

DEVELOPMENT AND FEASIBILITY OF OPEN-SOURCE HARDWARE  
AND SOFTWARE IN CONTROL THEORY APPLICATION

by

DEREK J. BLACK

B.S., Kansas State University, 2014

---

A THESIS

submitted in partial fulfillment of the  
requirements for the degree

MASTER OF SCIENCE

Department of Mechanical and Nuclear Engineering  
College of Engineering

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

2017

Approved by:

Major Professor  
Dr. Dale Schinstock

# Copyright

DEREK J. BLACK

2017

# Abstract

Control theory is a methodology investigated by many mechanical and electrical engineering students throughout most universities in the world. Because of control theory's broad and interdisciplinary nature, it necessitates further study by application through laboratory practice. Typically the hardware used to connect the theoretical aspects of controls to the practical can be expensive, big, and time consuming to the students and instructors teaching on the equipment. This is due to the fact that connecting various hardware components such as sensors, encoders, amplifiers, and motors can lead to data that does not fit perfectly the theoretical mold developed in the controls classroom, further dissuading students of the idea that there exists a connection between developed theoretical models and what is seen in practice.

There is a recent trend in universities wishing to develop open-source, inexpensive hardware for various applications. This thesis will investigate and conduct a multitude of experiments on an apparatus known as the Motorlab to determine the feasibility of such equipment in the field of control theory application. The results will be compared against time-tested hardware to demonstrate the practicality of open-source, inexpensive hardware.

# Table of Contents

List of Figures . . . . .	vii
List of Tables . . . . .	ix
Acronyms . . . . .	x
Acknowledgements . . . . .	x
1 Introduction . . . . .	1
2 Apparatus . . . . .	3
2.1 NERMLAB . . . . .	3
2.1.1 NERMLAB Hardware . . . . .	3
2.1.2 Position Sensor . . . . .	5
2.1.3 NERMLAB Parts . . . . .	8
2.1.4 NERMLAB Cost . . . . .	9
2.1.5 NERMLAB GUI . . . . .	10
2.2 Motorlab . . . . .	11
2.2.1 Motorlab Hardware . . . . .	12
2.2.2 Motorlab GUI . . . . .	12
3 System Identification . . . . .	14
3.1 Motor Resistance . . . . .	15
3.1.1 Resistance Estimation . . . . .	15
3.2 Motor Torque Constant and Back EMF . . . . .	16

3.2.1	Estimating the Back EMF Constant . . . . .	17
3.3	Mass Moment of Inertia Estimation . . . . .	19
3.3.1	Software Modeling of Mass Moment of Inertia . . . . .	19
3.3.2	Mathematical Approximation of Mass Moment of Inertia . . . . .	20
3.3.3	Lumped Mass Moment of Inertia . . . . .	21
3.4	Motor Inductance . . . . .	21
3.4.1	Inductance Estimate . . . . .	22
3.5	Viscous Friction . . . . .	23
4	Development of Mathematical Models for NERMLAB . . . . .	24
4.1	Electrical Dynamics . . . . .	24
4.2	Combined Dynamics - Electrical and Mechanical . . . . .	25
4.2.1	Position Models . . . . .	26
4.2.2	Speed Models . . . . .	28
4.2.3	Pendulum Model . . . . .	29
5	Approximating Friction of a BLDC Motor . . . . .	31
6	Experiment One . . . . .	35
6.1	Mathematical Model of a Closed-Loop Position Control System . . . . .	35
6.2	Experimental Results . . . . .	38
7	Experiment Two . . . . .	39
Bibliography	. . . . .	40
A	Experiment Documentation . . . . .	41
B	Part Drawings . . . . .	42
C	Laboratory Procedures . . . . .	46

