SOUTHWARD
С
С
    ---Program Calling Tree
С
с
      PD
С
        IPOSN
С
        SAMPLE
С
        CGAINS
С
        CONTRL
с
         RUNLSF
С
        PAUSE
С
        SETICK
С
С
    ---Declare Variables
С
      REAL G(13), F(13)
С
      LOGICAL*1 IDATA
С
      DIMENSION IDLIST(21)
С
      EXTERNAL SETICK
с
      COMMON /ADDR1/ IKSR, IKDR, INCH, ITCK
      COMMON /ADDR2/ IPSR, IPDR, IBPR, IADVEC, IADERR
      COMMON /ADDR3/ ICSR, IADR, IDAR, IADBUF
      COMMON /MISC2/ F,G,IZREF
      COMMON /TIME/ I, IEND
С
С
    ---Initialize the zero reference
С
      DATA IZREF/0/
С
С
    ---Initial PD control gains, and other parameters
С
      F(1) = 20.0
                                !Kp \
                                !Kd / PD controller gains
      F(2) = 0.5
С
```

```
F(3) = 1.0/(1.0 + 0.0)
                                  !position
       F(4) = 1.0 - F(3)
                                   !filter
С
       F(5) = 1.0/(1.0 + 100.0)
                                   !derivative
       F(6) = 1.0 - F(5)
                                   !filter
С
       F(7) = 19.895
                                   !input scale factor
       F(8) = 0.75936
                                   !output scale factor
С
                                   !maximum allowable error
       F(9) = 0.4
с
      F(10) = 2.0
                                   !Fsp (+ve static friction)
       F(11) = -2.0
                                   !Fsn (-ve static friction)
       F(12) = 10.0
                                   !EpsX (small +ve number)
       F(13) = 1.0
                                   !Eps (small +ve number)
С
С
    ---Set up address registers
С
       IPSW = "102
                                   !processor status word
       IKSR = "177560
                                   !keyboard status register
       IKDR = IKSR + 2
                              . !keyboard data register
       IPSR = IKSR + 4
                                   !printer status register
       IPDR = IKSR + 6
                                   !printer data register
       ICSR = "170420
                                   !real time clock status reg.
       IBPR = ICSR + 2
                                   !clock buffer preset reg.
       ICKV = "440
                                   !clock vector address
       IDAR = "170440
                                  !D/A base output addr.: ch.0
       IADR = "170400
                                  !A/D control status register
       IADBUF = IADR + 2
                                  !A/D buffer register
       IADVEC = "400
                                  !A/D done interrupt vector
       IADERR = IADVEC + 4
                                  !A/D error interrupt vector
       IBELL = "7
                                   !bell tone
       IZERO = "4000
                                   !zero value for D/A converter
С
С
    ---Protect System from crash
С
       IDLIST(1) = IKSR
                                   !keyboard status
       IDLIST(2) = IPEEK(IKSR)
       IDLIST(3) = IPSR
                                   !printer status
       IDLIST(4) = IPEEK(IPSR)
       IDLIST(5) = IDAR
                                   !zero D/A channel 0
       IDLIST(6) = IZERO
       IDLIST(7) = IDAR + 2
                                   !zero D/A channel 1
       IDLIST(8) = IZERO
       IDLIST(9) = IDAR + 4
                                   !zero D/A channel 2
       IDLIST(10) = IZERO
       IDLIST(11) = IDAR + 6
                                   !zero D/A channel 3
       IDLIST(12) = IZERO
       IDLIST(13) = IADR
                                   !A/D status
       IDLIST(14) = 0
       IDLIST(15) = IADVEC
                                   !A/D done interrupt vector
       IDLIST(16) = IPEEK(IADVEC)
       IDLIST(17) = ICSR
                                   !clock status
       IDLIST(18) = IPEEK(ICSR)
       IDLIST(19) = IPSW
                                   !processor status
       IDLIST(20) = IPEEK(IPSW)
       IDLIST(21) = 0
                                   !end of list marker
С
       CALL DEVICE (IDLIST)
С
С
    ---Format Statements
С
  500 FORMAT(/,/,/.2X, 'Discrete Time PD Control',
    + /,2X,'-----')
```

