

DEVELOPMENT OF A TEST PLATFORM FOR
EXPERIMENTAL TESTING OF
BICYCLE MODELS

By

RAYMOND PHILIP CAUDLE Jr.

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1991

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 1993

COPYRIGHT

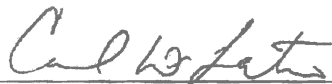
by

Raymond Philip Caudle Jr.

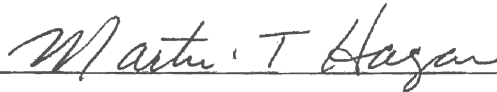
December 1993

DEVELOPMENT OF A TEST PLATFORM FOR
EXPERIMENTAL TESTING OF
BICYCLE MODELS

Thesis Approved :



Thesis Adviser



Dean of the Graduate College

ACKNOWLEDGMENTS

The success of graduate work is, in large, due to the support of the advisor. For his time, energy, encouragement and of course, his patience, I wish to thank my advisor Dr. Carl Latino. During the time I worked on this project, Dr. Martin Hagan spent numerous hours helping me understand the control theory he and his class developed for the bicycle. For his help, I thank Dr. Hagan.

Dr. Latino and Dr. Hagan have both devoted endless time and energy to teaching. I thank them for their dedication to teaching and remind them that their effort does not go unnoticed nor will it be forgotten.

The Oklahoma State University Department of Electrical and Computer Engineering has been very supportive of my education. In particular, Dr. Baker has been very helpful. To Dr. Baker and the department, I say "thank you".

For their efforts in proofreading, grammar and generally making this report readable I thank Mrs. R. Phil Caudle and Ms. Ann Sheffert.

Of course, these acknowledgments would not be complete without saying "thank you" to my friend Tom Crowell, TomNET, for his help with this project.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
Project Overview.....	1
Why Stabilize a Bicycle	2
History of the Project	3
State of the Project	4
Intuitive Explanation	6
Modeling Assumptions.....	11
II. MOTOR SUBSYSTEM.....	13
Chapter Overview	13
Motor Subsystem Overview	14
Physical Mount and Linkage	14
Power Drive	16
Pulse Width Modulation	17
Motor Model Development Overview	18
The Motor Model	19
Summary	25
III. SENSOR SUBSYSTEM	26
Overview of Sensors	26
Bicycle Lean Measurement	27
Inclinometer Hardware	28
Electrolytic Potentiometer	28
Simple Pendulum	29
Steering Angle Measurement	29
Basic Operation	30

IV. BICYCLE SUBSYSTEM	32
Important Components of the Bicycle Model	32
Relationship of the Bicycle Model to the Bicycle.....	33
Important Forces in the Bicycle Model	33
Overview of the Bicycle Model	33
Modeling Simplifications	35
Modeled Bicycle Geometry	36
Development of the Bicycle Equation	36
Bicycle Model Testing and Verification	39
Summary	40
V. MICROPROCESSOR	41
Onboard Microprocessor	41
Overview	41
Microprocessor Hardware	42
Microprocessor Software	43
Execution Rates	46
Software Testing	47
Summary	48
VI. CONTROL SYSTEM	49
Controls Method	49
Overview of Control System	50
Control System Simulation	52
Estimator	54
VII. RESULTS	56
Overview of Accomplishments	56
System Integration	57
Improved System Reliability	57
System Verification	58
General Recommendations	58
Improved System Results	60
REFERENCES	61

APPENDICES

A. MOTOR RESPONSE TEST DATA	62
B. MOTOR POWER DRIVE HARDWARE	74
C. MOTOR MOUNT	77
D. INCLINOMETER RESPONSE DATA (ELECT. POT.)	84
E. INCLINOMETER RESPONSE DATA (PENDULUM).....	91
F. ELECTROLYTIC POTENTIOMETER HARDWARE	100
G. SIMPLE PENDULUM HARDWARE	103
H. DERIVATION OF MODEL EQUATIONS	108
I. SIMULATION BLOCK DIAGRAMS FROM MATRIX _x	119
J. STATE SPACE MATRICES AND FINAL CONSTANTS	129
K. MATRIX _x SIMULATION RESULTS	133
L. BICYCLE TEST PLATFORM RESULTS	142
M. MAIN PROGRAM LISTING	152
N. MAIN INTERFACE HARDWARE	167

LIST OF FIGURES

Figure		Page
CHAPTER I		
1.1	Bicycle Test Platform	1
1.2	Important Forces	7
1.3	Bicycle Test System	8
1.4	The Bicycle as an Inverted Pendulum	10
1.5	The Modeled Bicycle	12
CHAPTER II		
2.1	The Motor Subsystem	13
2.2	Motor Hardware	16
2.3	Motor Switching	17
2.4	Actual Motor Response (Position)	20
2.5	Actual Motor Response (Velocity)	21
2.6	Motor Model	22
2.7	Actual vs. Simulated Motor Position	23
2.8	Actual vs. Simulated Motor Velocity	24
CHAPTER III		
3.1	Ideal Shaft Encoder Signals	30
3.2	Anomalous Shaft Encoder Signals	31

CHAPTER IV

4.1	The Bicycle System Model	32
4.2	Important Bicycle Forces	34
4.3	Front Wheel Geometry	35
4.4	Definition of Bicycle Steering Angle	37

CHAPTER V

5.1	Connection of the Onboard Computer	42
5.2	Bicycle Test Platform Software	44
5.3	One Iteration of the Bicycle Program	45

CHAPTER VI

6.1	Dual System Simulation Block Diagram	53
-----	--	----

CHAPTER I
INTRODUCTION
Project Overview

The goal of this project was to demonstrate vertical stabilization and develop a better understanding of the bicycle system using only steering control. In pursuit of this goal an attempt was made at replacing the human rider by a microprocessor, sensors and steering motor. Using the data received from the sensors the microprocessor calculates the appropriate signal to send to the steering motor. Because of space and testing constraints, the bicycle is tested on a motor driven roller system in an indoor laboratory. Together these components shown in Figure 1.1 function as a test platform which allows direct testing and demonstration of dynamic bicycle models. Figure 1.1 graphically depicts the Bicycle Platform.

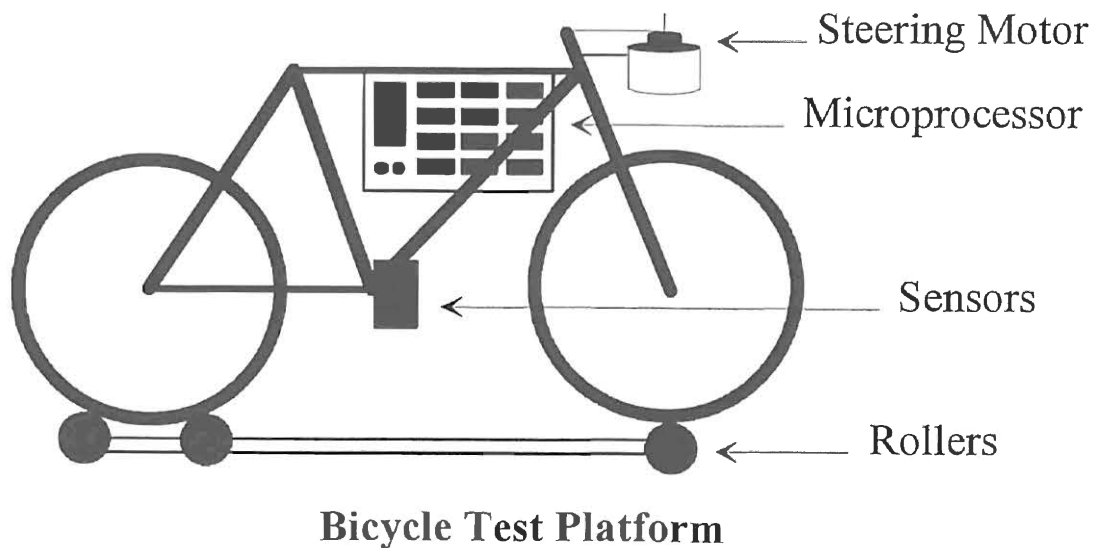


Figure 1.1

The microprocessor, although only one piece of the total bicycle system, has the complex task of replacing the experiential knowledge of the human rider. Numerous artificial intelligence methods ranging from expert systems to fuzzy logic could be employed to address the controller problem. The controls method of State Variable feedback has been chosen however, and artificial intelligence methods will not be considered. State variable feedback is generally more demanding from a mathematical perspective than artificial intelligence methods. It requires a set of dynamics equations governing the dominate physical aspects of the system. This rigor is appropriate in the pursuit of a more lucid understanding of the bicycle. The required equations which have formed the theoretical basis of this work have been previously developed by Dr. Martin Hagan and his Oklahoma State University graduate controls classes.

Why Stabilize a Bicycle

A desire for the quantification of the governing dynamic equations and a method of determining the actual lean angle of a bicycle was requested by Schwinn Bicycle Corp. in the spring of 1987. The object was to examine and quantify the important factors affecting the stability of a bicycle through the development of a test platform so that improvements in maneuverability and efficiency could be made. Current testing procedures rely heavily on qualitative information from expert bicycle riders. While this information is invaluable in the development of bicycles, the addition of the governing dynamics and important geometrical dimensions in a quantitative manner could further assist in the design of new bicycles.

The Bicycle Test Platform is also valuable as a teaching tool where students may develop control systems and can actually see the results in real time. The integrated nature of this project allows model development, simulation, direct testing and quantification of modeling errors. Further, by direct observation of system response,

students are provided checks against modeling mistakes. Finally, this physical extension of the mathematical simulation affords the student the opportunity of seeing what a pure mathematical simulation cannot provide: direct observation of system performance.

History of the Project

The Bicycle Test Platform was started as a senior design project under the Oklahoma State University Electrical and Computer Engineering Department in the spring of 1987. The first team assigned to this project was largely responsible for designing the original hardware set. As with most projects, the bicycle problem seemed simple at first, and its similarities to the inverted pendulum problem only further served to mask many of the complexities of the problem. As the controls model developed, simulations revealed inadequacies in the first hardware implementation. The motor could not tolerate the voltages necessary to stabilize the bike and the chosen microprocessor did not possess the required computational power.

Beginning in the spring of 1989, several teams were assigned to correct the existing problems. Hence, the hardware was improved; the microprocessor was upgraded; software was written, and Dr. Hagan's second graduate controls class completed its study of the bicycle model. The assigned teams completing this work had only a single semester (three months) to study the problem and then design and implement improvements. Each team required a significant amount of start up time to learn and to understand the problem only to run short of time and rush through the implementation of their improvements. This tended to create a continuity problem for the Bicycle Test Platform in that some improvements did not interface well with the existing hardware. By the fall of 1990, the Bicycle Test Platform had a number of partially working subsystems but still lacked system integration.

The primary requirement of the Bicycle Test System in the fall of 1990 was the verification and integration of the system. The following work has been in the direction of producing a reliable working test platform. This was accomplished by testing and verifying each subsystem of the bicycle, then making necessary corrections and adjustments.

State of the Project

A brief discussion of the state of the Bicycle Test System as of fall 1990 and the verification process used to isolate the major problems will help in understanding the test platform. For verification purposes, it is helpful to view this system as having two major components. The first component is a theoretical model which can be simulated on a workstation. The second component is an experimental system composed of all hardware, software and sensors necessary to physically operate a bicycle (referred to as the bicycle test platform). Each component may be used to verify the correct operation of the other. Verification tests may be performed one subsystem at a time or collections of subsystems may be tested so as to isolate any inconsistencies. At the beginning of this current work, it was assumed that the theoretical model was correct based on simulations and the reasonableness of the results the model provided. Thus, discrepancies between the theoretical model and experimental results would generally be assumed to be problems in the test platform. This testing procedure and assumption was used to verify correct operation of the software implementation of the model on the Bicycle Test Platform's microprocessor. Once the Test Platform's model was verified, the next step was to verify the remaining hardware, sensors and drive motor.

Prior to this work control models could be tested on the Test Platform but considerable maintenance and repair were required before and after each test. Experimental test results from the Test Platform were unreliable because bicycle sensors

provided the onboard microprocessor with inconsistent data. Consequently, experimental testing of the theoretical model had progressed as far as possible until hardware improvements were made.

As previously acknowledged, the two hardware subsystems of primary concern were the motor drive system and the sensors. Improvements of the motor hardware were mainly centered around constructing a more reliable power drive system with a faster response time. The existing drive hardware functioned, but it exhibited a sluggish response and lacked the required durability. The existing design seemed reasonable, so modifications were made. Improving the existing design produced a reliable motor drive system.

The second hardware system to be scrutinized was that of the sensors. In this case, the emphasis was placed on accuracy because estimates of unmeasured variables were calculated from these measurements. Sensor noise needed to be minimized and quantified so that appropriate measures could be taken in the estimation of other variables. In both cases, once the hardware modifications were completed, step response tests were conducted and theoretical model adjustments were made to account for the physical changes.

Once the hardware problems were addressed, the examination process began to focus on the theoretical model and how well it approximated the experimental bicycle. The estimation of unmeasured variables from step response tests produced reasonable results, yet the response to input noise was largely unknown. New simulations were conducted in which measurement noise was varied to study its effect on variable estimation. The basic conclusions to these tests were to increase the model's tolerance to measurement noise. Maximum limits of input noise were not specifically determined.

One notable difficulty alluded to throughout this section is the mechanical vibration problem. Slight misalignments in the rollers as well as in the roller drive motor

shaft inject a non-random vibration into the Test Platform. Vibration is also generated by a bouncing motion of the bicycle between the front and back rollers. This motion is generally not random and will be referred to as roller bounce. Both of these noise sources are problematic not only because of their effect on the sensors but also because of the constant pounding the hardware must endure. This problem has been only slightly improved by realignment of the motor shaft and the strategic placement of shock absorbing foam. Proper choice of sensors helped minimize the noise on the measurement signal. In regard to the physical deterioration the onboard microprocessor and associated hardware must endure, securely fastened hardware helps to minimize problems but occasional repair of broken connectors seemed unavoidable.

Throughout the testing of the bicycle subsystems, results collected through tests of individual subsystems were compared with results generated by simulation of appropriate models on Matrix_x. Matrix_x is a software package used in the development of controls models. First, individual measurements were verified, then entire systems and combinations of systems were verified. Inconsistencies between "real" and expected results were noted and corrective measures were implemented into the affected hardware or software. Attempts were made to correct for noise in the model where hardware corrections could not be reasonable made.

Intuitive Explanation

A condensed explanation of the important forces affecting the stability of the bicycle will be qualitatively developed. The bicycle control model is primarily concerned with balancing two torques created by gravity and centrifugal force. Figure 1.2(a) depicts the acting forces from a rear view of the bicycle; Figure 1.2(b) shows the same condition from a top-down view. The bicycle is assumed to be leaning and traveling around a circle of a constant radius at a constant velocity.

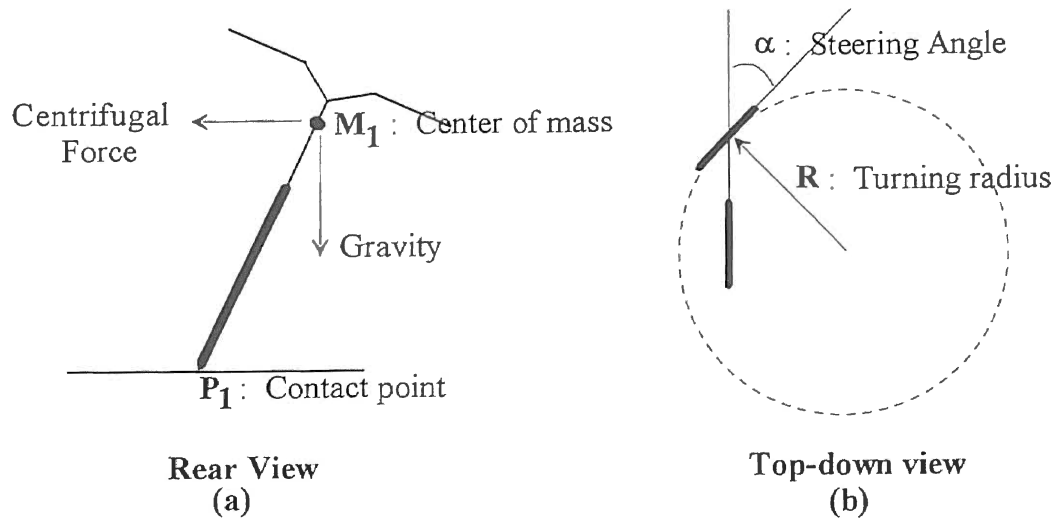


Figure 1.2

A leaning bicycle will fall due to gravity in the absence of a restoring torque. There are two ways to apply the restoring torque, through a rider's redistribution of weight (leaning) or through steering. When a rider leans, he shifts his weight to adjust the lever arm that the force of gravity acts upon, thereby balancing the torques. In this project, however, shifting weight will not be considered. A rider may also generate a restoring torque by steering in the direction of fall (the direction of turn as shown in Figure 1.2). Figure 1.2 provides an example of balanced torques though steering control alone.

Figure 1.2 illustrates a stable leaning state, but such a stable condition is not acceptable on the Test Platform because the bicycle cannot maintain a constant steering angle on the rollers. In fact, there are an infinite number of stable states for the bicycle

of which the vertical bicycle is the only sustainable condition acceptable on the Test Platform. To correct the leaning condition of Figure 1.2 the bicycle controller must steer the bicycle into a tighter turn causing the centrifugal force to increase and pull the bicycle upright. As the bicycle begins to correct, the steering angle is relaxed and is driven to zero (steering straight ahead) as the bicycle lean approaches vertical.

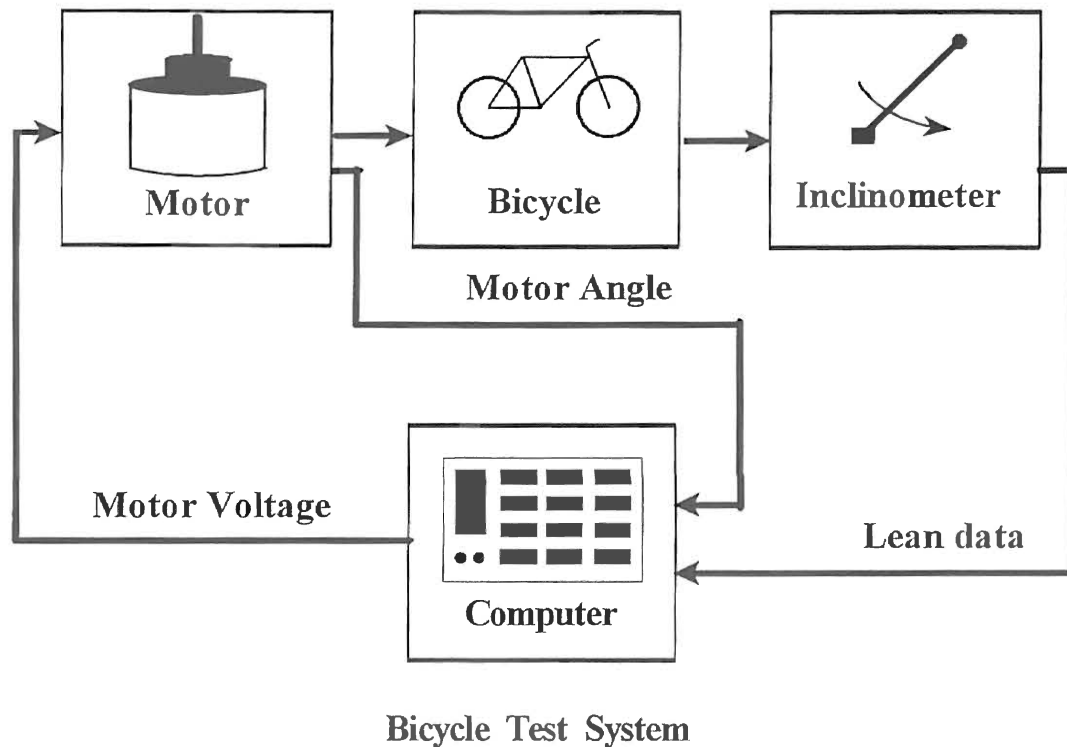


Figure 1.3

With the balancing act between gravity and centrifugal force addressed and the one acceptable stable state discussed, an overview of the entire system would be in order. The Bicycle Test Platform may be viewed as four subsystems: the motor, the microprocessor, sensors and a set of test rollers. Beginning with the inputs to the microprocessor, several sensors, including a motor shaft encoder and inclinometer,

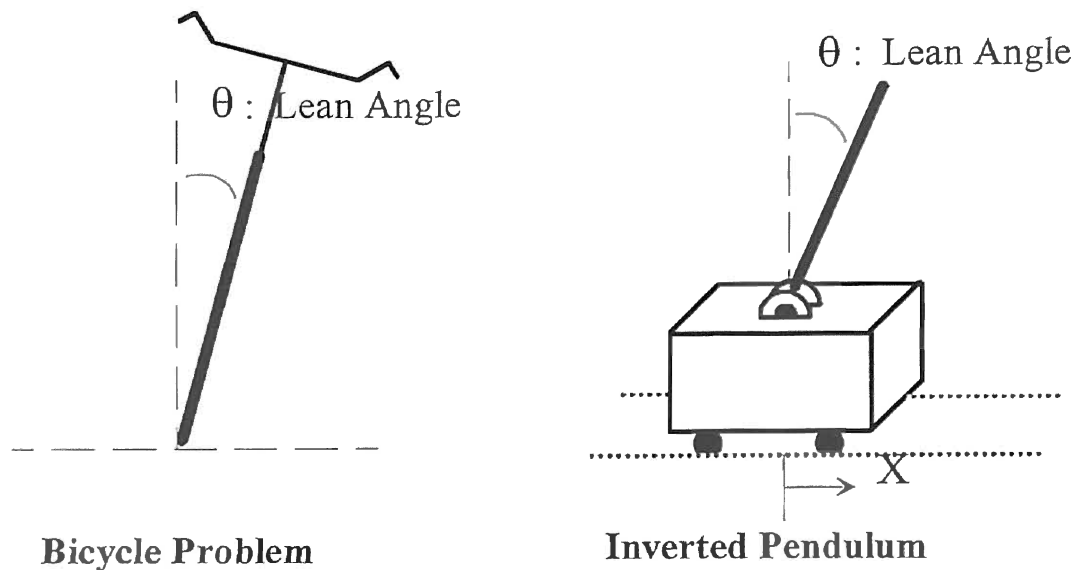
provide information about the bicycle steering angle and lean error angle respectively. These signals are conditioned and collected by the microcomputer mounted on the bicycle frame. From these measurements and knowledge of the bicycle model, a program executed on the microprocessor reconstructs unmeasured states of the bicycle and calculates the proper voltage to send to the motor. The effect of turning the motor is fed back to the microprocessor through the shaft encoder and inclinometer. Figure 1.3 graphically depicts this flow of events.

The microprocessor essentially has two major tasks, reconstruction of the unmeasured states of the bicycle and calculation of the proper motor voltage based on the previously determined states. As earlier indicated state variable feedback was chosen as the method of control. Implementation of this method requires knowledge of the state variables of the bicycle system so that the appropriate motor voltage may be calculated.

In the case of the bicycle controller, six states have been defined and are itemized in Chapter VI. Only the bicycle steering angle and the lean error angle, which is related to the lean angle, are measured. The unmeasured states must be reconstructed through the use of an estimator. The estimator calculates the "missing" state variables by use of the system equations and error adjustments.

The bicycle, when viewed from the front as shown in Figure 1.4, may be compared to the classic Inverted Pendulum problem. A usual demonstration of this problem involves placing a yardstick on end in the palm of the hand and moving the hand so as to balance the yardstick vertically. When it begins to fall, the hand is moved in that direction so as to realign the bottom with the top and minimize the toppling effect of gravitational force. The Inverted Pendulum shown in Figure 1.4 is a simplified version of the problem just described. In this case, the pendulum is hinged and affixed to a cart, thus constraining the pendulum to only fall left or right. As before, when the pendulum begins to fall to the left, the cart is also moved to the left. The comparison of

the bicycle to the inverted pendulum is an over simplification ignoring the complex steering geometries of the bicycle's front wheel. This is a reasonable starting basis for the development of a bicycle model from which a more accurate and increasingly complex model may be developed.



The Bicycle as An Inverted Pendulum

Figure 1.4

There are several important deviations of the bicycle from the inverted pendulum including the manner in which the critical measurement of lean angle (θ , theta) is made. The lean angle of the Inverted Pendulum is not difficult to measure directly due to the pendulum being hinged to the cart. A shaft encoder connected to this pivot point will enable the direct measurement of lean angle. In the case of the bicycle, the lean angle is not measured in the same manner due to the fact that the bicycle's wheels are free to translate across the roller platform. So, an inclinometer, which may be visualized as a

simple pendulum, measures an error angle instead of the lean angle. The error angle is a function of centrifugal accelerations as well as gravity. Bicycle lean angle must then be estimated by the microprocessor given the error angle measurement.

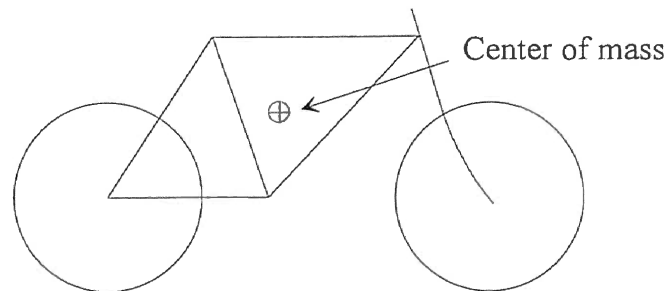
A second departure from the inverted pendulum involves the manner in which the correcting control is applied to the system. The inverted pendulum utilizes a motor to directly move the cart left or right. The bicycle utilizes a motor to turn the front wheel which indirectly affects the bicycle lean angle. The nature of these restoring forces is different; for the inverted pendulum the force is directly applied to the base of the pendulum from the cart linkage. For the bicycle the restoring force is the inertia of the bicycle being thrown away from the center of the circle the bicycle is traversing.

Modeling Assumptions

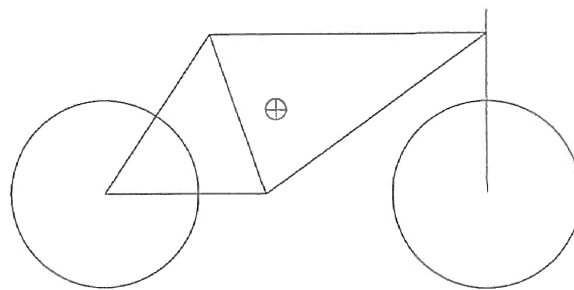
In the interests of producing a simple practical model that could be reasonably implemented, a series of assumptions were made. Previous works, which will be noted where appropriate, form the foundation of some of these assumptions. Other assumptions, made of necessity by the developers of the theoretical model, were tested within the limits of this system. A list of the most basic and important assumptions is provided here. Other more subtle or less important assumptions will be presented when appropriate.

- 1) Simplification of steering wheel geometry
- 2) Point mass acting at center of gravity
- 3) No gyroscopic effect of wheels
- 4) No wheel-roller friction
- 5) Small Angles (for linearized bicycle model)
- 6) Effect of inclinometer mass on bicycle not considered

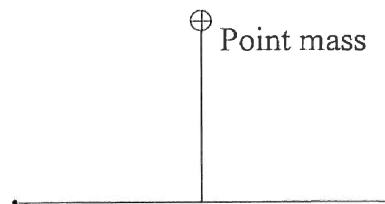
The first two assumptions are geometrical simplifications and were made for modeling purposes; however, important bicycle proportions were included so bicycle geometry was not ignored. The result of these assumptions is illustrated in Figure 1.5. The third assumption was shown to have only negligible effects on bicycle stability. The last three assumptions are discussed in Chapter IV.



a) The Actual Bicycle



b) Simplified Steering Geometry



c) Modeled Bicycle

The Modeled Bicycle

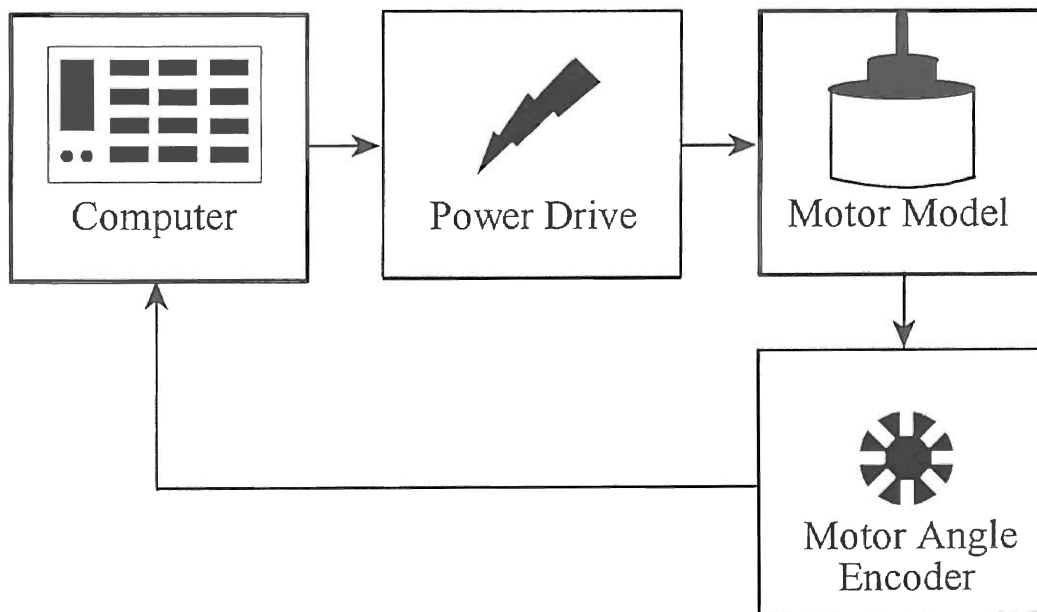
Figure 1.5

CHAPTER II

MOTOR SUBSYSTEM

Chapter Introduction

In the replacement of the human rider by a computer, steering actuation and control must also be accomplished through other means. The test bicycle achieves this by replacing the handlebars with a single DC motor and gear linkage operated by the onboard microprocessor. Four relevant areas of the Motor Subsystem are discussed in this chapter beginning with overview of the motor and how it is operated. The physical motor mount and linkage is explained, followed by a discussion of the power drive and Pulse Width Modulation scheme. Motor model development is explained and the chapter is concluded with a discussion of the verification process used to test the Motor Subsystem.



The Motor Subsystem

Figure 2.1

Motor Subsystem Overview

The Motor Subsystem consists of the four primary components shown in Figure 2.1. The bicycle onboard microprocessor shown as "Computer" in Figure 2.1, contains the Pulse Width Modulation subroutine needed to operate the power electronics module which in turn drives the motor. As the motor rotates, a shaft encoder provides information about the shaft position to the microprocessor. The "Motor Angle Encoder" and "Computer" blocks shown in Figure 2.1 are described in more detail in Chapters III and V respectively.

Correct operation of the Motor Subsystem is verified through comparison of experimental motor tests with Matrix_x simulations of the motor model. These tests are performed in isolation from other parts of the Bicycle Test Platform, so, possible errors in other subsystems will not affect verification of this subsystem.

Physical Mount and Linkage

A motor mount and linkage pair were in place on the bicycle at the beginning of this work. Problems observed in the correct measurement of motor shaft position lead to the discovery of four principal difficulties with the original mount. These problems include:

- 1) motor rotating within the mount
- 2) movement of the mount on the bicycle frame
- 3) slippage between sprocket and linkage belt
- 4) counter circuit errors.

The previous motor mount was constructed of wood and clamped around the motor and bicycle frame. With use, deterioration of the mount allowed the motor to rotate during tests. A similar deterioration problem occurred at the connection point between bicycle and the mount. A third slippage problem was detected between the

drive sprocket and the linkage belt. Motor direction changes and worn belts tended to intensify this problem. The counter circuit problem is discussed in the "Steering Angle Measurement" section of Chapter III but is listed here for completeness.

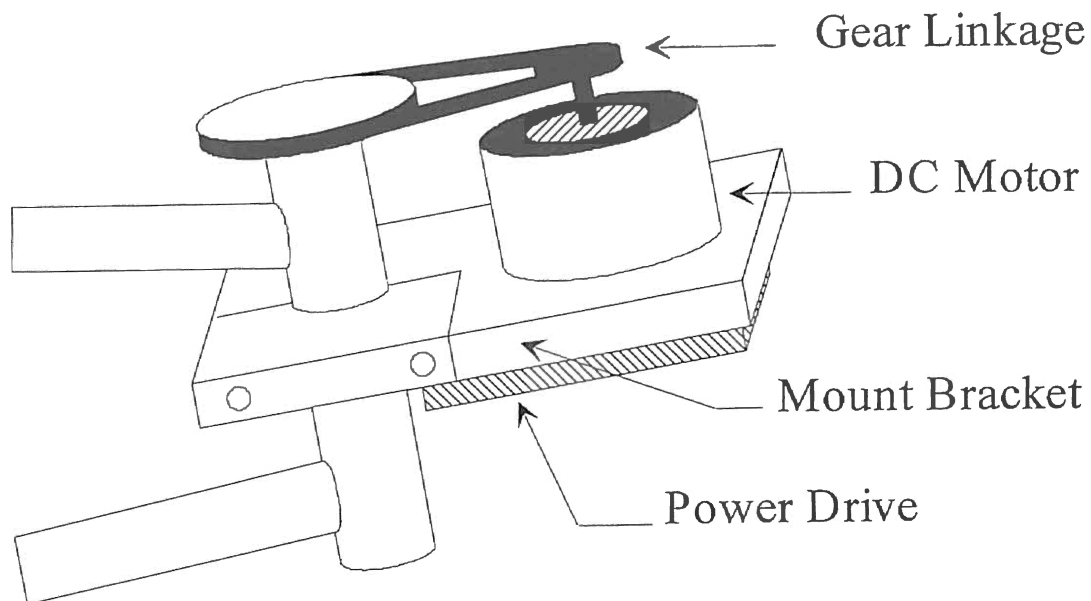
Plots of motor position with respect to time were used to detect these problems. One of these plots is reproduced in Appendix L. The important areas investigated on these plots were the places where the motor reversed directions because this was the condition that placed the greatest stress on the Motor Subsystem. While detection of errors from these plots was not difficult, specific diagnoses of a problem was very challenging. The difficulty in diagnosis was that these errors generally appeared intermittently in the collected data.

The physical connection of the motor to the bicycle was carefully considered because of the significant torques developed by the motor. A second area of concern in the design of the motor mount and linkage was the vibration created by the bicycle bouncing on the roller platform. Any mounting solution needed to address the issues of strength and durability.

The newly designed mount, sketched in Figure 2.2, is a reasonable compromise between complexity and durability. Special attention was given to the bracket's connection to the bicycle so that damage to the frame would be minimized. Due to the fact that the frame tube can be easily crushed or bent, a round clamp hole was cut into the mount bracket. When tightened, the bracket exerts pressure evenly around the tube. The motor position on the mount may be adjusted so that slack in the drive linkage may be reduced. This mount can be removed and transferred to another similar bicycle. The mount bracket design and detailed sizing information are provided in Appendix C.

A chain and sprocket linkage was chosen with a 6:1 gear ratio to increase the torque delivered to the steering shaft and also to improve accuracy of the steering angle (shaft encoder is mounted on the motor). The chain is allowed to slip off the steering

column sprocket after the wheel has rotated 60 degrees, thus limiting damage to the bicycle if the motor should continue to rotate.



Motor Hardware

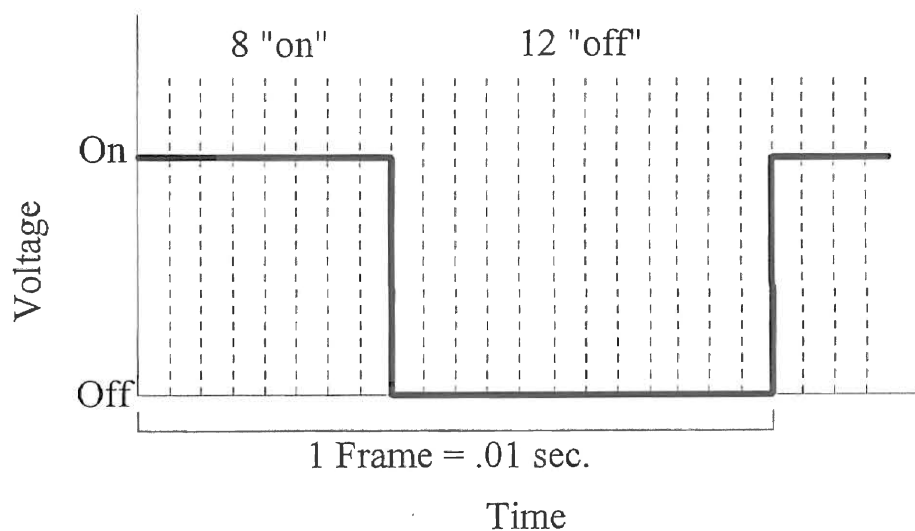
Figure 2.2

Power Drive

A fixed voltage is applied to the motor for a varying amount of time. If this switching scheme is carried out sufficiently faster than the motor response, then the time average of the switched voltage may be assumed constant. The actual switching is accomplished by a transistor bridge attached to the motor mount. Switching of the power electronics is controlled by a subroutine running on the microprocessor. Any noticeable hammering effect created by switching the motor is largely filtered out when the steering wheel and wheel-roller friction load are added to the motor.

Pulse Width Modulation

Motor switching is operated by a Pulse Width Modulation (PWM) subroutine which executes on the Bicycle Test Platform microcomputer. Once a voltage is calculated by the estimation and control portion of the software, it must be converted to a value that the PWM routine can use. Each one-hundredth of a second a new voltage is applied to the motor. These one-hundredth of a second frames are further subdivided into twenty equal divisions during which the motor may be on or off. For example, if forty volts were applied to the motor, it would be switched on during all of these twenty subdivisions.



Motor Switching

Figure 2.3

Application of twenty volts to the motor for one frame would require the motor to be on for the first ten subdivisions and off for the second ten subdivisions. Figure 2.3 graphically shows the switching configuration for a calculated motor voltage of 16 volts. The effective conversion is two volts per subdivision. Voltages between zero and forty

volts may therefore be applied in steps of two volts. The PWM routine is controlled by Routine A which is discussed along with general program timing in Chapter V.

Several major criteria for design of the electronic power driver include overvoltage protection and sizing devices to handle the required motor current. In this case diode over and undervoltage protection is very important because of the inductive properties of the motor. Transistors were chosen as switching devices because of their availability and were sized based on twice the motor rated current. Motor hardware schematics as well as device sizing details are provided in Appendix B.

Conformation of the proper operation of the described hardware was carried out by analyzing motor position with respect to time plots. These motor position tests as previously discussed provided reliable conformation of correct hardware operation. Quantitative tests were conducted on the power drive circuitry by recording current delivered to the motor for both rotational directions under loaded conditions (loaded conditions being the motor driving the steering wheel while on the roller platform). Motor current tests verified that the motor received rated current and that the Pulse Width Modulation subroutine was functioning correctly.

Motor Model Development Overview

The importance of the motor model to this system is twofold. A mathematical representation is extremely helpful in system verification especially when the system is composed of numerous subsystems. Problems in the Test Bicycle's response may be more easily pinpointed by comparison of actual test results to simulation of the model one subsystem at a time. Secondly, the model is important in the development and implementation of a control system. For the test bicycle, each major subsystem has been modeled and then linearized so that linear control laws may be applied. In the interests

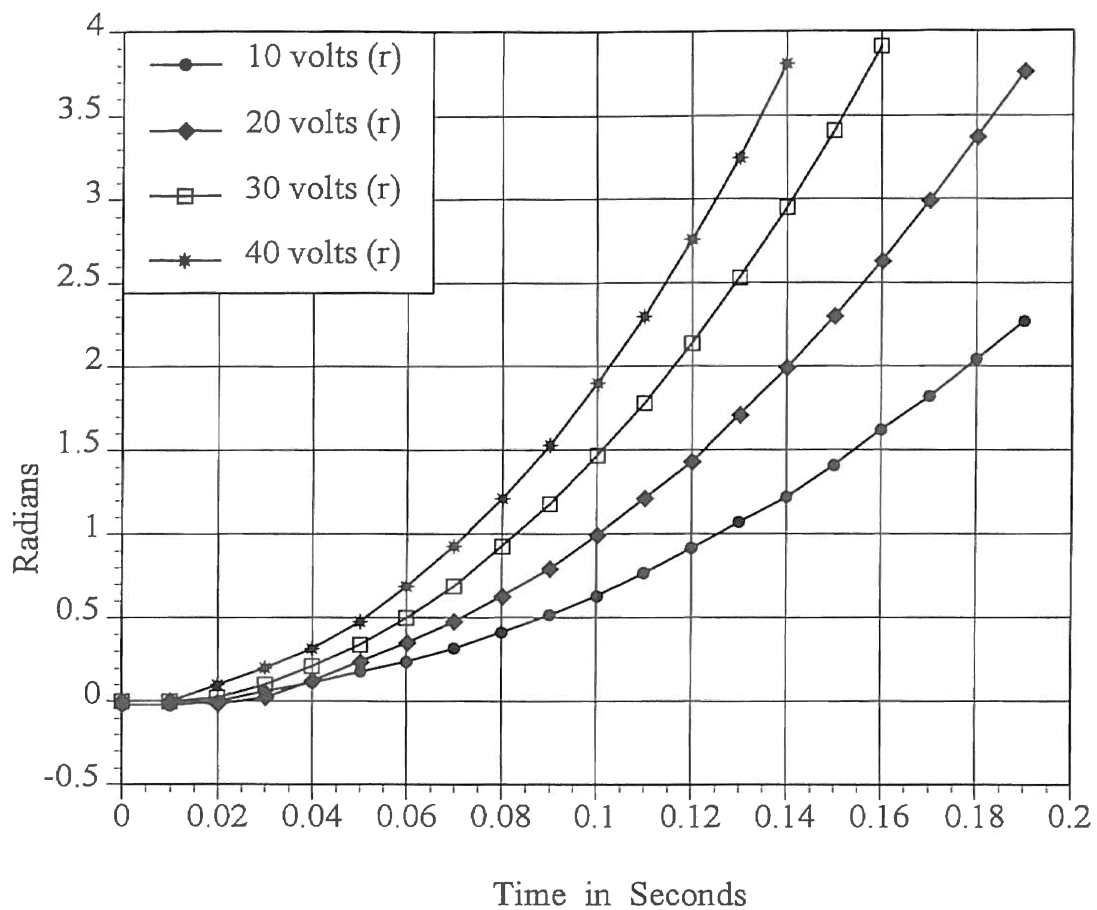
of simplicity and time, assumptions and simplifications have been made and are noted when appropriate. Error estimates have been formulated where appropriate.

The Motor Model

The motor model was developed from a basic second order model and from empirical data collected from the motor. Position verses time plots as well as velocity verses time plots were generated from step response tests. These tests consisted of applying a chosen voltage for a period of time and then reading the motor position. Velocity data was experimentally generated for the motor through interrupt driven timing on the microprocessor and measured position data. These tests were conducted with the motor under load turning both clockwise and counterclockwise. Figures 2.4 and 2.5 show motor position and velocity results respectively for the clockwise motor test at four voltages. These results present a reasonable picture of actual motor response due to the fact that the actual software and hardware were used.

One important nonrealistic effect of these tests is that the rollers were not in motion at the time of the tests. Effectively the wheel was turned while the bicycle was standing still. Due to difficulties in dynamic test repeatability and wear on the hardware, this static testing procedure was chosen as the motor test standard. The effect of this testing procedure was that actual motor test results will nearly always lag behind the simulated motor results. These tests, due to their simplicity and repeatability, are good indications of correct hardware operation and modeling reasonability even though numeric comparisons showed significant errors.

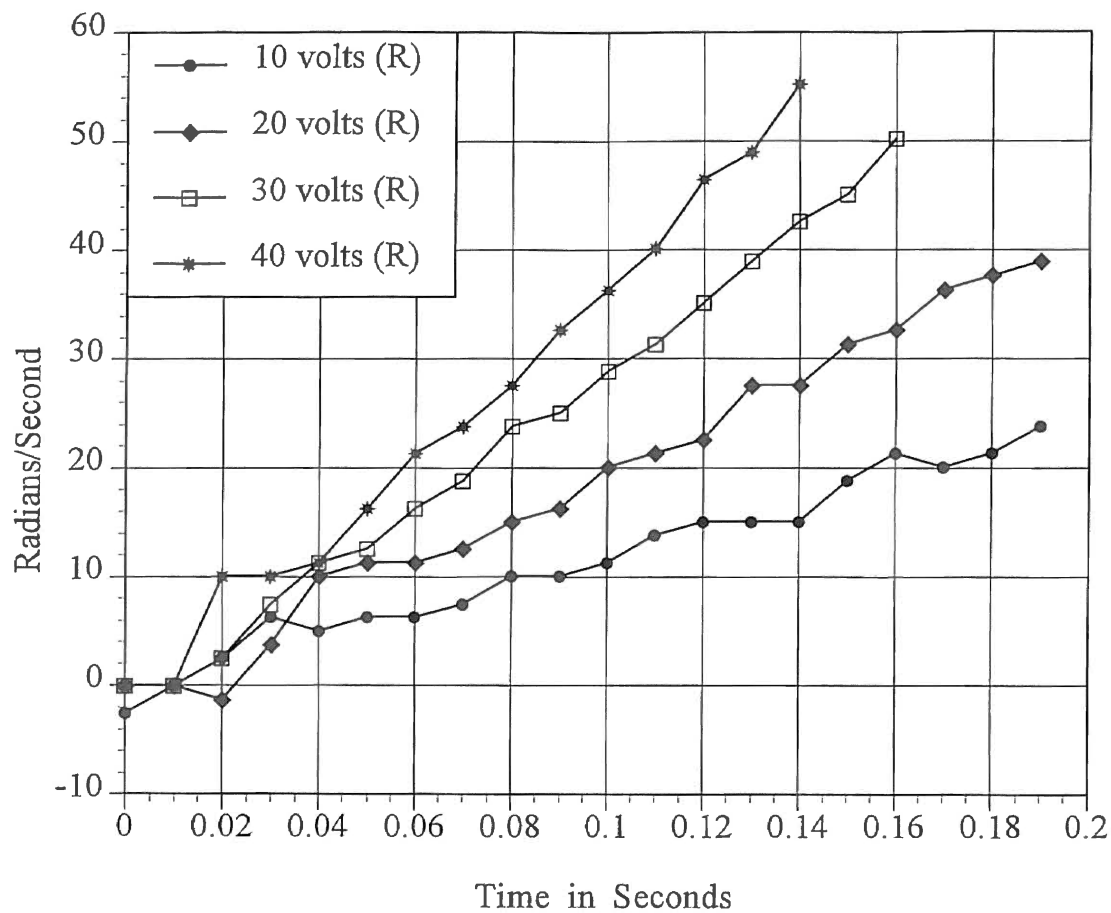
The results shown in Figure 2.5 reveal a significantly over damped system. These tests were performed under load so that the motor did not reach a steady-state velocity during a normal window of operation. These observed damping effects also serve to filter out any remaining surging effect of the Pulse Width Modulation driver.