D	Code	53
E	System Specifications	56

# List of Figures

2.1	NERMLAB . . . . .	4
2.2	Magnet and AS5047D . . . . .	6
2.3	Sensor Noise with Changing Speed . . . . .	7
2.4	Section View of Motorlab Assembly . . . . .	8
2.5	NERMLAB GUI . . . . .	10
2.6	Motorlab . . . . .	11
2.7	Motorlab GUI in MATLAB . . . . .	12
3.1	Motor Connection Configuration . . . . .	15
3.2	Measured Back EMF vs Speed . . . . .	18
3.3	Inductance Measurement Circuit . . . . .	22
4.1	Electrical and Mechanical Schematic of NERMLAB . . . . .	24
4.2	Position Model . . . . .	26
4.3	NERMLAB Speed Model . . . . .	28
4.4	NERMLAB Pendulum Model . . . . .	29
6.1	NERMLAB Model . . . . .	36
6.2	Block Diagram of Closed Loop Control System . . . . .	36
6.3	Block Diagram Reduction . . . . .	37
6.4	Base Load Inertia Step Responses . . . . .	38
7.1	Block Diagram of Closed Loop Speed Control System . . . . .	39
7.2	Block Diagram of Open Loop System (Speed Control) . . . . .	39

A.1	Back emf at various motor speeds . . . . .	41
B.1	Standoff for NERMLAB . . . . .	43
B.2	Magnet holder for NERMLAB . . . . .	44
B.3	Torque Transmission Shaft for NERMLAB . . . . .	45
D.1	Encoder Noise Experiment . . . . .	54
D.2	Back-EMF Plotter . . . . .	55

# List of Tables

2.1	Noise Experiment . . . . .	8
2.2	Motorlab expenditure report . . . . .	9
3.1	Motor Parameters Nomenclature . . . . .	14
3.2	Measured motor resistance . . . . .	16
3.3	Measured back voltage . . . . .	17
3.4	Measured Inertia Parameters . . . . .	19
3.5	Inductance Experiment . . . . .	23
E.1	NERMLAB Parameters . . . . .	56
E.2	Motorlab Parameters . . . . .	57

# Acronyms

<b>ARM</b>	Advanced RISC Machine
<b>DAEC</b>	Dynamic Angle Error Compensation
<b>MPU</b>	Microprocessor Unit
<b>BLDC</b>	Brushless DC
<b>GUI</b>	Graphical User Interface
<b>NERMLAB</b>	New Earth Robotics Motor Lab
<b>back-emf</b>	back electromotive force
<b>RPM</b>	Rotations Per Minute
<b>CAD</b>	Computer Aided Design
<b>SPI</b>	Serial Peripheral Interface Bus
<b>PWM</b>	Pulse Width Modulation
<b>JSON</b>	JavaScript Object Notation

# Acknowledgments

Enter the text for your Acknowledgements page in the acknowledge.tex file. The Acknowledgements page is optional. If you wish to remove it, see the comments in the etdrtemplate.tex file.

# Chapter 1

## Introduction

Current research indicates a growing need for laboratory components for introductory control theory classes. However, many hurdles like budget, class size, and space limitations arise when laboratories are appended to lectures in universities [1, 2]. The New Earth Robotics Motor Lab ([NERMLAB](#)) aims to address these concerns in reducing the overall cost imposed on instructors and students, as well as, minimizing the foot print of the hardware to allow students to take part in laboratories in a home environment. It is this home experimentation that allows students to engage in experimental learning, which is a methodology that aims at creating knowledge through wisdom, observation, and insight from experience. Experimental learning also provides an alternative learning mechanism for the traditional theoretical components that make up a standard engineering curriculum. Most experimental learning is achieved through a laboratory practicum that helps students connect the theoretical ideas developed in lecture with what is done in practice. As a result, students can gain further insight into the theory that might have gone unresolved without experimental learning [2].