```
95
```

```
510 FORMAT(/,2X,'Enter the INPUT A/D channel: ',$)
  530 FORMAT(/,2X,'Enter the OUTPUT D/A channel: ',$)
  540 FORMAT(12)
  550 FORMAT(/,/,10X,'----MAIN MENU-----',
        /,/,2X,'(I) nitialize the zero reference position',
        /,/,2X,'(G) ain settings for PD control & filters',
     +
         /,/,2X,'(S)tatus of Zero position',
     +
         /,/,2X,'(R)un the active PD controller',
     +
         /,/,2X,'(Q) uit the program',
     +
         /,/,2X,'Choose one of the above. . . ',$)
     +
  560 FORMAT(/,/,2X,'STATUS of Zero Reference. . .',
         /, 2X, '-----')
  570 FORMAT(/,2X,'INPUT zero reference: ',16)
  590 FORMAT(/,/,2X,'Exiting Control Routine. ..',5(/))
  600 FORMAT(/,/,2X,'Zero reference position NOT initialized')
с
С
     ---Calibrate the PAUSE subroutine
с
       IEND = 10000
                                    !set initial value
       IPR = 7
                                    !highest priority interrupt
С
   10 I = INTSET(ICKV, IPR, 1, SETICK) !attach to RTC vector
       IF (I .EQ. 0) GOTO 20
С
       WRITE(7,*)'INTSET error -- RTC vector, CODE = ',I
       GOTO 10
С
   20
      IRATE = "111
                                    !set up clock to generate
       ICOUNT = -16667
                                    !an interrupt after 1 tick
С
       CALL IPOKE (IBPR, ICOUNT)
       CALL IPOKE (ICSR, IRATE)
                                    !start the clock
С
       DO 30 I = 1,10000
                                    !do nothing loop while
   30 CONTINUE
                                    !waiting for one tick
С
       ITCK = IEND + 1
D
       WRITE(7, *) 'ITICK = ', ITCK
С
С
     ---Set zero (0) volts on all D/A outputs
С
       DO 40 I = 0, 6, 2
                                    !
          CALL IPOKE (IDAR+I, IZERO)
                                    !zero out all 4 channels
   40 CONTINUE
                                    !
С
С
     ---Get the INPUT A/D channels
С
       WRITE(7,500)
                                    !initial header
   50 WRITE (7,510)
                                    !get INPUT channel
       READ(5,540,ERR=50) INCH
       IF ((INCH.GT.15).OR.(INCH.LT.0)) GOTO 50
С
   70
       WRITE(7,530)
                                    !get D/A output channel
       READ(5,540,ERR=70) IDACH
       IF ((IDACH.LT.0).OR.(IDACH.GT.3)) GOTO 70
       IDAR = IDAR + IDACH * 2
С
С
     ---Print out the options menu
С
   80 WRITE(7,550)
                                    !print the menu options
       CALL PAUSE (50, ITCK)
                                    !wait for printing to finish
С
     ---Disable the keyboard
С
С
```

```
I = IPEEK(IKSR) .AND. "177477!clear keyboard interrupt
       CALL IPOKE (IKSR, I)
С
С
     ---Wait for keyboard input
с
   90 CALL IPOKE (IPDR, IBELL)
                                    !ring the bell
С
  100 I = IPEEK(IKSR) .AND. "200 !test bit 7 of IKSR
       IF (I .EQ. 0) GOTO 100
С
       IDATA = IPEEK(IKDR)
                                     !a key has been pressed
С
С
     ---Check for a valid character
С
       IF ((IDATA.NE.'I').AND. (IDATA.NE.'G').AND. (IDATA.NE.'S').AND.
           (IDATA.NE.'R').AND.(IDATA.NE.'Q')) GOTO 90
     +
с
       CALL IPOKE (IPDR, IDATA)
                                     !print the proper key
С
С
     ---Enable the keyboard
С
       I = IPEEK(IKSR) .OR. "100
                                   !set bit 6 of IKSR
       CALL IPOKE (IKSR, I)
       WRITE(7,*)' '
С
С
     ---Go to the proper place
С
       IF (IDATA .EQ. 'I') GOTO 110
       IF (IDATA .EQ. 'G') GOTO 120
       IF (IDATA .EQ. 'S') GOTO 130
IF (IDATA .EQ. 'R') GOTO 140
       IF (IDATA .EQ. 'Q') GOTO 150
с
С
     ---INITIALIZE the zero reference position
С
  110 CALL IPOSN
С
       GOTO 80
                                     !return to menu
с
с
     ---GAIN settings for PD control law
с
  120 CALL CGAINS
с
       GOTO 80
                                     !return to menu
С
С
     ---STATUS of Zero position
С
  130 WRITE (7,560)
                                     !initial header
       WRITE(7,570) IZREF
С
       GOTO 80
                                     !return to menu
с
С
     ---RUN the active control routine
С
  140 IF (IZREF .NE. 0) GOTO 145
С
       WRITE(7,600)
                                     !zero references
       GOTO 80
                                     !not initialized
С
  145 CALL CONTRL
с
       GOTO 80
                                    !return to menu
с
С
     ---QUIT the program
```

```
С
  150 WRITE(7,590)
                                  !exit message
       CALL PAUSE (60, ITCK)
       CALL EXIT
С
       END
С
C-----
C----- CGAINS ------
С
       SUBROUTINE CGAINS
С
C This subroutine allows the user to input the control
C gains for the PD control law, and the first order
С
  filter parameters.
С
С
    ---Definition of parameters F()
С
С
       F(1) = PD proportional gain (Kp)
С
       F(2) = PD derivative gain (Kd)
С
      F(3) = posn. filter parameter (alfa)
С
      F(4) = posn. filter parameter (1-alfa)
С
      F(5) = deriv. filter parameter (alfa)
с
      F(6) = deriv. filter parameter (1-alfa)
С
      F(7) = Input scale factor
С
      F(8) = Output scale factor
с
      F(9) = Maximum allowable error
с
      F(10) = Fsp (+ve static friction)
с
       F(11) = Fsn (-ve static friction)
С
       F(12) = Small positive number in control law (Eps)
С
       F(13) = Small positive number for error bound
С
С
    ---Declare Variables
с
       REAL G(13), F(13), SN
С
       COMMON /MISC2/ F,G,IZREF
С
С
     ---Format Statements
с
  100 FORMAT(/,/,2X,'INPUT the PD Control Gains',/,
         2X, '-----')
  110 FORMAT(/,2X,'1) Input PD Controller gains',
        /,2X,'2) Input Position Filter Parameters',
/,2X,'3) Input Derivative Filter Parameters',
     +
        /,2X,'4) Input INPUT/OUTPUT scale factors',
        /,2X,'5) Input maximum allowable error EMAX',
        /,2X,'6) Input Stiction compensation gains',
        /,2X,'7) Check Status of all gains',
         /,2X,'8) EXIT to main routine. .')
  120 FORMAT(/,2X,'Your Selection: ',$)
  130 FORMAT(I1)
  140 FORMAT(/,2X,A14,' = ',$)
  150 FORMAT(/,2X,'Enter # of Samples/Filter Time Const.: ',$)
  160 FORMAT(/,2X,'PD Controller Gains:',/,/,
         5X, 'Kp = ', F10.4, ' [N/m]',/,
    +
        5X, 'Kd = ', F10.4, ' [N-s/m]')
    +
  170 FORMAT(/,2X,A10,' Filter Parameters:',/,/,
         5X, '#Samples/Time Constant = ',F5.1,/,/,
         5X, ' alpha = ', F10.4,/,
         5X, '1-alpha = ', F10.4)
  180 FORMAT(/,2X,'Enter the ',A6,' scale: ',$)
```