**Steering Motor Dynamics,
Actual Motor Position Vs. Time,
Turning Right**

Figure 2.4

Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter and were unfiltered. Maximum motor voltage was 40 volts.

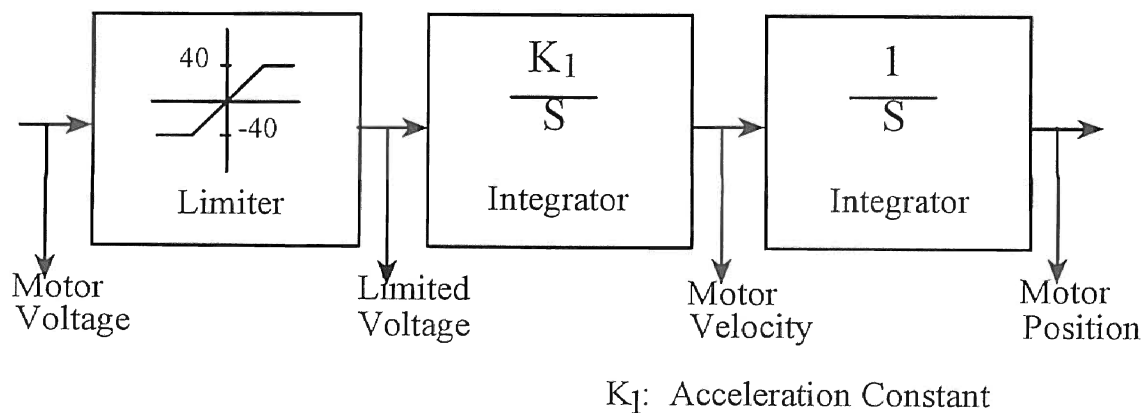


**Steering Motor Dynamics,
Actual Motor Velocity Vs. Time,
Turning Right**

Figure 2.5

Note: This test was conducted using actual motor drive hardware. Data were collected directly from the steering counter with samples .01 seconds apart. Velocity was calculated as the difference between positions divided by .01 seconds.

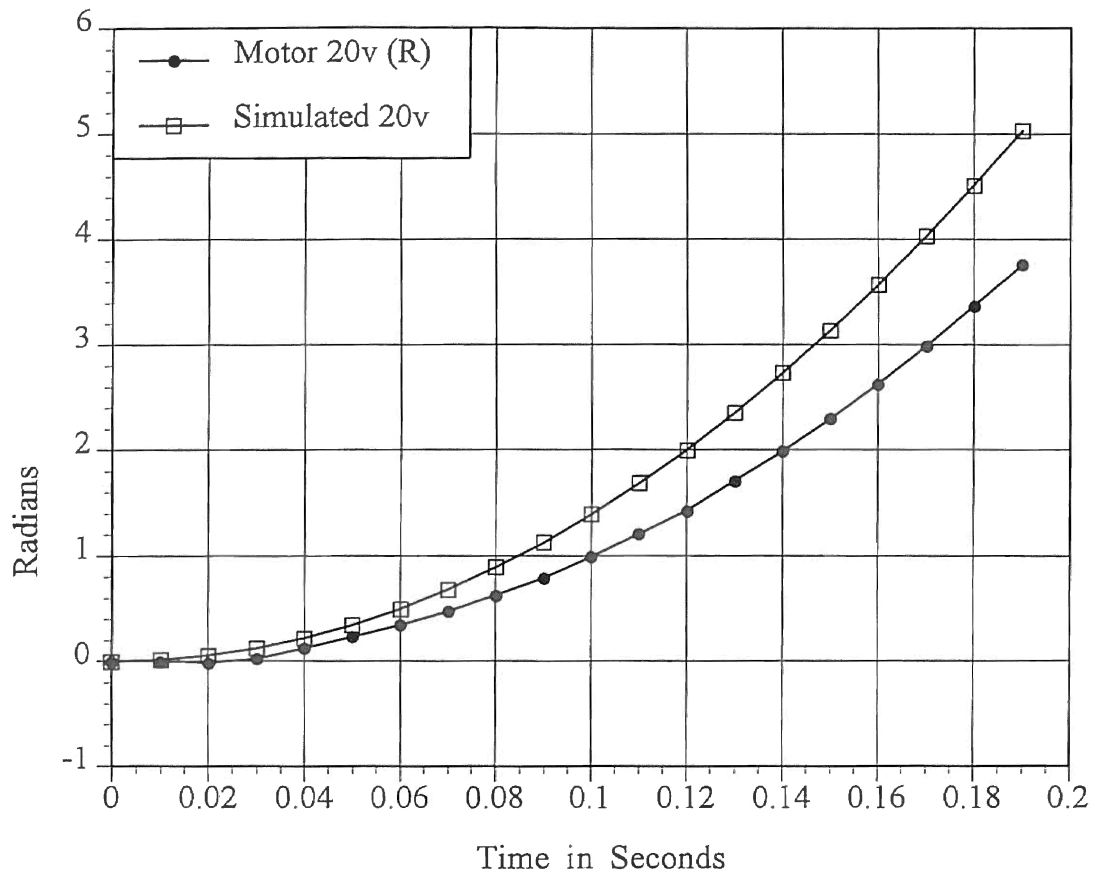
Visual inspection of these curves suggest that the nonlinear effects of the motor may be ignored for the motor model. Based on this observation, it was assumed that performance of the motor would be predicted accurately to the degree required when modeled by a second order integrator and a single gain constant. The back emf, armature constants and rotor inertia, and effects of pulse width modulation are not specifically accounted for in the model. The slope of each of these curves was calculated and averaged to produce a single motor acceleration constant noted as K_1 , in Figure 2.6. This figure also includes a limiter block not previously discussed. This block is necessary in the simulated system because control voltages greater than rated values can be generated by the model.



Motor Model

Figure 2.6

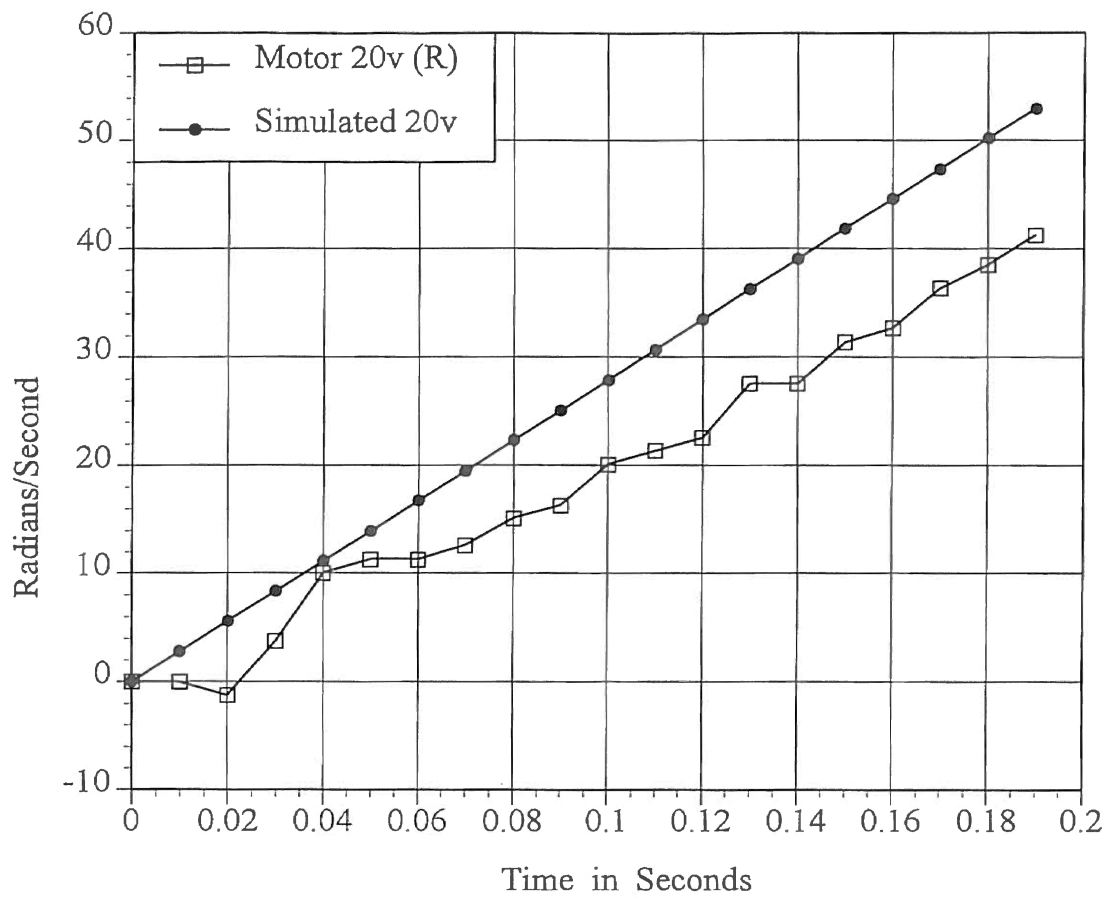
Analysis of the degree of linearity of the curves in Figure 2.5 was not carried out but comparison of the simulated model to the observed results was performed. The controls software package Matrix_x was used to simulate both the position and velocity profiles for a range of input voltages.



**Steering Motor Dynamics,
Actual and Simulated Motor Position Vs. Time,
Turning Right**

Figure 2.7

Note: This plot compares actual loaded motor position to simulated motor position at 20 volts. The Matrix_x model assumes the motor to be loaded.



**Steering Motor Dynamics,
Actual and Simulated Motor Velocity Vs. Time,
Turning Right**

Figure 2.8

Note: This plot compares actual loaded motor velocity to simulated motor velocity at 20 volts. The Matrix_x model assumes the motor to be loaded.


The same range of input voltages was applied to the actual hardware. The comparisons for an input voltage of 20 volts for position and velocity are shown in Figures 2.7 and 2.8 respectively.

As expected, the actual motor response lags behind the simulated motor. These are worst case results considering the load on the actual motor under these test conditions is significantly greater than it would be during a full bicycle test.

@ .10sec	Actual	Simulated	Difference	Percent Error
Position	.993 rad	1.395 rad	.402 rad	29%
Velocity	20.9 rad/s	27.9 rad/s	7.80 rad/s	28%

The important point of these results is that the model follows the actual motor to a close enough degree that the model is reasonable and the hardware may be verified as functioning correctly. Because an average slope was computed from eight tests performed turning in both rotational directions, error will be introduced. For this motor, this same test performed by turning counterclockwise will produce percent errors closer to 16% because the motor used is less efficient turning to the left.

Summary

This chapter examines the significant areas of the Motor Subsystem. The discussion begins with an overview of the motor and the important hardware and software required to operate it. The physical motor mount hardware is explained followed by a discussion of the power electronic drive. The software required to operate the power electronics is briefly discussed in the Pulse Width Modulation section. Motor model development is reviewed next and is supported by the test results that follow. 

Chapter III

Sensor Subsystem

Overview of Sensors

The riderless bicycle must replace the eyes and inner ear of the human with electrical or electromechanical sensors. All the input stimuli received by the human rider cannot be reasonably measured or accounted for by the robot bicycle; only the important variables need be measured. In the case of this test system, which assumes a constant forward velocity, two measurements are critical: bicycle lean angle and steering angle. Their derivatives are also important but it is assumed that these will not be measured but calculated. This test system is unable to directly measure bicycle lean angle. Instead, this angle is indirectly determined through other measurements and must be computed with the help of the onboard microprocessor. The second important measurement of steering angle is accomplished through the use of a shaft encoder. Both of these sensors have related hardware and software to access them, which must be verified. Correct operation of the hardware and software is again verified through comparison of experimental test results with simulations of the corresponding model. For the shaft encoder, the model is trivial so testing is more concerned with hardware verification.

Sensor accuracy is a general consideration. The Bicycle Test Platform measurement accuracy is of the utmost importance due to the requirement that other unmeasured variables must be determined from calculations using the few measured variables. Relatively minor measurement noise, when propagated through the real time estimation algorithms on the onboard microprocessor, can lead to significant controller errors. These problems led to the need to minimize noise and to quantify it. Tests performed on the sensors include noise quantification as well as function verification.

Bicycle Lean Measurement

The inclinometer, an electromechanical sensor, is utilized by the bicycle test platform to measure information about the lean angle of the bicycle. Other types of sensors could be employed to perform this task such as gyros and optical systems, but the inclinometer was chosen and has been theoretically proven to stabilize the bicycle. The sensor used does not directly measure the lean angle of the bicycle but instead measures an error angle which is arithmetically related to the lean angle. It is helpful to view the inclinometer as a sensor that measures accelerations including the acceleration due to gravity and the accelerations generated by centrifugal motion. Thus, at any point in time the inclinometer is measuring the combined effects of two different accelerations. For this reason, data from the inclinometer alone is not enough to determine bicycle lean angle. Special cases do exist such as when the bicycle only steers straight ahead; in this case centrifugal accelerations are zero and so only the effect of gravity is measured (assuming inclinometer dynamics are accounted for).

An electrolytic potentiometer and a simple pendulum were the inclinometers tested in response to practical measurement problems. Electrical and mechanical noise as well as dynamic response were the most significant criterion considered in the evaluation of the inclinometers tested. It followed that roller platform vibration was the most important problem to overcome. The electrolytic potentiometer version of the inclinometer exhibited greater accuracy and better dynamic response than the simple pendulum, but it was unacceptable because of its response to vibration. This sensor was shown to exhibit erratic response and resonance under full test platform experiments which included roller vibration. The simple pendulum version of the inclinometer was considerably more oscillatory and less accurate than the electrolytic potentiometer. The advantage of the simple pendulum was that it was more reliable under the vibration conditions encountered during a full platform test.

Inclinometer Hardware

Hardware configuration for both inclinometers is essentially the same and maybe divided into three pieces: the sensor element, an amplifier and an analog-to-digital converter. The actual sensor element in both cases is a variable resistor which has a resistance proportional to the angle of tilt from some null point. The amplifier adjusts the sensor's output voltage to a level appropriate for input to the analog-to-digital converter. This voltage output from the amplifier is proportional to the angle of tilt of the sensor element where the proportionality constant is modified by adjustment of the gain of the amplifier. Once a conversion is complete the, onboard microprocessor has access to the converted measurement through the microprocessor bus.

Electrolytic Potentiometer

The electrolytic potentiometer is basically a hollow toroidal shaped tube containing three separate copper conductors and an electrolytic solution. The three copper conductors are arranged so that depending upon the angle of tilt varying amounts of the conductors are electrically connected by the electrolytic solution. As the sensor is tilted, the resistance is proportionally varied. The electrolytic inclinometer used on the test platform was supplied with a driver module which excited the element. The driver module used an alternating current to avoid plating problems inside the tube. Specific hardware details of this sensor are provided in Appendix F.

This sensor was supplied with relatively detailed specifications and dynamic test response data that revealed an acceptable response time for the test platform with a minimum of oscillation. These test results could not be exactly reproduced due to lack of specific test setup details. Step response tests performed locally revealed very similar results compared to those received from the company. These results may be seen in Appendix D. Information concerning this sensor's response to a continuous vibrational

noise input was not supplied. The electrolytic inclinometer produced erratic results and tended to demonstrate resonant modes when subjected to noise response tests as part of the bicycle test platform and tests that isolated the sensor. The two tests involved were a background noise test and a dynamic bicycle drop test which involved allowing the bicycle to fall (all control and estimation routines off) while the rollers were in motion. The second test was helpful in presenting a realistic picture of the effect Test Platform vibration had on the sensor. Unfortunately, this test produces only a short time frame of data and is only marginally controlled. The background noise test was intended to quantify electrical distortion caused by the amplifier. The test procedure requires that the bicycle and all its surroundings be motionless. The conclusion reached from these results was that this sensor was not a good choice based on its response to the roller platform vibration.

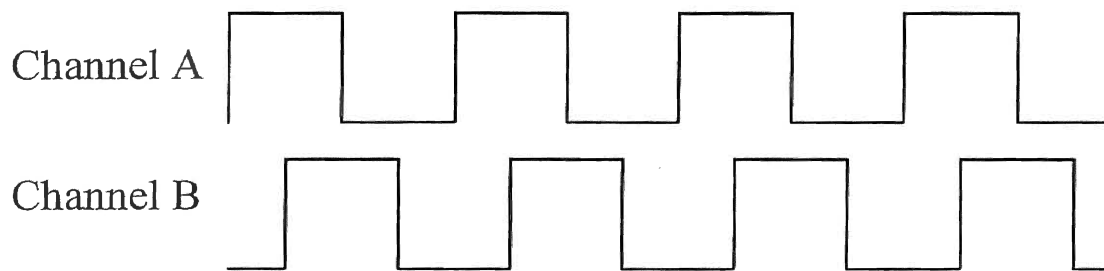
Simple Pendulum

The pendulum inclinometer was an experimental device constructed to study the reaction to mechanical noise. This sensor is a pendulum with an adjustable mass attached to a standard potentiometer that exhibited low friction characteristics. The potentiometer was mounted vertically allowing the pendulum to swing like the pendulum of a clock. Potentiometer resistance was a function of tilt angle and would fit into the existing hardware. The appeal of this simple device, despite its crudeness, was its relative insensitivity to vibration.

Steering Angle Measurement

An optical shaft encoder mounted on the steering motor provides digital information about the position and direction of rotation to a counter circuit which maintains the current motor position. Unlike the inclinometer, the shaft encoder recognizes a discrete number of shaft positions. For this sensor no analog to digital

conversion is required; only a count of the discrete encoder positions passed by the encoder is recorded by the counter. The number recorded by the counter is directly proportional to the steering wheel position and this conversion is made by the microprocessor after reading the count.



Constant Velocity Example

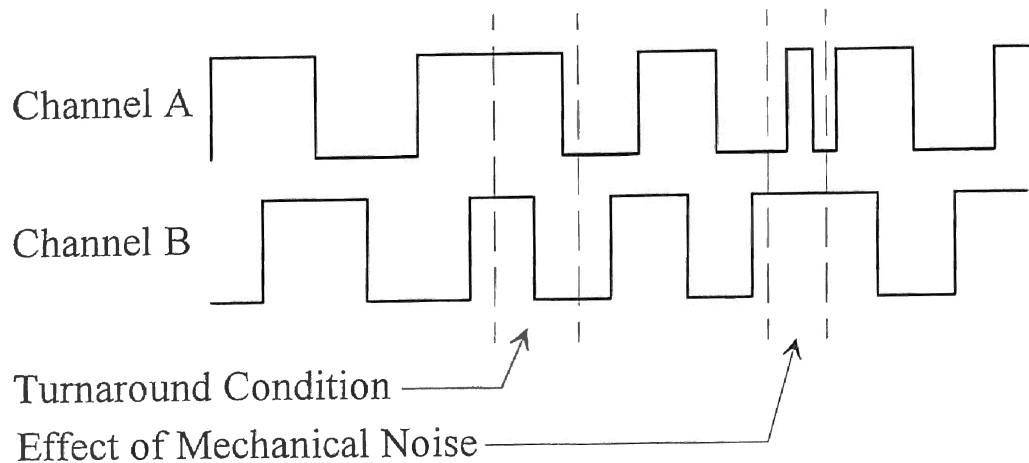
Ideal Shaft Encoder Signals

Figure 3.1

Basic Operation

The shaft encoder outputs two quadrature signals that are designated Signal A and Signal B. The shaft position and direction may be determined from these signals. When the shaft rotates in one direction, the signals A and B are square wave pulse trains with Signal A leading Signal B by 90 electrical degrees. If the encoder rotates in the opposite direction then Signal B will lead Signal A by 90 electrical degrees. Effectively, one signal is chosen to enable the count. The second signal causes the counter to count up or down. Therefore, while the counter is enabled a rising edge causes the counter to

count up and a falling edge causes the counter to count down. Figure 3.1 illustrates the ideal counting condition.



Anomalous Shaft Encoder Signals

Figure 3.2

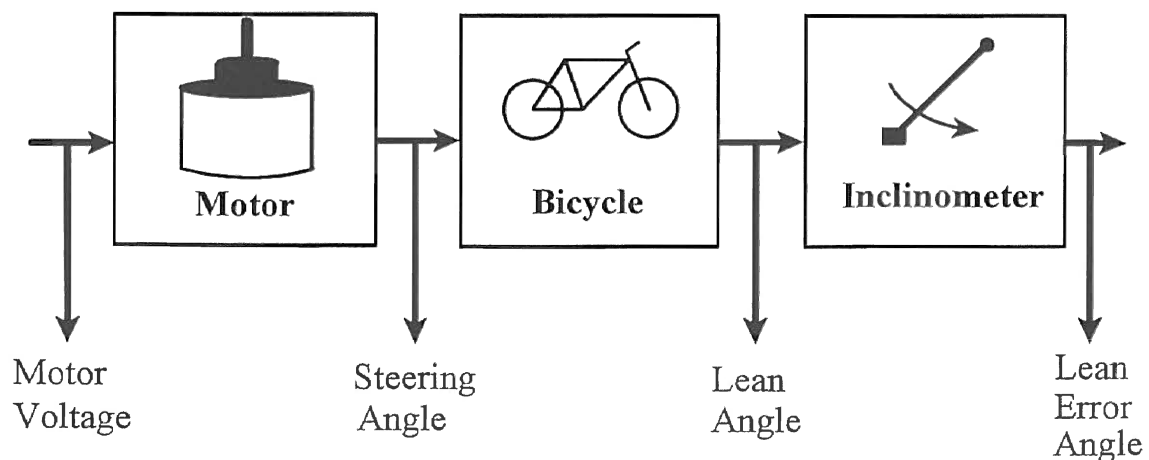
These details are important because they raise the question of what happens when the encoder changes direction. In the previous counter design there was a count problem at the turn around point. Problematic count conditions are shown in Figure 3.2. Often as the motor begins to reverse directions, glitches and incomplete pulse trains were misinterpreted by the counter as additional counts. This miscount problem is intensified by the fact that the motor may momentarily fluctuate from one direction to another before proceeding in one direction. This problem is difficult to detect in tests where the encoder is smoothly rotated a few times in each direction. Full system tests with vibration and drive chain chatter readily indicate the problem as a disjointed steering position curve.

CHAPTER IV

BICYCLE SUBSYSTEM

Important Components of the Bicycle System

The mathematical model of the Bicycle System may be viewed as three individual submodels: the motor, the bicycle and the inclinometer. Figure 4.1 pictorially shows these submodels as well as how they are related. Each block represents an equation of a physical device relating its input and output. This chapter covers the Bicycle subblock portion of the Bicycle System Model. Three important items concerning this model are presented. The relationship of the Bicycle Model to the system as a whole is discussed to help provide an overview. Second, important modeling considerations are reviewed in an intuitive development of the model. Finally, testing and verification of this model is explained and results are presented.



The Bicycle System Model

Figure 4.1

Relationship of the Bicycle Model to the Bicycle System

The Bicycle Model is an equation relating important geometric constants and variables to the state of the test bicycle. One reason for construction of the test platform was for the study of this equation. While this model is only one part of the total system, it is the only portion that describes the bicycle. All other portions of the model represent measurement or actuation devices that, while necessary to the test platform, are not the primary areas of study.

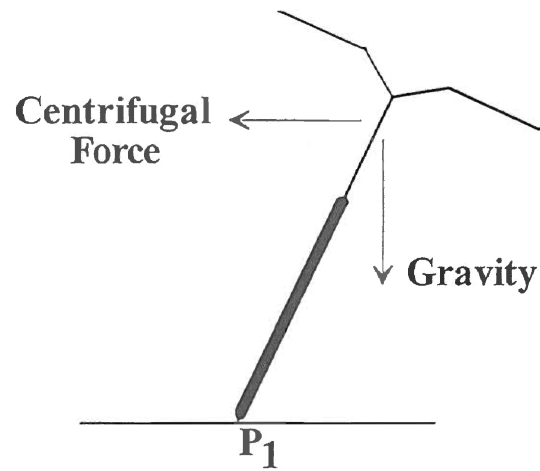
Important Forces in the Bicycle Model

The bicycle equation is concerned with modeling the effects of gravity and centrifugal force on the lean angle of the bicycle. Physical attributes deemed important in the modeling of these two forces have been included in the construction of the bicycle model. To simplify the actions of these forces, the total mass of the bicycle is assumed to be a point located at the center of mass. Figure 4.2 (a) depicts the bicycle with the two important forces; 4.2 (b) shows the free-body diagram of the bicycle with the forces acting through the center of mass. In this figure, Theta (θ) is the lean angle of the bicycle and l_1 is the distance from the ground to the center of mass of the bicycle.

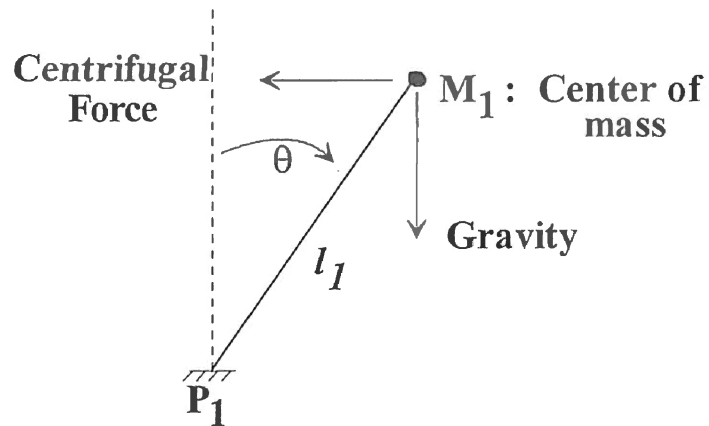
Overview of the Bicycle Model

Contained in the Bicycle Model are the important factors governing the stability of the bicycle apart from the effects of sensors and actuation hardware. This mathematical representation of the physical system is used in simulation work performed with the simulation tool Matrix_x on work stations. This model is also used by the onboard microprocessor in the estimation of unmeasured variables. Inherent in this model is the assumption that the classical Inverted Pendulum model represents the bicycle to the degree required to maintain the upright position of this system. While it is

the goal of this project to model the bicycle as accurately as possible, some less important factors have been omitted in the interest of simplicity and for practical implementation considerations.



(a) The Bicycle



(b) Free-body Diagram

Important Bicycle Forces

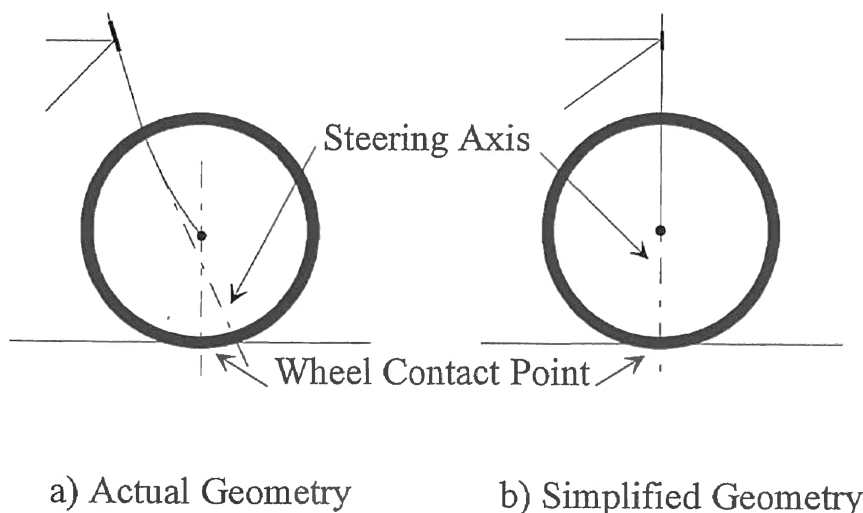
Figure 4.2

Modeling Simplifications

A series of simplifying assumptions were made in the construction of the Bicycle Model. Six of the more significant assumptions are listed below. The first two assumptions are considered the most significant.

- 1) Simplification of steering wheel geometry
- 2) Point mass acting at center of gravity
- 3) No gyroscopic effect of wheels
- 4) No wheel-roller friction
- 5) Small Angles (for Linearized Bicycle Model)
- 6) Effect of inclinometer mass on bicycle not considered

Assumptions 3, 4, 5 and 6 are considered minor assumptions and are listed here for completeness. Note, the Bicycle Model developed and presented in this chapter is the nonlinear model. This model must be linearized before the controls work of Chapter VI may be performed.



Front Wheel Geometry

Figure 4.3

The nonlinear model, developed by Mohammad B. Menhaj under the direction of Dr. Hagan, is listed in Appendix H along with an additional derivation presented because of its assistance with the intuitive understanding of the bicycle. It is not the intention of this chapter to derive or systematically dissect this model but to point out important implications and details that concern this work. Some of these important considerations have been previously mentioned but will be reexamined.

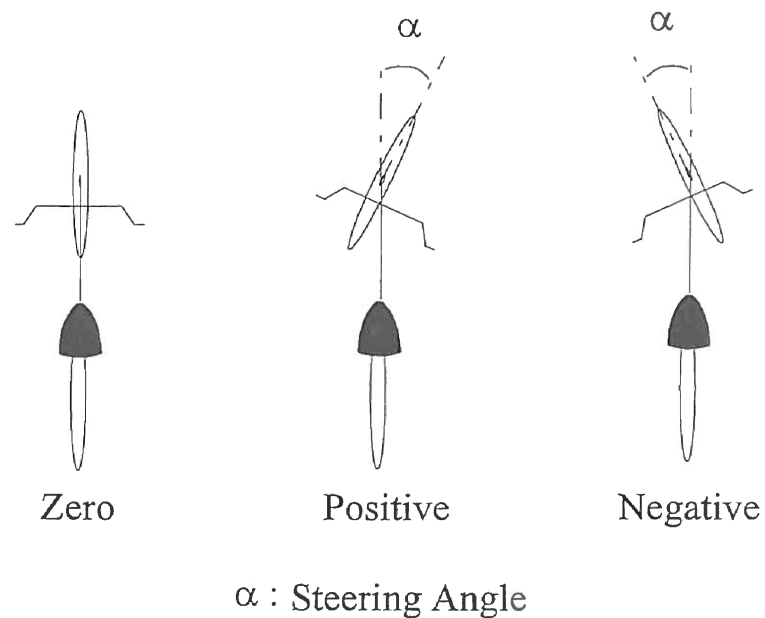
Modeled Bicycle Geometry

Wheelbase and vertical position of the center of mass are two geometrical considerations directly included in the model. Bicycle wheelbase, variable L_4 is measured from the contact point of the back wheel to the contact point of the front wheel. Realistically, however, the roller platform supports the back wheel by two rollers and so there are two contact points on the back wheel. This model assumes these two contact points may be treated as a single point so that the standard wheel wheelbase measurement may be used. The distance from the ground to the center of mass of the bicycle, variable L_1 , is determined by suspending the bicycle and using a geometric construction to estimate the center of mass of the bicycle. This measurement is performed with the motor and all control hardware affixed to the bicycle.

Development of the Bicycle Equation

The force of gravity causing the bicycle to fall may be counteracted by a centrifugal force as shown in Figure 4.2. Generation of this restoring force occurs when the rider, or in the bicycle test platform's case the steering motor, steers towards the direction of fall. Steering angle is defined as the angle the steering wheel makes with respect to the center line of the bicycle frame. This model development assumes the

simplified steering wheel geometry shown in Figure 4.3. Figure 4.4 shows an overhead view of the bicycle for both positive and negative steering angles.



Definition of Bicycle Steering Angle

Figure 4.4

It is helpful to consider centrifugal force and gravity as forces in a cursory discussion of the important factors included in the bicycle model; but, for a more in-depth development it is necessary to consider the angles at which these forces act. Torques are developed by these forces acting about the contact point of the wheel and the ground (point P_1 as shown in Figure 4.2). Hence, it is the torque developed by centrifugal force that is being manipulated to counter the torque developed due to the force of gravity. If the summation of torques is not zero, then the bicycle is falling. This implies that there is an angular acceleration, θ , which is accounted for in Equation (1).

$$\Sigma T = \ddot{\theta} I \quad (1)$$

In this equation the summation of torques is replaced by the two torques being summed. Using the Free-body Diagram shown in Figure 4.2 (b), Equation (2) may be written.

$$M_1 g l_1 \sin \theta - M_1 C_f l_1 \cos \theta = I \ddot{\theta} \quad (2)$$

Equation (2) relates the torque due to gravity and the torque due to centrifugal force, C_f , respectively to the angular rate of fall of the bicycle. Equation (2) may be manipulated, such as in Appendix H, to produce the Bicycle Equation. The Bicycle Equation is reproduced in Equation (3).

$$\ddot{\theta} = \frac{g}{l_1} \sin \theta - \frac{v^2}{l_1 l_4} \sin \alpha \cos \theta \quad (3)$$

g : Gravity

v : Bicycle Forward Velocity

l_1 : Vertical Position of the Center of Mass

l_4 : Bicycle Wheelbase

θ : Bicycle Lean Angle

α : Steering Angle

When the two principle torques are made equal, the bicycle is no longer falling and has reached a stable state. An infinite number of bicycle stable states exist which may be categorized as turning or traveling straight ahead. The bicycle roller test platform is not capable of sustaining the test bicycle in a turn, so the only stable state possible by the roller test platform is that state of the bicycle traveling in a straight line. Due to this physical constraint of the test platform, the goal is not to balance the torques

but to use the centrifugal torque to drive the bicycle upright quickly so that steering may be kept to a minimum. This material is considered further in Chapter VI.

Bicycle Model Testing and Verification

Chapters II and III dealt with the motor and inclinometer models respectively. The approach in these chapters has been to compare the theoretical model and actual test data for each block to verify the accuracy of the theoretical model. The motor and the inclinometer may each be examined in isolation so that inaccuracies in other blocks will not affect the block under examination. One exception to this "divide and conquer" strategy is the bicycle model because physical testing of this model necessitates the use of the motor and inclinometer. This inability to isolate and compare the bicycle model to its real counterpart represents the primary difference between the handling of this model and the aforementioned motor and inclinometer models.

Current testing of the bicycle model has been carried out by use of the full Test Platform. Before these were conducted, efforts were made to minimize errors in other parts of the model. Data received from these tests were information about the state of the bicycle such as lean angle and lean velocity. Data concerning other portions of the System Model were also taken during these tests and one full set is included in Appendix L. The difficulty with these tests was that unmodeled effects such as mechanical noise from the rollers and inclinometer anomalies tainted data so that conclusive observations were difficult to make.

The test bicycle appears to respond correctly when sensors provide reliable measurements to the controller. Qualitatively, test results suggest that the Bicycle Model is reasonable. Direct measurement of the lean angle of the bicycle, θ , would improve the accuracy of these experimental tests. The indirect measurement of lean angle and associated problems are covered in Chapter VI.

Summary

This chapter covers the important items of the Bicycle Model and its relationship to the complete system model. Significant modeling considerations are reviewed such as bicycle geometry and mass distribution. A simplified development of the Bicycle Equation is presented with emphasis on important forces. The testing and verification process used in the evaluation of this model is also discussed.

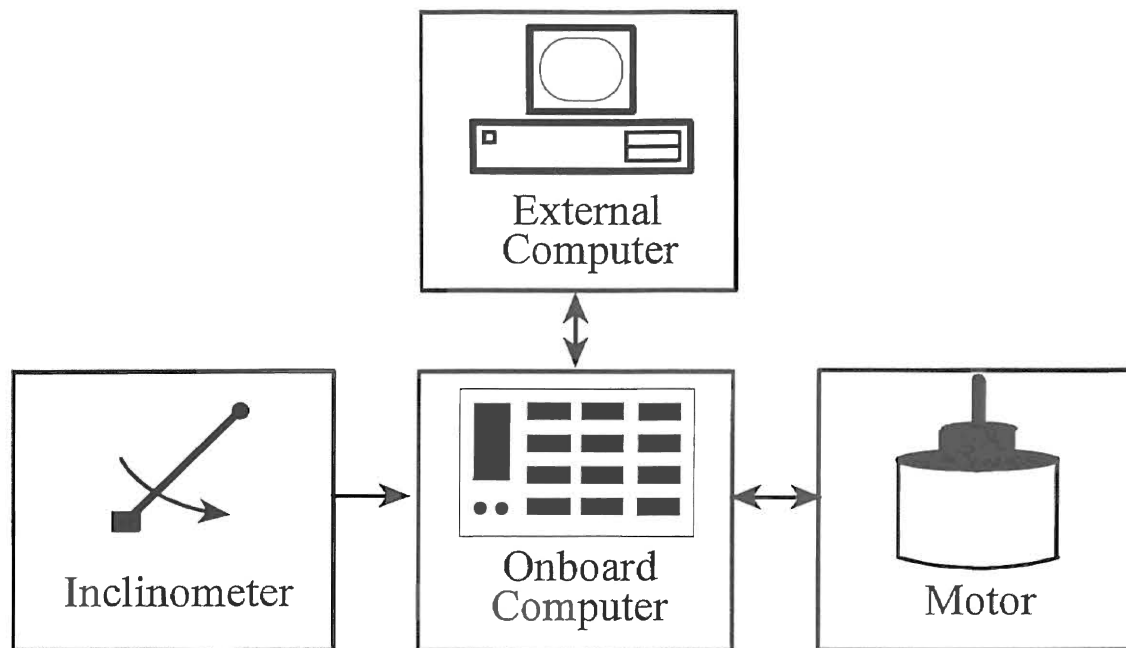
CHAPTER V MICROPROCESSOR SUBSYSTEM

Onboard Microprocessor

Continuing the human analogy, the riderless bicycle contains a single-board microprocessor to act in place of the human rider. Some of the same decisions made by the human rider must also be made by the bicycle computer such as what direction to steer and how fast. Specifically, the bicycle computer must administer to the tasks of taking measurements, estimating unmeasured variables, calculating motor voltage and operating the motor. This chapter will discuss the single-board microprocessor and the software written to execute on it. An overview of the microprocessor system will introduce its relationship to other test platform hardware. Second, the important hardware composing this single-board computer will be reviewed. Third, the software written for this microprocessor will be covered followed by the testing scheme used to verify correct operation of the software.

Overview

Before experimental Bicycle Platform tests, the onboard microprocessor is loaded with a program which contains a model of the Bicycle System. This program is downloaded from an external computer which is also used for editing, storage and data analysis. Once a Bicycle Platform Test is started, the external computer is no longer needed until another test is started. At the conclusion of a test, runtime data may be retrieved from the onboard microprocessor for storage and analysis. The relationship of the onboard microprocessor to external devices is illustrated in Figure 5.1.



Connection of the Onboard Computer

Figure 5.1

Microprocessor Hardware

A brief overview of the bicycle's single-board microprocessor is provided here as general information to improve the overall understanding of the test platform hardware. It is not the intention of this project to evaluate the performance of this computer or discuss hardware configuration beyond a list of major components and how these components are important to the Bicycle Test Platform. A brief list of the major single-board hardware components is noted here; for specific details on this hardware consult the Motorola user's manual MVME133/D1.

Main processor :	MC68020
Floating point coprocessor :	MC68881
Clock speed :	12.5 MHz
Memory, dynamic RAM :	1 Mb
Serial communication :	RS232C
Bus standard :	VME133
Operating system :	133XTBug

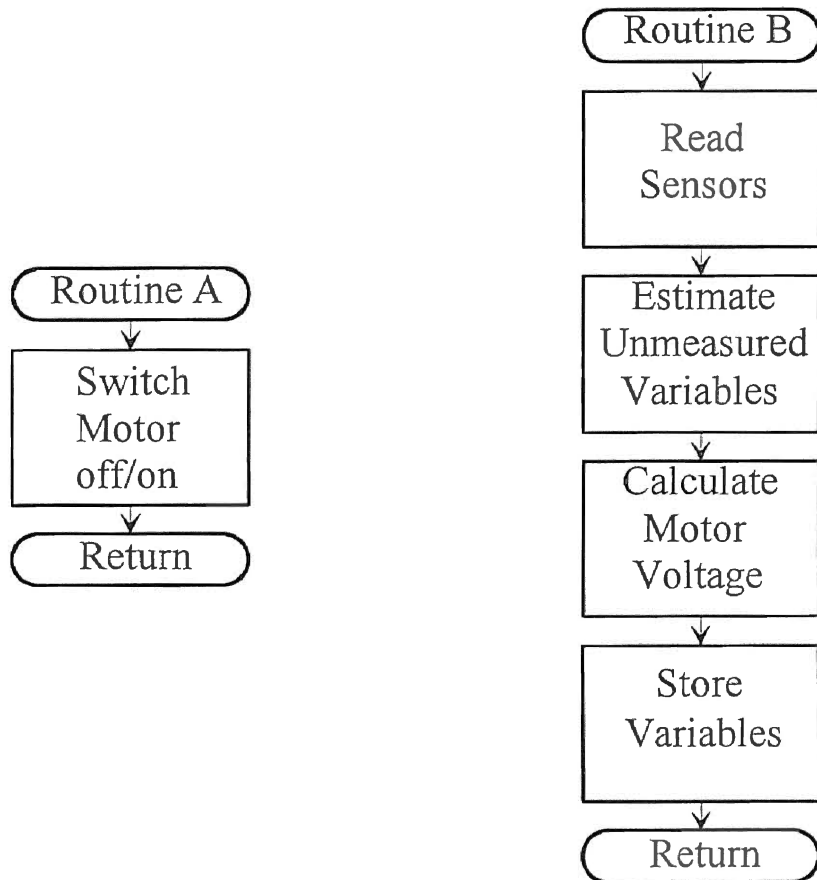
The bicycle test platform requires a considerable amount of computational power because of its use of an estimator to calculate unmeasured variables. Reasonably, this portion of the program requires the largest fraction of processor time. Essentially, the estimator is a string of floating point mathematical operations and requires the floating point coprocessor. After each iteration of the estimator, the variables calculated at that step of the program as well as the actual measurements are saved to memory for later analysis. The serial communication port serves as the link for program downloading from the external computer and the link for retrieving runtime data. Managing the serial communication operations as well as other low level tasks is the job of the resident operating system, 133XTBug. This operating system contains an assembler which is utilized during download to assemble the program into memory. Due to modifications made during platform tests, it has been helpful to retain Bug and the program in this development form. At whatever time it becomes appropriate, the operating system may be replaced by the platform test program.

Microprocessor Software

During one full iteration, the program loaded onto the bicycle microprocessor must complete all of the basic tasks consisting of operating the motor, reading the

sensors, estimating unmeasured variables and calculating the proper motor voltage.

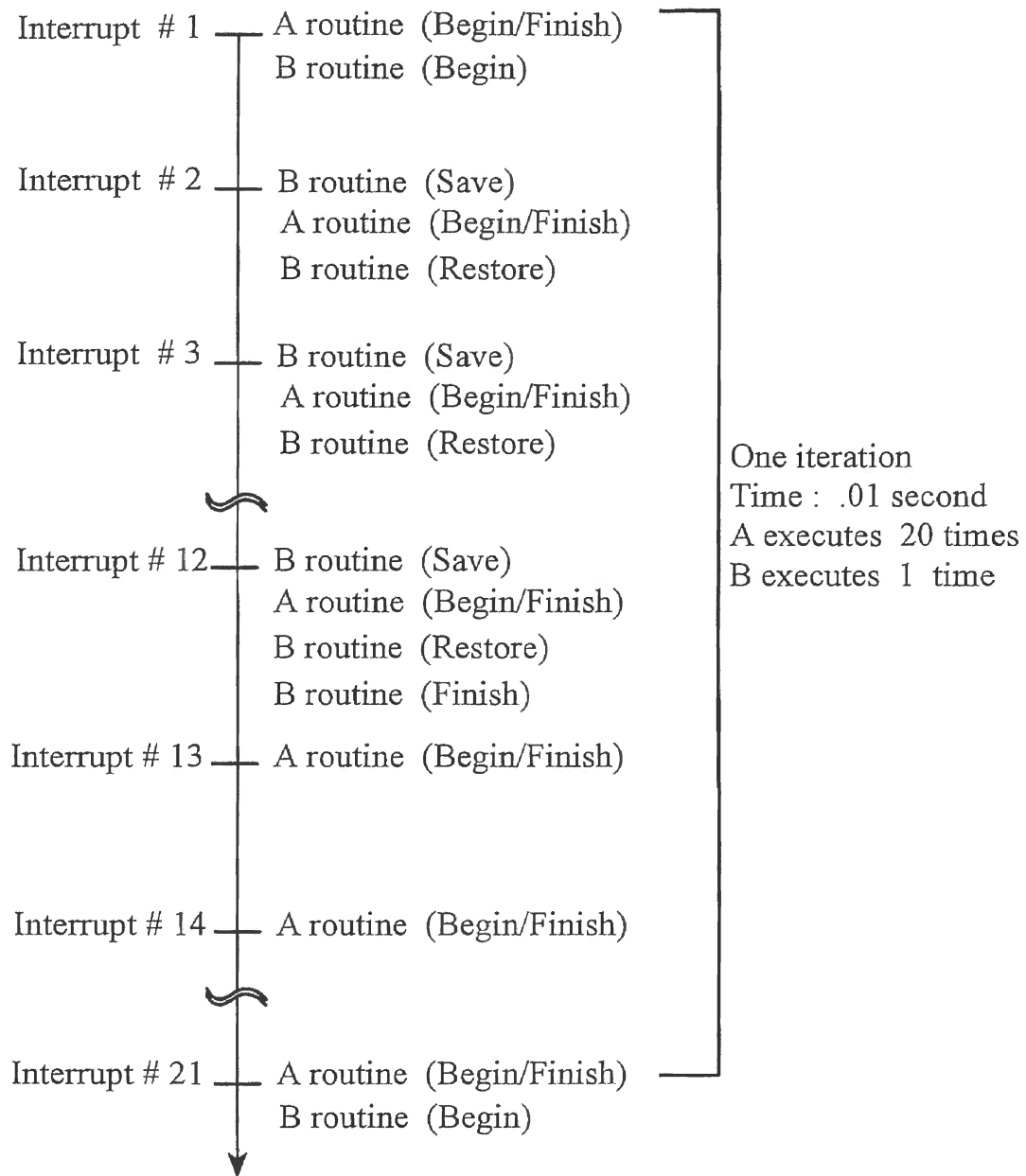
Figure 5.2 provides an overview of the described tasks.