Unfortunately, classroom sizes continue to grow in universities, and, a direct result of this, is increasing laboratory size. Since size means the cost per student increases due to the limited amount of equipment available, there is a desire for more affordable hardware [4]. A way to combat the issue would be to make laboratory hardware more portable, allowing for cheaper components, such as motors, motor drivers and the like to be used [2]. However,

utilzing cost-effective hardware in laboratory equipment can lead to poorly produced data, which does not adhere to theoretical models developed in lecture. While it is true that cheaper hardware does lead to less than desired data, it does allow greater access to students because of its cost. The goal of NERMLAB is to give students access to affordable equipment that can provide them with the experimental learning opportunities both in the classroom and at home. In addition, it is this aspect of portability that is of importance because students will be allowed to learn at their own pace, in a way that benefits them the most, and still achieve the same learning objective as that of a traditional on-campus laboratories [2].

This thesis will attempt to address the feasability of the cheaper NERMLAB alternative. Multiple experiments will be conducted as they appear in Appendix C and results will be compared to more expensive hardware, such as the Motorlab. Chapter 2 will describe the NERMLAB system apparatus and the various components that comprise it, as well as, comment on the differences between the NERMLAB and the older Motorlab system. Then, Chapter 3 will discuss system identification and characterization, which produces things such as the motor torque constant, inductance, and resistance. Chapter 4 will then develop the necessary mathematical models that are necessary for the experiments that make up chapters 5-7.

# Chapter 2

## Apparatus

This chapter will discuss two apparatus pieces used for conducting a series of five experiments included in this thesis and will, in detail, describe the purpose, design, and recreation of the equipment. Section 2.1 will describe the NERMLAB, including the hardware implementation, design of components, basic functionality, and use of the position sensor. In comparison, section 2.2 will describe the Motorlab, the model currently being used in Kansas State University’s engineering laboratories, and how it differs from NERMLAB.

### 2.1 NERMLAB

The NERMLAB is a reimplementation of older laboratory hardware created by Dr. Dale Schinstock and Dr. Warren White for Control of Mechanical Systems I at Kansas State University. This equipment allows users to connect the theoretical ideas of control theory with those in practice.

#### 2.1.1 NERMLAB Hardware

The NERMLAB consists of several key pieces of hardware, including: an STM32 Nucleo development board, motor driver, and a Brushless DC ([BLDC](#)) motor (Figure 2.1). The STM32 Nucleo houses a STM32F401RE Microprocessor Unit ([MPU](#)), which is a 32-bit pro-