```
190 FORMAT(/,2X, 'Calibration Scale Factors:',/,/,
     + 5X, 'Input Scale = ', F10.4, ' [Volt/m]',/,
         5X, 'Output Scale = ',F10.4,' [N/Volt]')
  200 FORMAT(/,2X,'Enter the maximum allowable error [m]: ',$)
  210 FORMAT(/,2X, 'Maximum Allowable Error:',/,/,
         5X, 'Emax = ',F10.4,' [m]')
  220 FORMAT(/,2X,'Stiction Compensation Gains:',/,/,
         5X, 'Fsp = ', F10.4, ' [N]',/,
         5X, 'Fsn = ', F10.4, ' [N]',/,
         5X, 'EpsX = ',F10.4,' [bits]',/,
         5X, 'EPS = ', F10.4, ' [bits]')
С
С
     ---Main routine
С
       WRITE(7,100)
   10 WRITE(7,110)
                                     !display menu options
   20 WRITE(7,120)
       READ(5,130,ERR=20) IOP
       IF ((IOP.LT.1).OR.(IOP.GT.8)) GOTO 20
С
С
     ---Goto the proper place
с
       GOTO (30, 40, 50, 60, 70, 80, 85, 90) IOP
С
С
     ---Input new PD controller gains
С
                                    !proportional gain
   30 WRITE(7,140) 'Kp [N/m] '
       READ(5,*,ERR=30) F(1)
С
   32
       WRITE(7,140) 'Kd [N-s/m]'
                                     !derivative gain
       READ (5, *, ERR=32) F (2)
С
       GOTO 10
                                     !return to CGAINS menu
С
С
     ---Input the filter parameters
С
   40 WRITE(7,150)
                                     !get position filter
       READ(5,*,ERR=40) F(3)
С
       F(3) = 1.0 / (1.0 + F(3))
       F(4) = 1.0 - F(3)
С
       GOTO 10
                                     !return to CGAINS menu
С
С
     ---Input the derivative filter parameters
С
   50 WRITE(7,150)
                                     !get derivative filter
       READ(5,*,ERR=50) F(5)
С
       F(5) = 1.0 / (1.0 + F(5))
       F(6) = 1.0 - F(5)
С
       GOTO 10
                                     !return to CGAINS menu
с
С
     ---Input the INPUT/OUTPUT scale factors
с
      WRITE(7,180) 'Input Cx (V/m]'!get input scale factor
   60
       READ(5,*,ERR=60) F(7)
С
   65
       WRITE(7,180) 'Outpt Cu [N/V]'!get output scale factor
       READ(5, *, ERR=65) F(8)
С
       GOTO 10
                                     !return to CGAINS menu
С
```

```
С
    ---Input the maximum allowable errror EMAX
С
  70 WRITE(7,200)
                                 !get EMAX
      READ(5,*,ERR=70) F(9)
с
      GOTO 10
С
С
    ---Input the stiction compensation gains
с
   80 WRITE(7,140) 'Fsp [N]'
                                 lget Fsp
      READ(5,*,ERR=80) F(10)
С
   81 WRITE(7,140) 'Fsn [N]'
                                 !get Fsn
      READ(5,*,ERR=81) F(11)
С
   82 WRITE(7,140) 'EpsX [bits]' !get Epsilon for X
      READ(5,*,ERR=82) F(12)
С
  83 WRITE(7,140) 'EPS [bits]'
                                 !get precision bound
      READ(5,*,ERR=83) F(13)
С
      GOTO 10
С
С
    ---Check the status of all gains
С
   85 WRITE(7,160) (F(I), I=1,2)
с
      SN = (1.0 / F(3)) - 1.0
      WRITE(7,170) 'Position ', SN, F(3), F(4)
С
      SN = (1.0 / F(5)) - 1.0
      WRITE(7,170) 'Derivative', SN, F(5), F(6)
С
      WRITE(7,190) F(7),F(8)
с
      WRITE(7,210) F(9)
с
      WRITE(7,220) F(10), F(11), F(12), F(13)
с
      GOTO 10
                                 !return to CGAINS menu
С
С
    ---EXIT to the main program
С
   90 RETURN
      END
С
C-
   C----- CONTRL -----
С
      SUBROUTINE CONTRL
с
  This is the control subroutine which sets up the
С
С
  clocked interrupt service control routine, and
С
  starts the control action.
С
      REAL G(13), F(13), FREQ, TICK, PER
      REAL EN, DEN, EO, DEO, VOLT
С
      LOGICAL*1 IDATA, TI1(8), TI2(8)
С
      EXTERNAL RUNLSF
                                 !to use as subroutine arg.
с
```