Bicycle Test Platform Software

Figure 5.2

The significant detail of this program is not the tasks themselves but the manner in which the program must perform them. Various routines must execute at different rates during a bicycle test. The model (estimation and control voltage calculation) is required to execute 100 iterations per second while the Pulse Width Modulation (PWM)



One Iteration of the Bicycle Program

Figure 5.3

routine for the motor executes 2000 times per second. Interrupts are thus required to maintain accurate timing. To resolve the execution rate conflict, an executive is employed to control the two routines as the microprocessor issues the interrupts. The motor routine is referred to as Routine A and must be performed 2000 times a second. Since this is the more frequently executed routine the interrupt is configured to occur at this rate. The estimation and control routine is referred to as Routine B and must be executed 100 times a second so, Routine B must be performed once for every twenty executions of Routine A. Figure 5.3 depicts one iteration of this program. In this case, the executive counts the iterations of Routine A so that when the count becomes 20, Routine B is then started. The executive has one further task to fulfill, that being the task of monitoring the execution of Routine B because it takes considerable time to execute so that it is frequently interrupted. The executive must record whether or not Routine B was executing when it was interrupted so that it may be properly restored.

Execution Rates

The choice of interrupt rate for Routine B was determined during the development of the control system and represents the rate at which the model samples the sensors. Increasing the sampling time will, at some point, cause the estimates of bicycle variables to fall behind the actual state of the bicycle. Thus, when the sensors are read on the next iteration of the program, the controller will have to "catch up" with the real state of the bicycle and will require decreasing the steering response time. If the program falls too far behind, the steering motor will simply be unable to respond quickly enough to keep the bicycle from falling down. On the other hand, decreasing the sampling time means that Routine B must execute in less time. Various sampling rates were simulated by Mohammad J. Menhaj of which the 100Hz sampling rate was found to be best compromise considering the hardware involved. This rate can be changed to speed up

execution of Routine B but the model must be adjusted to the sampling rate which is used in the construction of the estimator.

The choice of interrupt rate for Routine A was determined after processor time requirements for the estimation and controller were set. Desirable execution rates for the Pulse Width Modulation routine were experimentally found to be in the range of 2000Hz. In this case, the execution rate needs to be fast enough that switching the motor does indeed average out to the calculated voltage. If the execution rate of Routine A is decreased too far, operation problems with the motor can be observed.

Completion of Routine B results in the calculated voltage to apply to the motor for the next one-hundredth of a second. The value calculated by the controller is a voltage and must be converted to a number Routine A can use to switch the motor. Under the current interrupt configuration, each one-hundredth of a second is divided into twenty equal periods during each of which the motor may be on or off. If the motor is on for all twenty periods, the average voltage applied to the motor is forty volts. If the motor is on for less than the full number of periods, then the average voltage will be a proportion of the number of periods the motor is on. The Pulse Width Modulation scheme for the motor is covered in more detail in Chapter II.

Software Testing

Verification of correct software operation has been accomplished through comparison of simulated model results with results from execution of the on-board model. In these tests, measurements taken from bicycle platform sensors are replaced with constants. This procedure isolates the software testing from any possible hardware problems. The same constant values used to replace the sensor measurements in the bicycle software are used as inputs to Matrix_x model simulations. The calculated motor voltage results from these two independent tests should be equal at each iteration because

the same estimator and controller with the same sampling rate was used for both tests. Because models are first developed and tested on the simulation tool, inconsistencies in these test comparisons are generally considered to be problems in the bicycle test program. Most of these problems result from incorrect model constants. This testing procedure can be considered to reliably test model constants, mathematical procedure and iteration rate.

Summary

The bicycle's onboard microprocessor and its tasks are outlined in this chapter. Microprocessor hardware and how it is used by the controlling software is covered. Within the "Microprocessor Software" section, the major tasks of the program are covered along with the interrupt structure responsible for the rate of execution. The chapter is concluded with the testing scheme used to verify correct operation of the onboard microprocessor software.

CHAPTER VI
CONTROL SYSTEM
Controls Method

The bicycle microprocessor is tasked with the administrative duties of taking measurements, estimating variables and applying the correct response to maintain the upright state of the bicycle. Inherent in this approach is the assumption that the microprocessor and the controlling software will be able to perceive the state of the bicycle and will then be able to formulate the correct response. There are numerous approaches to the design of this software. Artificial Intelligence methods are currently popular in the implementation of numerous controllers. Approaches that deal with poorly defined problems or nondeterministic situations appear to solve difficult problems with relative ease. Unfortunately, these methods generally do not help in the understanding of poorly understood problems. In the case of the bicycle stability problem however, knowledge about the bicycle system is the primary goal. For these reasons, the mathematically rigorous method of State Variable feedback has been chosen as the controller.

State variable feedback requires the identification of important system variables and their relationship to other state variables along with system inputs. This invariably leads to the formulation of mathematical equations defining the crucial dynamic response of the system under study. For the bicycle system this means that equations must be written for each subsystem involved in the physical response of this system; this includes the motor, the sensors and the bicycle. Ideally, it would be desirable to study the bicycle alone without the steering motor or the sensors because the addition of this hardware invariably alters the system being studied. This point becomes increasingly

apparent when working with the controls equations. Each of the additional test platform devices add their response to the overall response of the system. These subsystems are not modeled perfectly, and so, some of their incorporated response is not removed from the overall response. Thus, some of these distortions are added into the bicycle response.

Overview of Control System

The task of the control system is to monitor sensor measurements and to apply knowledge about the system to maintain the bicycle vertically on the roller platform. Control system knowledge about the bicycle system is contained in the form of dynamic equations defining the response of chosen variables. One reasonable starting point for the development of the model is the identification of variables important to the state of the system being controlled. Theta, the lean angle of the bicycle, is one such variable. This variable is defined as the angle between the leaning bicycle and the vertical plane. A rider seated on a bicycle has been chosen as the reference point for our sign convention. Theta has been defined as positive when the bicycle leans to the rider's right. It is also important for the control system to monitor the rate of change of Theta referred to as $\dot{\theta}$. These are two variables that have been chosen as state variables in the control system. Information about these variables and how they relate to other state variables as well as to physical bicycle dimensions is contained in the Bicycle Equation. The derivation of this equation is contained in Appendix H and is discussed in Chapter IV.

The bicycle control model may be broken into three pieces, the steering motor, the inclinometer and the bicycle. Each of these pieces has two variables chosen as state variables and hence have been chosen to be controlled. Therefore, the bicycle system model contains six state variables; the state variables are listed here along with their definitions.

SubModel	Variable name	Description
Bicycle	Theta	Angular displacement of bicycle measured from vertical plane to leaning bicycle.
Bicycle	Thetadot	Angular lean velocity of bicycle, rate of change of Theta.
Motor	ThetaM	Angular displacement of the steering motor.
Motor	ThetaMdot	Angular velocity of steering motor, rate of change of ThetaM.
Inclinometer	Rho	Angular displacement of inclinometer measured from the vertical.
Inclinometer	Rhodot	Angular velocity of inclinometer, rate of change of Rho.

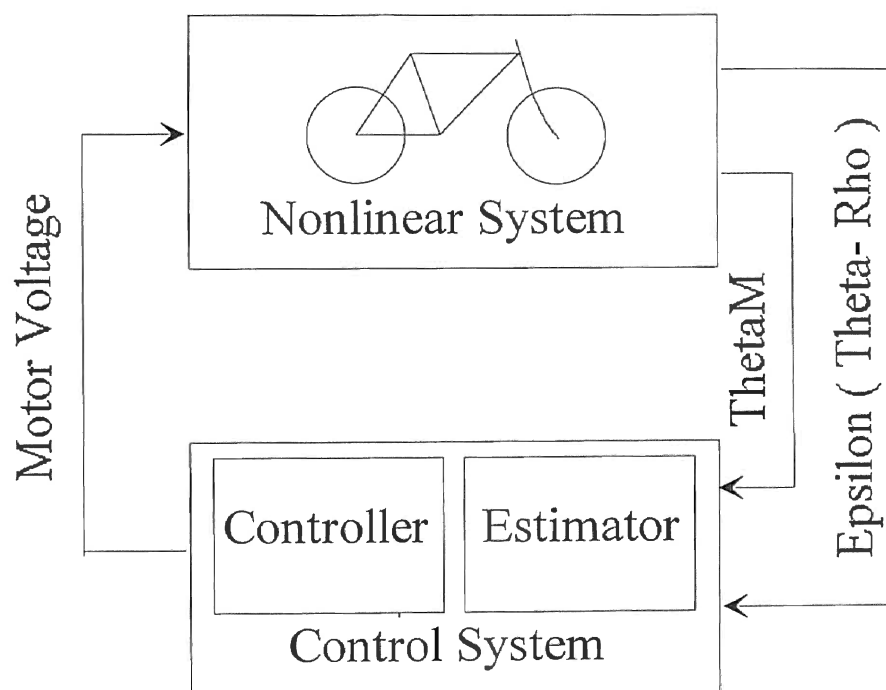
Given these six state variables the control system will calculate a response to send to the bicycle system; this response will be a voltage to send to the steering motor. The bicycle control system does not have access to all six of these measurements. Onboard sensors measure only Θ_M and ϵ (Theta - Rho). The second measurement of ϵ is due to the combined effects of gravity and centrifugal force acting on the inclinometer (for more information see chapter IV). Hence, the control system must estimate the unmeasured variable based on the three dynamics equations along with measurements from the two sensors.

Clarification of the term control system is appropriate at this point. While the control system is a nebulous term that can be applied to a myriad of methods of control, in this report, control system will apply to a combination of estimator and controller developed under the state variable methodology. The Estimator is used by the bicycle platform to calculate unmeasured state variables using system knowledge and measurement data from the sensors. These estimated state variables are then used by the controller to calculate a voltage to send to the steering motor.

Control System Simulation

Once the State Variables were chosen and the system equations were written, it was required that these equations be linearized so that State Variable design methods could be applied. The result of this design process was an estimator and controller pair to be used on the bicycle test platform. These two pieces of the control system were designed independently and then combined to form the full bicycle control system. Once the design process was completed, the control system was simulated with the use of Matrix_x before being loaded onto the bicycle microprocessor. The simulation involves the use of two sets of dynamics equations. The first set is the nonlinear equations

derived for the motor, the inclinometer and the bicycle. These equations are listed in Appendix H. The second set of equations used in the simulation are the first equations linearized plus error adjustment terms (error terms are discussed in the Estimator subsection). In the simulation, the first set of equations, the nonlinear equations, represent the modeled bicycle and the second set represent the estimator portion of the control system. During the simulation, the two sets of equations run in parallel



Dual System Simulation Block Diagram

Figure 6.1

with the estimator trying to emulate the nonlinear bicycle equations. State variable estimates from the estimator are sent to the controller which calculates the proper motor

voltage to upright the bicycle. Figure 6.1 illustrates the relationship of the two sets of equations. The simulation is started by giving the nonlinear system some nonzero initial condition such as an initial bicycle lean angle ($\Theta \neq 0$). The equations are then stepped forward in time to see if the control system can "catch up" with the nonlinear system and return the bicycle to vertical ($\Theta = 0$).

Measurement data sent into the control system and calculated motor voltage sent out are the only connections between the two parallel systems. The Estimator receives the motor position, Θ_M , and the lean error angle, $\Theta - \rho$. The Estimator uses these measured state variables to help in the reconstruction of all six state variables. These values are then used by the Controller to calculate the voltage to be sent to the motor.

Estimator

Certain properties of the derived model equations are used in the design of the control system. Assuming the bicycle is modeled perfectly, two important properties of this system are that it is fully observable and fully controllable. The design of the state variable estimator relies on the property of the state equations being observable. Essentially, if a system possesses this property, all of the defined state variables manifest themselves in the output of the system which in this case are the motor position and lean error angle measurements. Thus, observation of the two sensor measurements of motor position (motor position is proportional to steering angle) and lean error angle from the inclinometer will allow estimation of the unmeasured state variables.

A full State Estimator, which is employed by the bicycle control system, estimates all the defined state variables, including those directly measured by sensors. This redundancy allows the estimator to compare the measured states to the corresponding estimated values and to correct for estimation error. The error is multiplied by an error weight matrix which is subtracted from the calculation of the next

state variables. The error weight matrix, also referred to as the "k" matrix, may be adjusted to place more or less emphasis on the measurements depending on the level of accuracy achieved by the sensors.

The error weight matrix is calculated by use of the Riccati Equation. It is assumed for reasons of simplicity that bicycle dynamics do not change with time and so solution of the Riccati Equation is of the time invariant sort.

CHAPTER VII

RESULTS

Overview of Accomplishments

The Riderless Bicycle Test Platform includes a significant amount of theoretical work, software and hardware. A considerable quantity of this work existed prior to the beginning of this study. The primary challenge presented to this work was to test and assemble the existing material into an integrated platform. The result of this work addresses four key areas of the Riderless Bicycle Test Platform.

- 1) System Integration
- 2) Improved System Reliability
- 3) System Verification
- 4) General Recommendations

One of the ultimate goals of this system is to function as an experimental test platform for the development and verification of theoretical bicycle models. The original test platform was functional, but tests were difficult to perform due to an awkward configuration of hardware and software. Sections of the original platform were not properly verified. Hence, there were errors in the Test Platform.

This work has progressed by evaluating one subsystem at a time to verify proper operation or improve the existing design. Testing and verification began with those subsystems of the bicycle test platform believed to be the "weakest links" in the overall system. During verification and testing of each subsystem, important questions and desirable properties were identified. Where satisfactory improvements of the existing

hardware could not be currently implemented, recommendations we believed will correct these problems have been suggested.

System Integration

The improved Test Platform now allows the user to move through the entire design loop. The process may begin with the design of a mathematical model of the bicycle which may be tested with the simulation tool Matrix_x. This step of the design allows the designer to develop and test a State Variable control system for the Test Platform. Once the control system is complete, it can be transferred to the Bicycle Test Platform for observation of actual response. When the experimental test is concluded, recorded bicycle variables may be collected and plotted for further analysis.

Improved System Reliability

Regular failure of Test Platform hardware required considerable repair time and often produced unreliable test results. Hardware improvements were required if regular experiments were to be conducted on the Test Platform. Modifications to the hardware concentrated on increasing durability and on improving the sensor's ability to tolerate vibration from the Test Platform. The four major modifications implemented to improve system reliability were:

- 1) redesign of motor mount and linkage
- 2) improvement of motor power drive
- 3) modification of steering position counter
- 4) improvement of inclinometer hardware.

Testing efficiency has also been increased with these improvements as well as improvements to the platform software. Testing with the previous platform was a

difficult task because hardware required constant adjustment and repair. During the test, additional personnel were needed to monitor power supplies which were turned off when tests went awry. The improvement and reduction of hardware along with the addition of software now allows a single individual to perform tests in less time. Modifications to hardware intended to improve testing efficiency include:

- 1) emergency motor stop routine
- 2) improved data analysis software
- 3) overall reduction in discrete hardware
- 4) concentration of inclinometer hardware.

System Verification

System verification includes the individual testing of each subsystem for proper operation. The verification process also included the comparison of simulated tests from Matrix_x to static platform software tests. These tests suggested that both hardware and software functioned correctly with the possible exception of the inclinometer hardware. The primary issue revealed during sensor testing was "how should the sensors be tested?". Simple static tests discussed in Chapter III reveal reliable data but tend to overlook dynamic problems that occurred due to the significant vibration of the Test Platform. Dynamic bicycle sensor tests are difficult to repeat and the results vary considerably, but these tests suggest sensor problems not indicated by static tests. The continued use of the existing sensors will most likely require the use of a mixture of these sensor tests.

General Recommendations

The Riderless Bicycle Test Platform is a functioning test system. Models may be developed, simulated and tested on the laboratory test system. Experiments involving

changes to the bicycle model may be made at the theoretical level and may then be observed on the Test Platform. Improvements in some areas of the Test Platform are still needed. Based on the experience of this work, a series of recommendations may be made in order of perceived priority.

- 1) Measure Bicycle Angle Directly
- 2) Improve Data Collection and Analysis Software
- 3) Reduce Mechanical Noise

Direct measurement of bicycle lean angle would greatly reduce the complexities of the bicycle stabilization problem. This would change the problem to be solved because estimation of unmeasured variables would no longer be needed. If a sensor capable of direct lean angle measurement is not chosen to replace the inclinometer, then such a sensor would be helpful in testing the inclinometer.

The estimation of unmeasured states represents the single largest problem for vertical stabilization of the bicycle on the Test Platform assuming vibration cannot be reduced. Simulation results suggest that the Estimator portion of the Estimator-Controller pair is the weak link of this system. Direct measurement of Bicycle lean angle with an optical system would eliminate sensor dynamics and other complicating problems. Effectively, this "fast" direct measurement would leave only the Motor and Bicycle models to be simulated and allow more energy to be spent studying the bicycle's dynamics.

Design and implementation of software that automatically receives, sorts and plots recorded variables collected during a Platform test would greatly improve testing efficiency. Time spent writing this software would quickly be recovered during experiments in the time saved from not performing each of these tasks one at a time. The

addition of a new Windows plotting package did decrease data collection time but fell far short of a fully integrated software package.

The replacement of the roller platform with one with less vibration would not only help in the vertical stabilization of the bicycle, it would also extend the life of the hardware. The speed of the existing roller platform is limited due to increased vibration at increased speeds. Purchase of a professionally constructed roller set would be convenient, but the commercially available models are generally much narrower than the current roller platform. Thus, the sideways motion of the test bicycle is limited.

Improved System Results

The modifications to the Bicycle Test Platform discussed in this report have improved the overall experimental results of the Riderless Bicycle System. One indicator of experimental test success, bicycle test duration, has been increased. Bicycle test duration is defined as the time from which the bicycle is released to the time the emergency motor stop routine stops the test.

	Previous System	Current System
Average test duration	< 1 sec	2.3 sec
Best test duration	1.2 sec	3.8 sec

The increase in system performance shown here may generally be attributed to the improvements in platform hardware. While the overall performance of the bicycle on the Test Platform was disappointing, test results do indicate that improvements to the system have been accomplished.

References

- [1] Jones, D.E.H., "The Stability of the Bicycle," Physics Today, April 1970, pp. 35-40.

- [2] Nagai, M., "Analysis of Rider and Single-Track-Vehicle System; Its Application to Computer-Controlled Bicycles," International Federation of Automatic Control, Automatica, Vol. 19, No 6, pp. 737-740, 1983.

- [3] Mohammad B. Menhaj, "Riderless Bicycle Control Design," Oklahoma State University, August 1988.

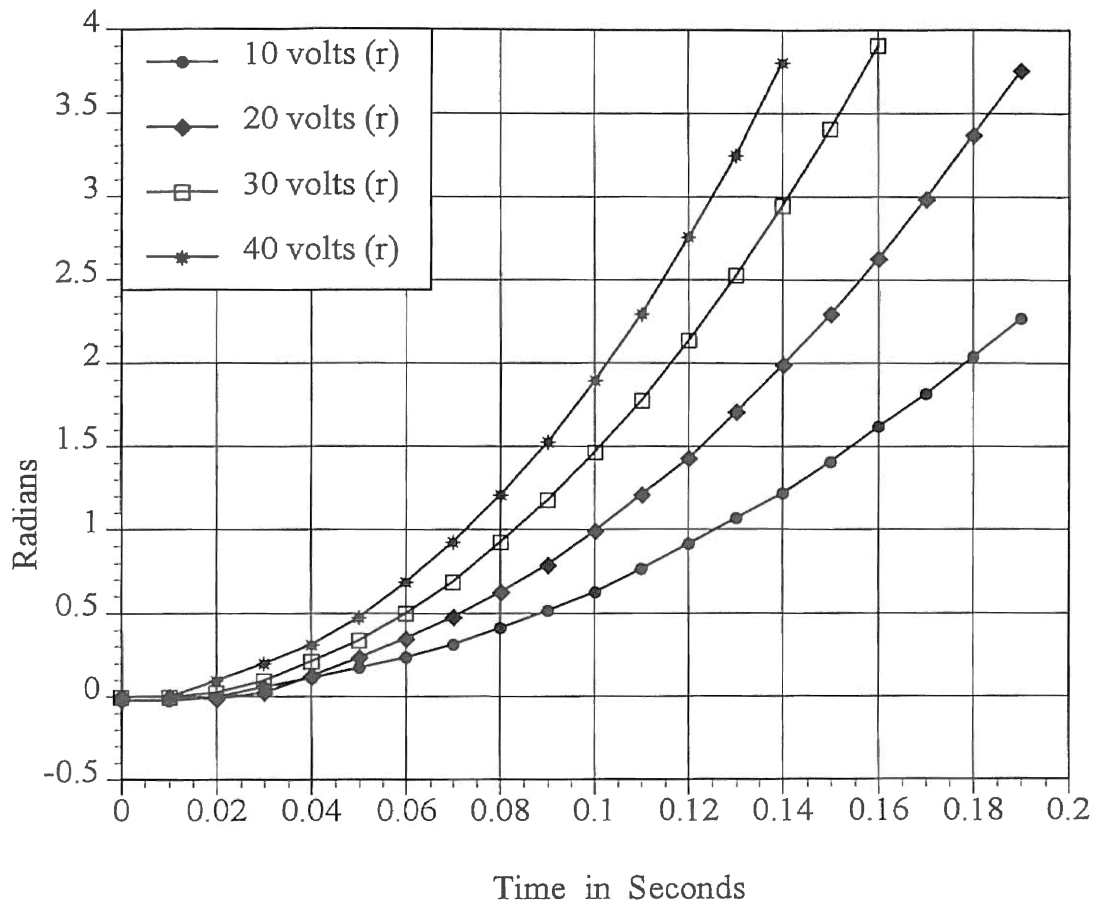
- [4] William, L. Brogran, "Modern Control Theory," Prentice Hall, Englewood Cliffs, New Jersey, 1991.

- [5] Benjamin C. Kuo, "Automatic Control Systems," Prentice Hall, Englewood Cliffs, New Jersey, 1987.

- [6] Gene F. Franklin, J David Powell, Michael L. Workman, "Digital Control of Dynamic Systems," Addison-Wesley, Reading, Massachusetts, 1990.

Appendix A

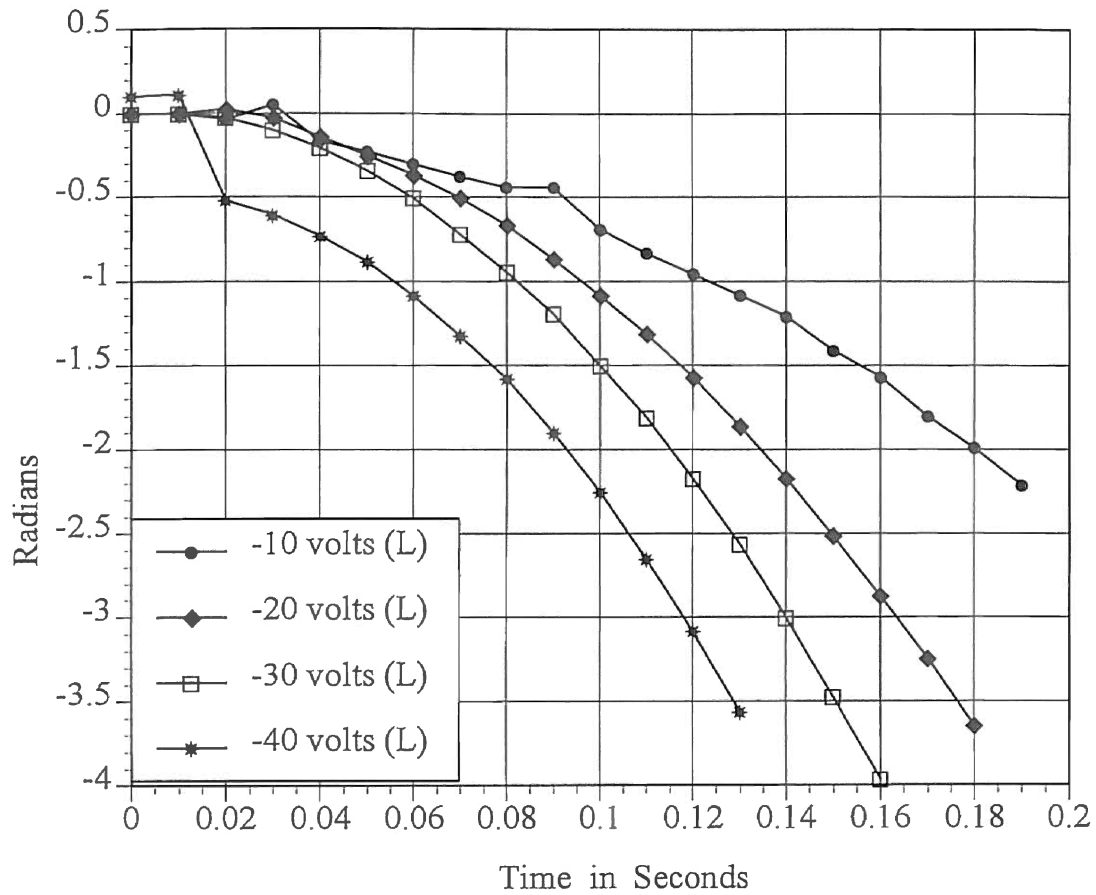
Motor Response Test Data



**Steering Motor Dynamics,
Actual Motor Position Vs. Time,
Turning Right**

Figure A - 1

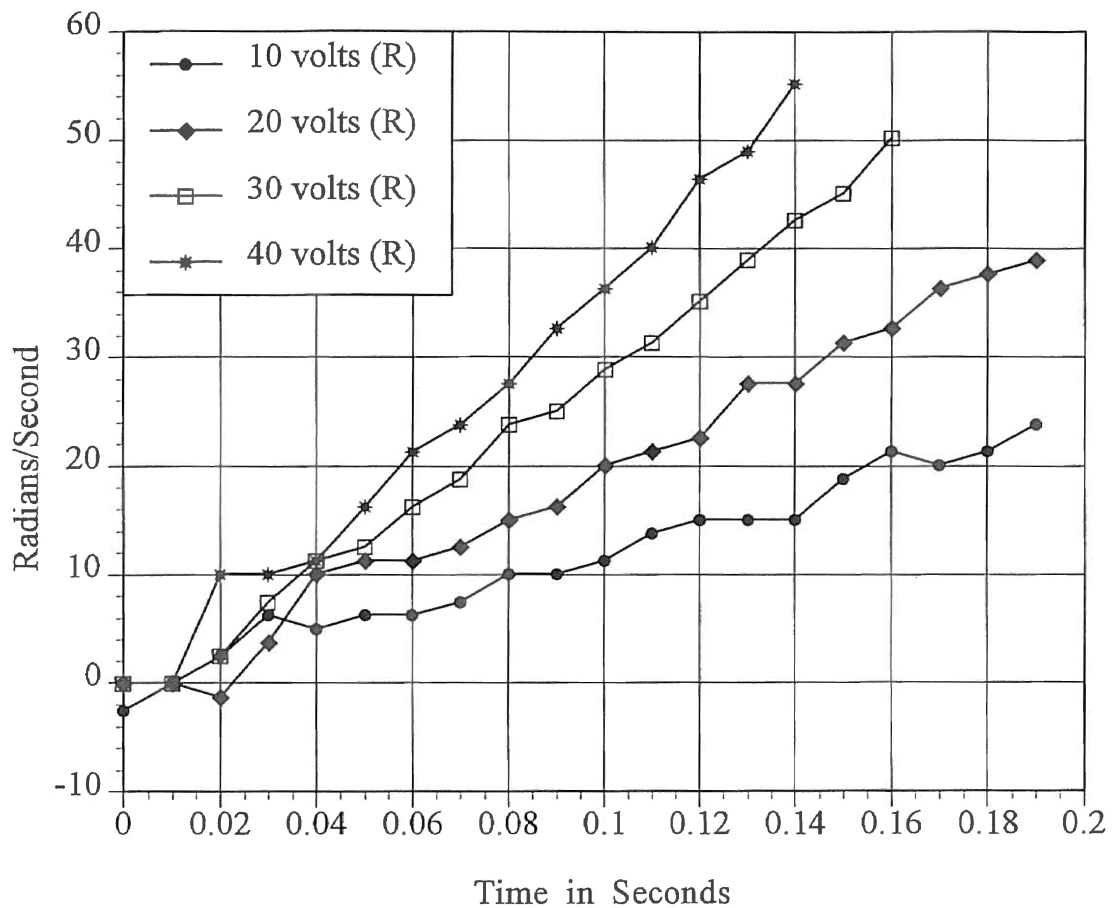
Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter; data were not filtered. Maximum motor voltage is 40 volts.



**Steering Motor Dynamics,
Actual Motor Position Vs. Time,
Turning Left**

Figure A - 2

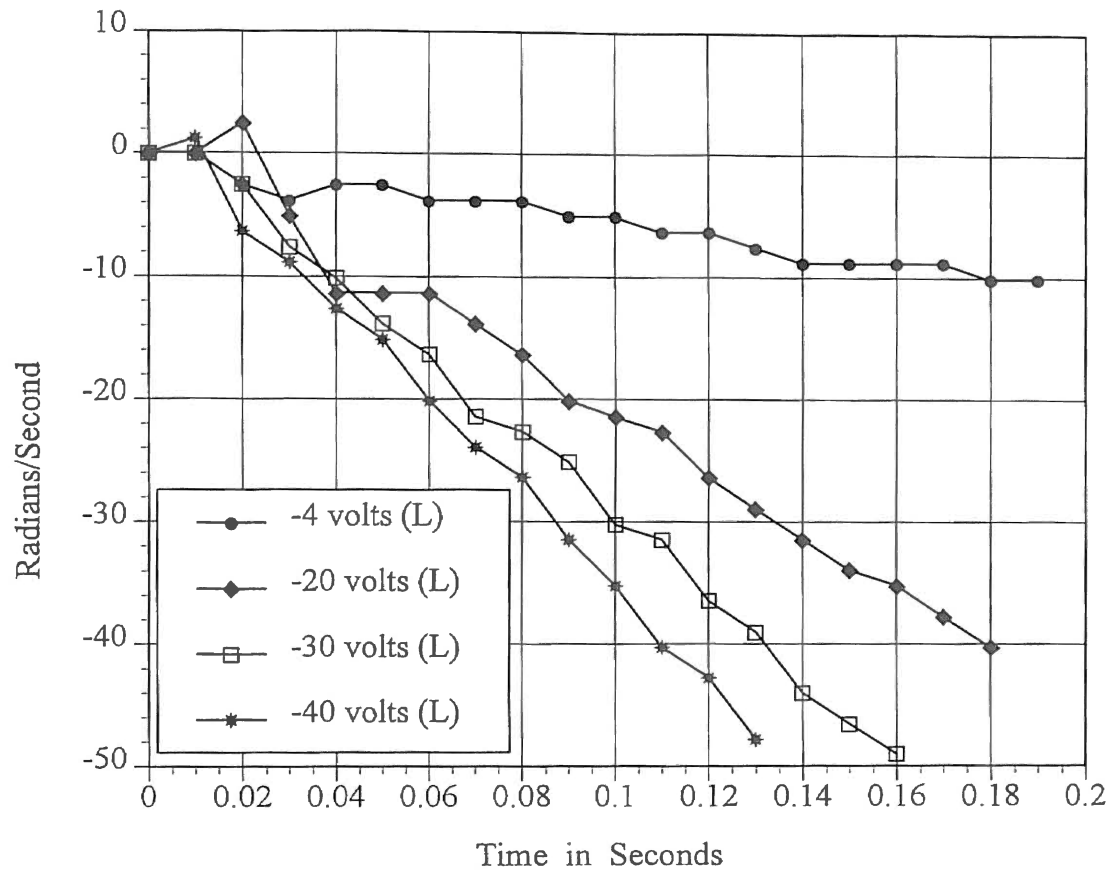
Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter; data were not filtered. Minimum motor voltage is -40 volts.



**Steering Motor Dynamics,
Actual Motor Velocity Vs. Time,
Turning Right**

Figure A - 3

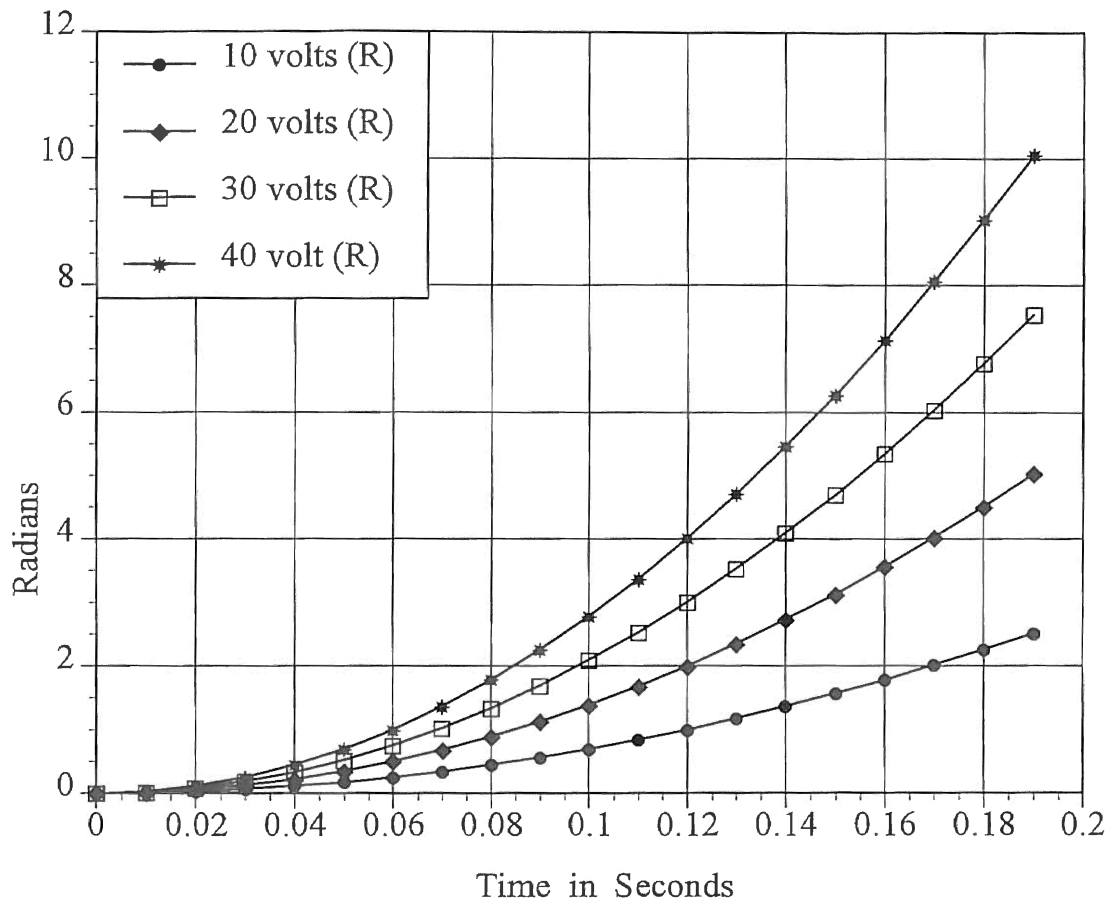
Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter with samples .01 second apart. Velocity was calculated as the difference between the positions divided by .01 second.



**Steering Motor Dynamics,
Actual Motor Velocity Vs. Time,
Turning Left**

Figure A - 4

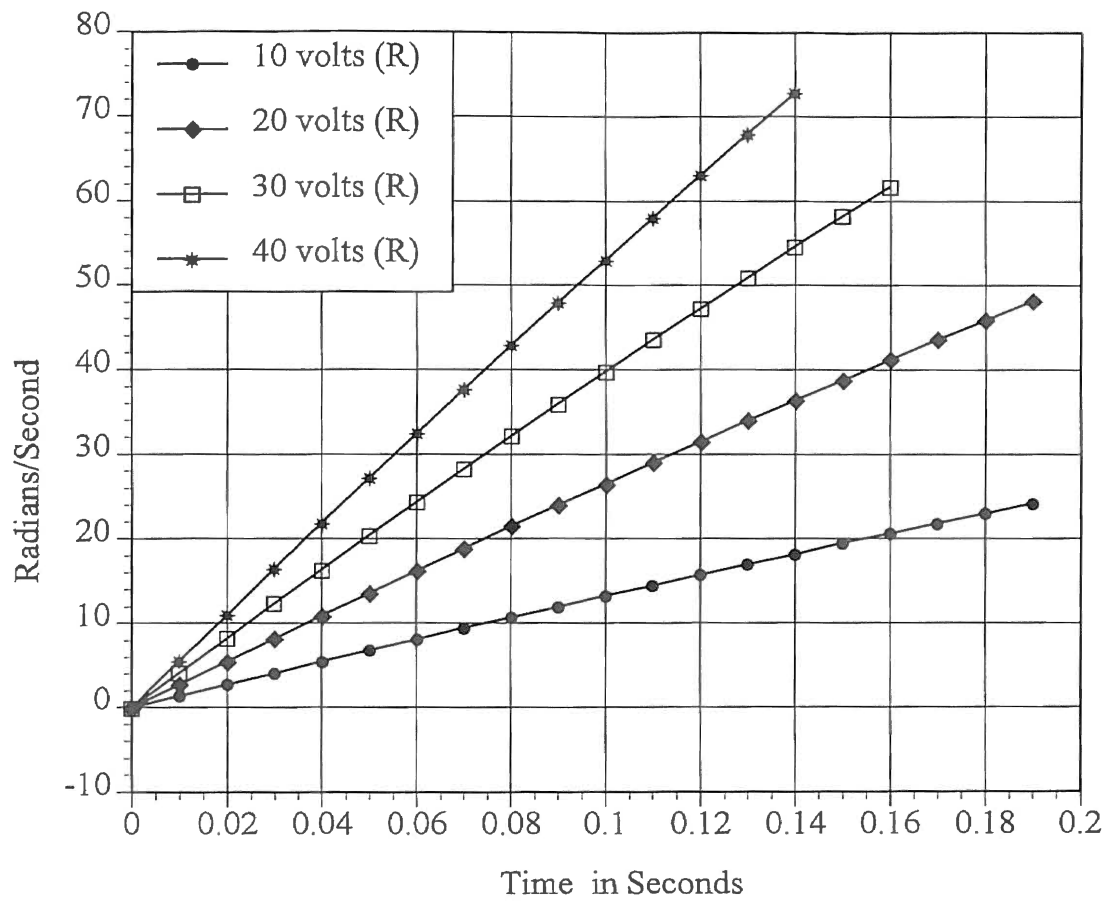
Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter with samples .01 second apart. Velocity was calculated as the difference the between positions divided by .01 second.



**Steering Motor Dynamics,
Simulated Motor Position Vs. Time,
Turning Right**

Figure A - 5

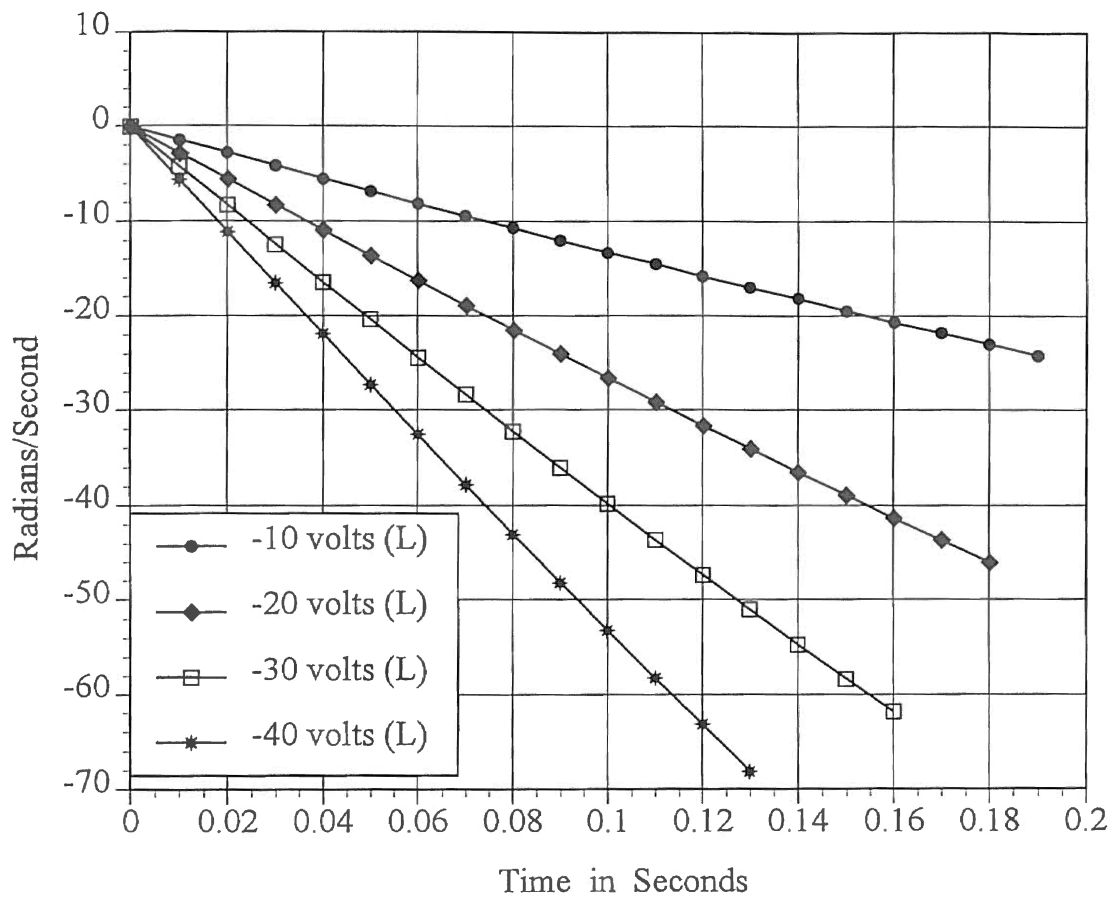
Note: These data were generated by the motor model simulated at various test voltages with the simulation tool Matrix_x.



**Steering Motor Dynamics,
Simulated Motor Velocity Vs. Time,
Turning Right**

Figure A - 6

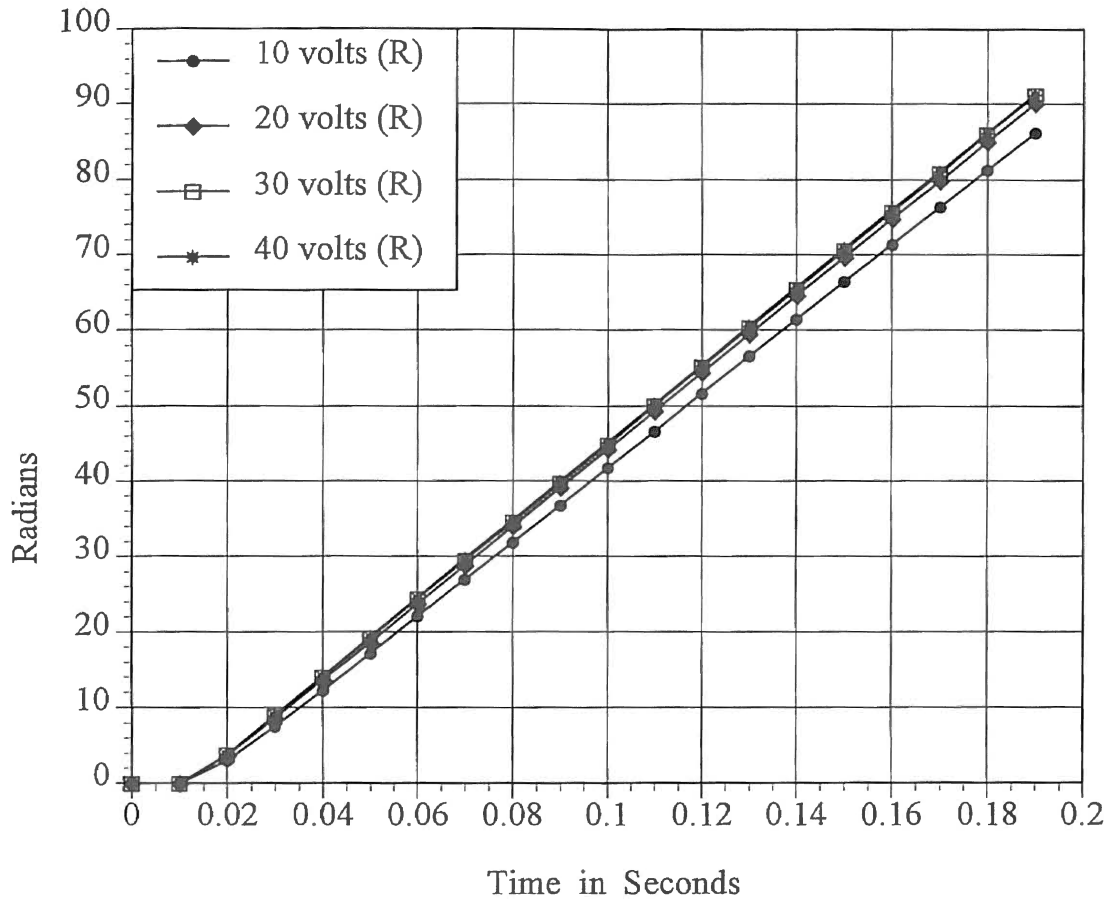
Note: These data were generated by the motor model simulated at various test voltages on the simulation tool Matrix_x.