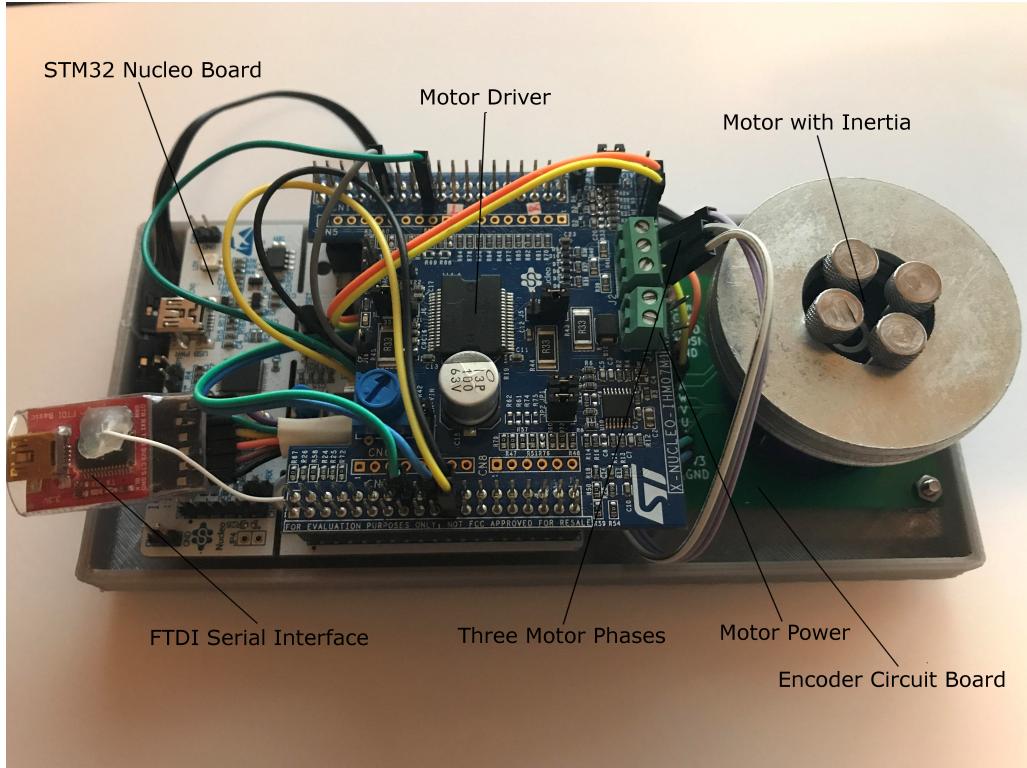


Figure 2.1: NERMLAB

cessor with an 84 MHz clock speed and up to 512 Kbytes of flash memory. The STM32 Nucleo also allows Arduino shields and other STM boards to be attached for added functionality.

A motor driver was required to drive a brushless DC motor. As a result, an X-Nucleo-IHM07M1 (a three-phase brushless DC motor driver) was selected to be the primary driver for the NERMLAB. The X-Nucleo has a nominal operating voltage of 8V-48 VDC with a 2.8 A peak current output, which is sufficient to drive a BLDC gimbal motor, such as the RCTIMER GBM2804, which is the primary motor used in this thesis.

The RCTIMER GBM2804 is a 100 turn BLDC motor that has a hollow shaft which allows placement of a position sensor for feedback control purposes. Motor specifications were not given by the manufacturer of this motor, so chapter 3 details the experiments that were conducted to find the various parameters needed to adequately model the entire NERMLAB system.

### 2.1.2 Position Sensor

The main purpose of the NERMLAB is to conduct control laboratory experiments. To accomplish this, feedback via sensor readings is necessary, and the typical way to do position and speed control is to use position feedback via an encoder. An encoder is a device that converts angular position of a motor shaft to an analog or digital signal that can be processed by an MPU. In the case of the NERMLAB, an on-axis magnetic encoder is used to do position feedback. Special equipment had to be designed in order to use this type of encoder, and will be detailed in section 2.1.3.

The encoder that is being used consists of 14-bit on-axis magnetic rotary position sensor chip, specifically the AS5047D by AMS <sup>1</sup>. The position sensor chip provides high resolution absolute angle measurements through a full 360 degree range <sup>2</sup>. In addition to the fast absolute angle measurement system that the position sensor provides, it also has Dynamic Angle Error Compensation ([DAEC](#)) that provides position control systems with near 0 latency [3].

The AS5047D chip is a magnetic sensor that utilizes the Hall-effect. The chip works by taking the Hall sensors and converting the perpendicular magnetic field on the surface of the chip to a voltage. The voltage signals are filtered and amplified in order to calculate the angle of the magnetic vector. In order for position measurements to be taken, a small diametrically opposed magnet must be placed on the shaft of the equipment being measured. The magnet and AS5047D are contactless, meaning there is a small air gap between the chip and magnet. As the magnet rotates above the chip (Figure 2.2), angle measurements are calculated and transmitted through the chip [3].