```
COMMON /ADDR1/ IKSR, IKDR, INCH, ITCK
       COMMON /ADDR2/ IPSR, IPDR, IBPR, IADVEC, IADERR
       COMMON /ADDR3/ ICSR, IADR, IDAR, IADBUF
       COMMON /MISC1/ ICHAN, PER, IFAST
       COMMON /MISC2/ F,G,IZREF
       COMMON /STATE/ EO, DEO, EN, DEN, VOLT
С
С
    ---Format Statements
C
  100 FORMAT(/,/,2X,'RUN the Clocked Control Routine',
    + /,2X,'-----')
  110 FORMAT(/,2X, 'Enter the SAMPLING FREQUENCY (Hz.): ',$)
  120 FORMAT(/,2X,'ERROR. . . [Sampling frequency is too high.]')
  130 FORMAT(/,2X,'ERROR. . .[Sampling frequency is too low.]')
  140 FORMAT(/,2X,'ERROR. . .[Sampling rate TOO HIGH for system.]')
  150 FORMAT(/,2X, 'Actual sampling freq. = ',E11.4, ' Hz.')
  160 FORMAT(/,2X,'Actual sample period = ',E11.4,' Sec.',/,/)
  170 FORMAT(/,2X, 'Press ANY KEY to STOP active control. . .')
  180 FORMAT(/,2X, 'Are you ready to begin control [Y,N]. . . ',$)
  185 FORMAT(/,2X,'1) PD control only',
             /,2X,'2) PD w/Stiction compensation')
  186 FORMAT(/,2X, 'Choose one [1,2]: ',$)
  187
      FORMAT(I1)
  190 FORMAT('+', 5X, 'Error: ', F10.4, 5X, 'Output: ', F10.4)
  220 FORMAT('+',2X,'Turn OFF the Line Time Clock. . .')
  230 FORMAT(/,2X,'ERROR. . .[System too far from origin]')
  240 FORMAT(/,2X,'Is system near the origin [Y,N]. . . ',$)
С
с
     ---Initialize Variables
с
       IBELL = "7
       IPR = 7
                                    !highest priority interrupt
       IFAST = 0
                                    !reset too-high flag
       FMAX = 250.0
                                    !set max. sampling frequency
       VOLT = 0
                                    !initial voltage
С
С
     ---Convert real gains to controller gains
С
       G(1) = F(1) / (F(8) * F(7)) !Kp \setminus
       G(2) = F(2) / (F(8) * F(7)) !Kd / PD controller gains
С
       G(3) = F(3)
                                    !position
       G(4) = F(4)
                                    !filter
С
       G(5) = F(5)
                                    !derivative
       G(6) = F(6)
                                    Ifilter
С
       G(9) = F(9) * F(7) * 204.8 !EMAX
       G(9) = ABS(G(9))
C
       G(10) = -(F(10) / F(1)) * F(7) * 204.8
       G(11) = -(F(11) / F(1)) * F(7) * 204.8
       G(12) = ABS(F(12))
С
       G(10) = G(10) - G(12)
       G(11) = G(11) + G(12)
С
       G(13) = ABS(F(13))
С
С
     ---Set up clocked ISR
С
       WRITE(7,100)
                                   !intial header
С
       WRITE(7,185)
```

```
5 WRITE(7,186)
                                    !choose desired control alg.
       READ(5,187,ERR=5) IOP
       IF ((IOP.NE.1).AND.(IOP.NE.2)) GOTO 5
С
       GOTO (10,15) IOP
С
  10 I = INTSET(IADVEC, IPR, 1, RUNLSF) ! attach to A/D done vector
       IF (I .EQ. 0) GOTO 20
С
       WRITE(7,*) 'INTSET error -- A/D Vector, CODE = ',I
       GOTO 10
С
  15 I = INTSET(IADVEC, IPR, 2, RUNLSF) ! attach to A/D done vector
       IF (I .EQ. 0) GOTO 20
С
       WRITE(7,*) 'INTSET error -- A/D Vector, CODE = ',I
       GOTO 15
С
   20 I = INTSET(IADERR, IPR, 3, RUNLSF) ! attach to A/D error vector
       IF (I .EQ. 0) GOTO 30
С
       WRITE(7,*) 'INTSET error -- A/D Error Vector, CODE = ',I
       GOTO 20
С
С
     ---Input the sampling frequency
С
   30 WRITE(7,110)
                                    !get the sampling frequency
       READ(5, *, ERR=30) FREQ
С
       IF ((FREQ.LE.FMAX).AND.(FREQ.GT.0.)) GOTO 35
       WRITE(7,120)
                                     !sampling frequency too high
       GOTO 30
С
     ---Calculate the best base clock rate
С
С
   35 IR = 1
                                    !start at highest clock rate
с
   40 TICK = (10.0**(7-IR))/FREQ
с
       IF (TICK .LT. 32767.) GOTO 45! integer out of range
       IR = IR + 1
                                    !next lower base frequency
       IF (IR .LE. 7) GOTO 40
С
       WRITE(7,130)
                                    !sampling frequency too low
       GOTO 30
С
     ---Calculate the ticks for IBPR
С
С
   45 ITICK = IFIX(-1.0*(TICK+0.5))!nearest integer
С
С
     ---Calculate the actual sampling frequency and period
С
       FREQ = (10.0**(7-IR))/FLOAT(-ITICK)
С
       PER = 1.0 / FREQ
       G(5) = G(5) / PER
С
       WRITE(7,150) FREQ
       WRITE(7,160) PER
С
С
     ---Set up the clock status and A/D status registers
С
   50 IRATE = (IR * 8) + 3
       ICHAN = "40140 + INCH * (2**8)!set up input sample
```