**Steering Motor Dynamics,
Simulated Motor Velocity Vs. Time,
Turning Left**

Figure A - 7

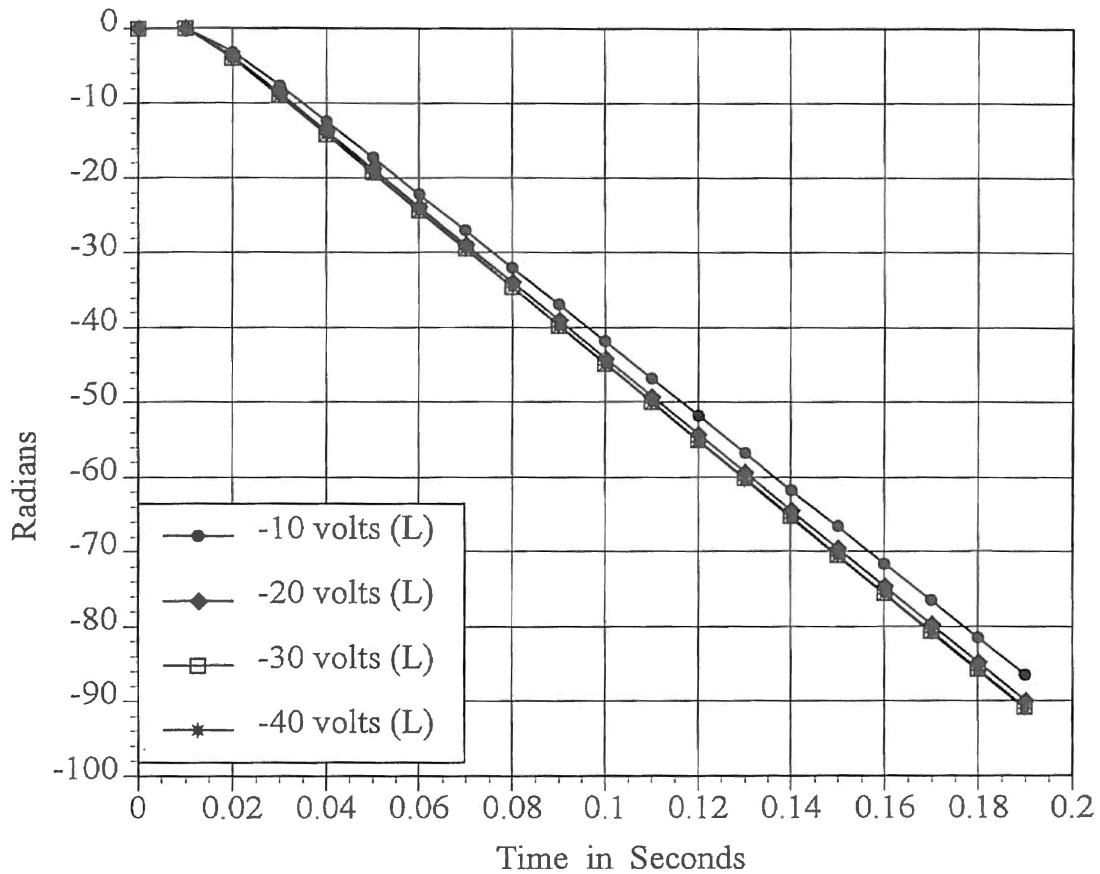
Note: These data were generated by the motor model simulated at various test voltages with the simulation tool Matrix_x.



**Steering Motor Dynamics,
Actual Unloaded Motor Position Vs. Time,
Turning Right**

Figure A - 8

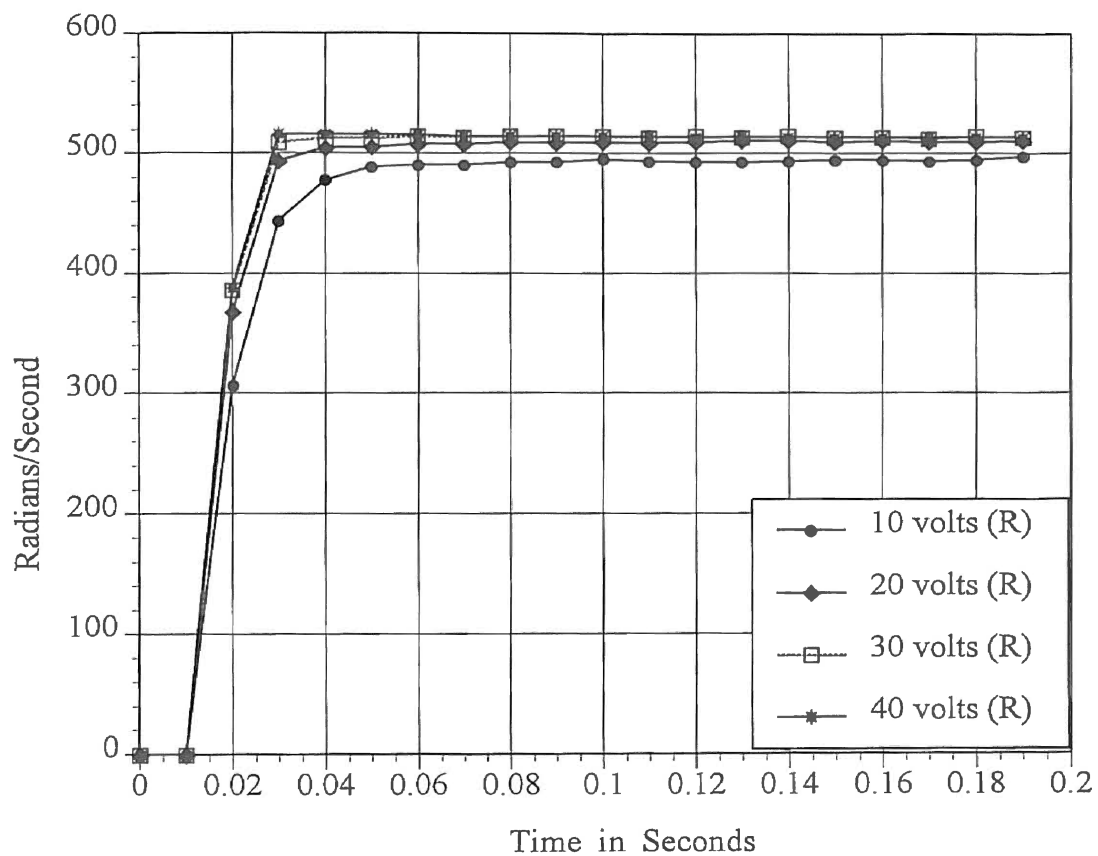
Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter; data were not filtered. The motor was disconnected from the steering wheel for this unloaded motor test.



**Steering Motor Dynamics,
Actual Unloaded Motor Position Vs. Time,
Turning Left**

Figure A - 9

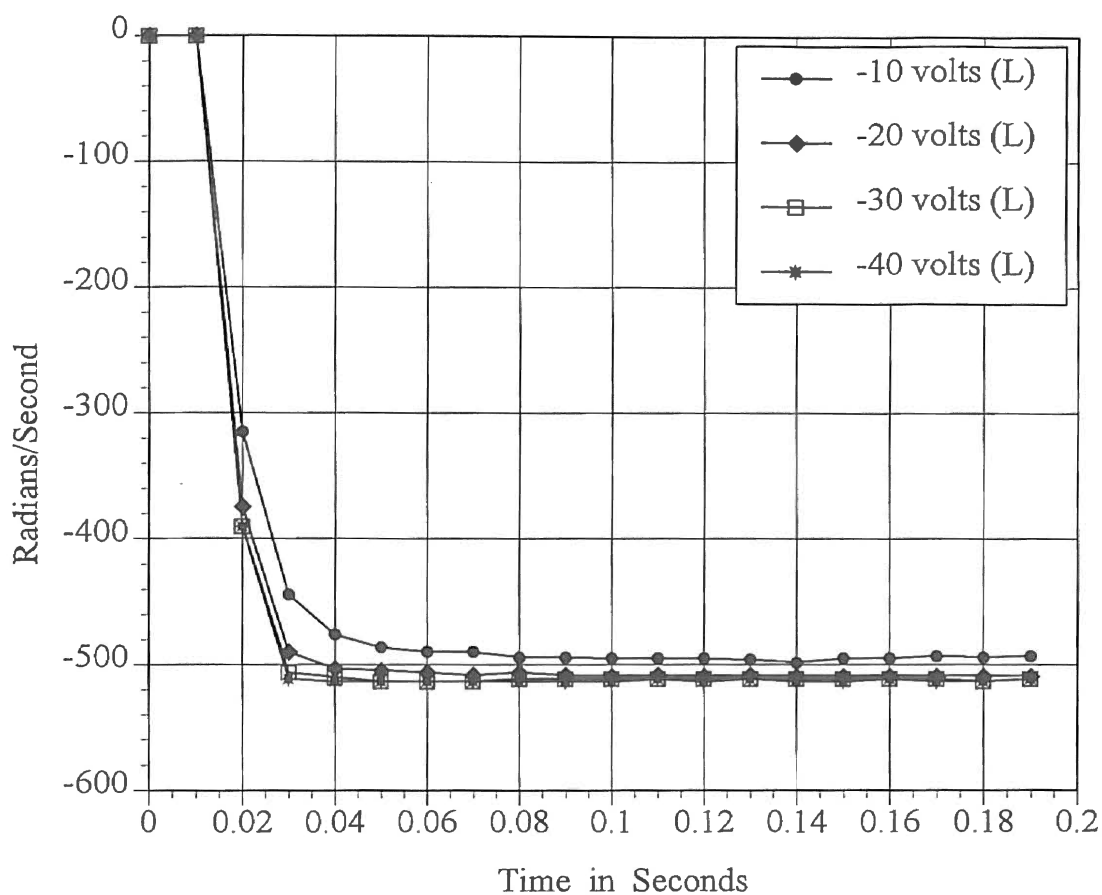
Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter; data were not filtered. The motor was disconnected from the steering wheel for this unloaded motor test.



**Steering Motor Dynamics,
Actual Unloaded Motor Velocity Vs. Time,
Turning Right**

Figure A - 10

Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter; data were not filtered. The motor was disconnected from the steering wheel for this unloaded motor test.



**Steering Motor Dynamics,
Actual Unloaded Motor Velocity Vs. Time,
Turning Left**

Figure A - 11

Note: This test was conducted using actual motor driver hardware. Data were collected directly from the steering counter; data were not filtered. The motor was disconnected from the steering wheel for this unloaded motor test.

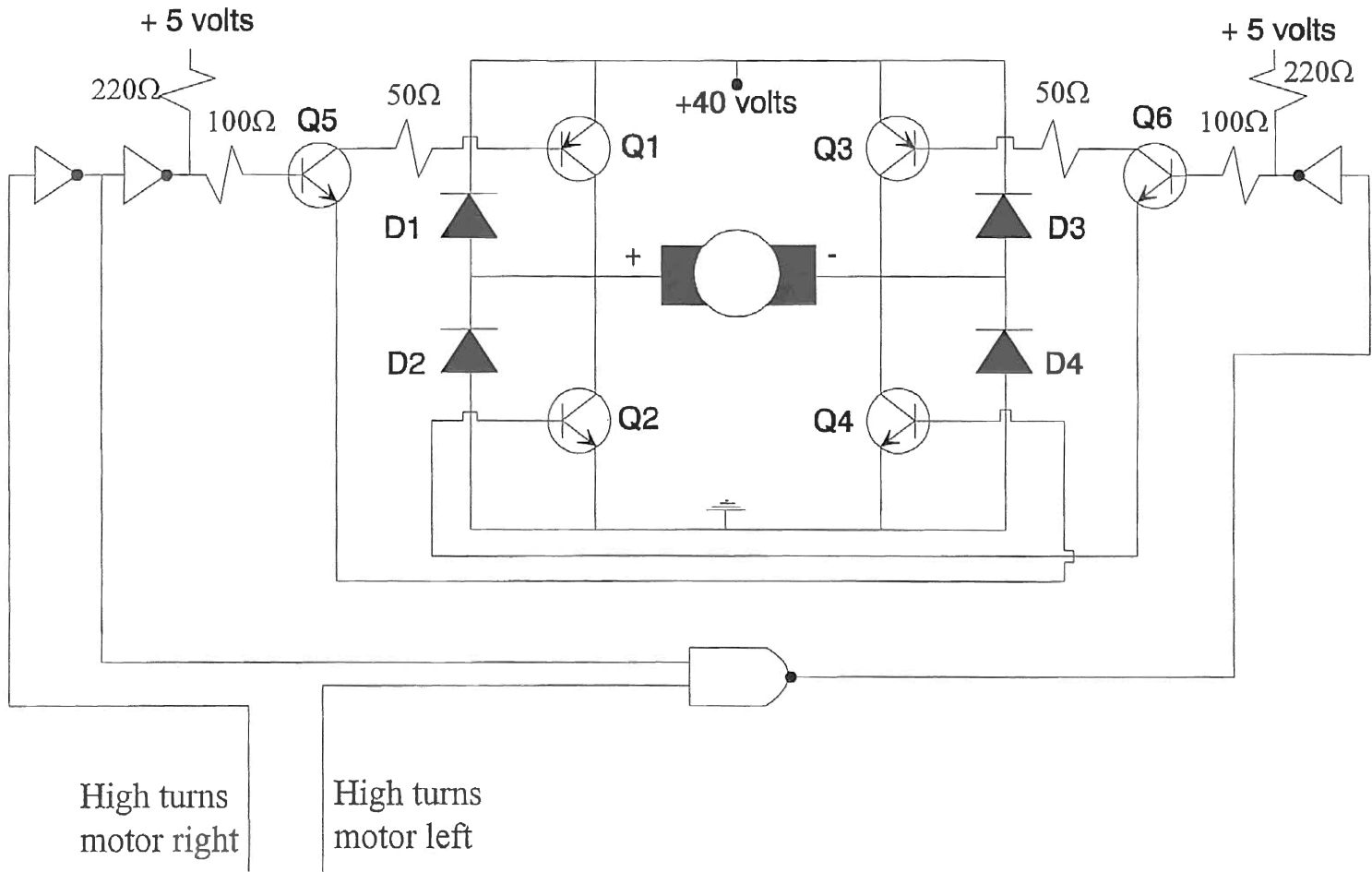
Appendix B

Motor Power Drive Hardware

**Motor Power Drive
Parts List**

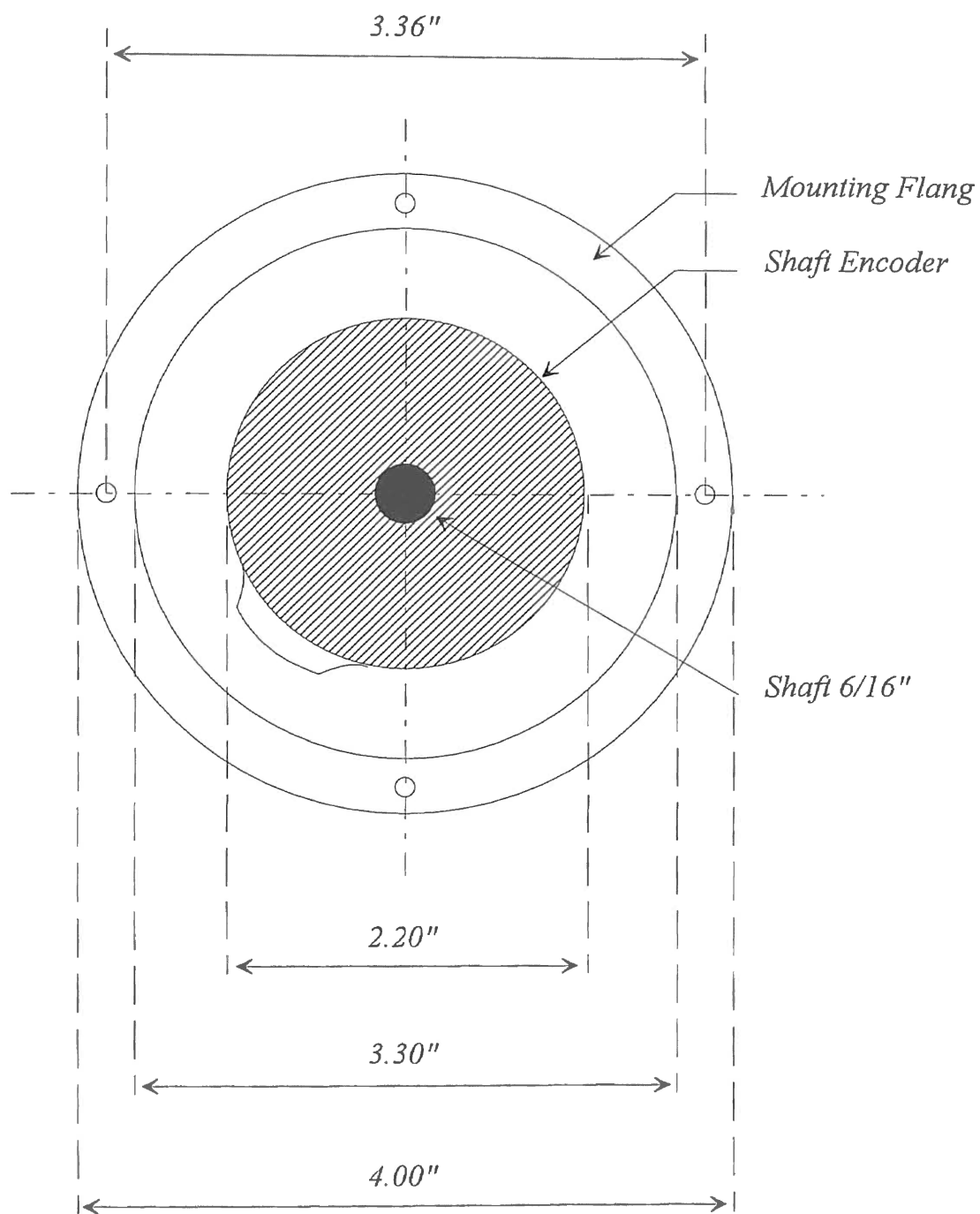
D ₁ - D ₄	Diode	1N3889R
Q ₁ & Q ₃	PNP Transistor	SK9136
Q ₂ & Q ₄	NPN Transistor	SK9134
Q ₅ & Q ₆	NPN Transistor	2N3053
50 ohm R	Resistor	5 watts
1K, 10K R	Resistor	1/4 watt

Motor Power Drive Circuit

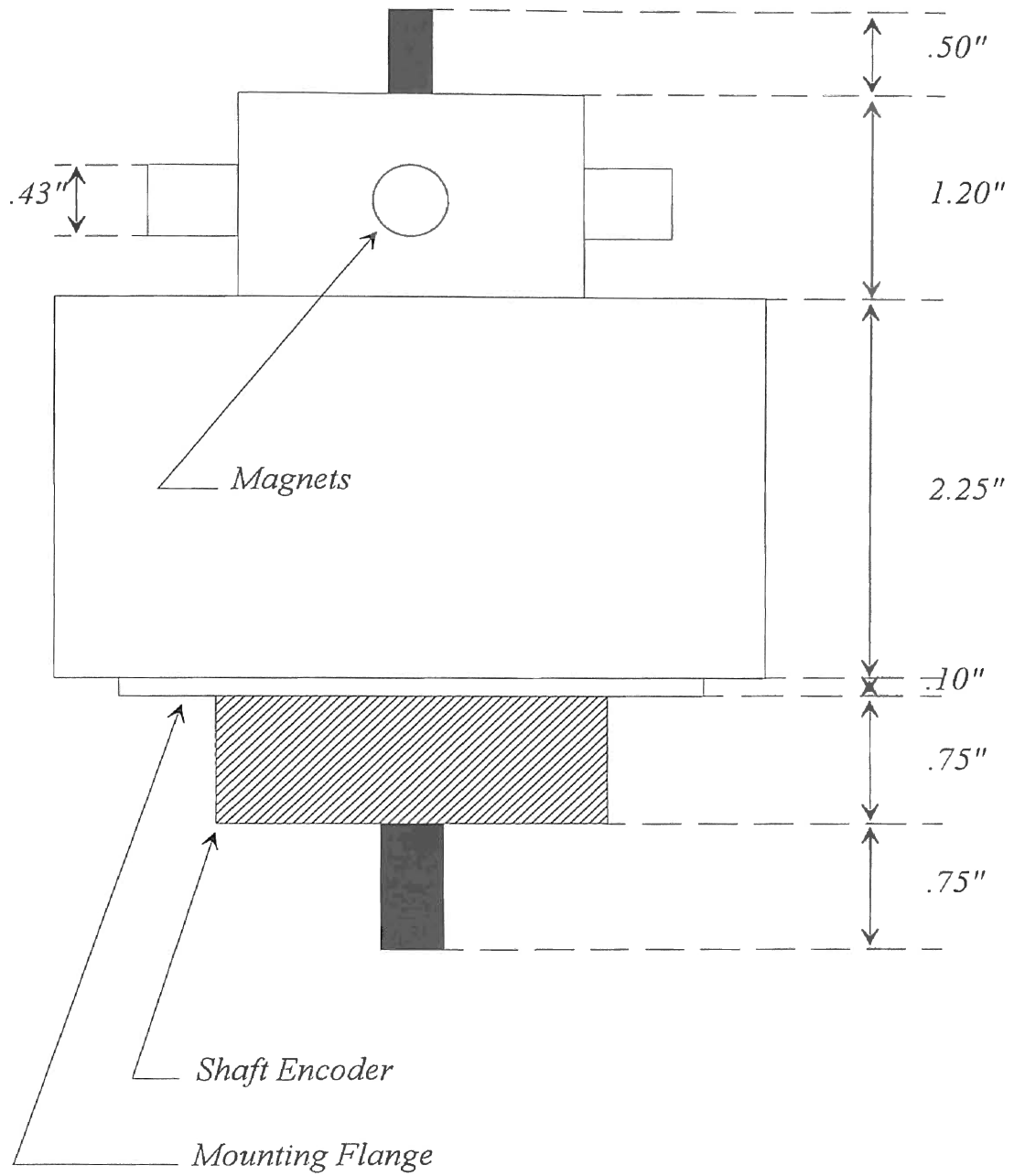


Appendix C

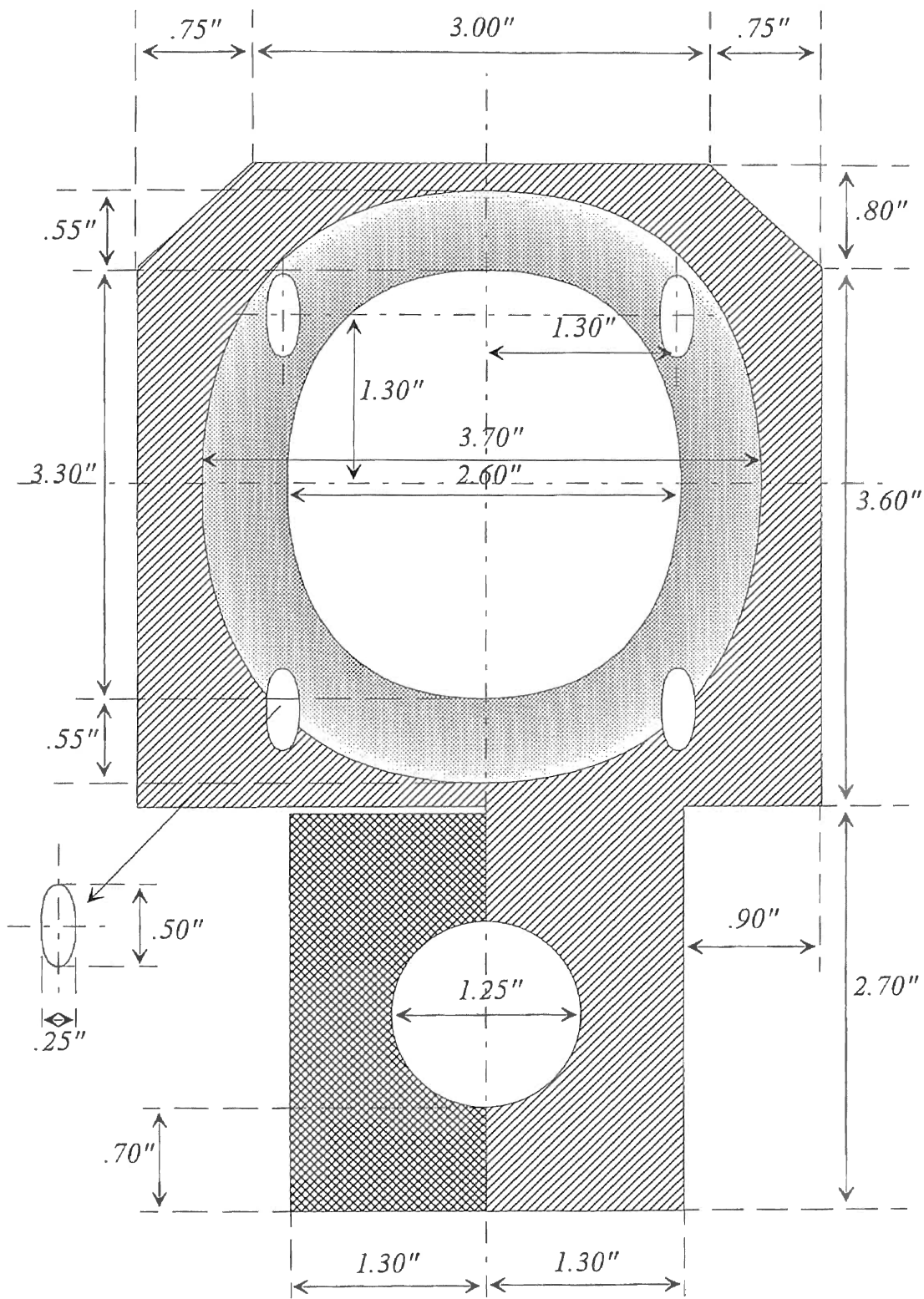
Motor Mount Specifications



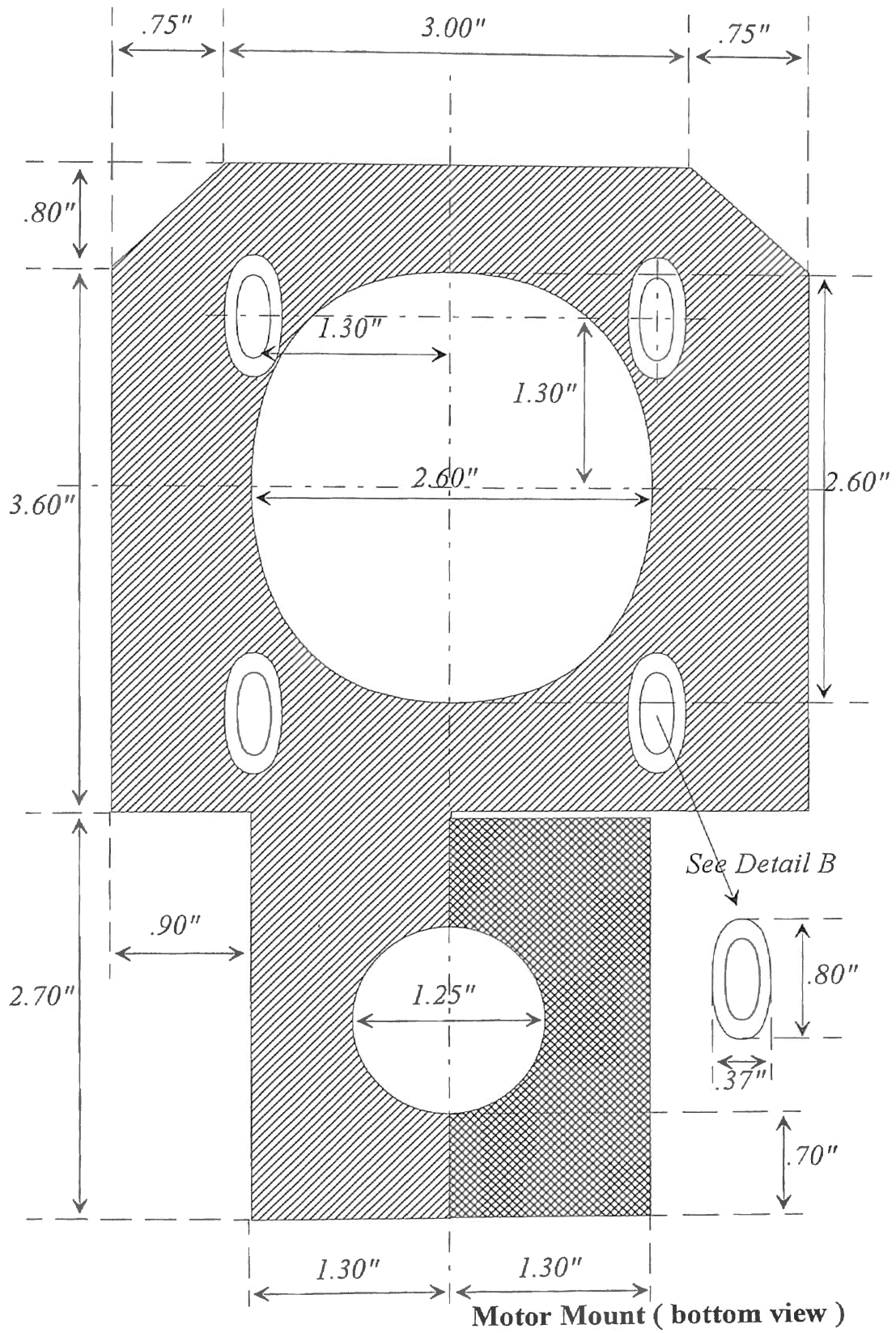
Motor Dimensions (Bottom View)

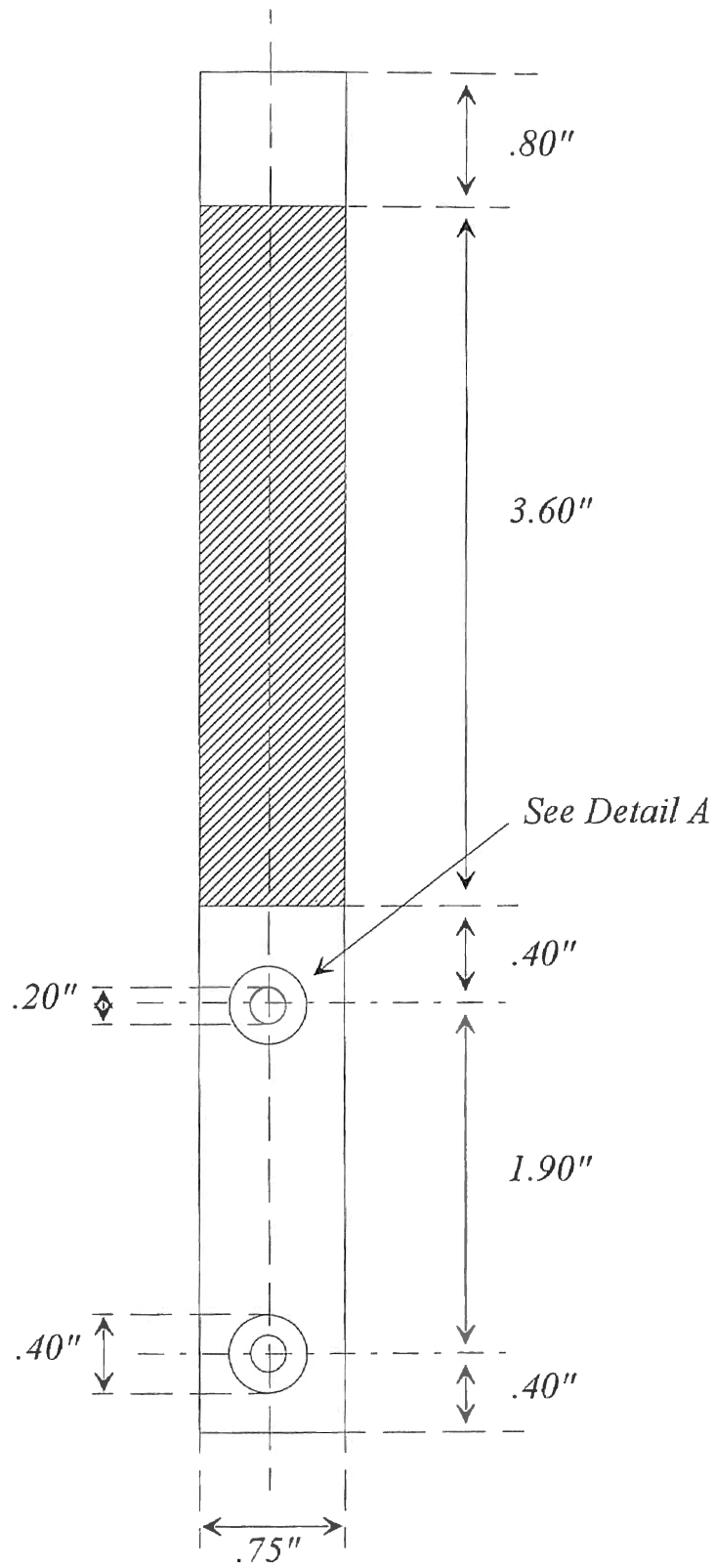


Motor Dimensions (Side View)

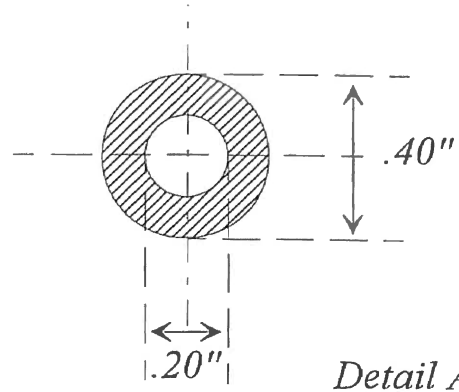


Motor Mount (top view)





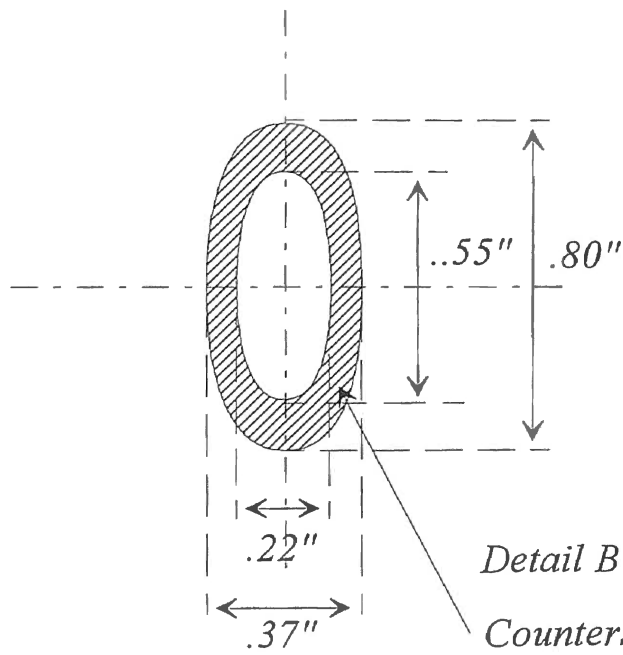
Motor Mount (Side View)



Detail A

Countersink depth .50"

2 X Scale



Detail B

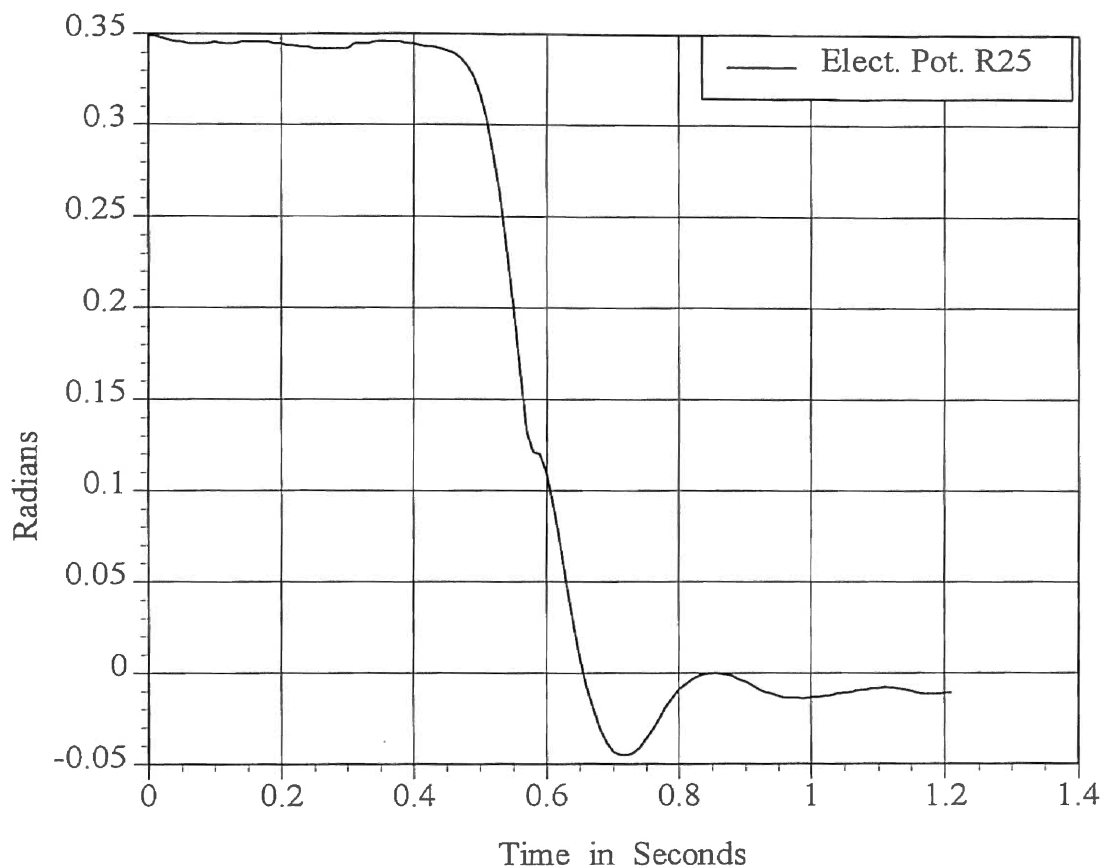
Countersink depth .25"

2 X Scale

Motor Mount (Detail A&B)

Appendix D

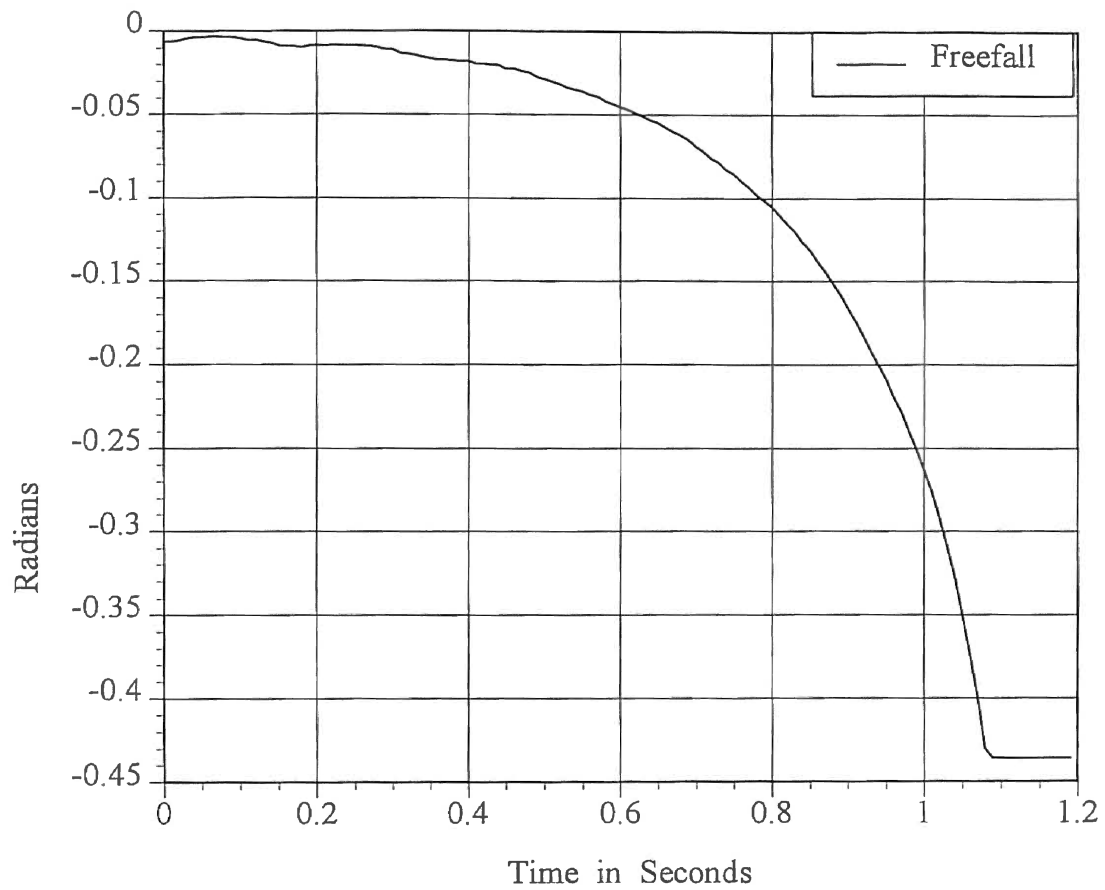
Inclinometer Electrolytic Potentiometer Test Data



**Inclinometer Dynamics,
Forced Fall Test, Position Vs Time
Electrolytic Potentiometer**

Figure D - 1

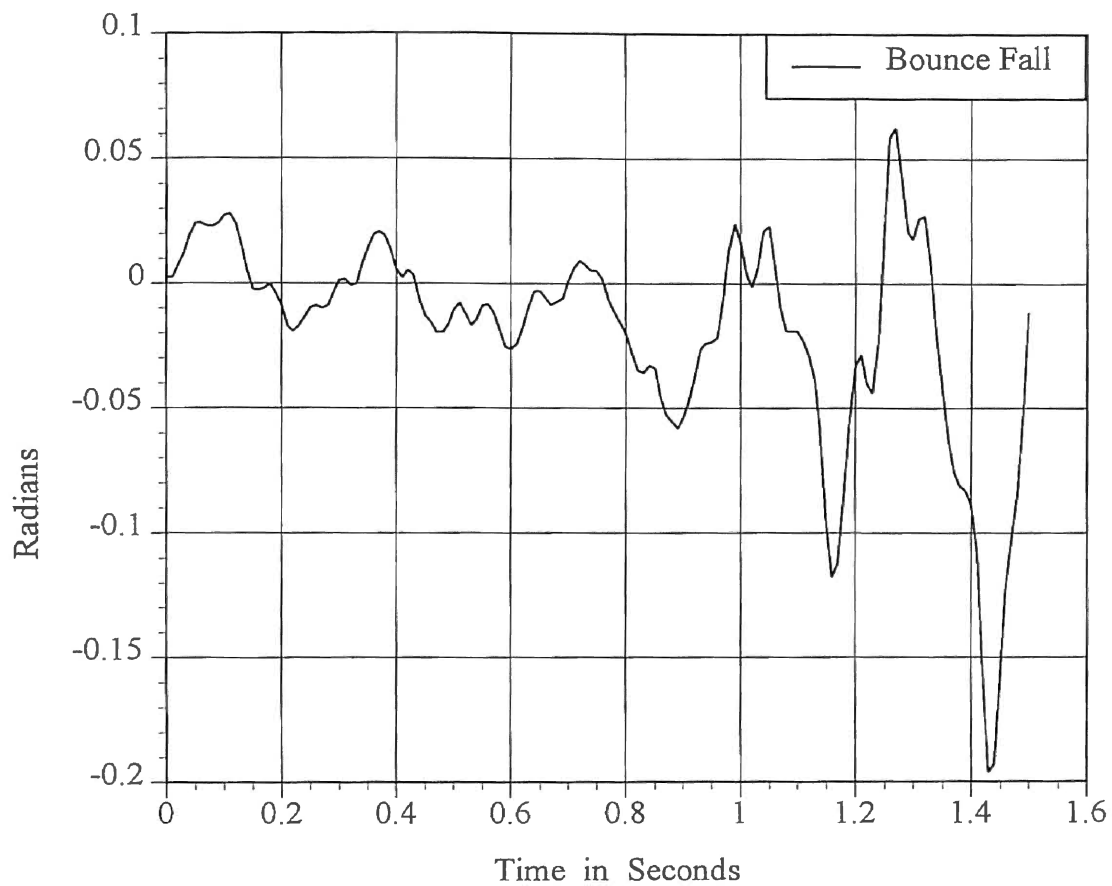
Note: In this test the Electrolytic Potentiometer was concentrically mounted on a rotating shaft and its rotation was forced through 25 degrees. Damping ratio and natural frequency were determined from this test.



**Inclinometer Dynamics,
Bicycle Free Fall Test, Position Vs Time
Electrolytic Potentiometer**

Figure D - 2

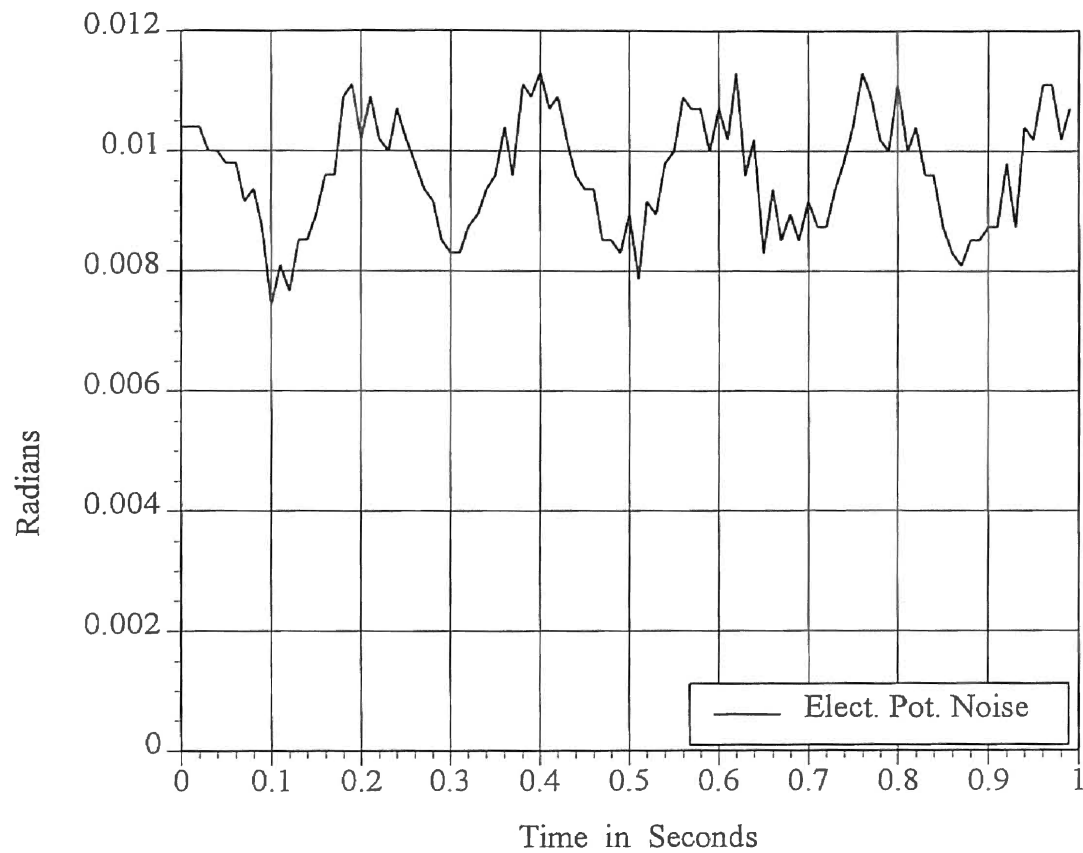
Note: In this test the Electrolytic Potentiometer was mounted on the bicycle which was allowed to fall without vibration from the roller platform. This test was used as a comparison to the same test with the roller platform in motion.