### Sensor Output

The AS5047D has multiple input/output types that can be used for feedback and chip programming. A Serial Peripheral Interface Bus ([SPI](#)) is the main input to the chip that allows a one time programming operation to be carried out. The chip also outputs an ABI

---

<sup>1</sup>AMS is an Austrian analog sensor and semi-conductor manufacturer

<sup>2</sup>These chips typically provide a maximum resolution of 2000 steps/revolution in decimal mode and 2048 steps/revolution in binary mode

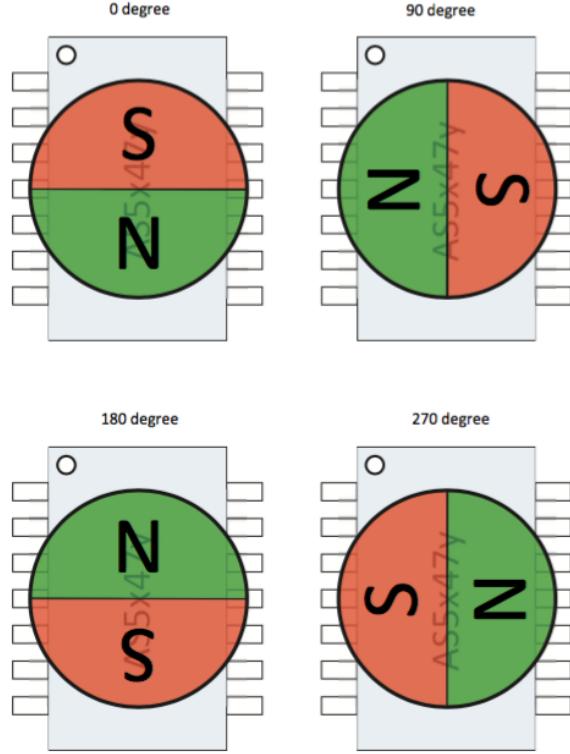


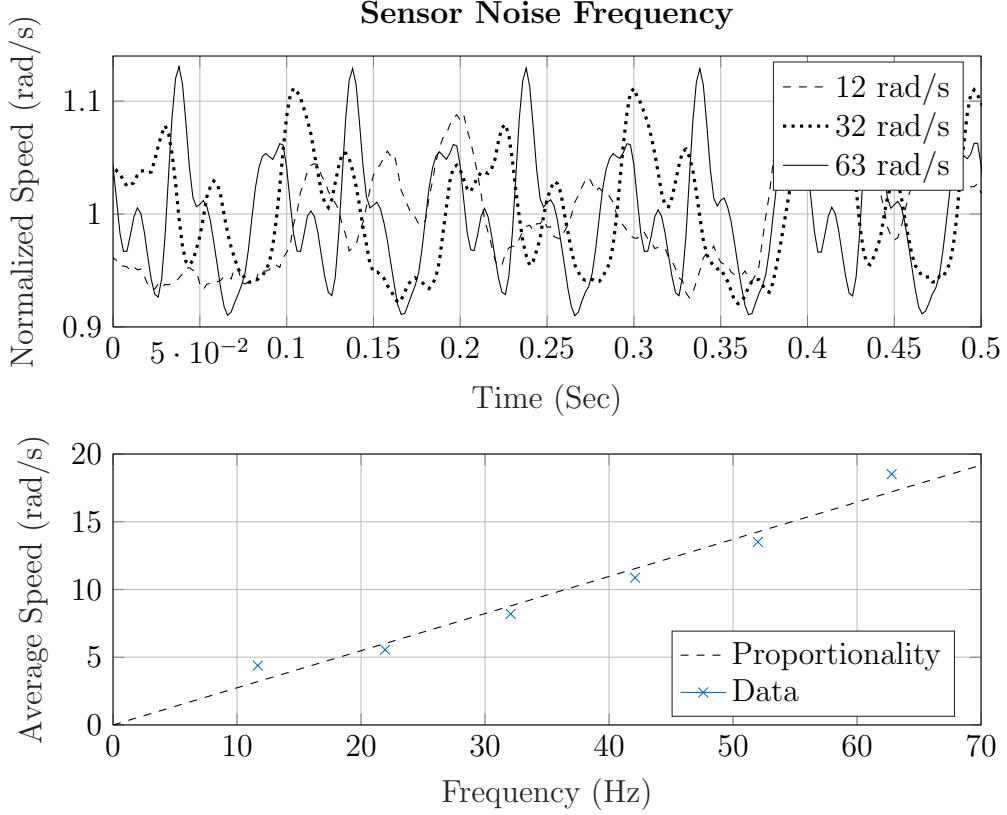
Figure 2.2: Magnet and AS5047D [3]