```
С
С
    ---Make sure the line time clock is turned off
С
                                   !get the first time
   55 CALL TIME(TI1)
       CALL PAUSE(60, ITCK)
                                   !wait for 1 second
       CALL TIME(TI2)
                                   !get the second time
С
       IF ((TI1(8).EQ.TI2(8)).AND.(TI1(7).EQ.TI2(7))) GOTO 60
С
       CALL IPOKE (IPDR, IBELL)
       WRITE(7,220)
                                    imessage to user
       GOTO 55
С
С
     ---Ready to begin active control
С
   60 WRITE(7,170)
                                   !user instructions
       WRITE(7,180)
                                   !stop message
       CALL PAUSE (40, ITCK)
                                   !wait for terminal
С
С
     ---Disable keyboard and wait for input [Y,N]
С
       I = IPEEK(IKSR) .AND. "177677 !reset bit 6 of IKSR
       CALL IPOKE(IKSR, I)
С
   65 I = IPEEK(IKSR) .AND. "200
                                         !test bit 7 of IKSR
       IF (I .EQ. 0) GOTO 65
С
       IDATA = IPEEK(IKDR) !a key has been pressed
С
       IF ((IDATA.EQ.'Y').OR.(IDATA.EQ.'N')) CALL IPOKE(IPDR, IDATA)
       IF (IDATA .EQ. 'Y') GOTO 70
       IF (IDATA .EQ. 'N') GOTO 85
С
       CALL IPOKE (IPDR, IBELL)
       GOTO 65
С
С
    ---Disable the printer
С
   70 I = IPEEK(IPSR) .AND. "177677!reset bit 6 of IPSR
       CALL IPOKE (IPSR, I)
С
С
     ---Set up initial conditions for the system
С
   72 CALL SAMPLE(INCH, INVAL, IFLAG) !get initial position
с
       EN = (IZREF - INVAL) !initial error
       DEN = 0.0
С
     ---Check for initial state too far from origin
С
С
       IF (ABS(EN) .LE. G(9)) GOTO 74
С
       IDATA = IPEEK(IKDR)
                                   !reset the keyboard status
С
       I = IPEEK(IPSR) .OR. "100 !set bit 6 of IPSR
       CALL IPOKE (IPSR, I)
С
       WRITE(7,*)' '
       WRITE(7,240)
       GOTO 65
С
     ---Start clock and ISR
С
С
   74 CALL IPOKE (IBPR, ITICK)
```