**Inclinometer Dynamics,
Free Fall with Roller Bounce Test, Position Vs Time
Electrolytic Potentiometer**

Figure D - 3

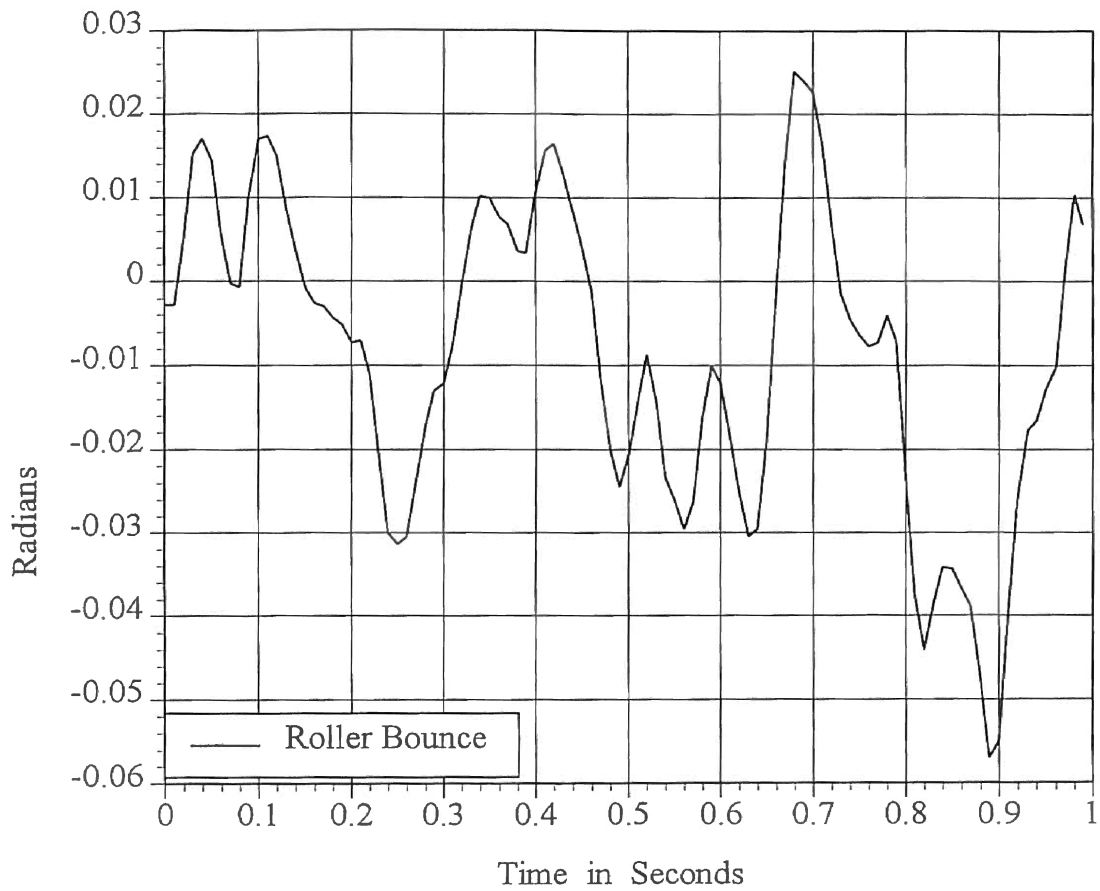
Note: In this test the Electrolytic Potentiometer was mounted on the bicycle which was allowed to fall with vibration from the roller platform. This test was used to show the effect of roller vibration on the inclinometer sensor.



**Inclinometer Dynamics,
Background Noise Test, Position Vs Time
Electrolytic Potentiometer**

Figure D - 4

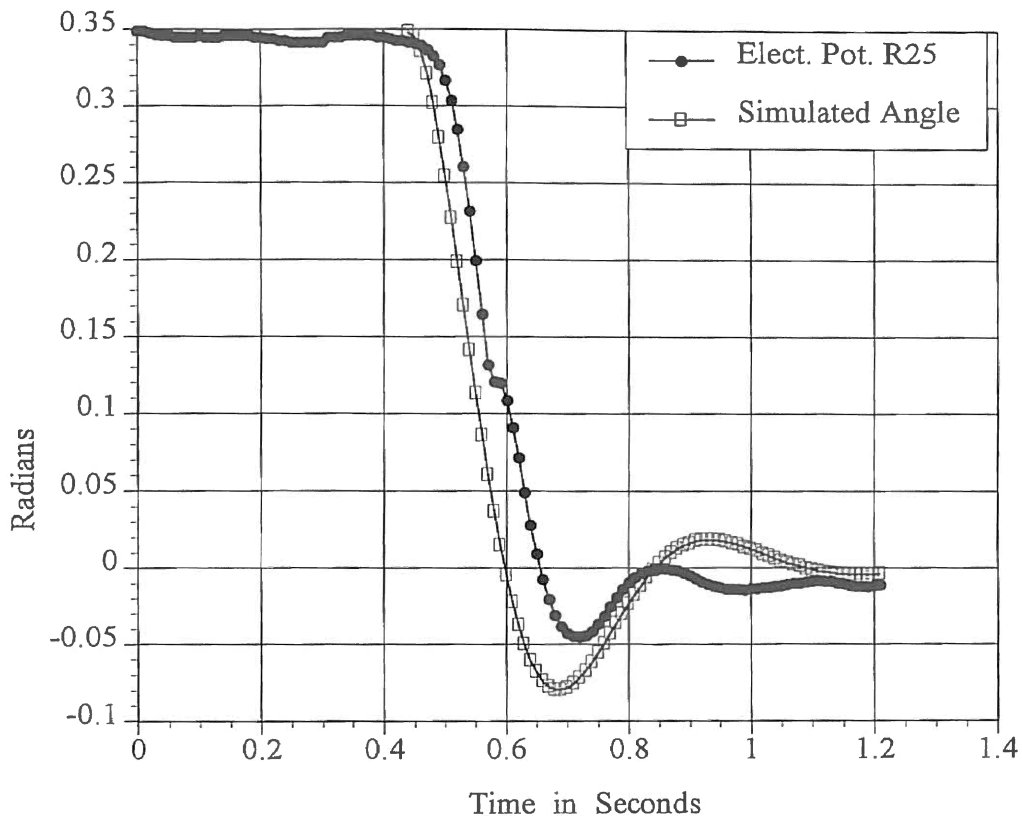
Note: In this test the Electrolytic Potentiometer was mounted on the motionless bicycle. This test was used to determine the intensity of background noise the inclinometer and driver circuit would read with no input.



**Inclinometer Dynamics,
Roller Bounce Noise Test, Position Vs Time
Electrolytic Potentiometer**

Figure D - 5

Note: In this test the Electrolytic Potentiometer was mounted on the bicycle with the rollers in motion. This test was used to study the effect of roller platform vibration on the inclinometer.



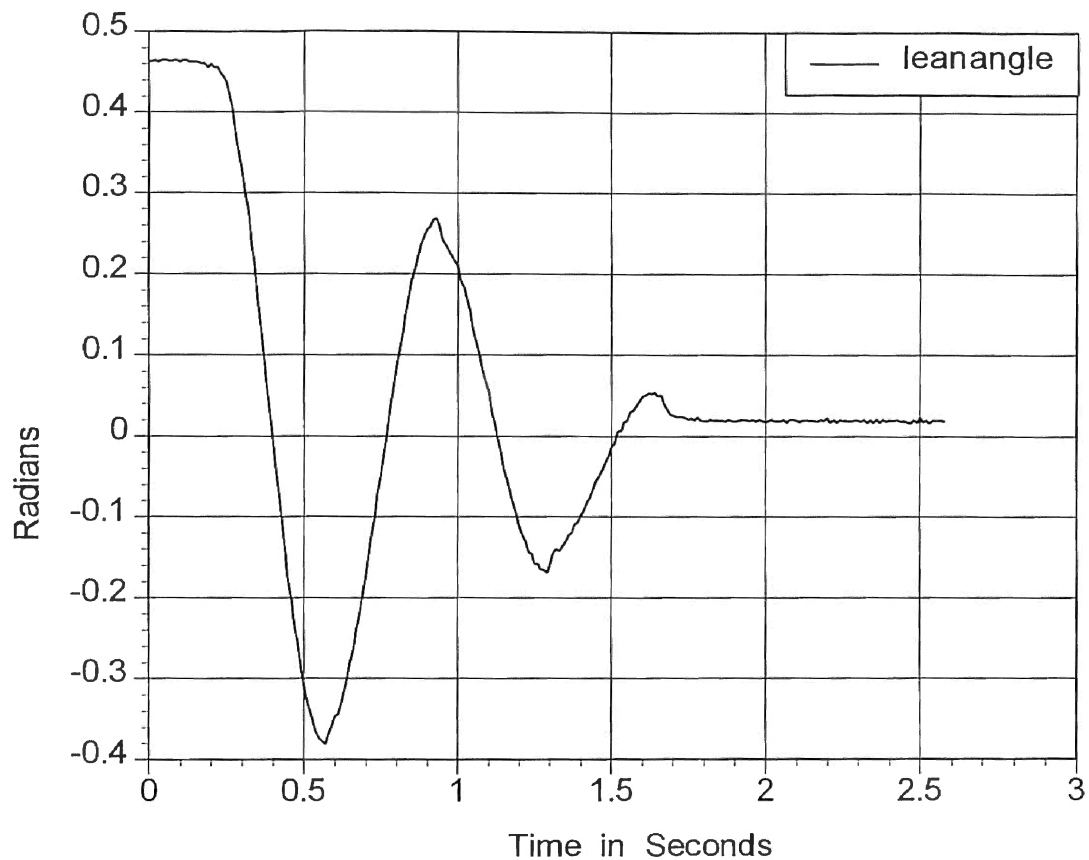
**Inclinometer Dynamics,
Actual and Simulated Fall, Position Vs Time
Electrolytic Potentiometer**

Figure D - 6

Note: In this test the Electrolytic Potentiometer was concentrically mounted on a rotating shaft and its rotation was forced through 25 degrees. Here actual test data were compared to the Matrix_x simulation of inclinometer position Vs. time.

Appendix E

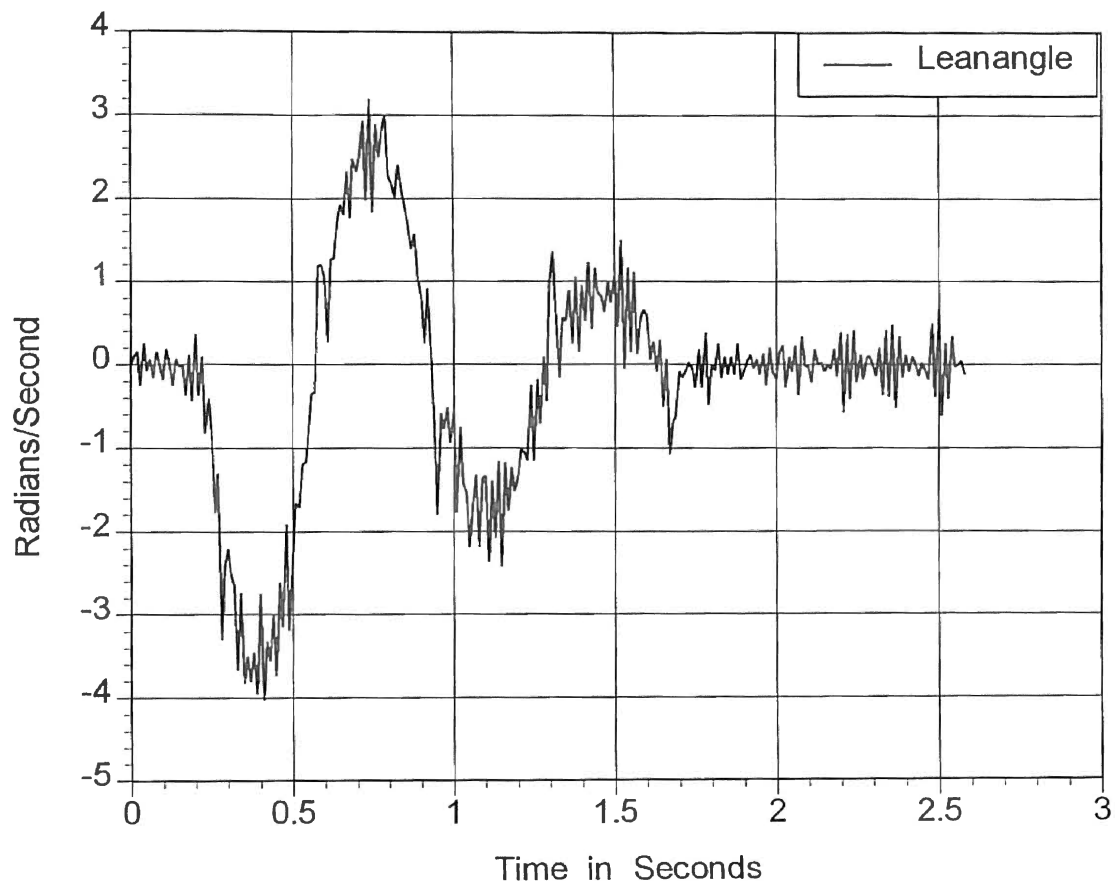
Inclinometer Simple Pendulum Test Data



**Inclinometer Dynamics,
Free Fall Test, Position Vs. Time,
Simple Pendulum**

Figure E -1

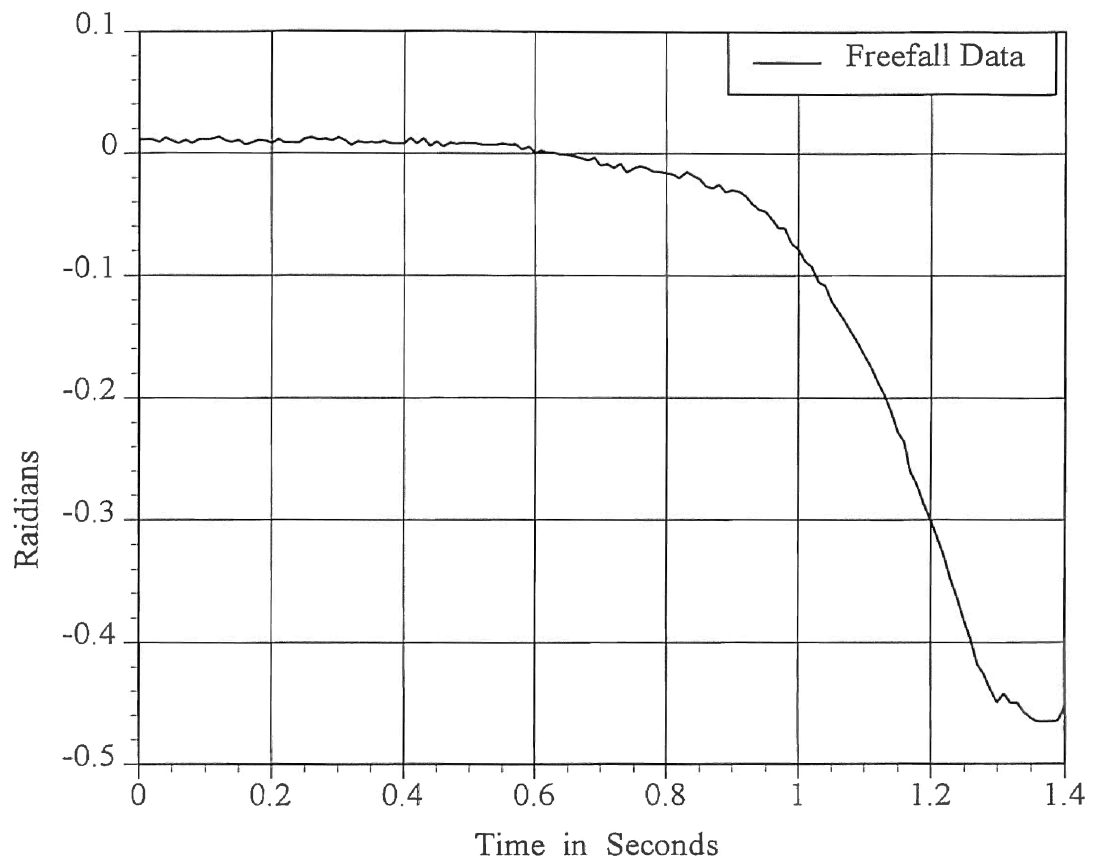
Note: The Simple Pendulum for this test was mounted apart from the Bicycle. In this test the pendulum was dropped from a displacement of 25 degrees (.44 radians) and allowed to come to rest.



**Inclinometer Dynamics,
Free Fall Test, Velocity Vs. Time,
Simple Pendulum**

Figure E - 2

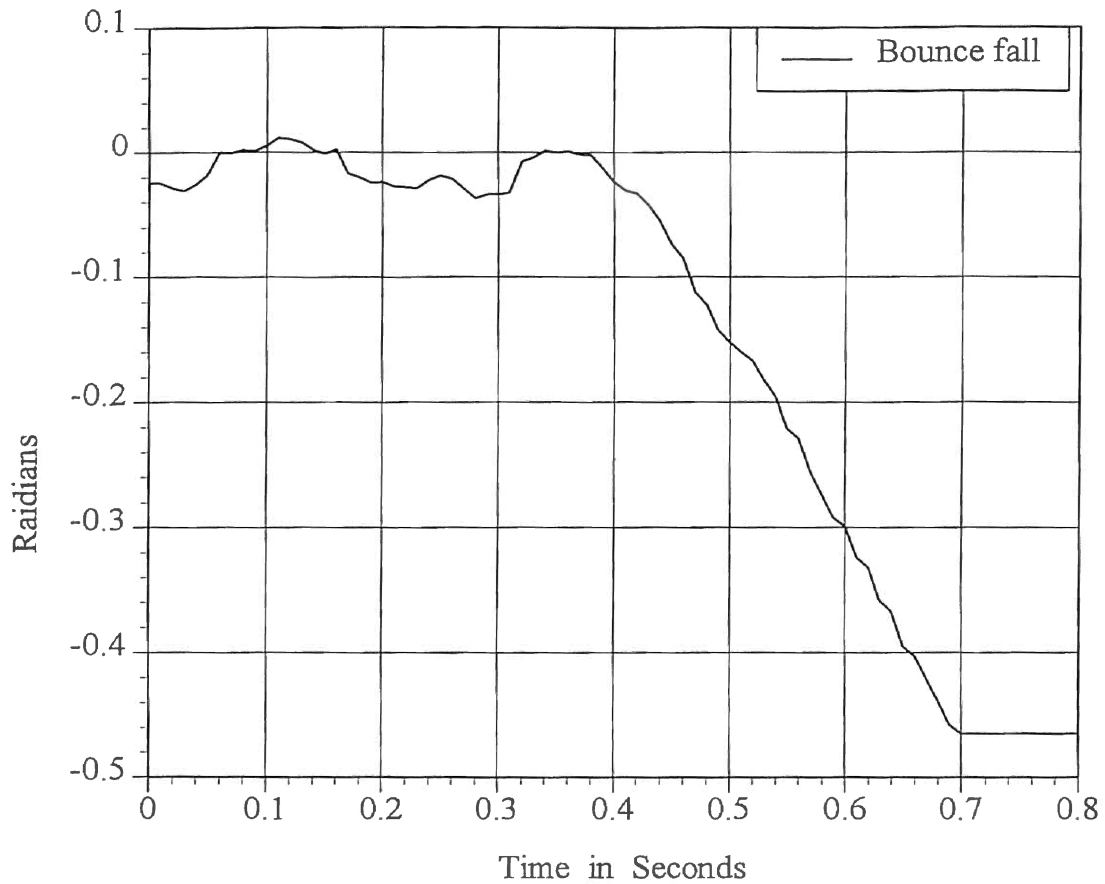
Note: The Simple Pendulum for this test was mounted apart from the Bicycle. In this test the pendulum was dropped from a displacement of 25 degrees (.44 radians) and allowed to come to rest.



**Inclinometer Dynamics,
Bicycle Free Fall Test, Position Vs. Time,
Simple Pendulum**

Figure E - 3

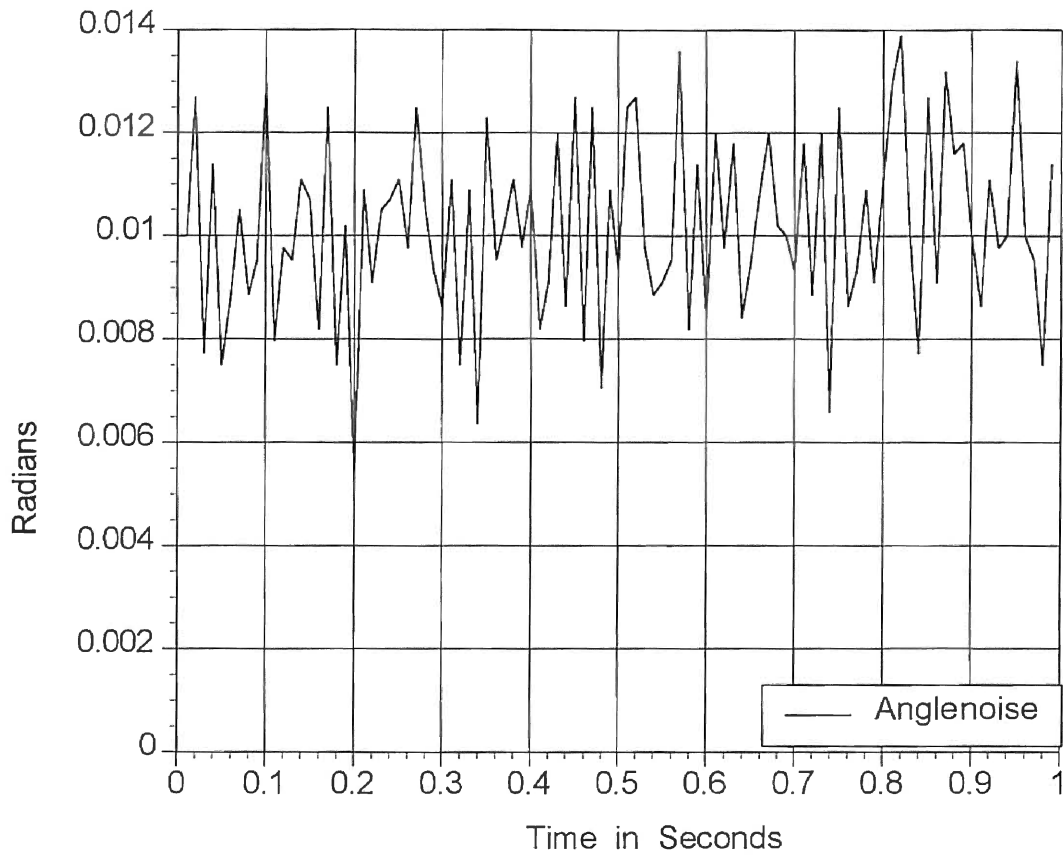
Note: The Simple Pendulum for this test was mounted on the bicycle which was allowed to fall without vibration from the roller platform. This test may be compared to the same test with the roller platform in motion.



**Inclinometer Dynamics,
Free Fall with Roller Bounce Test, Position Vs. Time,
Simple Pendulum**

Figure E - 4

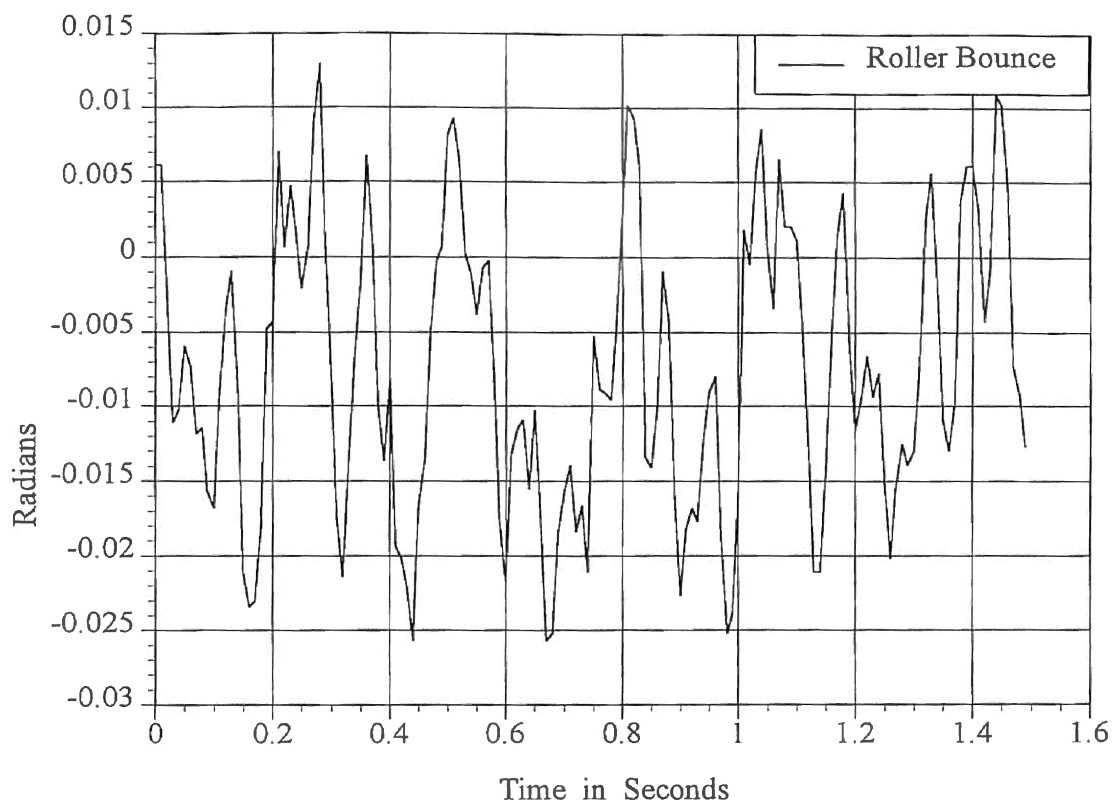
Note: The Simple Pendulum for this test was mounted on the bicycle which was allowed to fall without vibration from the roller platform. This test may be compared to the same test with the roller platform in motion.



**Inclinometer Dynamics,
Background Noise Test, Position Vs. Time,
Simple Pendulum**

Figure E - 5

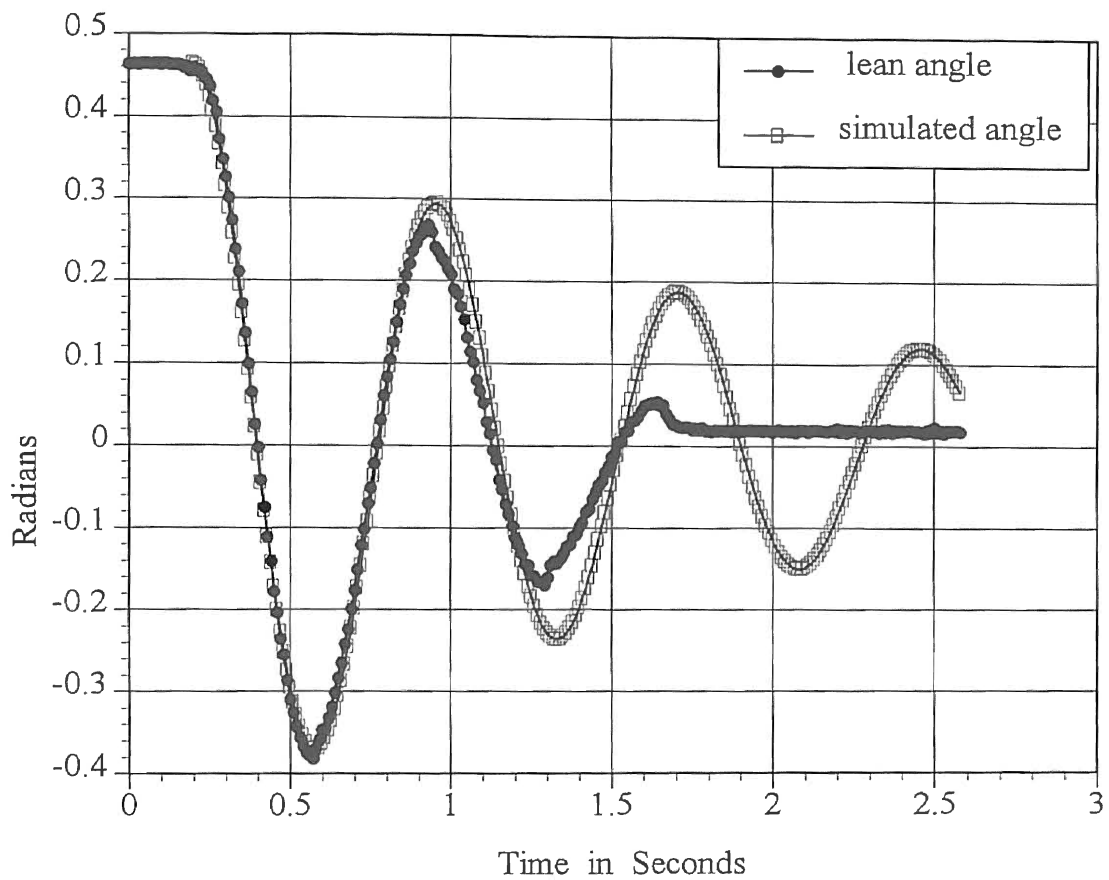
Note: The Simple Pendulum for this test was mounted on the motionless bicycle. This test was used to determine the intensity of background noise the inclinometer and driver circuit would read with no input.



**Inclinometer Dynamics,
Roller Bounce Noise Test, Position Vs. Time,
Simple Pendulum**

Figure E - 6

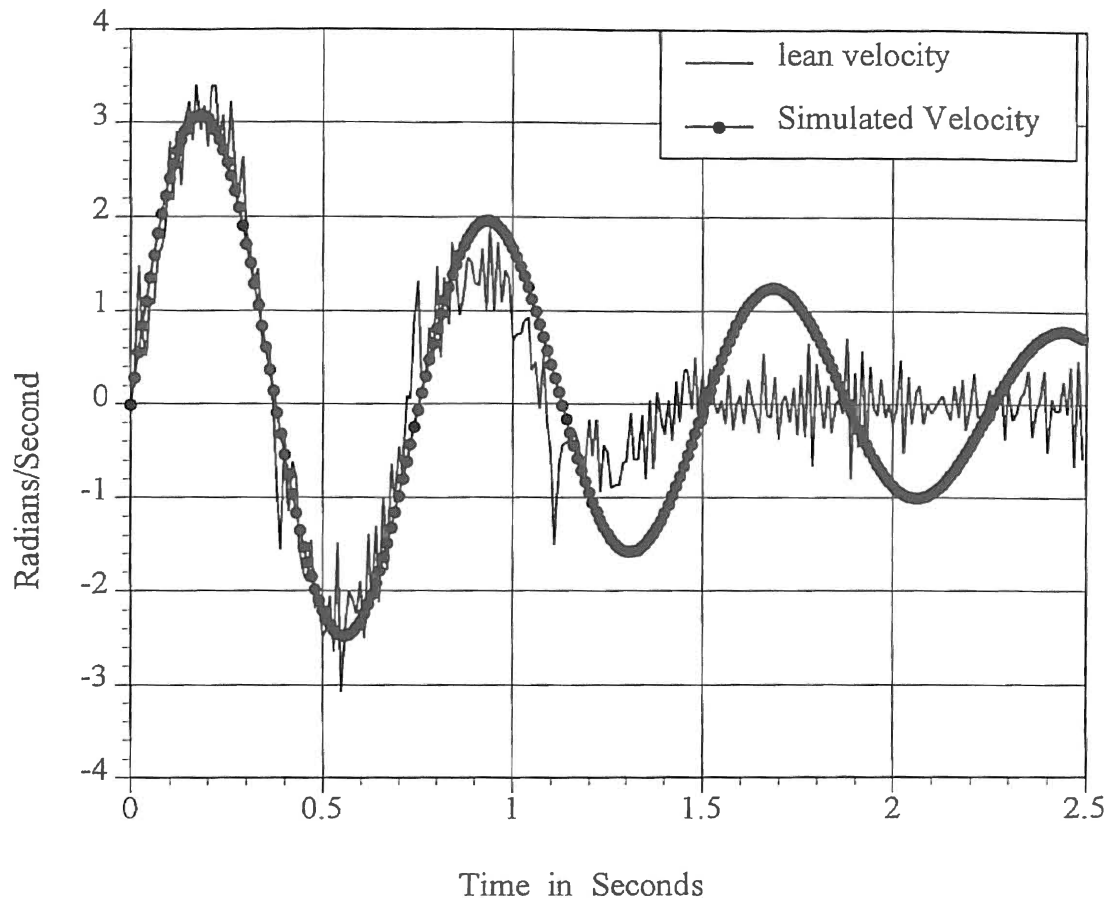
Note: The Simple Pendulum for this test was mounted on the bicycle with the rollers in motion. This test was used to study the effect of roller platform vibration on the inclinometer.



**Inclinometer Dynamics,
Actual and Simulated Fall, Position Vs. Time,
Simple Pendulum**

Figure E - 7

Note: The Simple Pendulum for this test was allowed to fall through 25 degrees (.44 radians) and come to rest. The actual data were compared to a Matrix_x simulation of the inclinometer.



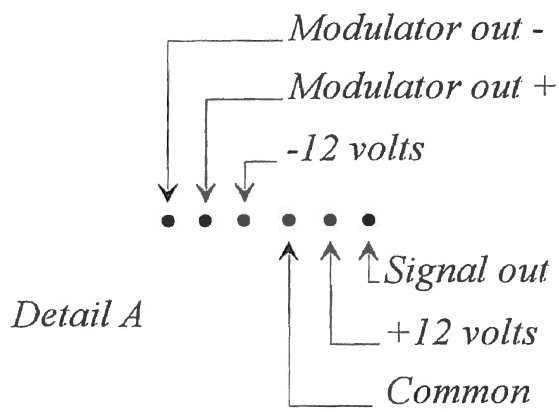
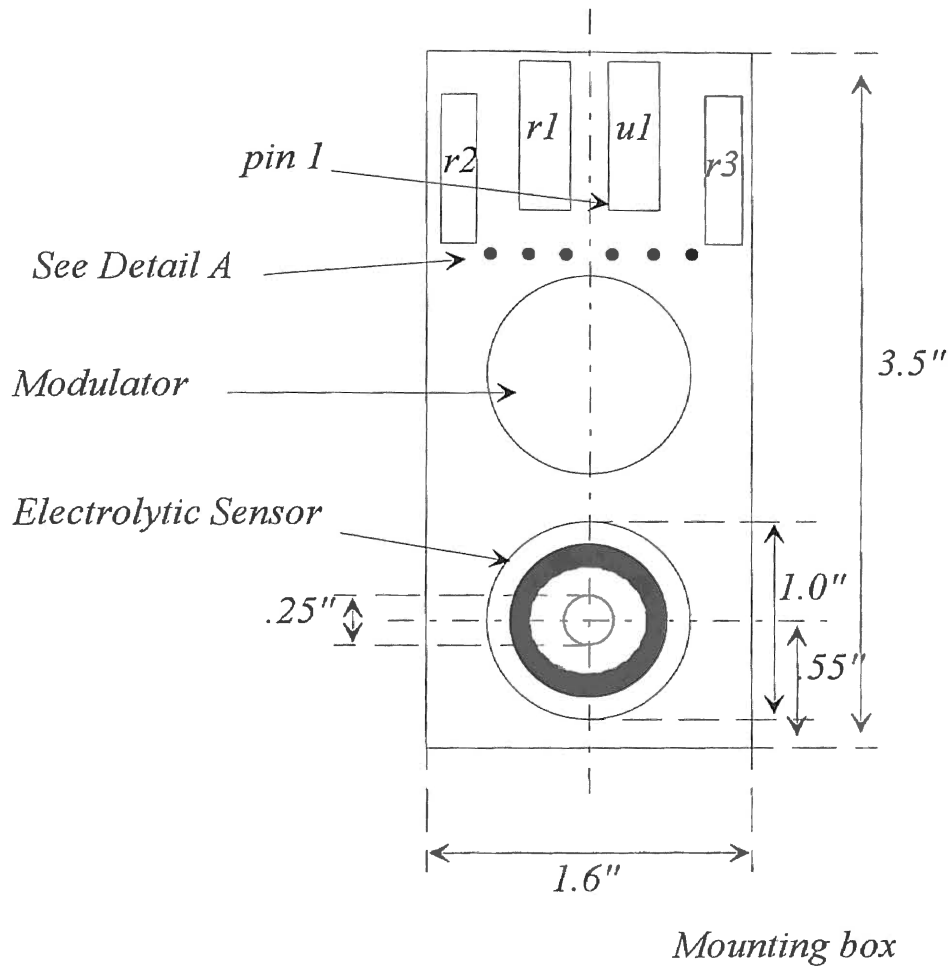
**Inclinometer Dynamics,
Actual and Simulated Fall, Velocity Vs. Time,
Simple Pendulum**

Figure E - 8

Note: The Simple Pendulum for this test was allowed to fall through 25 degrees (.44 radians) and come to rest. The actual data were compared to a Matrix_x simulation of the inclinometer.

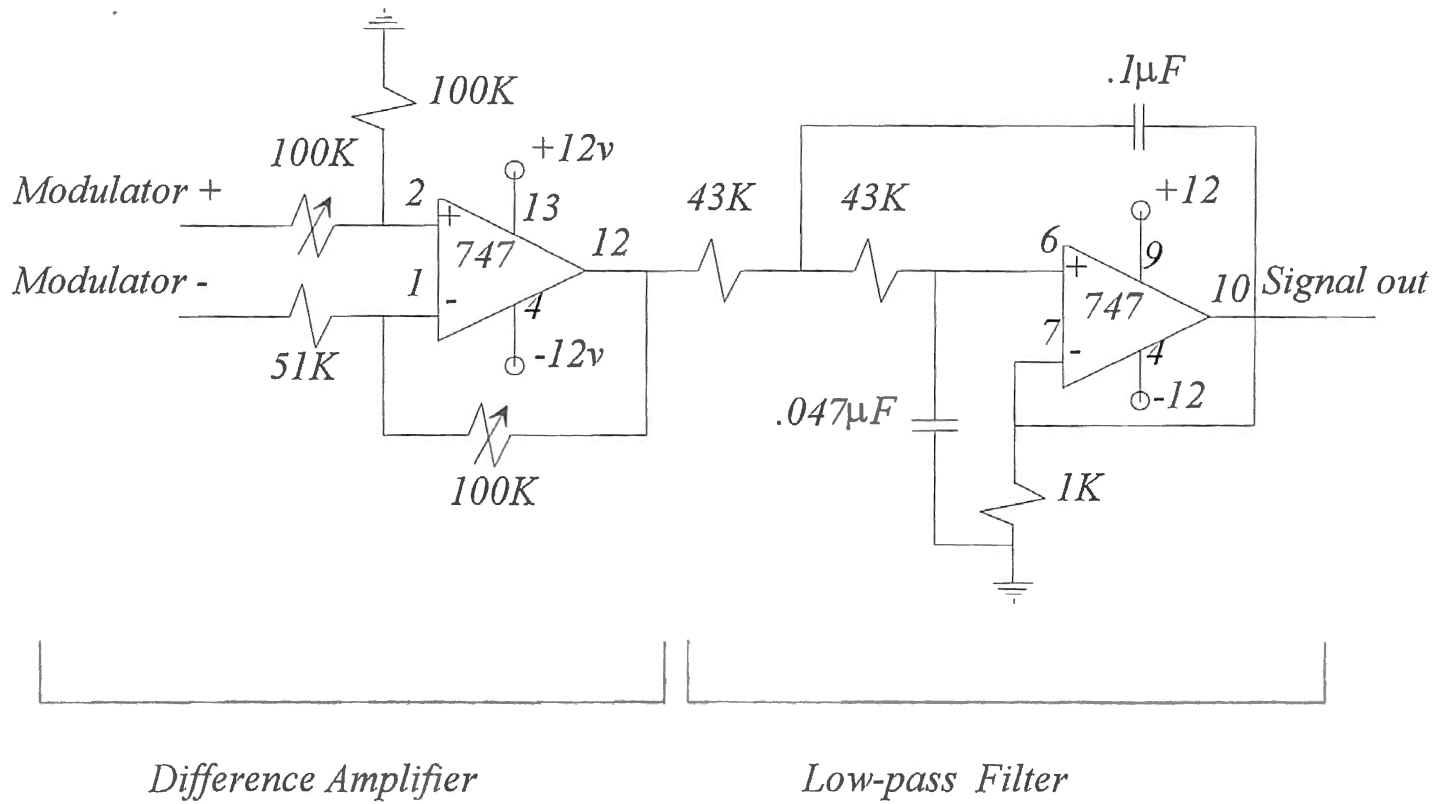
Appendix F

Inclinometer Electrolytic Potentiometer Hardware



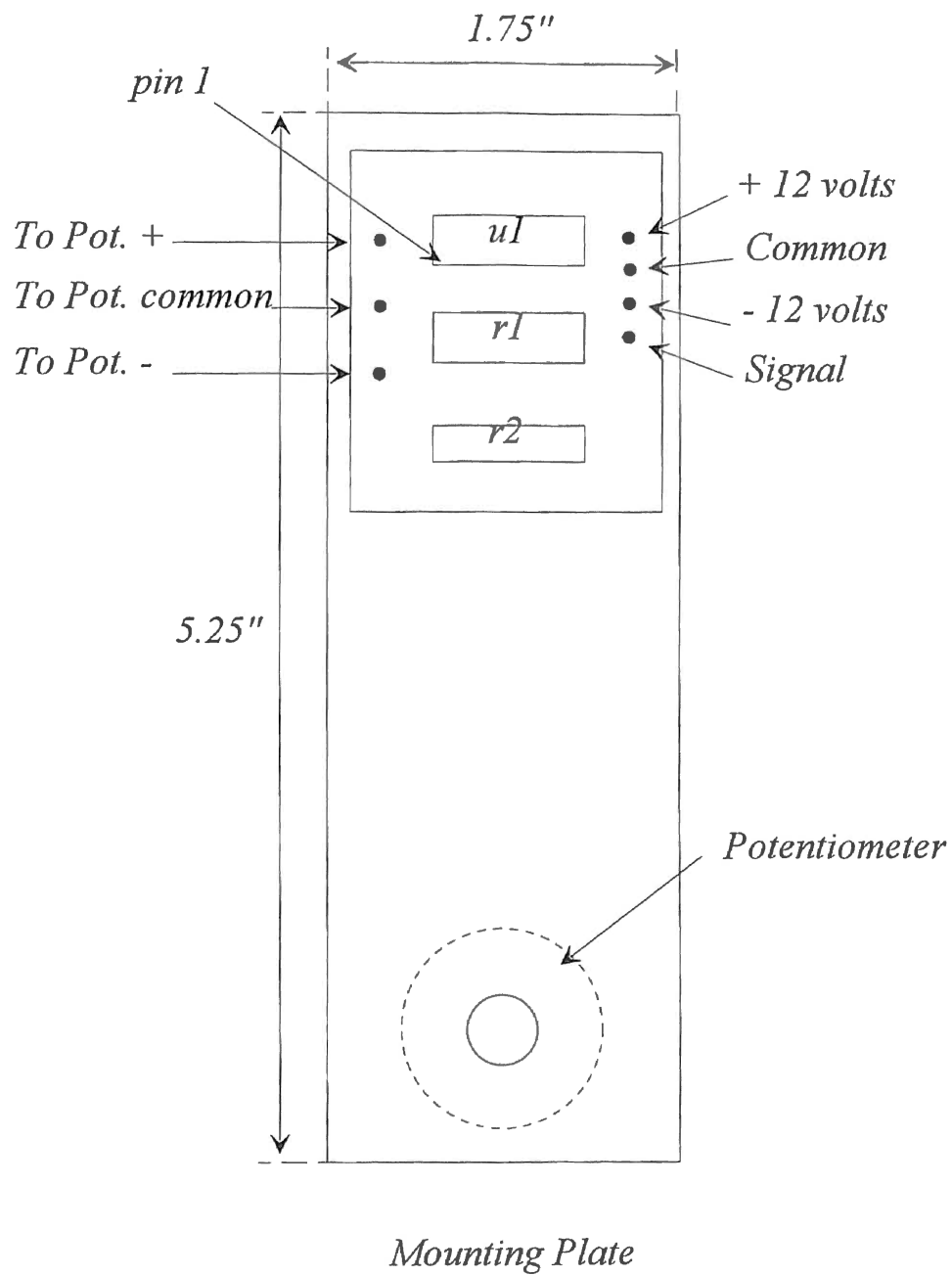
Electrolytic Potentiometer (Rear View)

Signal Conditioning (Electrolytic Potentiometer)