and Pulse Width Modulation ([PWM](#)) signal that can be used in feedback measurements. In the case of the NERMLAB, the ABI output is the chosen signal type to be used for encoder readings. The ABI is an incremental type signal that has two 90 degree offset signals that indicate motor direction. To determine the motor's position, one only needs to count the number of pulses coming from the chip either from the leading or falling edge of the signal. From there its possible to use equation [2.1](#) to come up with the position in radians, where  $n_{resol}$  is the resolution of the encoder output and  $n_{count}$  is the current encoder count.

$$\theta_{rad} = 2\pi \frac{n_{count}}{n_{resol}} \quad (2.1)$$

## Sensor Noise

Figure 2.3: Sensor Noise with Changing Speed



While the AS5047D does have a high resolution output in comparison to other cheap encoders, the chip does suffer from measurement inaccuracies. These inaccuracies could be a result of the magnet's rotation or the tolerance and fit of the magnet holder. It is when doing speed control that these inaccuracies show up on the NERMLAB as superimposed noise on the response. It was found through experimentation that the frequency of the noise was proportional to the speed of the rotor. The NERMLAB was run at incremental speeds, and the frequency of the noise was tabulated in table 2.1. It is evident from figure 2.3 that as the speed of the rotor is increased, the frequency of the noise increases proportionally, as described by equation 2.2.

$$f = 0.2741v \quad (2.2)$$

Table 2.1: Noise Experiment

Average Speed (rad/s)	Frequency (Hz)	Input Voltage (V)
11.67	4.386	1.5
21.93	5.55	2.5
32.07	8.196	3.5
42.10	10.869	4.5
52.00	13.510	5.5
62.79	18.518	6.5

While the exact problem is not pinpointed in this thesis, the measurement inaccuracy does not greatly affect the experimental results and is more so another feature that must be realized when doing system analysis.