```
103
```

```
CALL IPOKE (ICSR, IRATE)
                               !start clock ticking
      CALL IPOKE (IADR, ICHAN)
                                 !start A/D conversion
С
    ---Wait for keyboard input or too-fast error
С
С
      I = IPEEK(IPSR) .OR. "100 !set bit 6 of IPSR
      CALL IPOKE (IPSR, I)
С
      WRITE(7,*) ' '
      WRITE(7,*) ' '
      WRITE(7,*) ' '
      WRITE(7,*) ' '
  80 IF (IFAST .NE. 0) GOTO 85
      WRITE(7,190) EN,VOLT
с
      I = IPEEK(IKSR) .AND. "200 !test bit 7 of IKSR
      IF (I .EQ. 0) GOTO 80
С
С
    ---Enable the keyboard and printer
С
  85 I = IPEEK(IKSR) .OR. "100 !set bit 6 of IKSR
      CALL IPOKE (IKSR, I)
С
      I = IPEEK(IPSR) .OR. "100 !set bit 6 of IPSR
      CALL IPOKE (IPSR, I)
с
С
    ---Check for too-fast error
С
      IF (IFAST .EQ. 0) GOTO 95
      IF (IFAST .EQ. 2) GOTO 90
С
      WRITE(7,*)' '
      WRITE(7,140)
                                 !print an error message
      GOTO 95
С
  90 WRITE(7,*)' '
      WRITE(7,230)
                                 !print an error message
С
С
    ---Turn off clock and A/D converter
С
  95 CALL IPOKE (IADR, 0)
      CALL IPOKE(ICSR, 0)
      WRITE(7,*)' '
с
С
    ---Zero output voltage
С
      IZERO = "4000
      CALL IPOKE (IDAR, IZERO)
с
      RETURN
      END
С
C---
         _____
C----- RUNLSF -----
С
      SUBROUTINE RUNLSF(ID)
С
C This is an interrupt service routine to get clocked samples
C of the input signal and compute the PD output control voltage.
С
С
    ---Definition of parameters G()
С
```

С G(1) = proportional gain С G(2) = derivative gainс G(3) = filter parameter (alfa)с G(4) = filter parameter (1-alfa)С G(5) = deriv. filter parameter (alfa) С G(6) = deriv. filter parameter (1-alfa) С REAL G(13), F(13), VOLT, PER, VMAX REAL EO, DEO, EN, DEN С COMMON /ADDR3/ ICSR, IADR, IDAR, IADBUF COMMON /MISC1/ ICHAN, PER, IFAST COMMON /MISC2/ F,G,IZREF COMMON /STATE/ EO, DEO, EN, DEN, VOLT С DATA VMAX/2047.5/ С С ---Select entry: ID = 1 . . . A/D Sample Done [PD control] С ID = 2 . . . A/D Sample Done [PD w/Comp] С ID = 3 . . A/D Sample Error с IF (ID .EQ. 3) GOTO 90 С С <<<<< Calculate PD Errors >>>> С С ---Get the new input sample С EN = G(3) \* (IZREF - IPEEK(IADBUF)) + G(4) \* EOс IF (ABS(EN) .GT. G(9)) GOTO 90!out of range С CALL IPOKE (IADR, ICHAN) !reset A/D status register с ---Calculate derivative of the input error С С DEN = G(5) \* (EN - EO) + G(6) \* DEOс С ---Shift new variables into the old variables С EO = ENDEO = DENС С ---Use these approximated errors for the PD control law С С <<<<< Calculate PD Control Law >>>>> С С ---Calculate the PD control law С IF (ID .EQ. 2) GOTO 10 С VOLT =  $G(1) \star EN + G(2) \star DEN$ GOTO 80 С С ---Compute the compensation force С 10 IF ((-EN.GE.G(10)).AND.(-EN.LE.G(11))) GOTO 20 VOLT = G(1) \* EN + G(2) \* DENGOTO 50 20 IF (-EN .LE. G(13)) GOTO 30 VOLT = -G(1) \* G(11) + G(2) \* DENGOTO 50 30 IF (-EN .GE. -G(13)) GOTO 40 VOLT = -G(1) \* G(10) + G(2) \* DENGOTO 50

```
40 CONTINUE
        VOLT = G(2) * DEN
   50 CONTINUE
С
С
    ---Check for voltage out of range
С
   80 IF (VOLT .LT. -VMAX) VOLT = -VMAX
       IF (VOLT .GT. VMAX) VOLT = VMAX
С
       IVOLT = IFIX(VOLT + 2048.0)
С
       CALL IPOKE (IDAR, IVOLT)
                                  !apply the voltage
С
       RETURN
С
С
    ---A/D Sample error
С
   90 CALL IPOKE (IADR, 0)
                                  !turn off A/D
      CALL IPOKE(ICSR, 0)
                                  !turn off RTC
С
      IFAST = 1
                                  !set the too-fast flag
С
       IF (ABS(EN) .GT. G(9)) IFAST = 2!out of range
С
       RETURN
       END
```

## **B.4 Common Control Modules**

The following subroutines are common to both the PID and PD control programs.

These are IPOSN, PAUSE, SETICK, and SAMPLE.

```
С
C--
      ------
C----- IPOSN ------
С
      SUBROUTINE IPOSN
С
C This subroutine allows the user to define the initial
C zero reference position of the input signal.
С
      REAL G(10), F(10)
С
      LOGICAL*1 IDATA
С
      COMMON /ADDR1/ IKSR, IKDR, INCH, ITCK
      COMMON /MISC2/ F,G,IZREF
с
С
    ---Format Statements
С
 100 FORMAT('+', 5X, 'INPUT Sample: ', I4)
 110 FORMAT(/,/,2X,'INITIALIZE the Reference Position',/,
      2X, '----',/,/,
       2X, 'Move the system to its reference',/,/,
    +
      2X, 'position to define the ZERO.',/,/,
    +
       2X, 'Press ''X'' to define the INPUT reference . . .',/,/,/)
С
    ---Main Routine
С
С
      WRITE(7,110)
                             !print the instructions
```

```
!wait for CRT
      CALL PAUSE (100, ITCK)
С
С
    ---Turn off the keyboard
С
      I = IPEEK(IKSR) .AND. "177477!clear keyboard interrupt
      CALL IPOKE (IKSR, I)
С
С
    ---Start Sampling
С
  10 CALL SAMPLE(INCH, INVAL, IFLAG) !get INPUT A/D sample
      IF (IFLAG .NE. 0) GOTO 20
      IZREF = INVAL
С
      WRITE(7,100) IZREF
С
      CALL PAUSE(7, ITCK)
                              !wait for 7 ticks
С
С
    ---Check for keyboard input
С
      I = IPEEK(IKSR) .AND. "200 !test bit 7 of IKSR
      IF (I .EQ. 0) GOTO 10
С
      IDATA = IPEEK(IKDR)
                               !a key has been pressed
С
      IF (IDATA .EQ. 'X') GOTO 20
      GOTO 10
с
С
    ---Turn on the keyboard
С
  20 I = IPEEK(IKSR) .OR. "100
      CALL IPOKE (IKSR, I)
С
      RETURN
      END
С
C-
    _____
   ----- PAUSE ------
C-
С
      SUBROUTINE PAUSE (NTICKS, ITCK)
С
С
  This subroutine is approximately calibrated to pause
C in ITCK intervals, as specified on input. There are
C 60 ticks in one second.
С
      DO 20 I = 1,NTICKS
С
        DO 10 J = 1, ITCK
                               !one tick per outer
        CONTINUE
  10
                               !do loop step on (I)
С
  20 CONTINUE
С
      RETURN
      END
с
C-
                                      C----- SETICK -----
С
      SUBROUTINE SETICK(ID)
С
C This is an interrupt service routine that is called
```