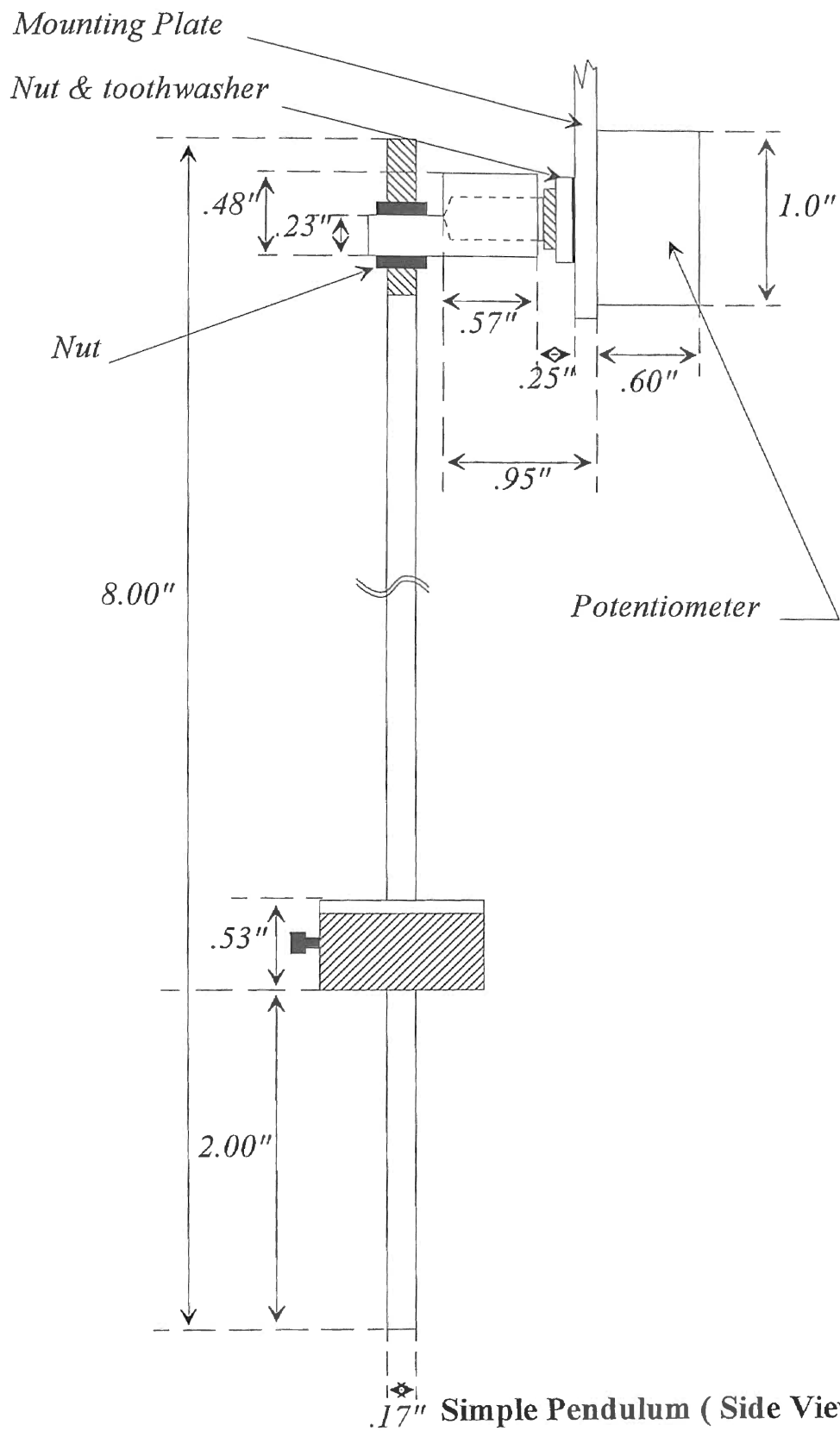


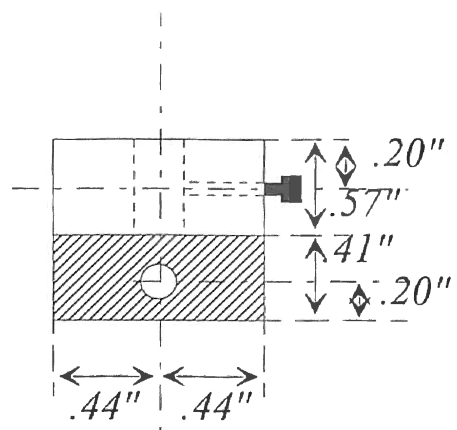
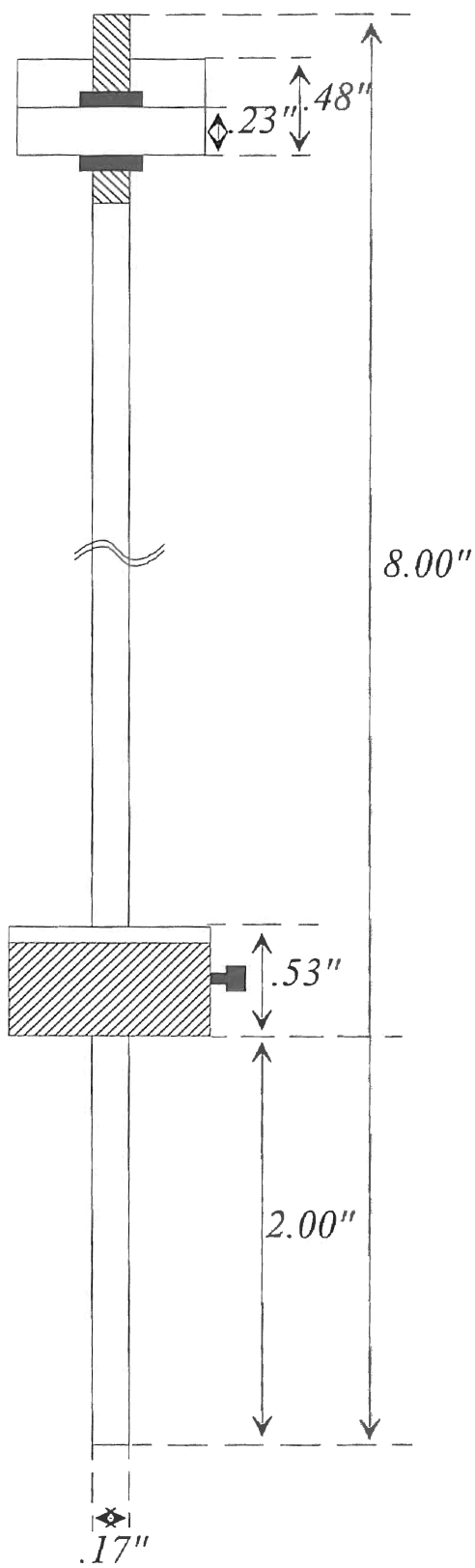
Appendix G

Inclinometer Simple Pendulum Hardware

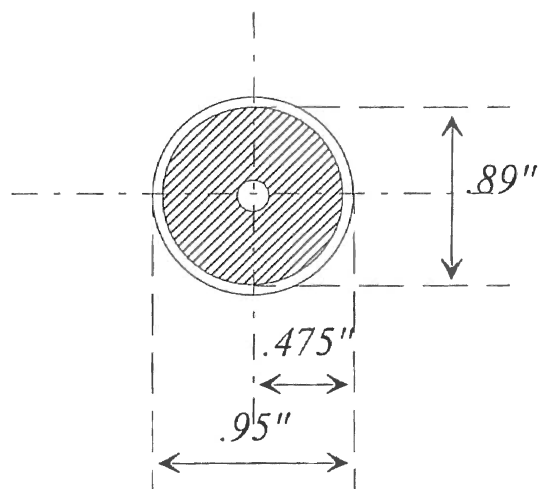


Simple Pendulum (Front View)



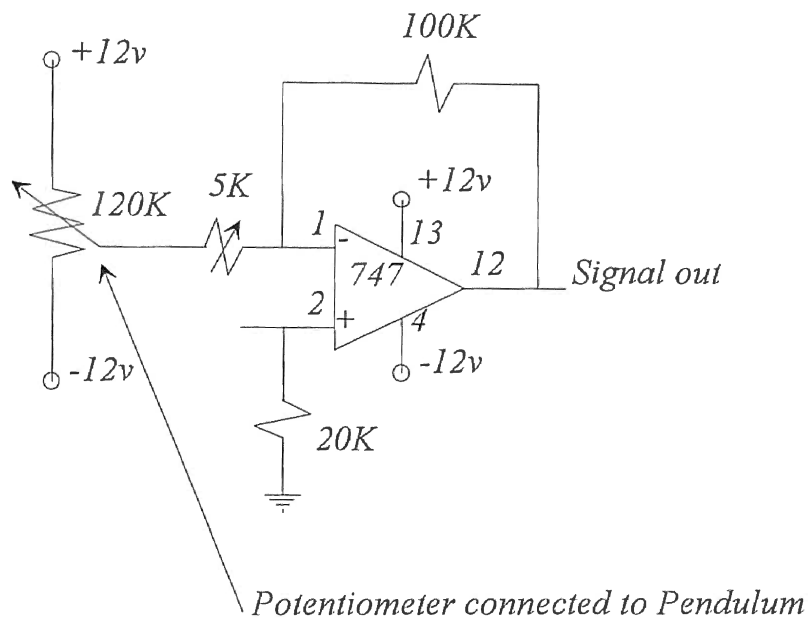


Mount bracket



Pendulum mass

Simple Pendulum



Signal Conditioning (Simple Pendulum)

Appendix H

Derivation of Model Equations

Bicycle Constants

- l_1 : Length from ground to bicycle center of mass.
- l_2 : Length from ground to pivot point of inclinometer.
- l_3 : Pendulum length.
- l_4 : Bicycle wheelbase.
- g : Gravity
- m_1 : Bicycle mass
- m_2 : Pendulum mass
- v : Bicycle velocity
- R : Bicycle turning radius
- ω_n : Inclinometer natural frequency
- z : Inclinometer damping ratio

Bicycle Systems Variables

- α : Bicycle steering angle (steering straight ahead, $\alpha = 0$)
- ϵ : Inclinometer error angle measurement

Bicycle State Variables

- θ : Bicycle lean angle (vertical bicycle, $\theta = 0$)
- $\dot{\theta}$: Bicycle lean velocity
- θ_M : Steering motor angle (steering straight ahead, $\theta_M = 0$)
- $\dot{\theta}_M$: Steering motor velocity
- ρ : Pendulum lean angle (pendulum straight down, $\rho = 0$)
- $\dot{\rho}$: Pendulum lean velocity

Derivation Variables

- E_K : Bicycle kinetic energy
- E_U : Bicycle potential energy
- v_1 : Velocity of bicycle mass perpendicular to forward v
- v_2 : Velocity of pendulum mass perpendicular to forward v
- x : Horizontal displacement of pendulum mass
- y : Vertical displacement of pendulum mass

Derivation of Bicycle and Inclinometer System

Reprinted from

Riderless Bicycle Control Design

by

Mohammad B. Menhaj

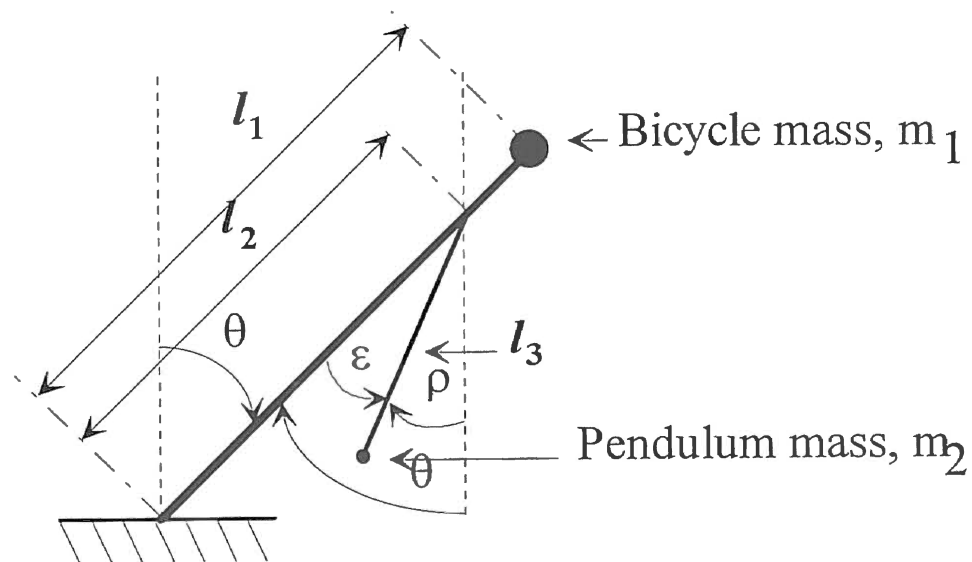


Diagram of Bicycle Constants

Figure H-1

$$\epsilon = \theta - \rho \quad (\text{H-1})$$

Kinetic Energy

$$E_K = \frac{1}{2}(m_1 v_1^2) + \frac{1}{2}(m_2 v_2^2) \quad (\text{H-2})$$

$$v_1 = l_1 \frac{d\theta}{dt} \quad (\text{H-3})$$

To find v_2 determine the coordinates of mass m_2

$$x = l_2 \sin \theta - l_3 \sin \rho \quad (\text{H-4})$$

$$y = l_2 \cos \theta - l_3 \cos \rho \quad (\text{H - 5})$$

$$\frac{dx}{dt} = (l_2 \cos \theta) \frac{d\theta}{dt} - (l_3 \cos \rho) \frac{d\rho}{dt} \quad (\text{H - 6})$$

$$v_2^2 = \left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 \quad (\text{H - 7})$$

$$v_2^2 = (l_2 \frac{d\theta}{dt} \cos \theta - l_3 \frac{d\rho}{dt} \cos \rho)^2 + (-l_2 \frac{d\theta}{dt} \sin \theta + l_3 \frac{d\rho}{dt} \sin \rho)^2 \quad (\text{H - 8})$$

$$\begin{aligned} v_2^2 = & l_2^2 \left(\frac{d\theta}{dt}\right)^2 \cos^2 \theta + l_3^2 \left(\frac{d\rho}{dt}\right)^2 \cos^2 \rho - 2l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \cos \theta \cos \rho \\ & + l_2^2 \left(\frac{d\theta}{dt}\right)^2 \sin^2 \theta + l_3^2 \left(\frac{d\rho}{dt}\right)^2 \sin^2 \rho - 2l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \sin \theta \sin \rho \end{aligned} \quad (\text{H - 9})$$

$$v_2^2 = l_2^2 \left(\frac{d\theta}{dt}\right)^2 + l_3^2 \left(\frac{d\rho}{dt}\right)^2 - 2l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} [\cos \theta \sin \rho + \sin \theta \cos \rho] \quad (\text{H - 10})$$

$$v_2^2 = l_2^2 \left(\frac{d\theta}{dt}\right)^2 + l_3^2 \left(\frac{d\rho}{dt}\right)^2 - 2l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \cos(\theta - \rho) \quad (\text{H - 11})$$

$$\begin{aligned} E_k = & \frac{1}{2} m_1 \left(l_1 \frac{d\theta}{dt}\right)^2 + \frac{1}{2} m_2 \left[l_2^2 \left(\frac{d\theta}{dt}\right)^2 + l_3^2 \left(\frac{d\rho}{dt}\right)^2 \right. \\ & \left. - 2l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \cos(\theta - \rho)\right] \end{aligned} \quad (\text{H - 12})$$

Potential Energy

$$E_U = -m_1 g l_1 [1 - \cos \theta] - m_2 g [l_2 [1 - \cos \theta] - l_3 [1 - \cos \rho]]$$

Due to gravity with $\theta=0$, $\rho=0$ reference point

$$+m_1 \frac{v^2}{R} l_1 \sin \theta + m_2 \frac{v^2}{R} [l_2 \sin \theta - l_3 \sin \rho] \quad (\text{H - 13})$$

Due to centrifugal force with $\theta=0, \rho=0$, reference point

Lagrangian

$$L = E_K - E_U \quad (\text{H - 14})$$

$$\begin{aligned} L = & \frac{1}{2} m_1 l_1^2 \left(\frac{d\theta}{dt} \right)^2 + \frac{1}{2} m_2 l_2^2 \left(\frac{d\theta}{dt} \right)^2 + \frac{1}{2} m_2 l_3^2 \left(\frac{d\rho}{dt} \right)^2 - m_2 l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \cos(\theta - \rho) \\ & + m_1 g l_1 [1 - \cos \theta] + m_2 g l_2 [1 - \cos \theta] - m_2 g l_3 [1 - \cos \rho] \\ & - \frac{m_1 v^2}{R} l_1 \sin \theta - m_2 \frac{v^2}{R} l_2 \sin \theta + m_2 \frac{v^2}{R} l_3 \sin \rho \quad (\text{H - 15}) \end{aligned}$$

Lagrange's Equations of Motion

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] - \frac{\partial L}{\partial \theta} = \quad (\text{H - 16})$$

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\rho}} \right] - \frac{\partial L}{\partial \rho} = \quad (\text{H - 17})$$

For θ

$$\frac{\partial L}{\partial \theta} = m_1 l_1^2 \frac{d\theta}{dt} + m_2 l_2^2 \frac{d\theta}{dt} - m_2 l_2 l_3 \frac{d\rho}{dt} \cos(\theta - \rho) \quad (\text{H - 18})$$

$$\frac{\partial L}{\partial \theta} = (m_1 l_1^2 + m_2 l_2^2) \frac{d\theta}{dt} - m_2 l_2 l_3 \frac{d\rho}{dt} \cos(\theta - \rho) \quad (\text{H - 19})$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] &= (m_1 l_1^2 + m_2 l_2^2) \frac{d^2 \theta}{dt^2} - m_2 l_2 l_3 \frac{d\rho}{dt} \cos(\theta - \rho) \\ &\quad + m_2 l_2 l_3 \frac{d\rho}{dt} \sin(\theta - \rho) \left[\frac{d\theta}{dt} - \frac{d\rho}{dt} \right] \quad (\text{H - 20}) \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] &= (m_1 l_1^2 + m_2 l_2^2) \frac{d^2 \theta}{dt^2} - m_2 l_2 l_3 \frac{d\rho^2}{dt^2} \cos(\theta - \rho) \\ &\quad + m_2 l_2 l_3 \frac{d\rho}{dt} \frac{d\theta}{dt} \sin(\theta - \rho) - m_2 l_2 l_3 \left(\frac{d\rho}{dt} \right)^2 \sin(\theta - \rho) \quad (\text{H - 21}) \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial \theta} &= m_2 l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \sin(\theta - \rho) + m_1 g l_1 \sin \theta + m_2 g l_2 \sin \theta \\ &\quad - m_1 \frac{v^2}{R} l_1 \cos \theta - m_2 \frac{v^2}{R} l_2 \cos \theta \quad (\text{H - 22}) \end{aligned}$$

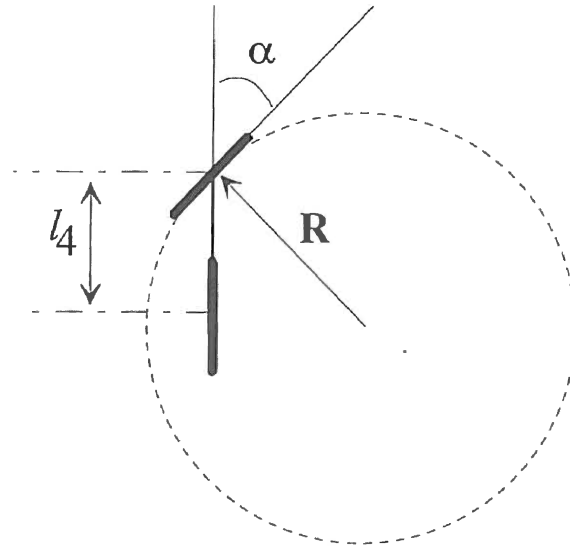
$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\theta}} \right] - \frac{\partial L}{\partial \theta} &= 0 = (m_1 l_1^2 + m_2 l_2^2) \frac{d^2}{dt^2} \\ &\quad - m_2 l_2 l_3 \left[\frac{d^2 \rho}{dt^2} \cos(\theta - \rho) + \left(\frac{d\rho}{dt} \right)^2 \sin(\theta - \rho) \right] \\ &\quad - (m_1 l_1 + m_2 l_2) g \sin \theta + (m_1 l_1 + m_2 l_2) \frac{v^2}{R} \cos \theta \quad (\text{H - 23}) \end{aligned}$$

For this equation assume that the mass m_2 is very small in comparison with mass m_1 .

$$m_1 l_1^2 \frac{d^2 \theta}{dt^2} - m_1 l_1 g \sin \theta + m_1 l_1 \frac{v^2}{R} \cos \theta = \quad (\text{H - 24})$$

$$\frac{d^2 \theta}{dt^2} - \frac{g}{l_1} \sin \theta + \frac{v^2}{l_1 R} \cos \theta = \quad (\text{H - 25})$$

R is related to the steering angle.



Bicycle Top-down view

Figure H - 2

If the angle α is small the R in this diagram is almost equal to the R to the Center of mass.

$$R = l_4 / \sin \alpha \quad (\text{H - 26})$$

$$\frac{d^2 \theta}{dt^2} - \frac{g}{l_1} \sin \theta + \frac{v^2}{l_1 l_4} \sin \alpha \cos \theta = 0 \quad (\text{H - 27}) \quad \text{BICYCLE EQUATION}$$

For ρ

$$\frac{\partial L}{\partial \dot{\rho}} = m_2 l_3^2 \frac{d\rho}{dt} - m_2 l_2 l_3 \frac{d\theta}{dt} \cos(\theta - \rho) \quad (\text{H - 28})$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\rho}} \right] &= m_2 l_3^2 \frac{d^2 \rho}{dt^2} - m_2 l_2 l_3 \frac{d^2 \theta}{dt^2} \cos(\theta - \rho) \\ &\quad + m_2 l_2 l_3 \frac{d\theta}{dt} \sin(\theta - \rho) \left(\frac{d\theta}{dt} - \frac{d\rho}{dt} \right) \quad (\text{H - 29}) \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\rho}} \right] &= m_2 l_3^2 \frac{d^2 \rho}{dt^2} - m_2 l_2 l_3 \frac{d^2 \theta}{dt^2} \cos(\theta - \rho) \\ &\quad + m_2 l_2 l_3 \left(\frac{d\theta}{dt} \right)^2 \sin(\theta - \rho) - m_2 l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \sin(\theta - \rho) \quad (\text{H - 30}) \end{aligned}$$

$$\frac{\partial L}{\partial \rho} = -m_2 l_2 l_3 \frac{d\theta}{dt} \frac{d\rho}{dt} \sin(\theta - \rho) - m_2 g l_3 \sin \rho + m_2 \frac{v^2}{R} \cos \rho \quad (\text{H - 31})$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{\rho}} \right] - \frac{\partial L}{\partial \rho} &= m_2 l_3^2 \frac{d^2 \rho}{dt^2} - m_2 l_2 l_3 \left[\frac{d^2 \theta}{dt^2} \cos(\theta - \rho) - \left(\frac{d\theta}{dt} \right)^2 \sin(\theta - \rho) \right] \\ &\quad + m_2 g l_3 \sin \rho - m_2 \frac{v^2}{R} l_3 \cos \rho = 0 \quad (\text{H - 32}) \end{aligned}$$

If we again use the approximation

$$R = l_4 / \sin \alpha \quad (\text{H - 26})$$

$$\begin{aligned} \frac{d^2 \rho}{dt^2} - \frac{l_2}{l_3} \left[\frac{d^2 \theta}{dt^2} \cos(\theta - \rho) - \left(\frac{d\theta}{dt} \right)^2 \sin(\theta - \rho) \right] + \frac{g}{l_3} \sin \\ - \frac{v^2}{l_3 l_4} \sin \alpha \cos \rho = 0 \quad (\text{H - 33}) \end{aligned}$$

We may want to add damping to the inclinometer equation.

$$\frac{d^2\rho}{dt^2} + \beta \frac{d\rho}{dt} + \frac{g}{l_3} \sin \rho = \frac{v^2}{l_3 l_4} \sin \alpha \cos \rho$$
$$+ \frac{l_2}{l_3} \left[\frac{d^2\theta}{dt^2} \cos(\theta - \rho) - \left(\frac{d\theta}{dt} \right)^2 \sin(\theta - \rho) \right] + \beta \frac{d}{dt} \quad (\text{H - 34})$$

Equation H - 34 is referred to as the **INCLINOMETER EQUATION**.

Derivation of the Bicycle Equation
via summation of torques at bicycle wheelbase

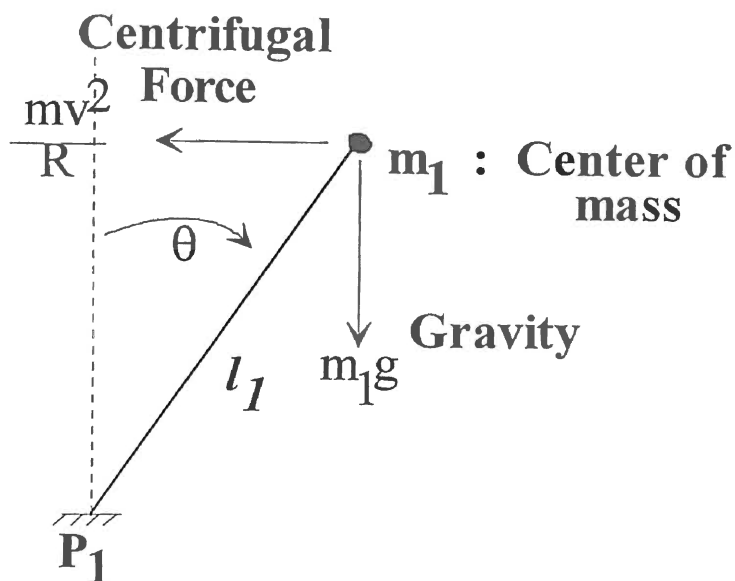


Figure H - 3

$$\tau = mr^2 a \quad (\text{H - 35})$$

Torque equation

τ : torque

m : mass

a : acceleration

$$m_1 l_1^2 \frac{d^3 \theta}{dt^3} = m_1 g l_1 \sin \theta - m_1 v^2 / R l_1 \cos \theta \quad (\text{H - 36})$$

Summation of gravity and centrifugal force about the point P_1 for the dynamic case.

$$l_1^2 \frac{d^3\theta}{dt^3} = gl_1 \sin \theta - v^2/Rl_1 \cos \theta \quad (\text{H - 37})$$

Bicycle mass m_1 cancels out of the equation.

$$R = l_4 / \sin \alpha \quad (\text{H -26})$$

substitution for turning radius

$$l_1 \frac{d^3\theta}{dt^3} = gl_1 \sin \theta - v^2/(l_4 / \sin \alpha)l_1 \cos \theta \quad (\text{H - 38})$$

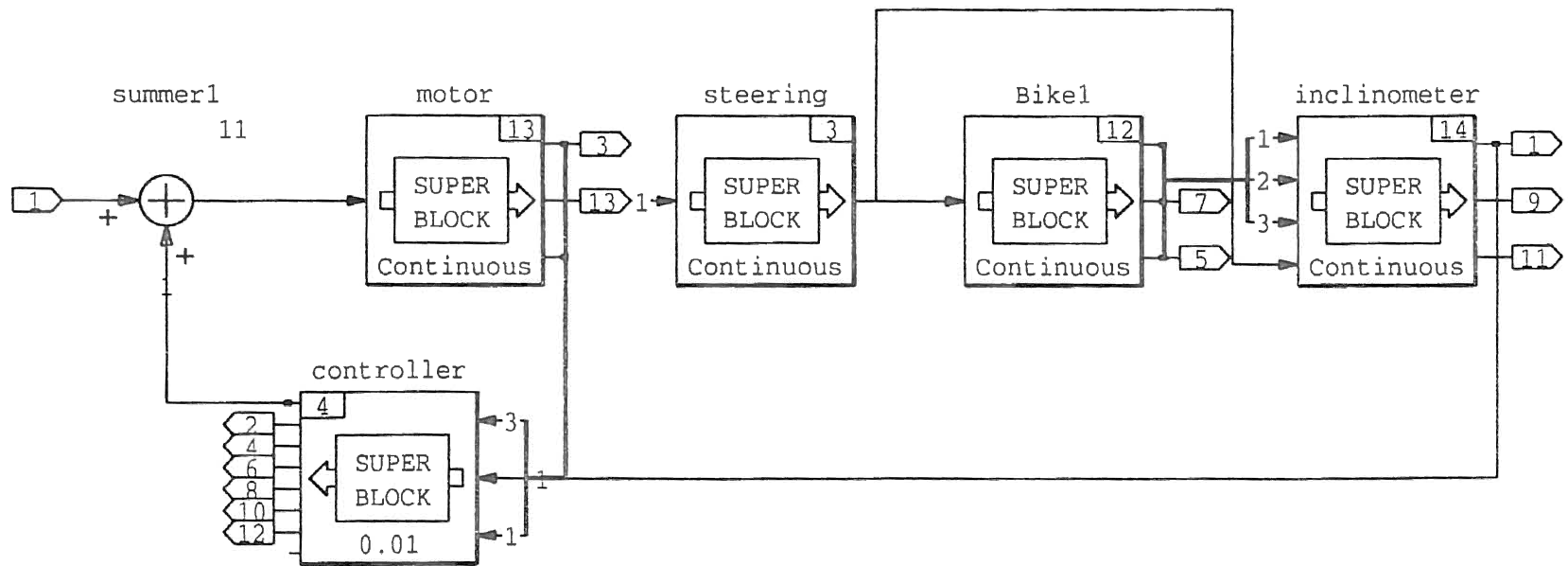
Rearranging terms yields the Bicycle Equation.

$$\frac{d^3\theta}{dt^3} = \frac{g}{l_1} \sin \theta - \frac{v^2}{l_1 l_4} \sin \alpha \cos \theta \quad (\text{H - 39})$$

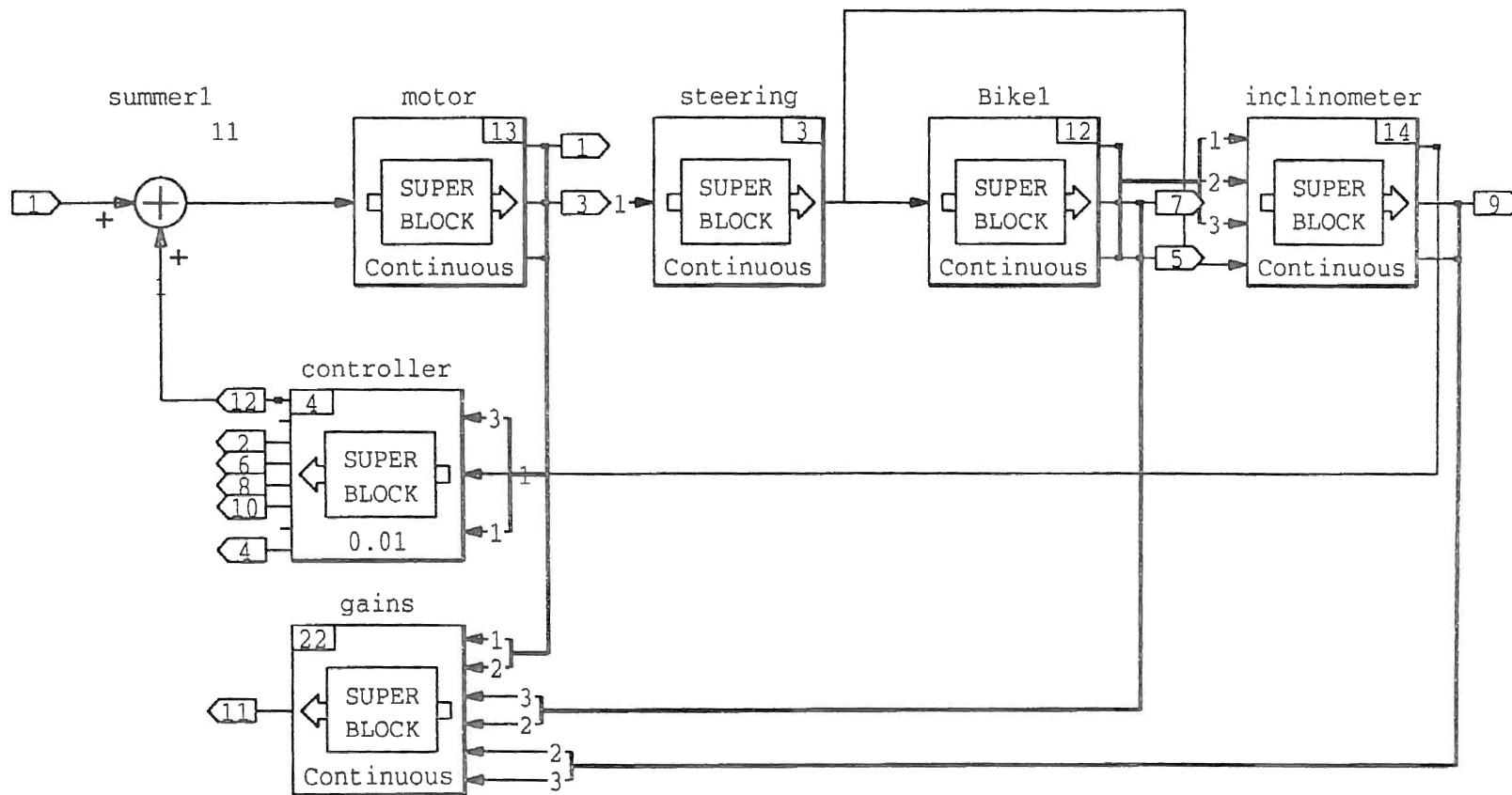
Equation H - 39 is equivalent to equation H - 27, so this is an alternative derivation of the Bicycle Equation.

Appendix I

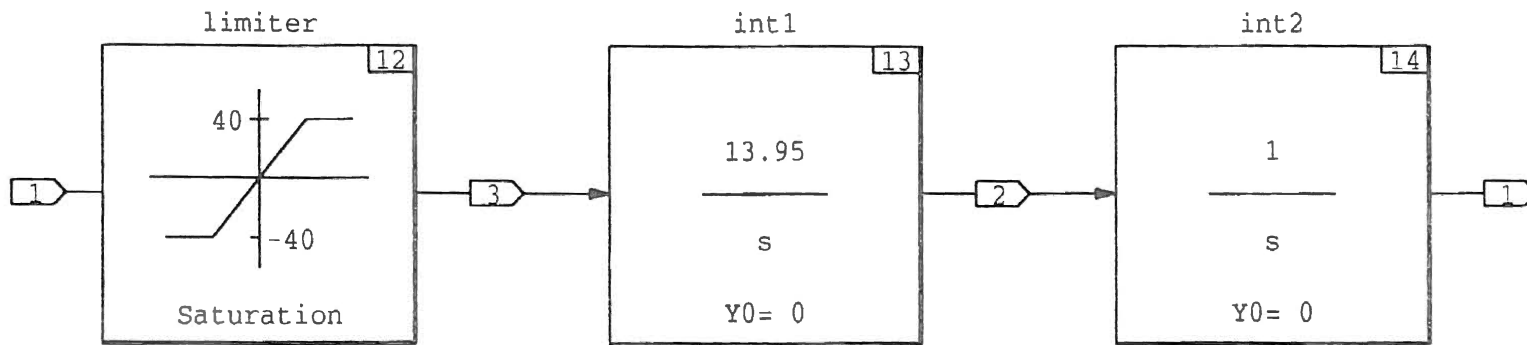
Simulation Block Diagrams from Matrix_x



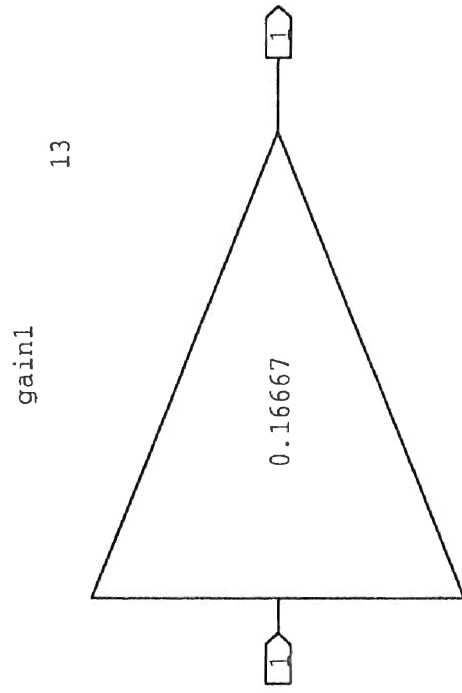
Matrix_x block diagram of the Parallel Systems



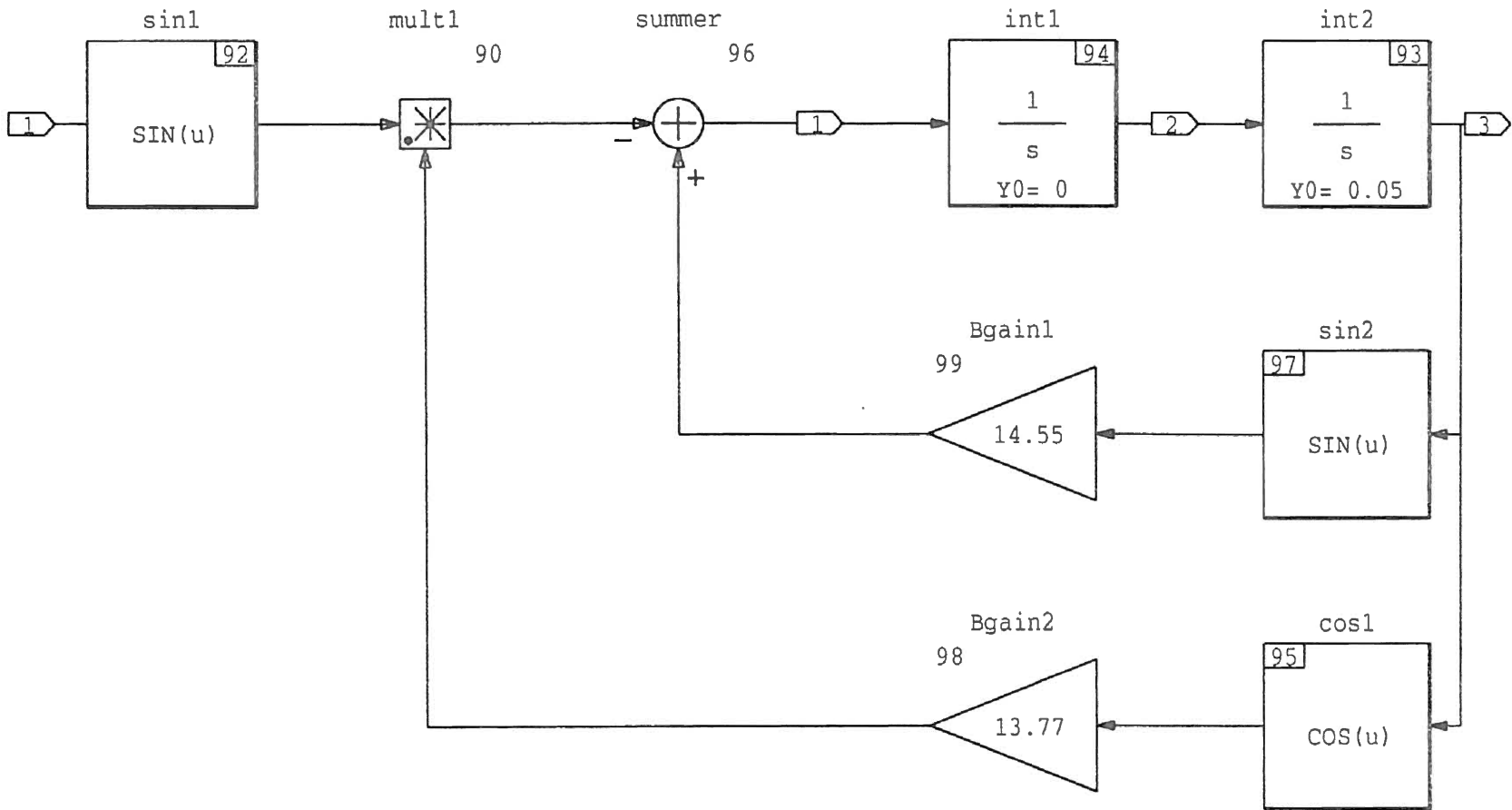
Matrix_x block diagram of the Parallel Systems



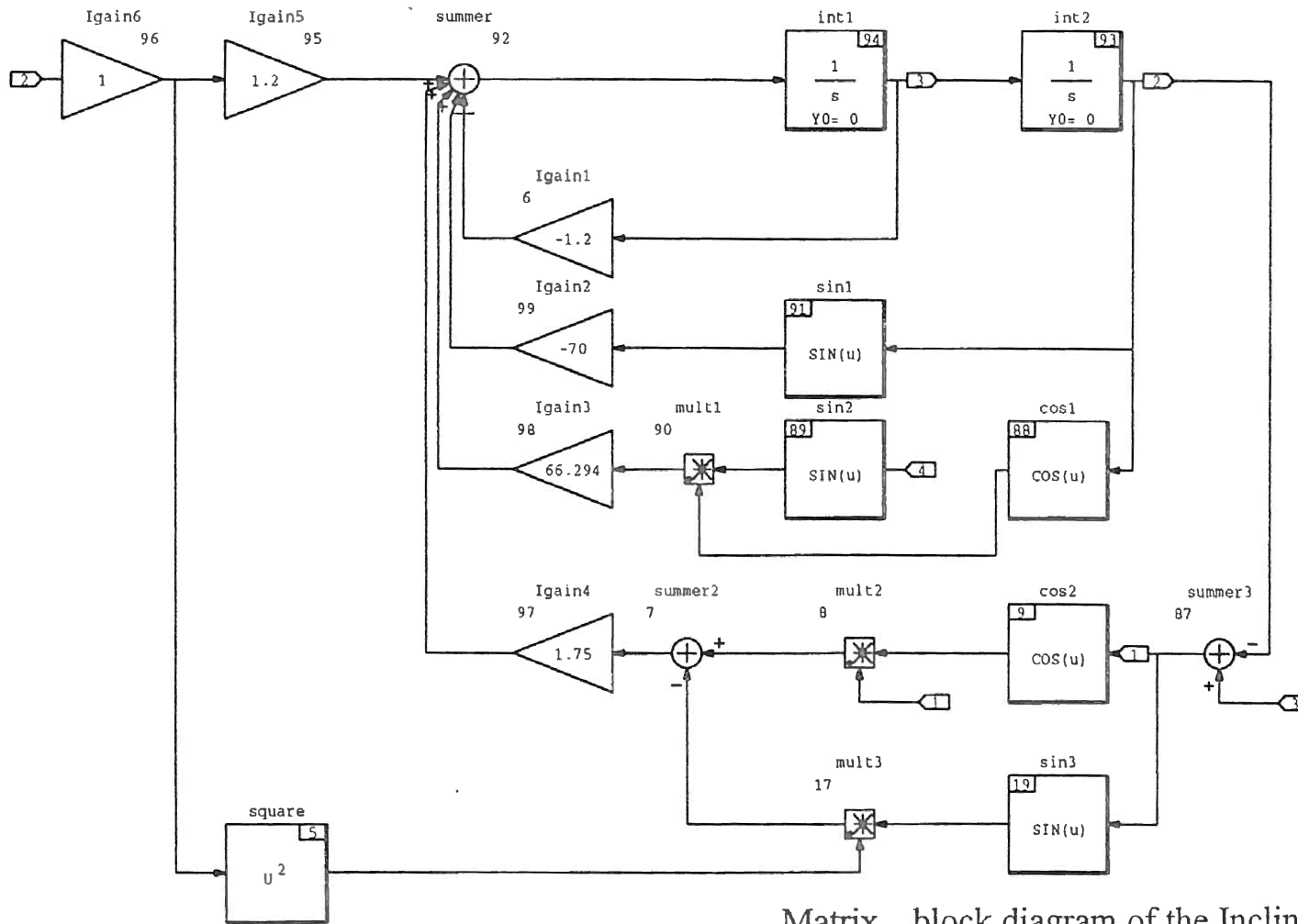
Matrix_x block diagram of the Motor Subsystem



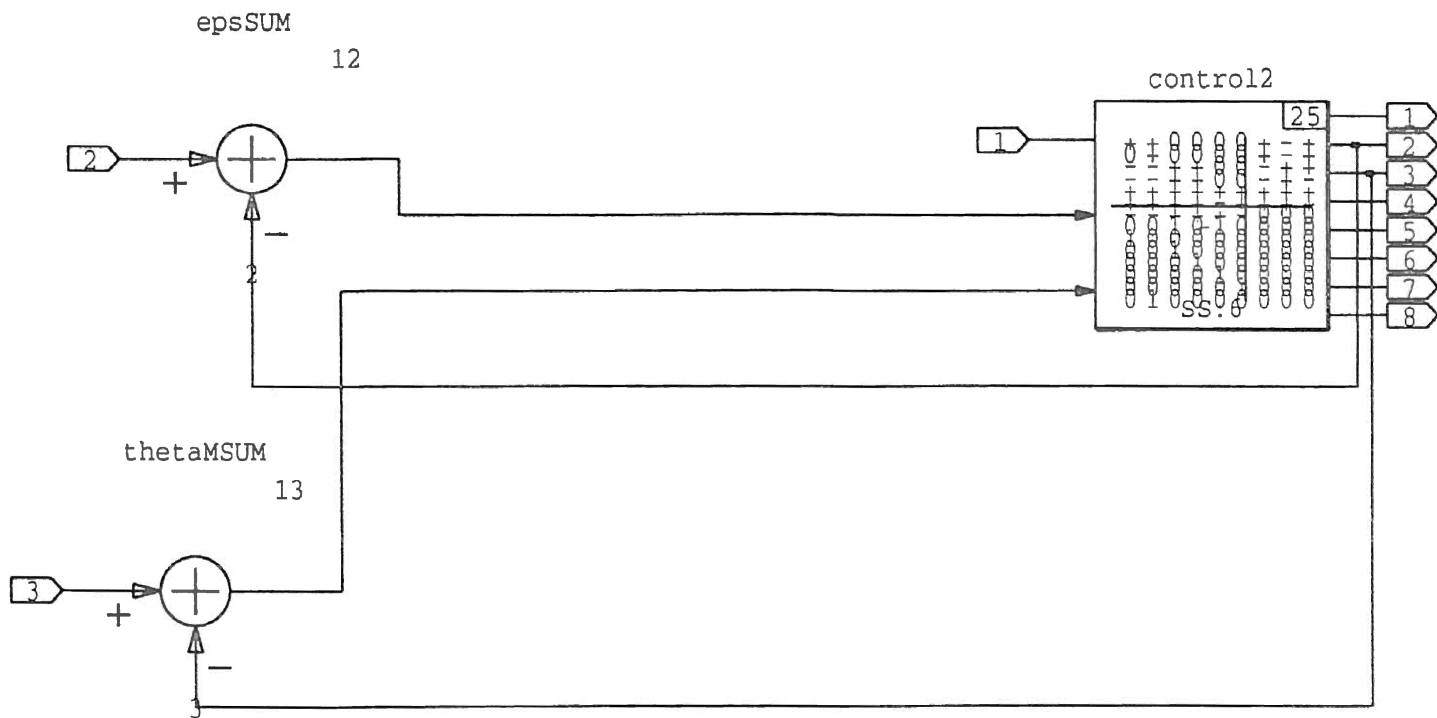
Matrix_x steering ratio block



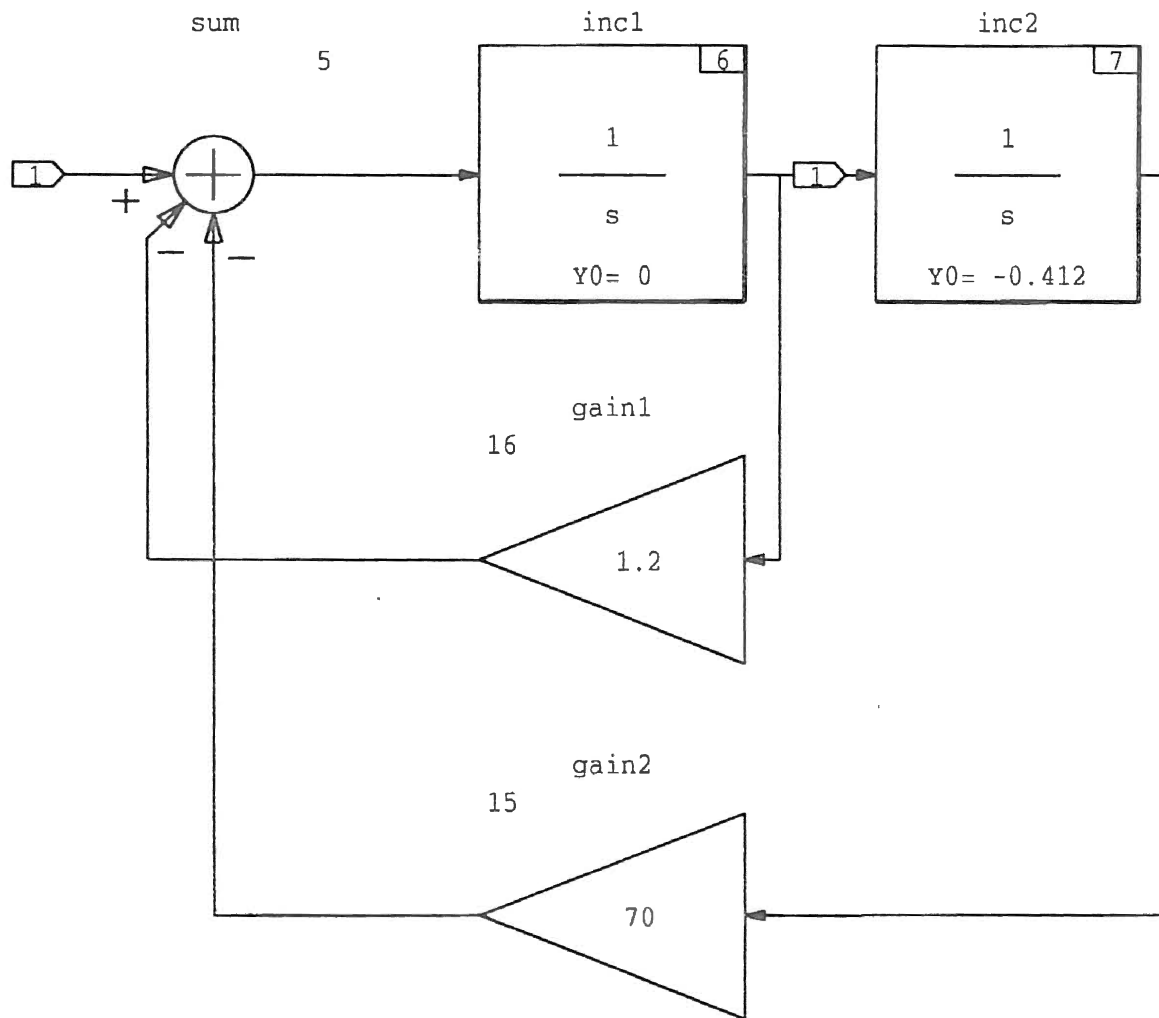
Matrix_x block diagram of the Bicycle Subsystem