### 2.1.3 NERMLAB Parts

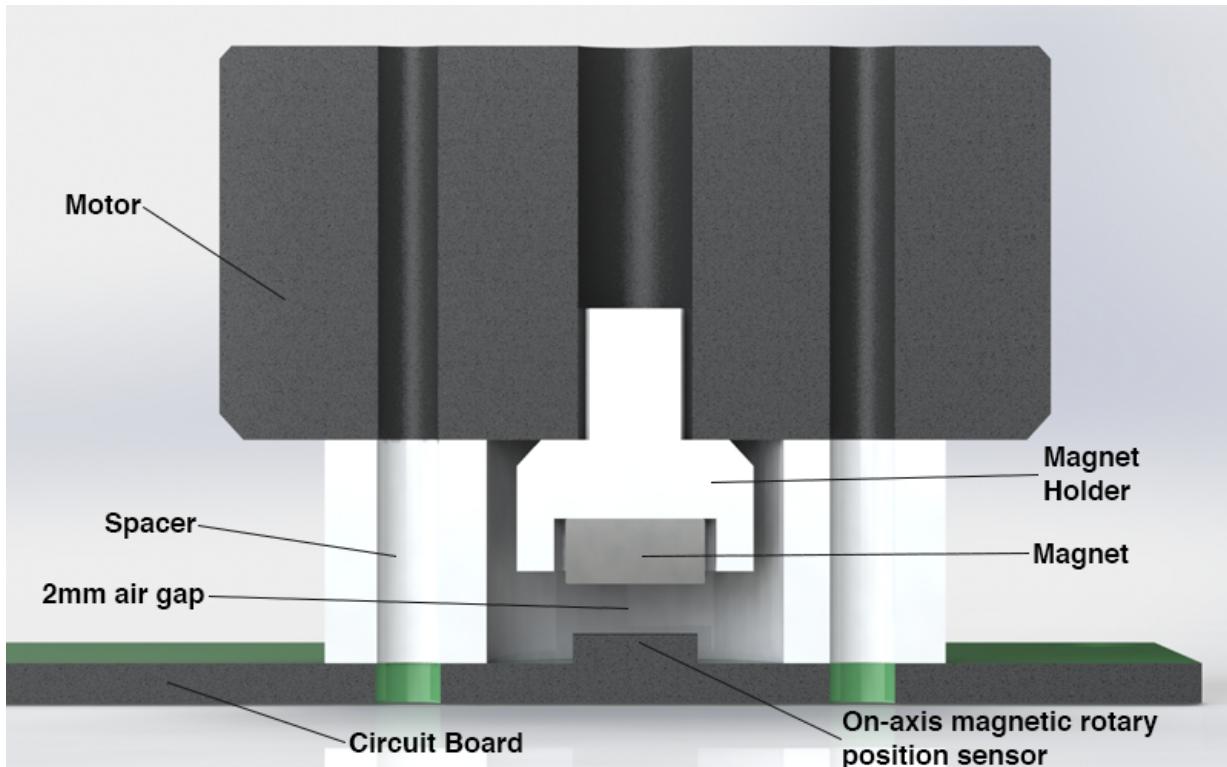


Figure 2.4: Section View of Motorlab Assembly

Along with the hardware mentioned in section 2.1.1, three components were needed to be developed in order to bring the NERMLAB to fruition: a printed circuit board that houses the on-axis magnetic rotary position sensor, a spacer to put distance between the circuit board and the motor, and a magnet holder, which holds one diametrically opposed magnet <sup>3</sup>. Both the spacer and magnet holder, which can be seen in figure 2.4, had to be 3D printed in order to achieve the required specifications of the apparatus setup. Detailed drawings of these two parts can be found in Appendix B, figures B.1 and B.2, if reproduction is desired. Along with the two 3D printed parts, a printed circuit board had to be designed to allow the position sensor to communicate with the rest of the hardware.

Because of variability in resolution of current 3D printers, care was given to the design of the magnet holder <sup>4</sup>. A spline was used for both the shaft of the magnet holder and the section that holds the magnet itself. The spline allowed for greater tolerances in the parts, meaning the magnet holder could be easier to press fit into the motor, and likewise allowed easier removal of the diametrically opposed magnet.

#### 2.1.4 NERMLAB Cost

Table 2.2 lists the components that make up the NERMLAB system and their associated price at the writing of this thesis.

Table 2.2: Motorlab expenditure report

Component	Brand/Manufacture	Cost
BLDC Motor	RCTIMER GBM2804	11.94 USD
Position Sensor	AS5047D AMS	4.21 USD
ST32 Nucleo	STMicroelectronics	10.12 USD
X-Nucleo-IHM07M1	STMicroelectronics	9.80 USD
Magnet	-	3.00 USD
Printed Circuit Board	-	30.00 USD
<b>TOTAL COST</b>		<b>69.07 USD</b>

---

<sup>3</sup>Diametrically opposed meaning the north and south poles of the magnet are in-plane as opposed to top/bottom poles. Reference figure 2.2 for further clarification

<sup>4</sup>Because of this variability in resolution, the magnet holder was printed in iterations, varying the diameter.