```
107
```

```
C when the real time programmable clock has generated
C an interrupt after one tick (1/60 sec.).
С
      COMMON /TIME/ I, IEND
С
      IF (ID .EQ. 1) GOTO 10
                                !check for entry error
С
      WRITE(7,*) 'SETICK entry error, ID = ', ID
      CALL EXIT
С
   10 IEND = I
      I = 10000
                                !reset do loop counter
с
      RETURN
      END
;
;-----
     ----- SAMPLE ------
;
    .TITLE SAMPLE
;
; This is a FORTRAN compatible subroutine called with
;
    CALL SAMPLE (ICHAN, IVAL, IFLG)
;
; This subroutine gets a sample from the specified A/D converter
; channel ICHAN. The data read from the A/D is passed back in 'IVAL'.
; The flag IFLG is returned to the main program with the status:
                   0 = NO ERRORS DETECTED
           1
;
   IFLG = -1 = CHANNEL NUMBER OUT OF RANGE (0-15)
;
                   2 = A/D SAMPLING ERROR
           1
;
;
    .GLOBL SAMPLE
                           ;This makes SAMPLE a global symbol
;
    LC=.
                          ;Find current assembly location
    .=1000+LC
                          ;Start assembling at location 1000
;
    CSR=170400
                          ;Define A/D status register address
    DBR=CSR+2
                          ;Define A/D buffer register address
SAMPLE:
   R5 points to the first address of a table of arguments
;
    @(R5) is the first data (= # of arguments passed)
;
    @2(R5) is the first argument passed
;
;
    MOV
            02(R5), ICHAN
                          ;Put first argument in ICHAN
                           ;Put first argument in TEMP also
    MOV
            02(R5), TEMP
            #177760, ICHAN ;Clear bits 4 to 15
    BIC
                          ;Clear the FLG
    CLR
            FLG
            TEMP, ICHAN
                          ;Check for ICHAN > 15
    CMP
                          ;Branch to ERR1 if ICHAN > 15
    BNE
            ERR1
START: SWAB ICHAN
                          ;Swap high byte for low byte
    MOV
            ICHAN, @#CSR ;Set the A/D Control Status Register
            #1,@#CSR
                          ;Start the A/D conversion
    BIS
WAIT: BIT
                          ;Test bit 7 for A/D done
            #200,@#CSR
                           ;Wait if not set
    BEQ
            WAIT
;
    MOV
            @#DBR,@4(R5) ;Put sampled data into argument table
            #100000,@#CSR ;Test bit 15 for A/D sampling error
    BIT
                      ;Branch to ERR2 if bit 15 is set
    BNE
            ERR2
            FLG,@6(R5)
                         ;Put flag back into argument table
DONE: MOV
```