Matrix_x block diagram of the Inclinometer



Matrix_x block diagram of the Estimator



Pendulum test model

Appendix J

State Space Matrices and Final Constants

Bicycle State Variables

- θ : Bicycle lean angle (vertical bicycle, $\theta = 0$)
- $\dot{\theta}$: Bicycle lean velocity
- θ_M : Steering motor angle (steering straight ahead, $\theta_M = 0$)
- $\dot{\theta}_M$: Steering motor velocity
- ρ : Pendulum lean angle (pendulum straight down, $\rho = 0$)
- $\dot{\rho}$: Pendulum lean velocity

Important Control System Matrices

- A : State dynamics of bicycle system, (6x6)
- B : Control input matrix, (6x1)
- C : Observation matrix of system, (2x6)
- F : Controller feedback gain matrix, (1x6)
- K : Error weight matrix, (6x2)
- R_{uu} : Riccitti weighting matrix for input, controller design, (1x1)
- R_{xx} : Riccitti weighting matrix for states, controller design, (6x6)
- Q_{xx} : Riccitti state noise intensity matrix, observer design, (6x6)
- Q_{yy} : Riccitti observation noise intensity matrix, observer, (2x2)

$$L1 := 2.2$$

$$z := .0717$$

$$V := 10$$

$$L2 := .8$$

$$Wn := 8.366$$

$$G := 32$$

$$L3 := .4571$$

$$a := .1667$$

$$L4 := 3.3$$

$$I_{\text{gain1}} \quad -2 \cdot z \cdot Wn = -1.2$$

$$B_{\text{gain1}} \quad \frac{G}{L1} = 14.545$$

$$I_{\text{gain2}} \quad \frac{-G}{L3} = -70.007$$

$$B_{\text{gain2}} \quad \frac{V^2}{-L1 \cdot L4} = -13.774$$

$$I_{\text{gain3}} \quad \frac{V^2}{L3 \cdot L4} = 66.294$$

$$I_{\text{gain4}} \quad \frac{L2}{L3} = 1.75$$

$$I_{\text{gain5}} \quad 2 \cdot z \cdot Wn = 1.2$$

$$A(6,1) \quad \frac{1}{L3 \cdot L4} - \frac{1}{L1 \cdot L4} \cdot V^2 \cdot a = 8.755$$

$$A(4,1) \quad \frac{V^2 \cdot a}{-L1 \cdot L4} = -2.296$$

$$A(6,3) \quad \frac{G}{L1} = 14.545$$

$$A(4,3) \quad \frac{G}{L1} = 14.545$$

$$A(6,4) \quad 2 \cdot z \cdot Wn = 1.2$$

$$A(6,5) \quad \frac{-G}{L3} = -70.007$$

$$A(6,6) \quad -2 \cdot z \cdot Wn = -1.2$$

'A' Matrix (Simple Pendulum)

.0000	1.0000	.0000	.0000	.0000	.0000
.0000	-1.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	1.0000	.0000	.0000
-2.2960	.0000	14.5450	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	1.0000
8.7550	.0000	14.5450	1.2000	-70.0000	-1.2000

'B' Matrix

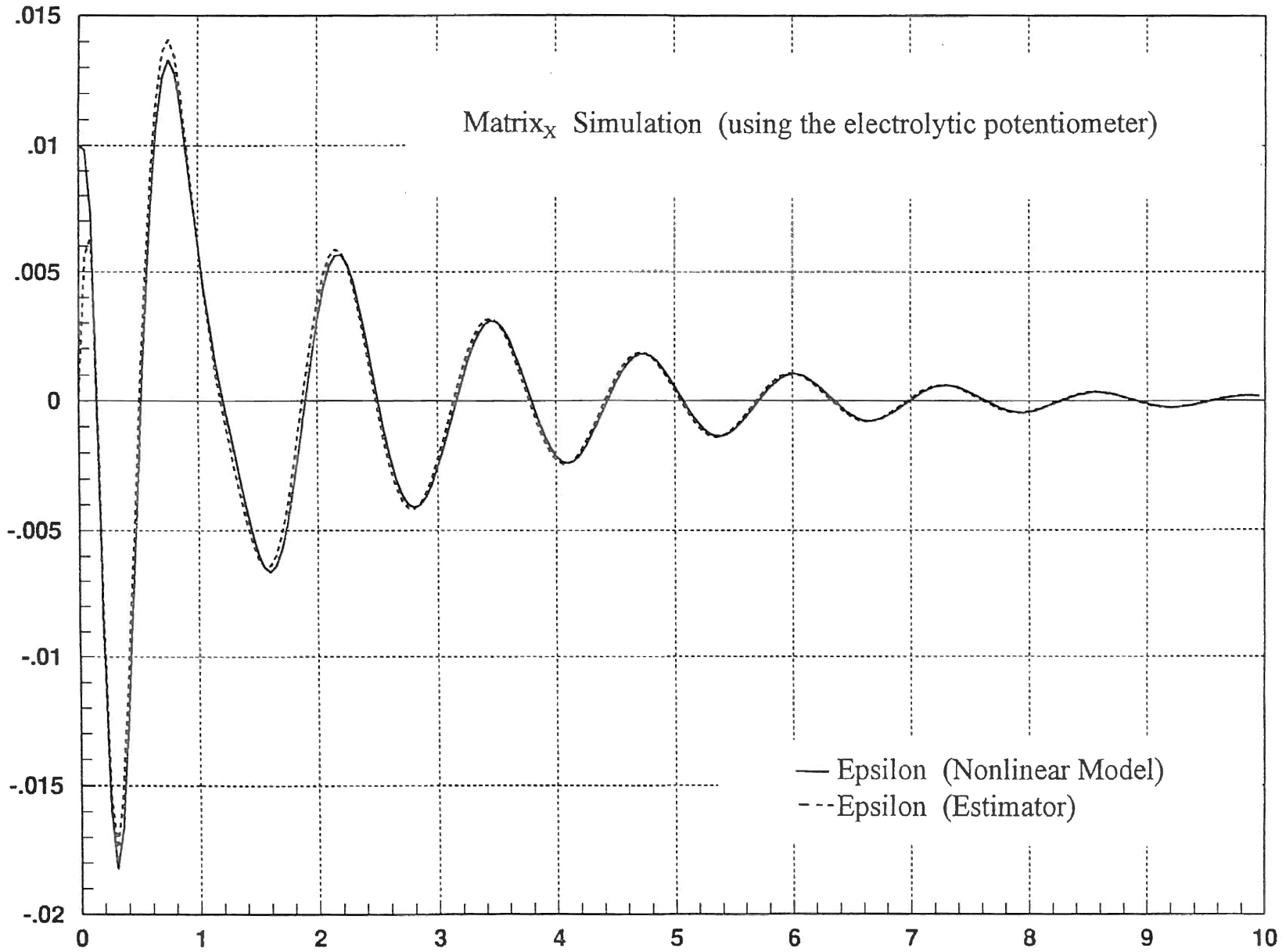
.0000
13.9500
.0000
.0000
.0000
.0000

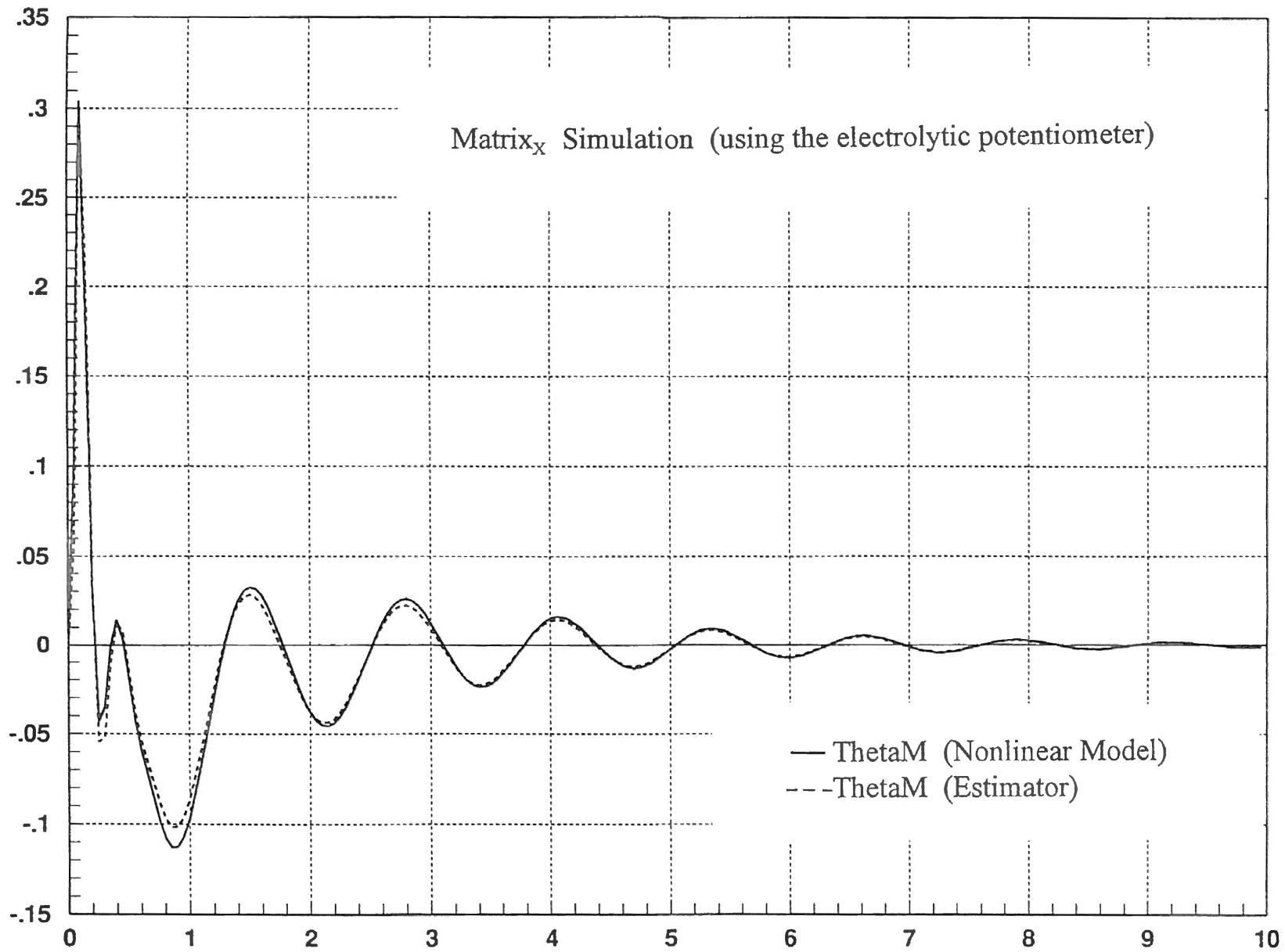
'C' Matrix

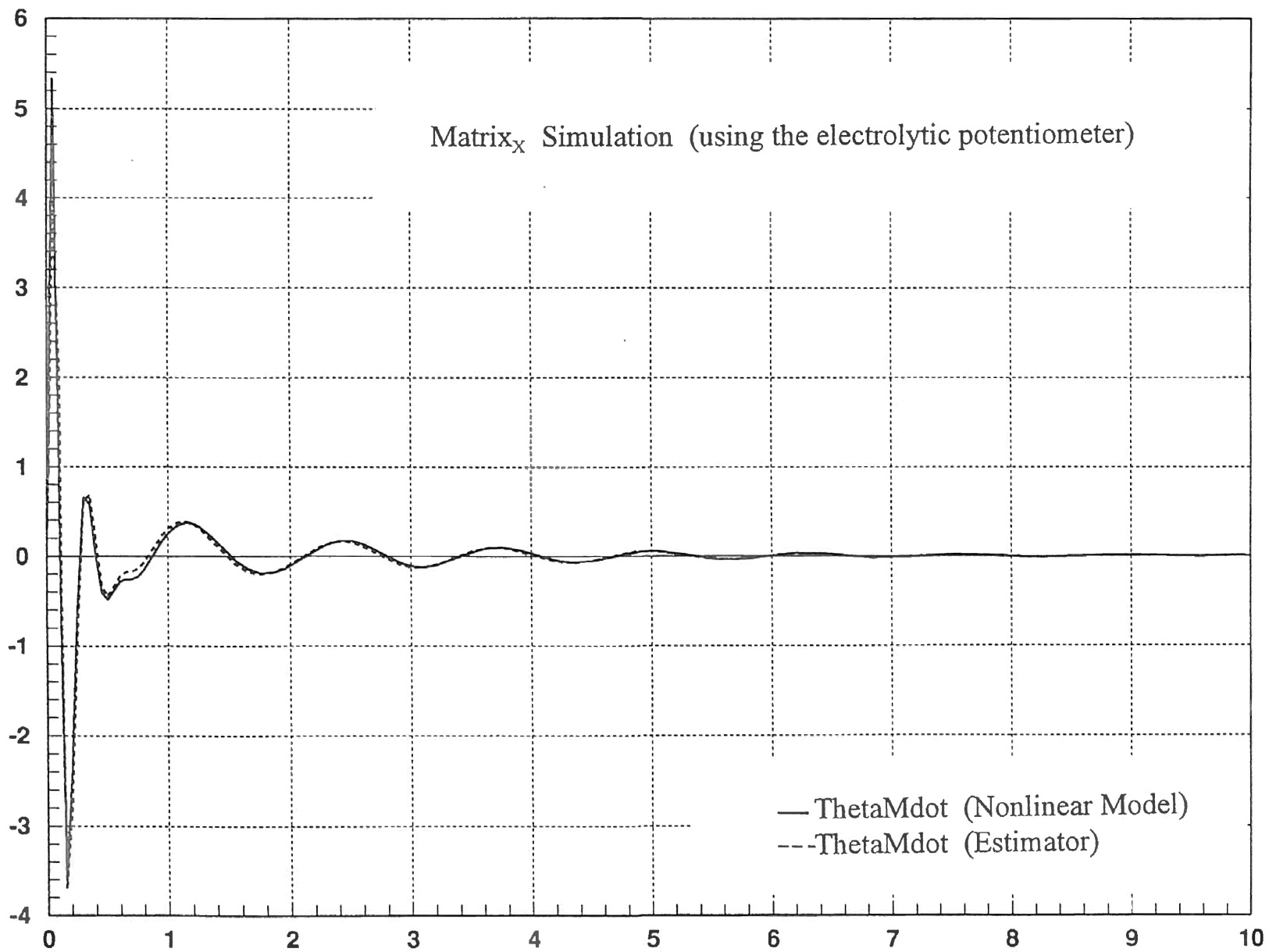
0	0	1	0	-1	0
1	0	0	0	0	0

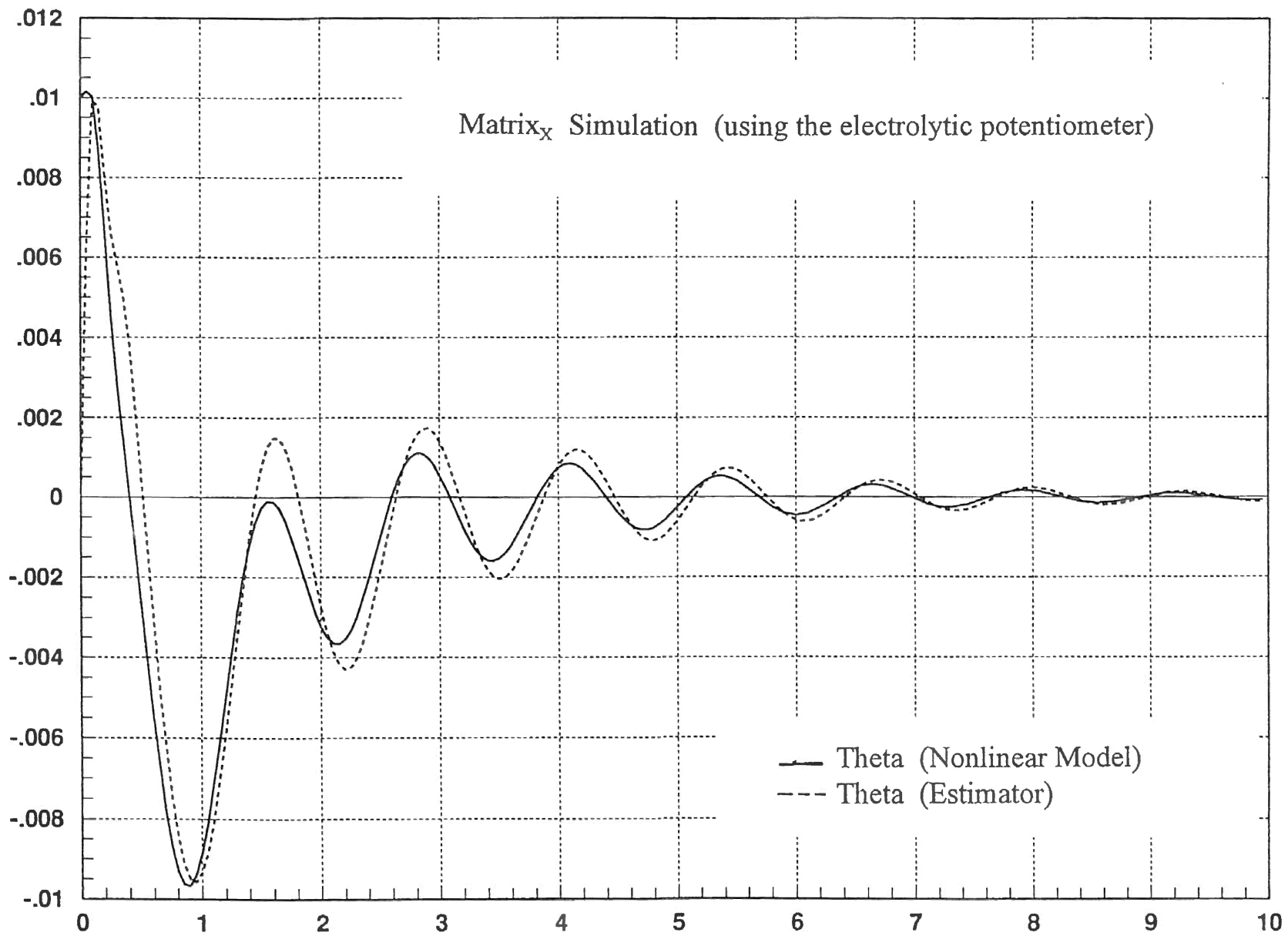
Appendix K

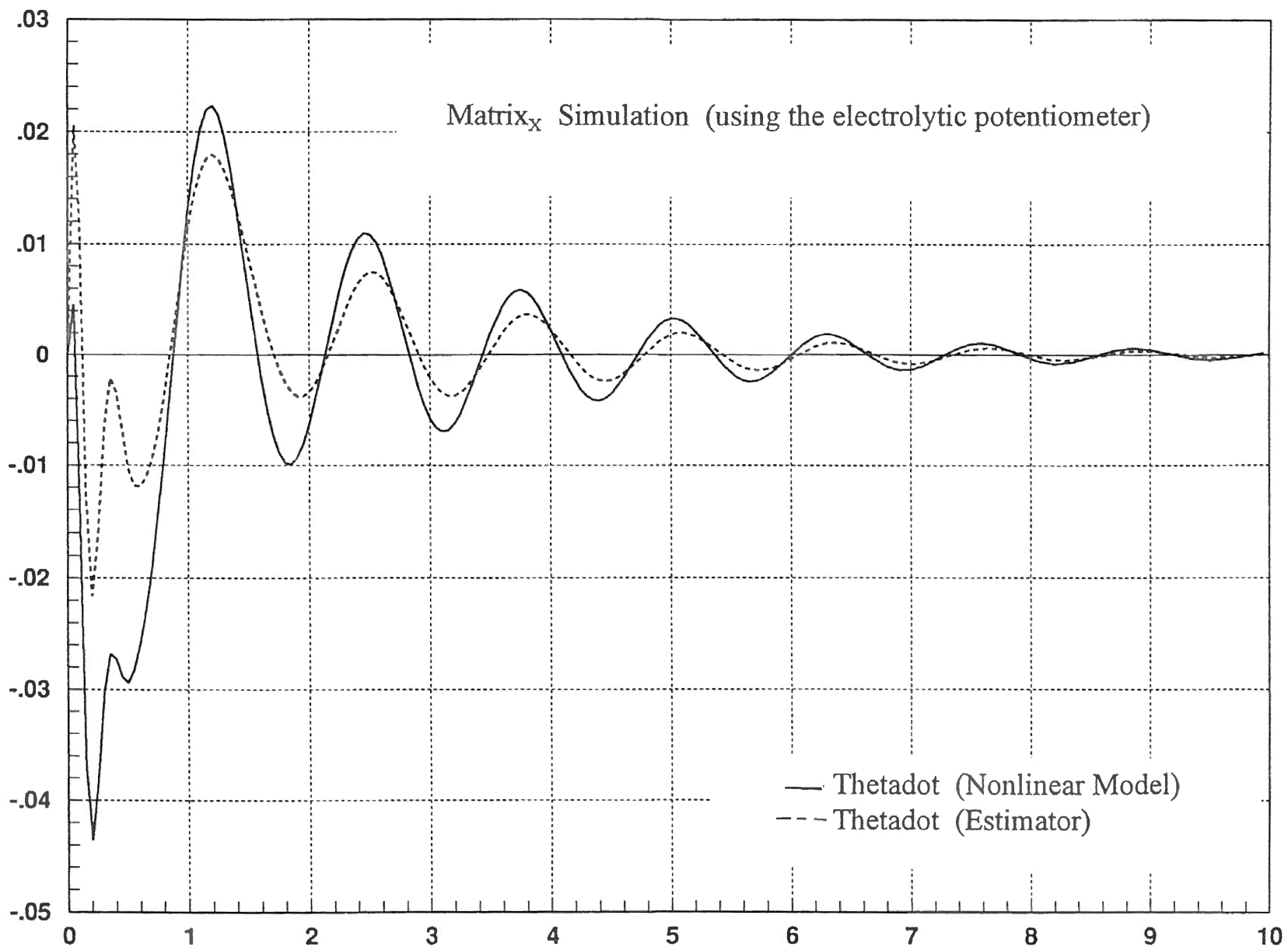
Matrix_x Simulation Results

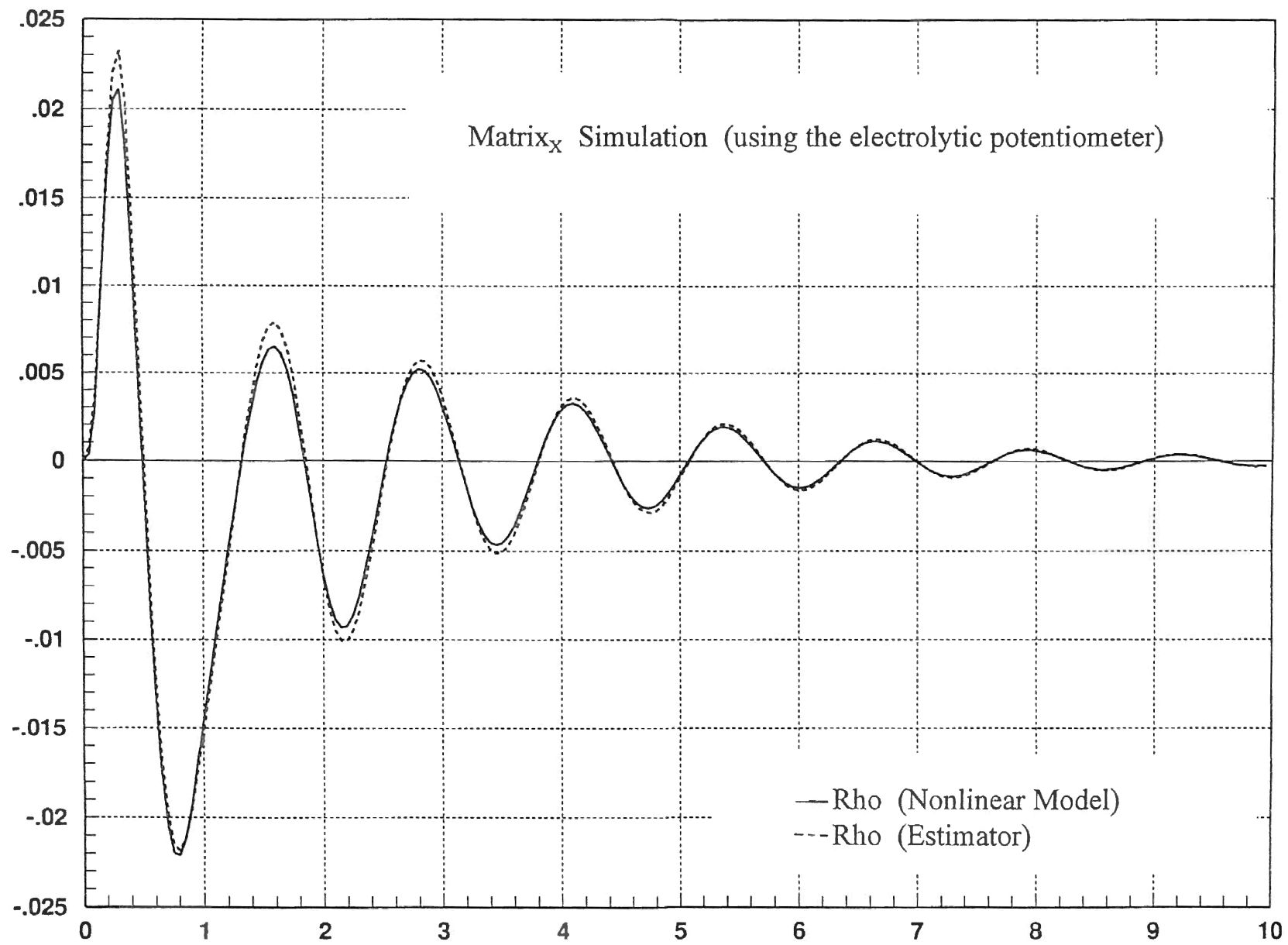


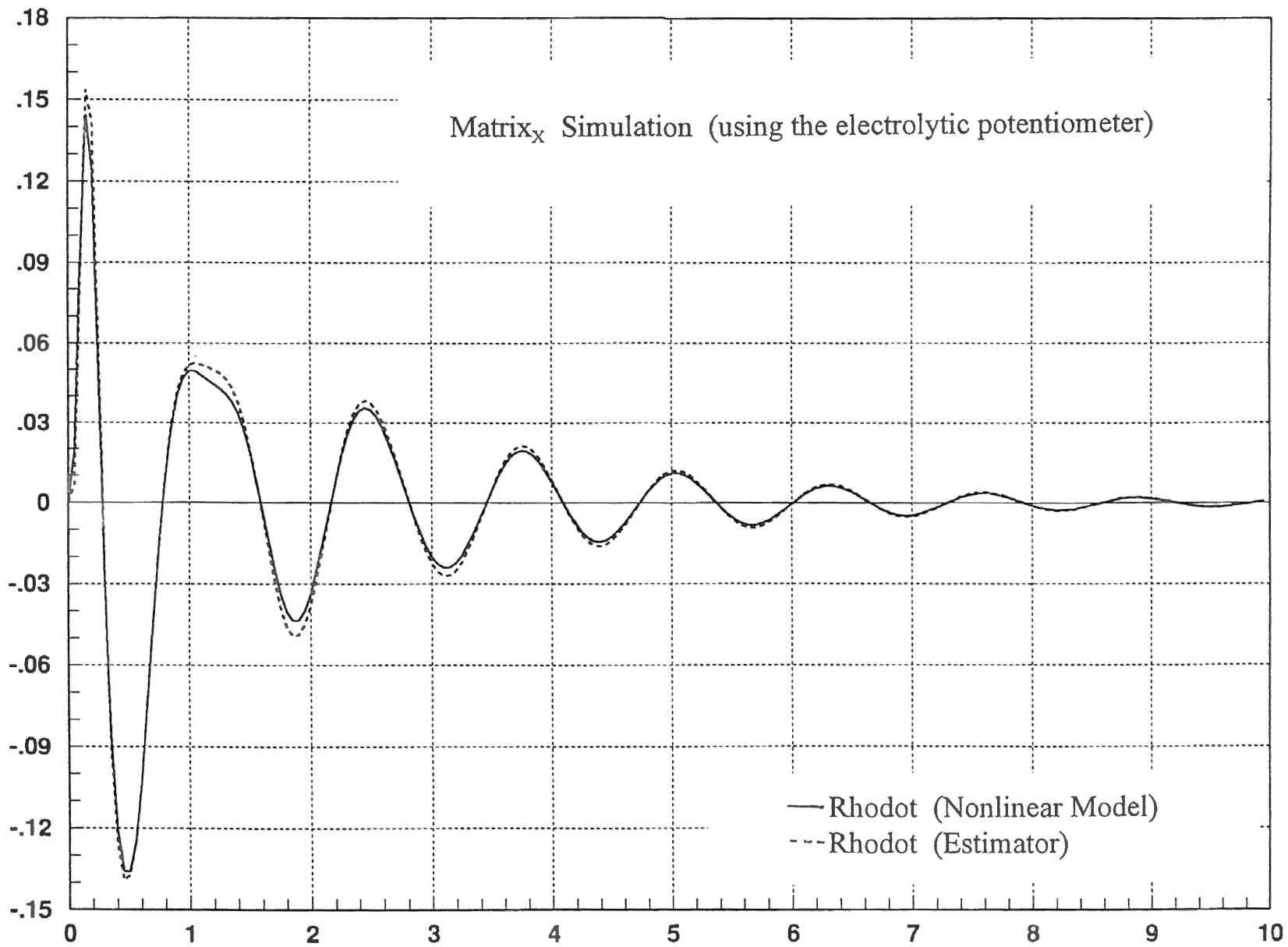


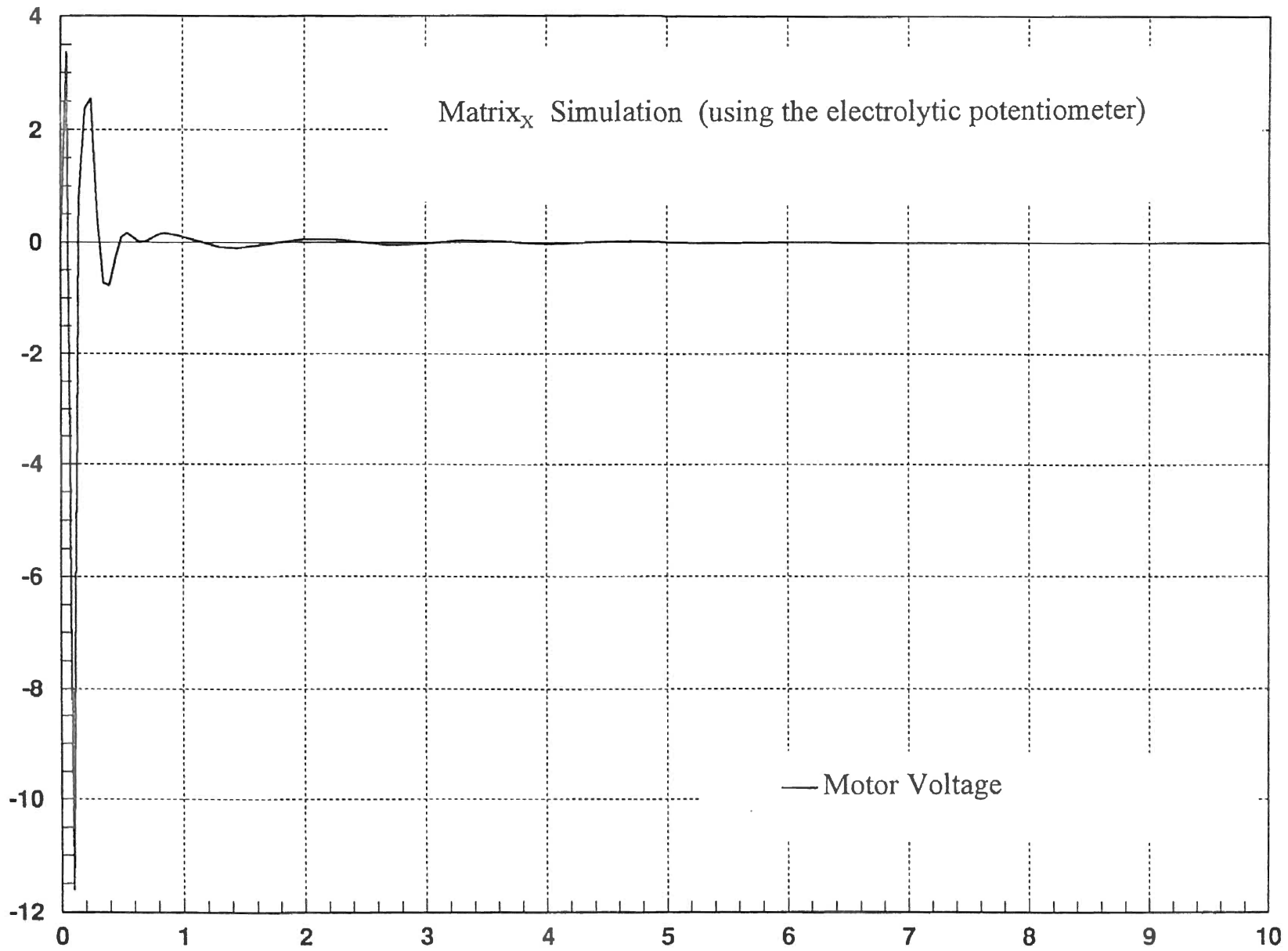






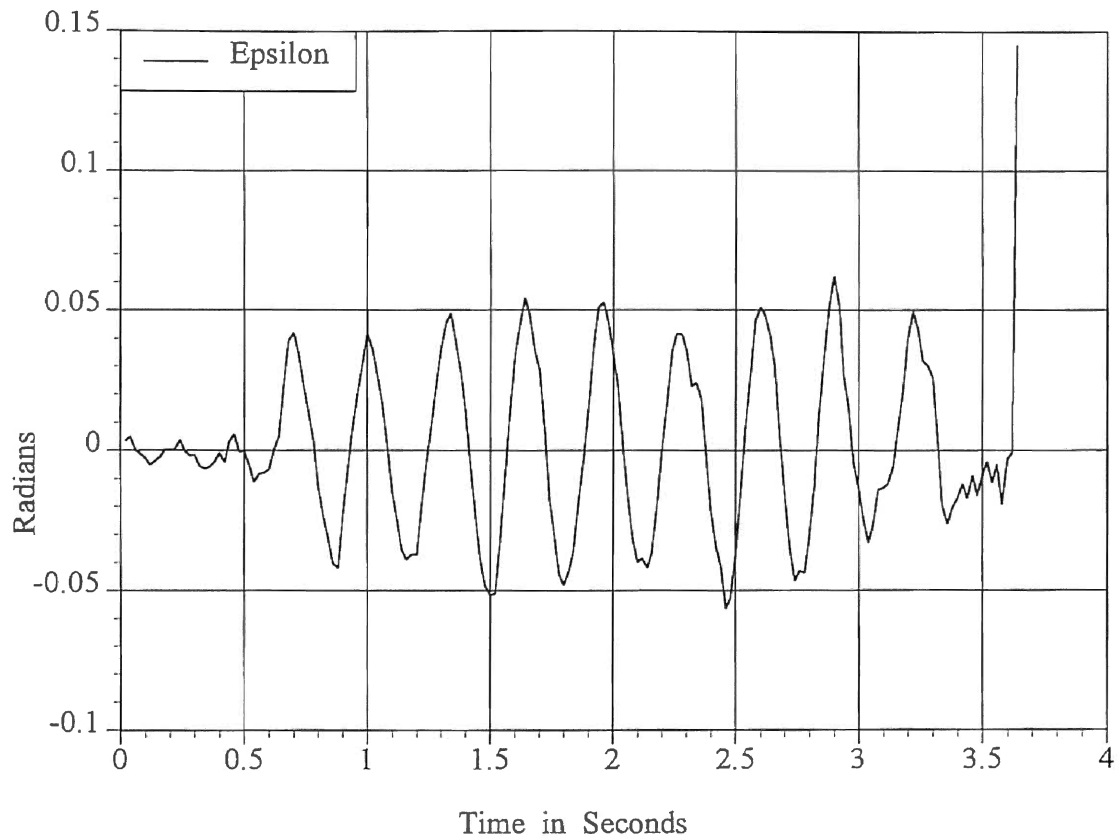






Appendix L

Bicycle Test Platform Results



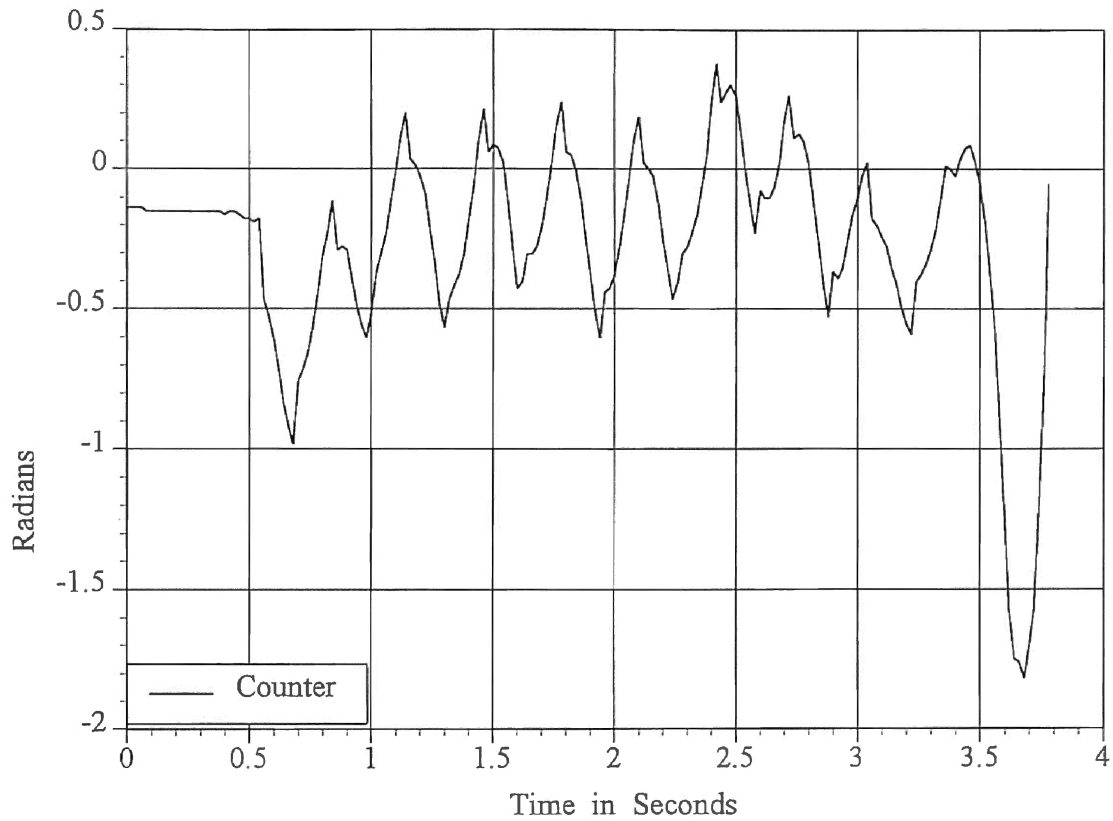
Filtered Bicycle Error Angle,

Epsilon,

Full Bicycle Platform Test

Figure L - 1

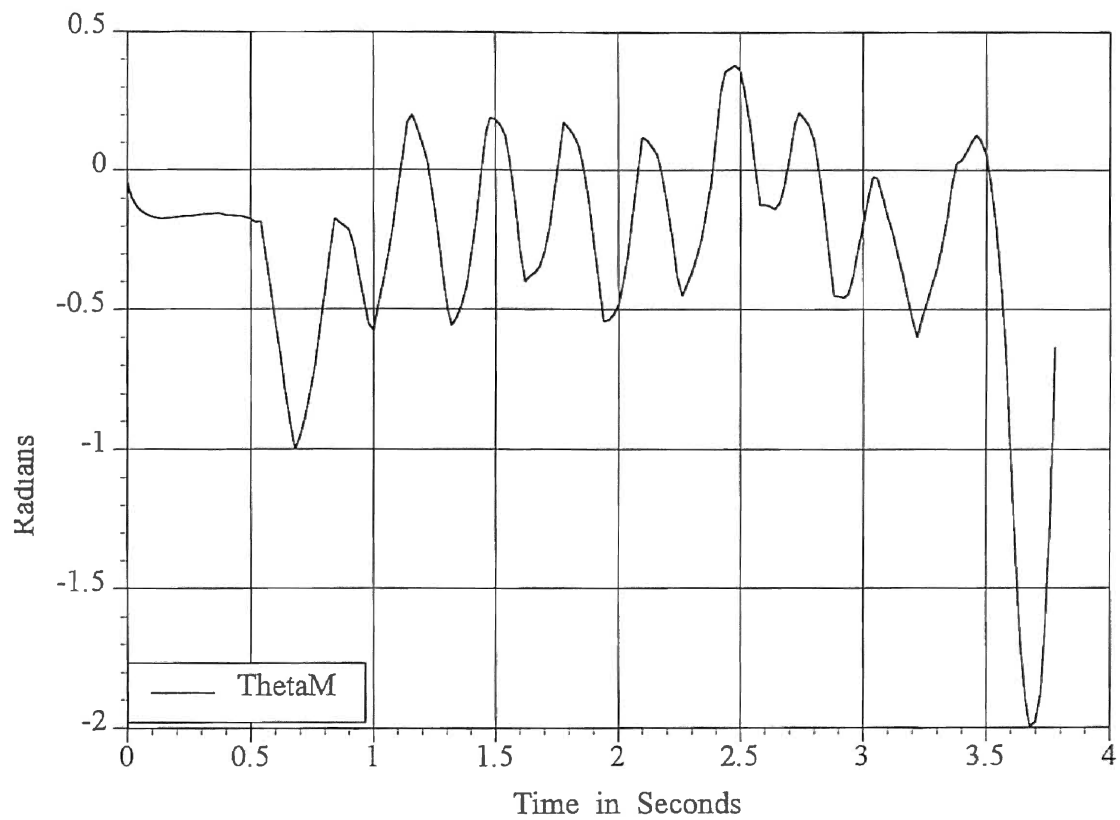
This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.



**Steering Motor Counter,
Full Bicycle Platform Test**

Figure L - 2

This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.