```
RTS PC
                         ;Return from subroutine to calling program
;
  ERROR HANDLING ROUTINES
;
ERR1: BIS
          #1,FLG
                   ;Set bit 0 of FLG
   BR
           START
                        ;Go back to start
ERR2: BIS
          $2,FLG
                        ;Set bit 1 of FLG
   BR
           DONE
                        ;Go back to DONE
;
; LOCAL VARIABLES
ICHAN: .WORD 0
TEMP: .WORD 0
FLG: .WORD 0
;
   .END
```

.

References

.

# References

- Amontons, G., 1699, Histoire de l'Academie Royale des Sciences avec les Memoires de Mathematique et de Physique, p. 206.
- Armstrong, B., 1988, "Friction: Experimental Determination, Modeling and Compensation," IEEE International Conference on Robotics and Automation, Vol. 3, pp. 1422-1427.
- Barbashin, E.A., 1970, Introduction to the Theory of Stability, Wolters-Noordhoff Publishing, The Netherlands, pp. 66-68.
- Bowden, F.P., and Tabor, D., 1950, The Friction and Lubrication of Solids, Oxford, Clarendon Press.
- Bowden, F.P., and Tabor, D., 1964, The Friction and Lubrication of Solids Part II, Oxford, Clarendon Press.
- Canudas, C., Astrom, K.J., and Braun, K., 1987, "Adaptive Friction Compensation in DC-Motor Drives," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 6, pp. 681-685.
- Cheok, K.C., Hu, H., and Loh, N.K., 1988, "Modeling and Identification of a Class of Servomechanism Systems With Stick-Slip Friction," *Transactions of the ASME*, Vol. 110, pp. 324-328.
- Coulomb, C.A., 1785, Memoires de Mathematique et de Physique de l'Academie Royale des Sciences, p. 161.
- Craig, J., 1988, Adaptive Control of Mechanical Manipulators, Addison-Wesley Publishing Company Inc.
- Davison, C. St. C., 1957, Wear, Vol. 1, pg. 155.
- Euler, L., 1750, Histoire de l'Academie Royale des Sciences et Belles-Lettres, 1748. Berlin, 1750, p. 122.

- Filippov, A.F., 1964, "Differential Equations with Discontinuous Right-Hand Side," American Mathematical Society Translations, Vol. 42, Series 2, pp. 199-231.
- Gilbart, J.W., and Winston, G.C., 1974, "Adaptive Compensation for an Optical Tracking Telescope," *Automatica*, Vol. 10, pp. 125-131.
- Gogoussis, A., and Donath, M., 1987, "Coulomb Friction Joint and Drive Effects in Robot Mechanisms," *IEEE*, pp. 828-836.
- Hahn, W., 1963, Theory and Application of Lyapunov's Direct Method, Prentice-Hall, Englewood Cliffs, New Jersey, pg. 35.
- Hahn, W., 1967, Stability of Motion, Springer-Verlag, New York Inc., pp. 369-371.
- Halliday, D., and Resnick, R., 1978, *Physics, Parts 1 & 2*, Third Ed., John Wiley & Sons, New York, pp. 97-101.
- Hartog, J.P. Den, 1930, "Forced Vibrations with Combined Coulomb and Viscous Damping," Transactions of the ASME, Vol. 53, pp. 107-115.
- Karnopp, D., 1985, "Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 107, pp. 100-103.
- Kolston, P.J., 1988, "Modeling Mechanical Stick-Slip Friction Using Electrical Circuit Analysis," *Transactions of the ASME*, Vol. 110, pp. 440-443.
- Kubo, T., Anwar, G., and Tomizuka, M., 1986, "Application of Nonlinear Friction Compensation to Robot Arm Control," *IEEE International Conference on Robotics* and Automation, Vol. 2, pp. 722-727.
- McShane, E.J., 1947, Integration, Princeton University Press, Princeton, pp. 188-217.
- Miller, F., 1977, College Physics, 4th Ed., Harcourt Brace Jovanovich, Inc., New York, pg. 88.
- Rabinowicz, E., 1951, "The Nature of the Static and Kinetic Coefficients of Friction," Journal of Applied Physics, Vol. 22, No. 11, November, pp. 1373-1379.

- Rouche, N., Habets, P., and Laloy, M., 1977, Stability Theory by Lyapunov's Direct Method, Springer-Verlag, New York, pp. 345-346.
- Shaw, S., 1986, "On the Dynamic Response of a System with Dry Friction," Journal of Sound and Vibration, Vol. 108(2), pp. 305-325.
- Siljak, D.D., 1969, Nonlinear Systems, The Parameter Analysis and Design, John Wiley & Sons, Inc., New York, pp. 539-544.
- Solncev, Yu. K., 1951, "On the Stability in the Sense of Lyapunov of the Equilibrium Positions of a System of two Differential Equations in the Case of Discontinuous Right-Hand Sides," Moskovski Gos. Universitet, Uchenve Zapiski, Vol. 148, Mat. 4, pp. 144-180.
- Southward, S., 1986, A Modified Luenberger Observer for the Inverted Pendulum, MS Thesis, Michigan State University, East Lansing, Michigan.
- Southward, S., 1989, DIFFEQ User's Guide, Double Precision, VMS Version 2.0.
- Tomlinson, G.R., and Hibbert, J.H., 1979, "Identification of the Dynamic Characteristics of a Structure with Coulomb Friction," *Journal of Sound and Vibration*, Vol. 64, No. 2, pp. 233-242.
- Tou, J., and Schultheiss, P.M., 1953, "Static and Sliding Friction in Feedback Systems," Journal of Applied Physics, Vol. 24, No. 9, pp. 1210-1217.
- Townsend, W.T., and Salisbury, J.K., 1987, "The Effect of Coulomb Friction and Stiction on Force Control," *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 883-889.
- Tustin, A., 1947, "The Effects of Backlash and of Speed-Dependent Friction on the Stability of Closed-Cycle Control Systems," Journal of the Institute of Electrical Engineers, Vol. 94(2A), pp. 143-151.
- Utkin, V.I., 1977, "Variable Structure Systems with Sliding Modes," *IEEE Transactions* on Automatic Control, Vol. AC-22, No. 2, April, pp. 212-222.
- Vidyasagar, M., 1978, Nonlinear Systems Analysis, Prentice-Hall, Englewood Cliffs, New Jersey.

- Vinci, L. da, circa 1452-1519, S.K.M. III, The Forster Manuscripts, Library of the Victoria and Albert Museum, South Kensington, 1923-1930, 72 r.
- Vinci, L. da, circa 1452-1519, S.K.M. II, The Forster Manuscripts, Library of the Victoria and Albert Museum, South Kensington, 1923-1930, 133 r.
- Vinci, L. da, circa 1452-1519, Codex Atlanticus, Ambrosiana Library, Milan, 1894-1904, 198 v.
- Walrath, C.D., 1984, "Adaptive Bearing Friction Compensation Based on Recent Knowledge of Dynamic Friction," *Automatica*, Vol. 20, No. 6, pp. 717-727.
- Yang, S., and Tomizuka, M., 1987, "Adaptive Pulse Width Control for Precise Positioning Under Influence of Stiction and Coulomb Friction," *IEEE*, pp. 188-193.
- Yoshizawa, T., 1966, Stability Theory by Lyapunov's Second Method, The Mathematical Society of Japan, Gakujutsutosho Printing Co., Ltd. Tokyo, pp. 1-10.

28,200 = 24,600+56000 - X - 3,900

This Disentulior