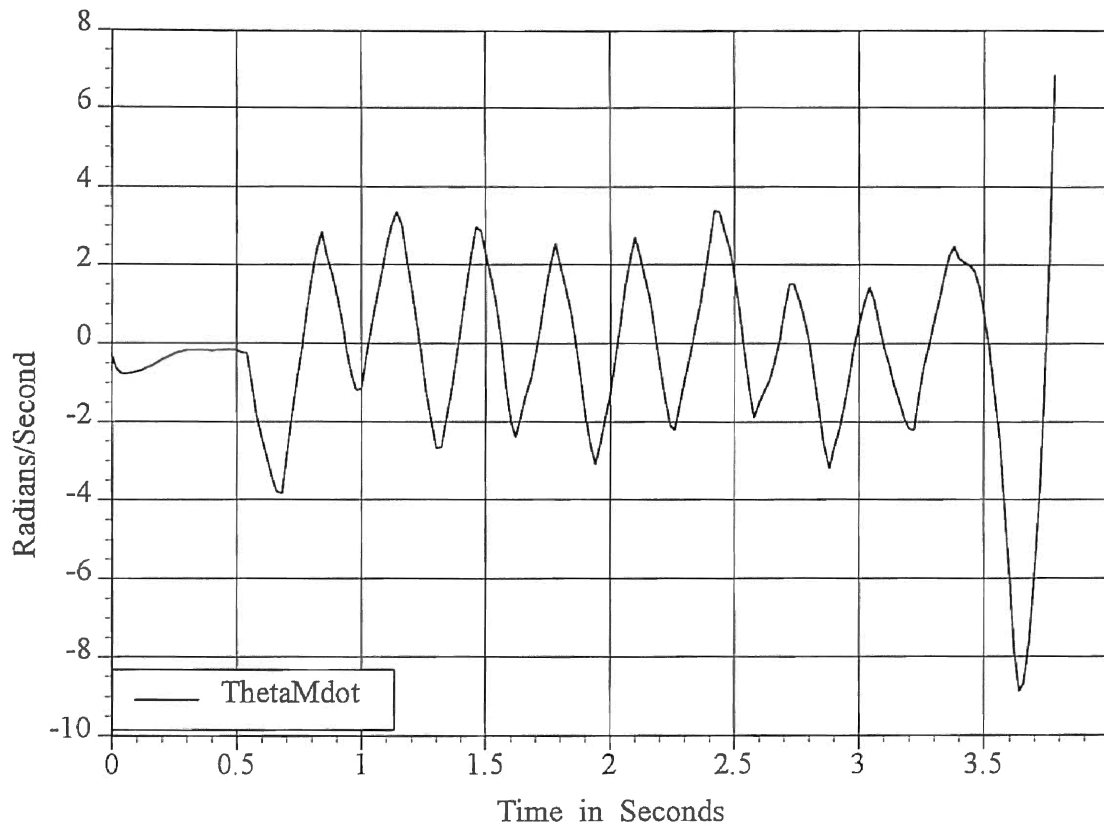


State Variable, Θ_M ,

Full Bicycle Platform Test

Figure L - 3

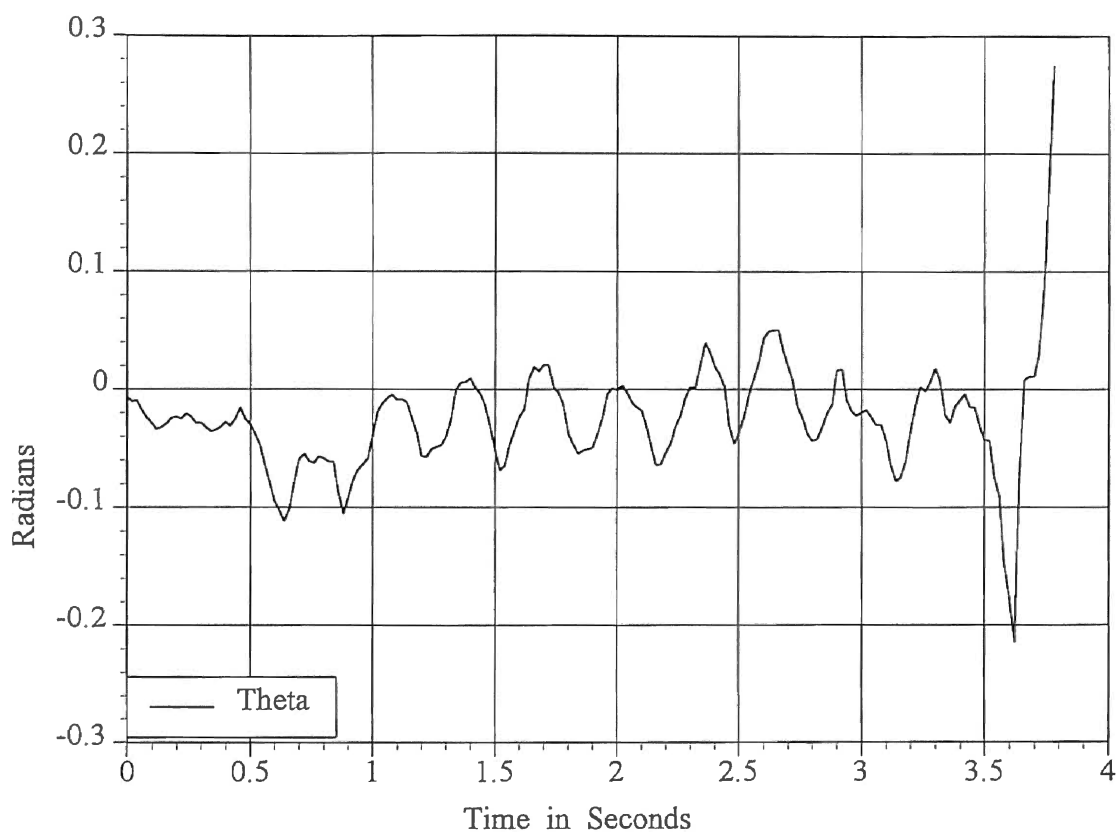
This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.



**State Variable, $\Theta\dot{m}$,
Full Bicycle Platform Test**

Figure L - 4

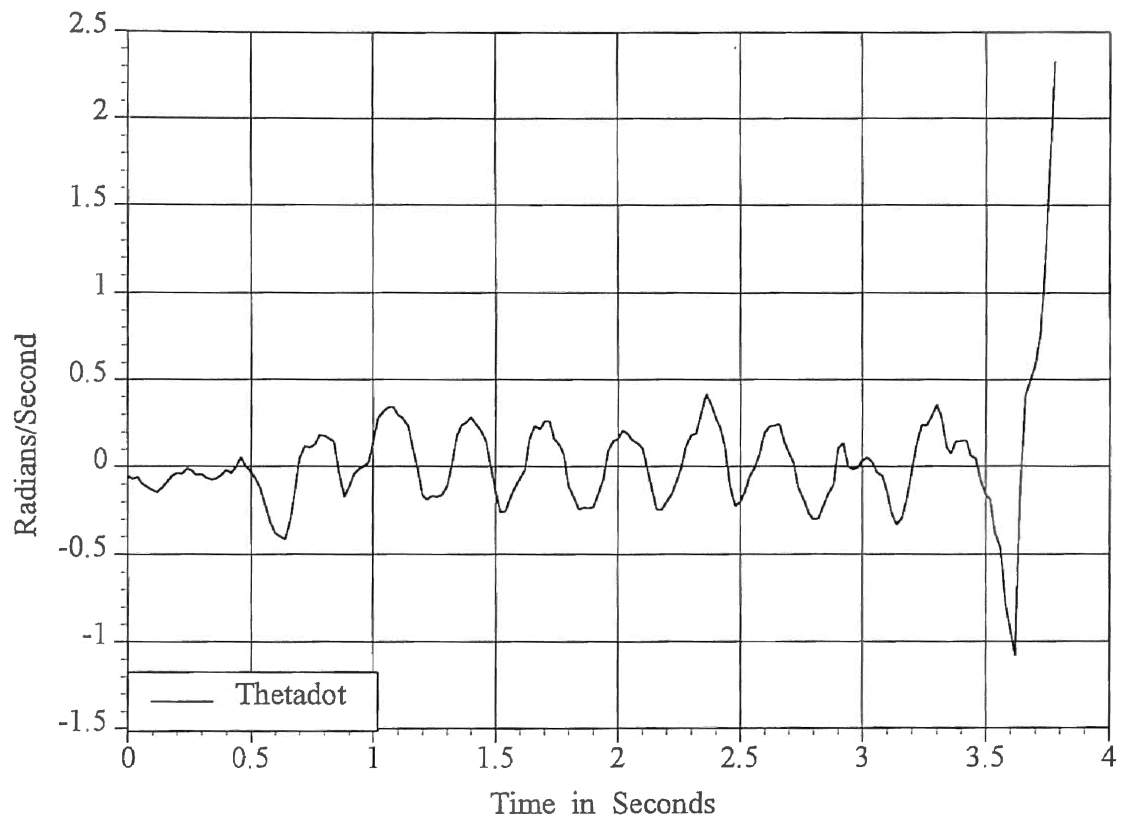
This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.



**State Variable, Theta,
Full Bicycle Platform Test**

Figure L - 5

This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.

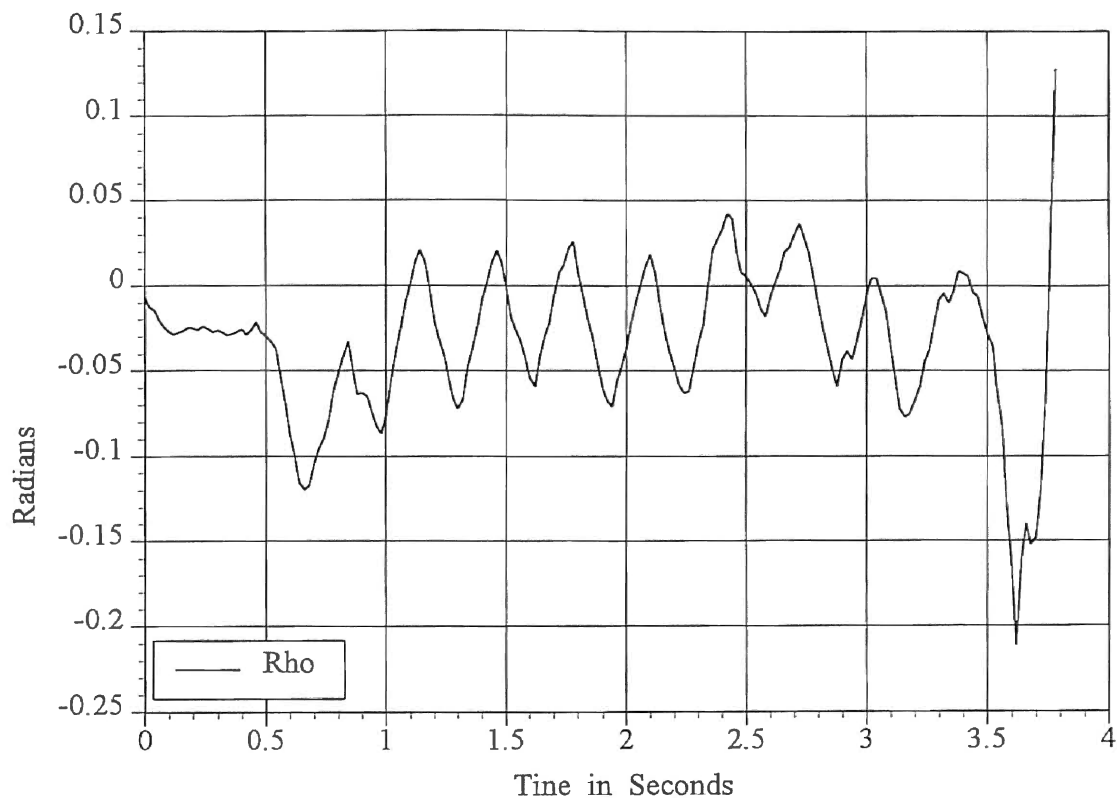


State Variable, Thetadot,

Full Bicycle Platform Test

Figure L - 6

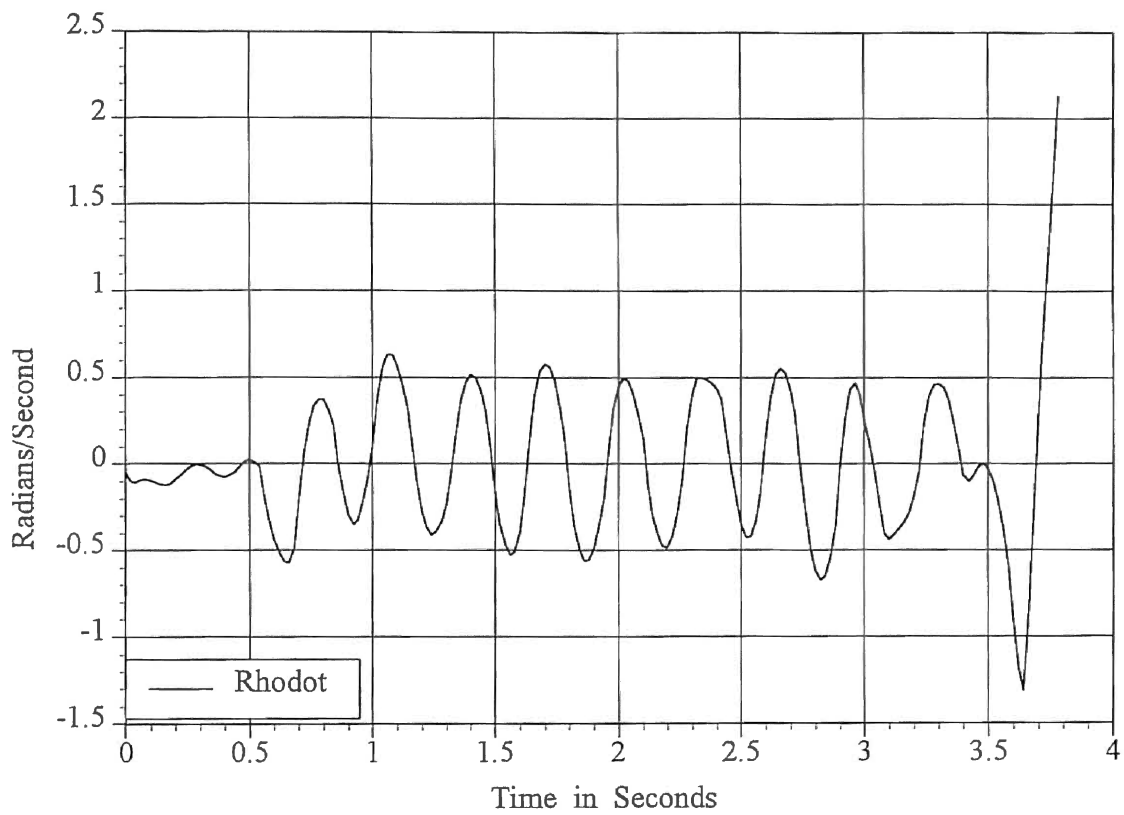
This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.



State Variable, ρ ,
Full Bicycle Platform Test

Figure L - 7

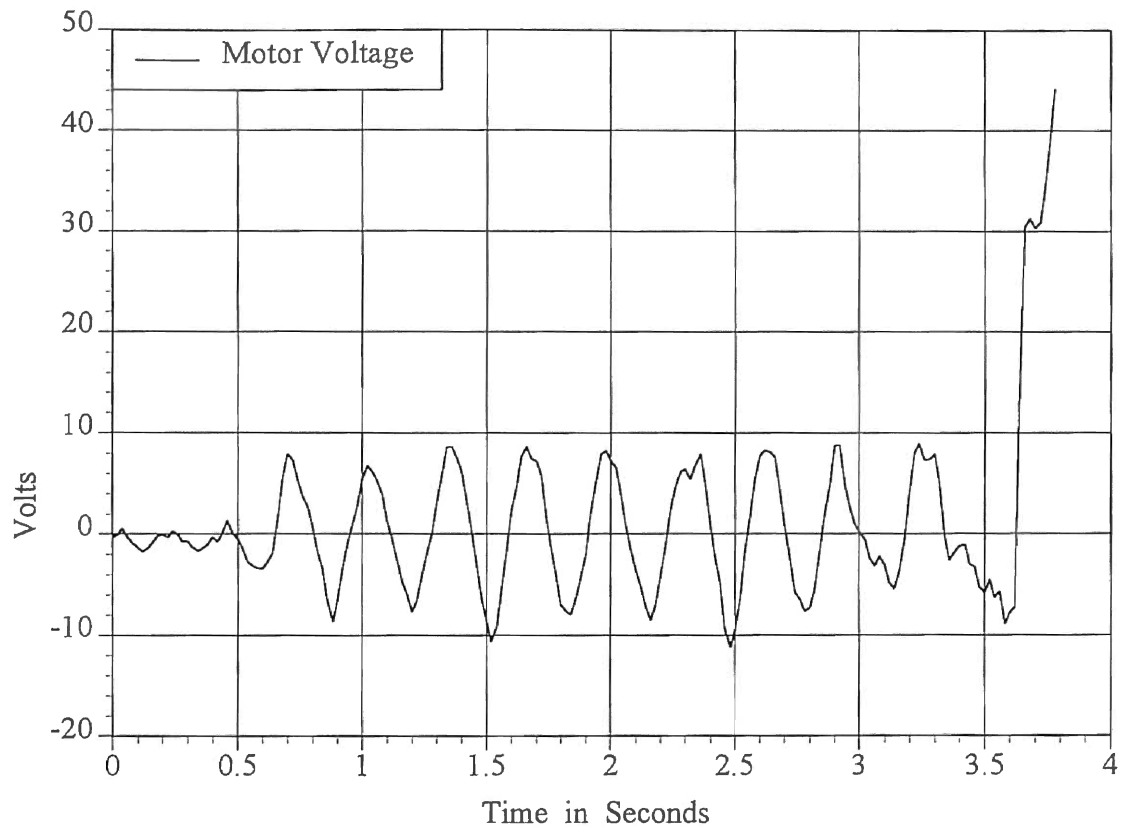
This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.



**State Variable, Rhodot,
Full Bicycle Platform Test**

Figure L - 8

This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.



Calculated Motor Voltage

Full Bicycle Platform Test

Figure L - 9

This Test was conducted using the Electrolytic Potentiometer to sense the bicycle error angle. A software filter was used in an attempt to reduce the platform noise sent to the control system.

Appendix M

Bicycle Test Platform Main Program Listing

Bicycle Control Program
Original Program Written By Bicycle Design Group 3
with modifications from
ROBBI 4
and additional modifications.

mm 5000;s	Load estimator and controller constants
1.0	A(1, 1)
1.0e-2	A(1, 2)
0.0	A(1, 3)
0.0	A(1, 4)
3.07e-1	K(1, 2)
0.0	K(1, 1)
0.0	A(2, 1)
1.0	A(2, 2)
0.0	A(2, 3)
0.0	A(2, 4)
0.0	A(2, 5)
2.103	K(2, 2)
0.0	K(2, 1)
1.0009	A(3, 3)
1.0e-2	A(3, 4)
4.56e-2	K(3, 2)
1.044	K(3, 1)
-2.79e-2	A(4, 1)
1.841e-1	A(4, 3)
1.0009	A(4, 4)
3.2e-1	K(4, 2)
6.1823	K(4, 1)
1.7e-3	A(5, 1)
1.48e-2	A(5, 3)
9.743e-1	A(5, 5)
9.4e-3	A(5, 6)
4.56e-2	K(5, 2)
5.579e-1	K(5, 1)

3.231e-1	A(6, 1)
1.7e-3	A(6, 2)
2.8974	A(6, 3)
1.231e-1	A(6, 4)
-5.022	A(6, 5)
8.66e-1	A(6, 6)
3.201e-1	K(6, 2)
-1.485e-1	K(6, 1)
-7.7883	F(1, 1)
-1.1133	F(1, 2)
5.71613e+1	F(1, 3)
1.33227e+1	F(1, 4)
1.0e-2	B(2, 1)
4.0	Max. steering angle
1.0e-2	Sampling rate
0.0	Offset
.	
mm 6000;l	Load initial values of variables
0	Epsilon (measurement) in radians
0	Steering angle (measurement) in radians
0	BEXE
1	BPEN
15	MOTCNT
0	MOTCNT1
15	BCNT
15	BCNT1
1	DFLAG
1	DFLAG1
0	WRAP FLAG
0	error code
40000	End of data buffer
0	ThetaM
0	ThetaMdot
0	Theta
0	ThetaM
0	Rho
0	Rhodot
0	ThetaM(K + 1)


```

ori.b #$10,$fff80015
equal
move.b $fff80017,d5
andi.l #$f0,d5
ori.b #$04,d5
asl #2,d5
movea.l d5,a2
lea 20000,a1
move.l a1,(a2)
move.b #3d,$fff80025
move.b #2,d5
andi.b #$f0,$fff8001d
or.b d5,$fff8001d
ori.b #$10,$fff80009
bra *

```

```

.
mm 14000;di
move.w ($100006),d1
trap #15

```

```

mm 20000;di
subq.l #1,$601c
cmpi.l #0,$6014
beq $20100
subq.l #1,$6014
cmpi.l #1,$6024
beq $20050
move.w ($100004),d1
bra 20200

```

```

.
mm 20050;di
move.w ($100002),d1
bra 20200

```

```

.
mm 20100;di
move.w ($100006),d1
bra 20200

```

which is set equal to 3d(hex) and Scale is equal to the of \$fff8001d, the Prescaler. If the Prescaler is equal to 2 the Scale is equal to 10.

Begin at next address.

This subroutine addresses location 100006 which turns off the motor. Trap #15 returns control to the Bug operating system.

{Routine A: executes at 2Khz}
 Subtract one from BNT1.
 If MOTCNT1 = 0 then go to 20100 to turn off motor.
 Else subtract one form MOTCNT1.
 If DFLAG1 = 1 then go to 20050 (turn right).
 Else (turn left).

{when MOTCNT1 > 0 and DFLAG = 0}
 (turn right)

{when MOTCNT1 = 0}
 Turn the motor off.

<pre> mm 20200;di cmpi.l #0,\$601c bgt 20300 move.l \$6010,\$6014 move.l \$6018,\$601c move.l \$6020,\$6024 cmpi.l #0,\$6008 beq 20250 move.w #fa,\$602c andi.b #\$ef,\$fff80009 move.w (\$100006),d1 trap #15 </pre>	<pre> { always reached } If BCNT1 > 0 then go to 20300 Else let MOTCNT1 = MOTCNT let BCNT1 = BCNT let DFLAG = DFLAG If BEXE = 0 then go to 20250 Else the B routine has taken longer than the .01 second frame to execute. set error flag to #FA . Turn off the motor. Call Bug, stop execution of program. </pre>
<pre> mm 20250;di move.l #1,\$600c rte </pre>	<pre> { When BCNT1 = 0 and BEXEC = 0 } Let BPEND = 1 Restore execution of the B routine </pre>
<pre> mm 20300;di cmpi.l #0,\$6008 beq 20400 rte </pre>	<pre> { When BCNT > 0 } If BEXE = 0 then goto 20400 Else restore execution of the B routine. </pre>
<pre> mm 20400;di cmpi.l #1,\$600c beq 20500 rte </pre>	<pre> { When BCNT1 > 0 and BEXE = 0 } If BPEND = 1 then goto 20500 Else restore execution of the B routine. </pre>
<pre> mm 20500;di move.l #0,\$600c move.l #1,\$6008 move.l #0,d3 move.w (\$100000),d3 </pre>	<pre> { When BCNT1 > 0 and BEXEC = 0 and BPEND = 1 } Let BPEND = 0 Let BEXEC = 1 Read the motor shaft encoder. Clear redgester d3. Read counter latches into redgester d3. ThetaM is calculated in radians. ThetaM = (d3 * 2 * Pi) / (500) </pre>

```

fmovecr #0,fp1
fmove.w d3,fp0
fmul.x fp0,fp1
fmul.s #2,fp1
fdiv.s #500,fp1
fmove.s fp1,$6004

```

Put Pi into floating point register #1.
Move Count into floating point #0.

```

move.w #0,d4
move.w #0,$1fffe00

```

Store the steering angle

```

fabs.x fp1,fp3
fsub.s $6074,fp1
fdiv.s $50a8,fp1
fmove.s fp1,$6078

```

Send convert command to A/D.

Clear register d4.

Initialize the A/D converter.

Calculate steering angle velocity.

This data is not directly used in this control system but is used in analysis.

Velocity = (X2 - X1) / dT

```

fcmp.s $50a4,fp3
fbge.l $14000
bra 20600

```

Over steer motor stop routine.

Compare steering angle to max. angle.

Turn motor off if current angle has exceeded this maximum.

```

.
mm 20600;di
move.w $1fffe00,d4
bmi $20600
cmp.w #$4000,d4
blt $20700
ori.w #$8000,d4
bra 20700

```

Read the inclinometer (A/D).

Move data into register d4

branch on minus to 20600

If the value read from the A/D is less than 4000 then epsilon is positive, goto 20700.

Else, convert the number to 2's complement and goto 20700.

```

.
mm 20700;di
fmovecr #0,fp1
fmove.w d4,fp0
fdiv.s #1.365E+2,fp0
fmul.x fp0,fp1
fdiv.s #180,fp1
fmove.s fp1,$6000
bra 22800

```

Convert the read data in register d4 to an angle in radians.

Epsilon = (d4 * Pi) / (180* inc. constant)

Store Epsilon in 6000

```

.
mm 22800;di

```

Calculate ThetaMerror

```

fmove.s $6004,fp1
fmove.s $6034,fp2
fsub.x fp2,fp1
fmove.s fp1,$6064
fmove.s $6000,fp1
fmove.s $603c,fp2
fmove.s $6044,fp3
fsub.x fp3,fp2
fsub.x fp2,fp1
fmove.s fp1,$6068
bra 23000

```

```

mm 23000;di
fmove.s $6034,fp1
fmove.s $5000,fp3
fmul.x fp3,fp1
fmove.s $6038,fp2
fmove.s $5004,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $603c,fp2
fmove.s $5008,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6040,fp2
fmove.s $500c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6064,fp2
fmove.s $5010,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6068,fp2
fmove.s $5014,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s fp1,$604c

```

ThetaMerror = measured steering angle
- calculated steering angle

Store ThetaMerror in 6064

Calculate Epsilonerror

Epsilonerror = measured Epsilon
- calculated Epsilon

note: calculated Epsilon = Theta - Rho

Store Epsilonerror in 6068

Estimator Portion of program

Calculate ThetaM(K + 1)

This is the estimated next state (position)
of the motor angle.

Store ThetaM (K + 1) in 604c.


```

bra 24000
.
mm 24000;di
fmove.s $6034,fp1
fmove.s $5018,fp3
fmul.x fp3,fp1
fmove.s $6038,fp2
fmove.s $501c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $603c,fp2
fmove.s $5020,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6040,fp2
fmove.s $5024,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6044,fp2
fmove.s $5028,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6064,fp2
fmove.s $502c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6068,fp2
fmove.s $5030,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $606c,fp2
fmove.s $50a0,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s fp1,$6050
bra 25000

```

Calculate $\Theta_{\dot{M}}(K + 1)$
This is the estimated next state (velocity)
of the motor angular velocity.

Store $\Theta_{\dot{M}}(K + 1)$ in 6050

```

.
mm 25000;di

```

Calculate $\Theta(K + 1)$

```

fmove.s $603c,fp1
fmove.s $5034,fp3
fmul.x fp3,fp1
fmove.s $6040,fp2
fmove.s $5038,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6064,fp2
fmove.s $503c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6068,fp2
fmove.s $5040,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s fp1,$6054
bra 26000

```

This is the estimated next state (position)
of the bicycle lean angle.

Store Theta(K + 1) in 6054

```

.
mm 26000;di
fmove.s $6034,fp1
fmove.s $5044,fp3
fmul.x fp3,fp1
fmove.s $603c,fp2
fmove.s $5048,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6040,fp2
fmove.s $504c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6064,fp2
fmove.s $5050,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6068,fp2
fmove.s $5054,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1

```

Calculate Thetadot(K + 1)

This is the estimated next state (velocity)
of the bicycle lean velocity.

```
fmove.s fp1,$6058
bra 27000
```

Store Thetadot(K + 1) in 6058

```
.
mm 27000;di
fmove.s $6034,fp1
fmove.s $5058,fp3
fmul.x fp3,fp1
fmove.s $603c,fp2
fmove.s $505c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6044,fp2
fmove.s $5060,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6048,fp2
fmove.s $5064,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6064,fp2
fmove.s $5068,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6068,fp2
fmove.s $506c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s fp1,$605c
bra 28000
```

Calculate Rho(K + 1)

This is the estimated next state (position)
of the inclinometer lean angle.

```
.
mm 28000;di
fmove.s $6034,fp1
fmove.s $5070,fp3
fmul.x fp3,fp1
fmove.s $6038,fp2
fmove.s $5074,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
```

Store Rho(K + 1) in 605C.

Calculate Rhodot(K + 1)

This is the estimated next state (velocity)
of the inclinometer lean velocity.

```

fmove.s $603c,fp2
fmove.s $5078,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6040,fp2
fmove.s $507c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6044,fp2
fmove.s $5080,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6048,fp2
fmove.s $5084,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6064,fp2
fmove.s $5088,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6068,fp2
fmove.s $508c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s fp1,$6060
bra 29000

```

Store Rhodot($K + 1$) in 6060.

```

mm 29000;di
fmove.s $604c,fp1
fmove.s $5090,fp3
fmul.x fp3,fp1
fmove.s $6050,fp2
fmove.s $5094,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6054,fp2
fmove.s $5098,fp3

```

Controller portion of program

This part of the program calculates the next voltage to send to the motor.

The estimated next states of the model are multiplied by their corresponding feed back gain to calculate the motor voltage.

Calculated voltages larger than rated voltage (+40 to -40 volts) are clipped at 31000.

```

fmul.x fp3,fp2
fadd.x fp2,fp1
fmove.s $6058,fp2
fmove.s $509c,fp3
fmul.x fp3,fp2
fadd.x fp2,fp1
fadd.s $50ac,fp1
fmove.s fp1,$606c

```

Store the motor voltage in 606C.

```

move.l $604c,$6034
move.l $6050,$6038
move.l $6054,$603c
move.l $6058,$6040
move.l $605c,$6044
move.l $6060,$6048
move.l $6004,$6074

```

This block of code moves the next bicycle state just calculated into the present state. When B routine is called on the next iteration these values will be used to calculate the next state.

```
fcmp.s #0,fp1
```

Motor voltage is still in fp1.

(e = motor voltage)

```
fbgt $29100
```

If $e > 1$ then goto 29100.

```
fabs.x fp1
```

Else take the absolute value of e and

```
move.l #0,$6020
```

set DFLAG = 1 (left).

```
bra $29200
```

goto 29200

```
.
```

```
mm 29100;di
```

{ When $e > 0$ }

```
move.l #1,$6020
```

set DFLAG = 0 (right).

```
bra $29200
```

goto 29200

```
.
```

```
mm 29200;di
```

Overrun error checking

```
fmove.l fp1,d1
```

```
cmp.l #7FFFFFFF,d1
```

```
beq 29400
```

If an overrun occurs goto 29400 and

```
cmp.l #7FFFFFF0,d1
```

turn the motor off.

```
beq 29400
```

```
fcmp.s #2,fp1
```

If $|e| <$ then goto 29400 and do not turn the motor on next iteration.

```
fblt 29400
```

```
fcmp.s #40,fp1
```

If $|e| >$ then goto 29300 and turn the motor on 100% for the next iteration.

```
fbgt $29300
```

```

fdiv.s #40,fp1
fmove.l $6018,fp2
fsub.l #1,fp2
fmul.x fp1,fp2
fadd.s #1,fp2
fmove.l fp2,d1
move.l d1,$6010
bra $31000

```

```

.
mm 29300;di
move.l $6018,d1
move.l d1,$6010
bra $31000

```

```

.
mm 29400;di
move.l #0,$6010
bra $31000
.

```

```

mm 31000;di
movea.l $6030,a1

```

```

move.l $6000,(a1)+
move.l $6004,(a1)+
move.l $604c,(a1)+
move.l $6050,(a1)+
move.l $6054,(a1)+
move.l $6058,(a1)+
move.l $605c,(a1)+
move.l $6060,(a1)+
move.l $606c,(a1)+
move.l $6078,(a1)+

```

```

fmove.s $606c,fp1
fcmp.s #40,fp1
fbgt 31100
fcmp.s #-40,fp1
fblt 31200

```

Else calculate MOTCNT

$$\text{MOTCNT} = (|e|/40) * (\text{BCNT} - 1) + 1$$

This is the number of iterations of the A routine will leave the motor on during the next iteration of the B routine.

{ When $|e| > 40$ }

Set MOTCNT = BCNT

(leave the motor on all the time during the next .01 second)

{ When $|e| < 2$ }

Set MOTCNT = 0

(Do not turn the motor on during the next .01 second)

This is the variable storage routine.

Set the A1 register to the next RAM address (this starts at \$40000).

Store Epsilon (auto. increment address)

Store Steering counter

Store ThetaM(K + 1)

Store ThetaMdot(K + 1)

Store Theta(K + 1)

Store Thetadot(K + 1)

Store Rho(K + 1)

Store Rhodot(K + 1)

Store motor voltage

Store steering velocity

If $e > 40$ then goto 31100

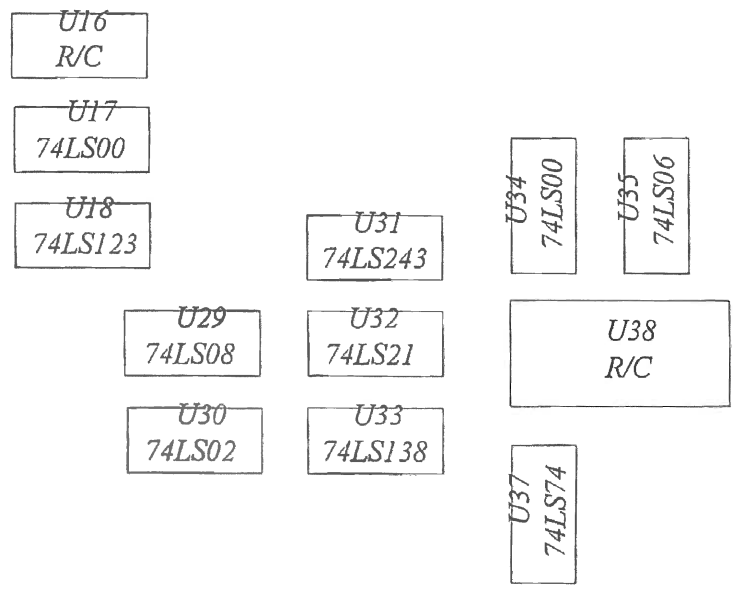
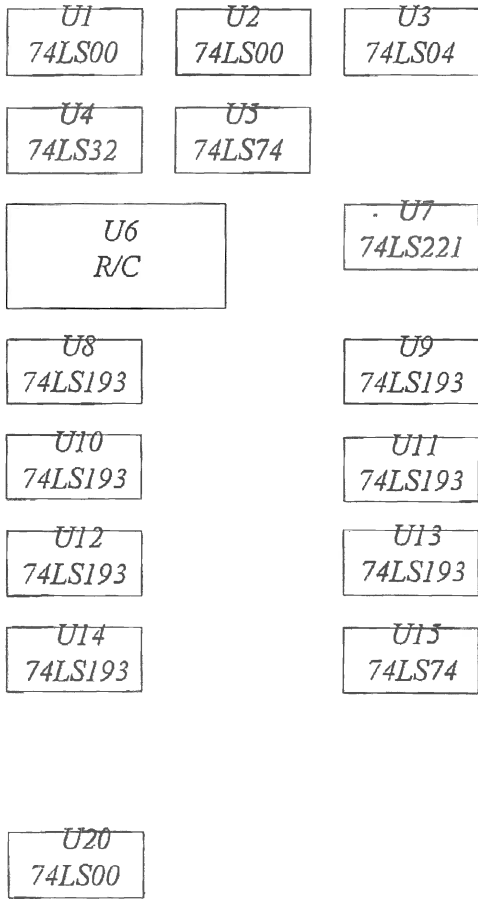
If $e < -40$ then goto 31200

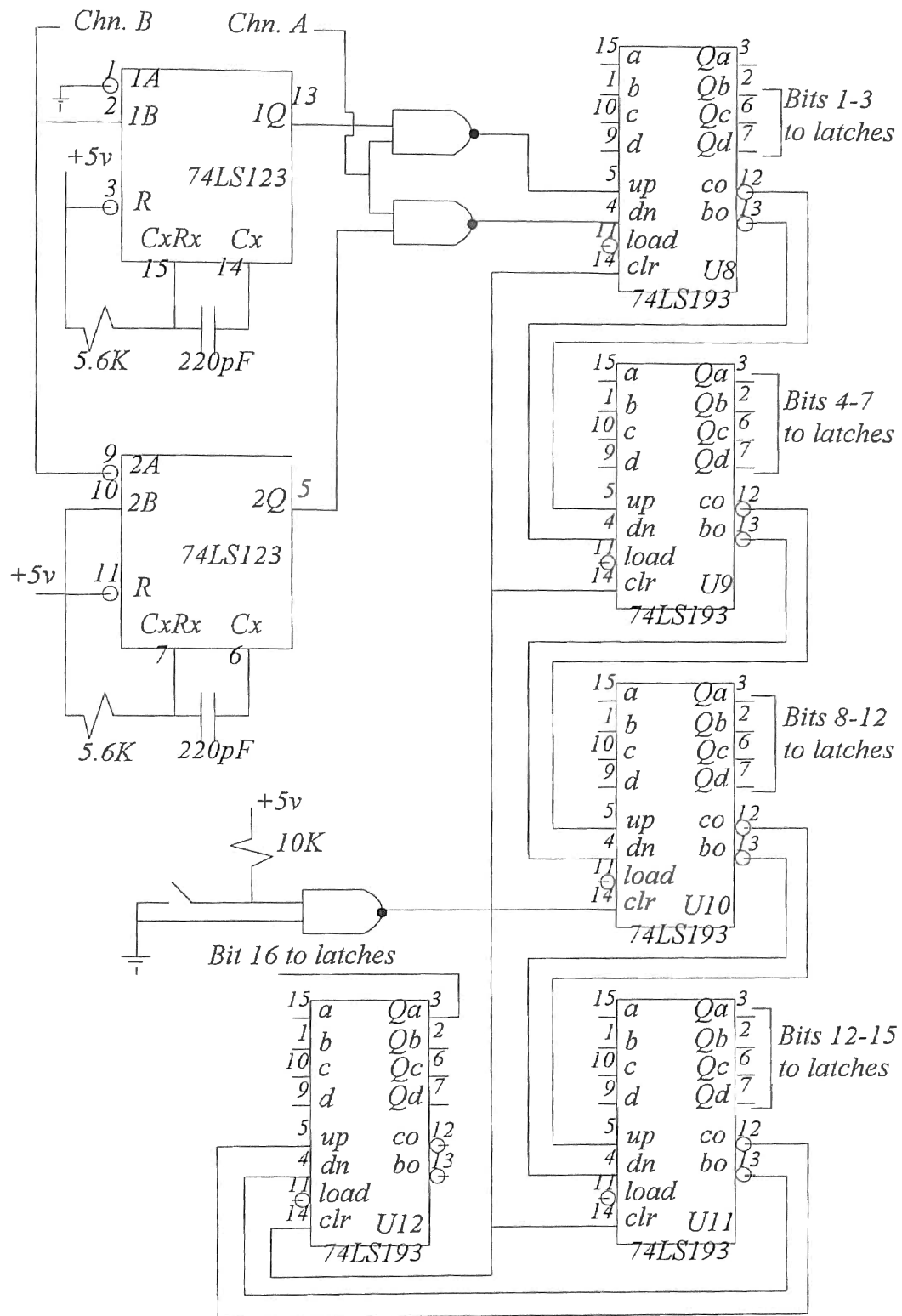
bra 31400	Else goto 31400
.	
mm 31100;di	{ When e > 40 } Motor voltage clip
fmove.s #40,fp1	Let e = 40 volts
fmove.s fp1,\$606c	
bra 31400	
.	
mm 31200;di	{ When e < -40 } Motor voltage clip
fmove.s #-40,fp1	Let e = - 40 volts
fmove.s fp1,\$606c	
bra 31400	
.	
mm 31400;di	If the a1 redgester has reached the end of
cmpa.l #\$ffff0,a1	memory, reset \$6030 to 40000 and begin
blt \$31500	storing new data over old data and set
movea.l #\$40000,a1	wrap flag (6028).
move.l #-1,d5	
move.l d5,\$6028	
move.l a1,\$6030	
move.l #0,\$6008	
rte	
.	
mm 31500;di	Set BEXE = 0 because the B routine is done.
move.l a1,\$6030	Load next storage address into 6030.
move.l #0,\$6008	
rte	
.	

Appendix N

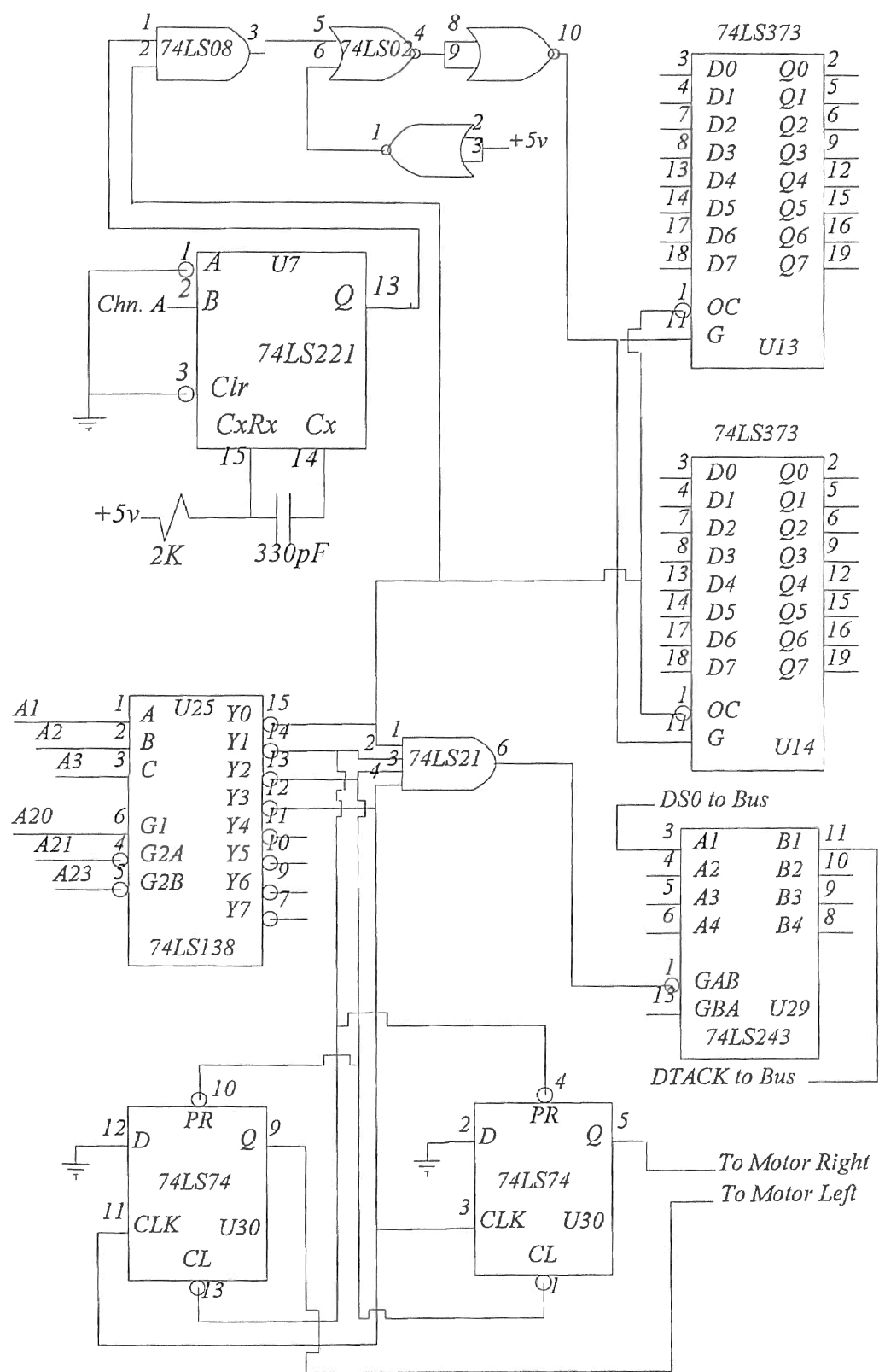
Overall System Block Diagram and Main Interface Hardware

Chip Layout

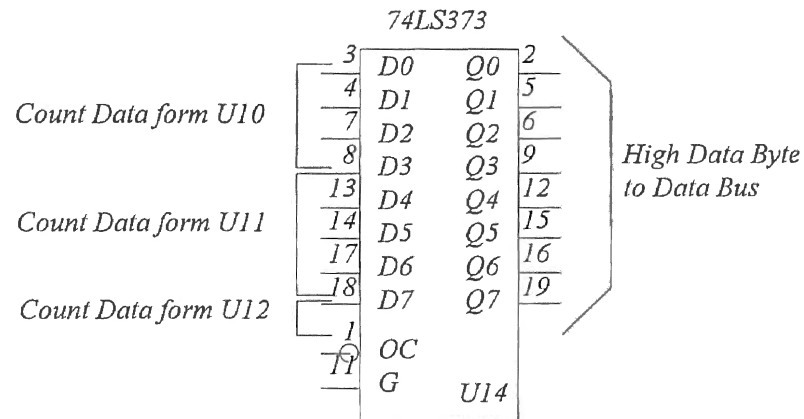
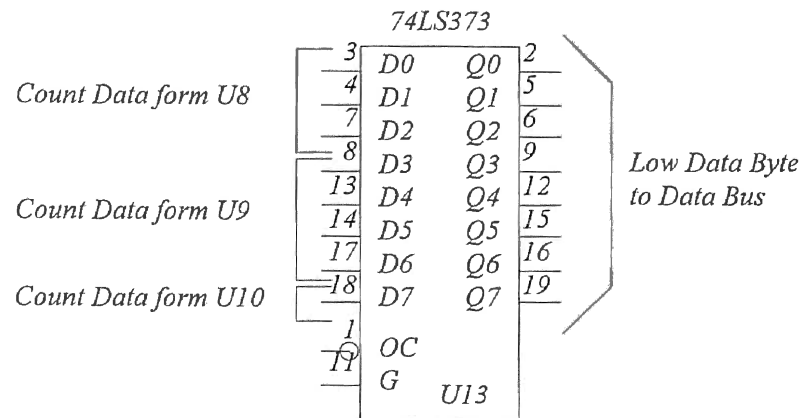




Steering Position Counter



Memory Decoder Circuit



Latch Detail from Counter

VITA 2

Raymond Philip Caudle

Candidate for the Degree of

Master of Science

Thesis: DEVELOPMENT OF A TEST PLATFORM FOR EXPERIMENTAL
TESTING OF BICYCLE MODELS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Tulsa, Oklahoma, June 21, 1967, the son of Dr. Raymond Philip and Wilma M. Caudle; one sister, Carolyn Marie Caudle.

Education: Graduated from Holland Hall School, Tulsa, Oklahoma, May 1986; received Bachelor of Science Degree in Electrical Engineering from Oklahoma State University, Stillwater, Oklahoma, May 1991; completed the requirement for the Master of Science Degree at Oklahoma State University, December 1993.