## 2.1.5 NERMLAB GUI

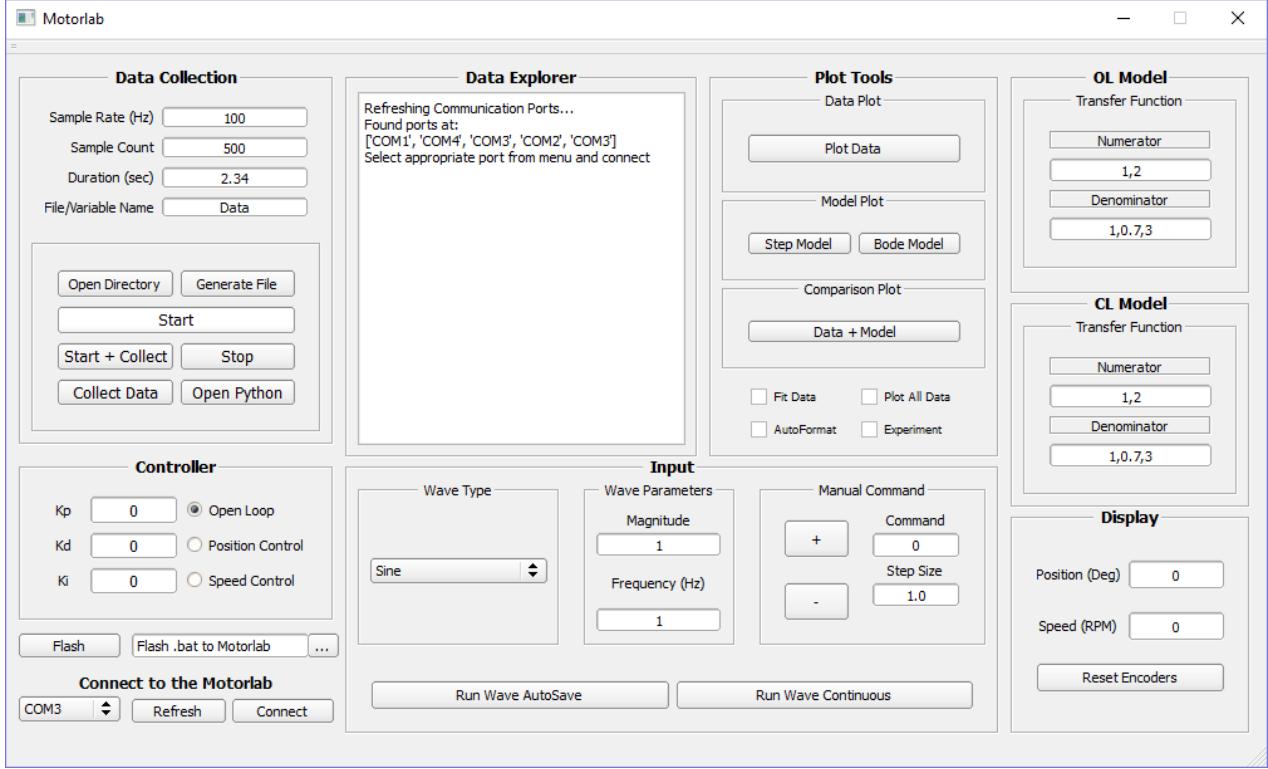


Figure 2.5: NERMLAB GUI

The NERMLAB GUI is an alternative way of interacting with the NERMLAB without having to hardcode values and re-flashing every time an experiment needs to be run. The user simply enters the desired settings into the GUI, which then triggers serial events in the back-end. The GUI is coded in python 2.7, utilizing a variety of different libraries such as the PyQt4 framework that allows cross platform development of GUI applications, matplotlib for plotting purposes, and variety of signal processing toolboxes for employing frequency response and model responses.

JavaScript Object Notation ([JSON](#)) is the main communication protocol that is used to allow back and forth communication between the NERMLAB and GUI interface. The GUI sends out JSON messages whenever a user triggers an event, and the back-end code of the NERMLAB then sees the object in its buffer, which is decoded into key-value pairings that can be processed.

Much of the NERMLAB GUI is still in development and is left as future work. Features that still need implementation would include: being able to flash hex and binary files directly to the NERMLAB system from the GUI, allowing python code to interact with the GUI/NERMLAB system, and better plotting capabilities for instant data visualization.

## 2.2 Motorlab

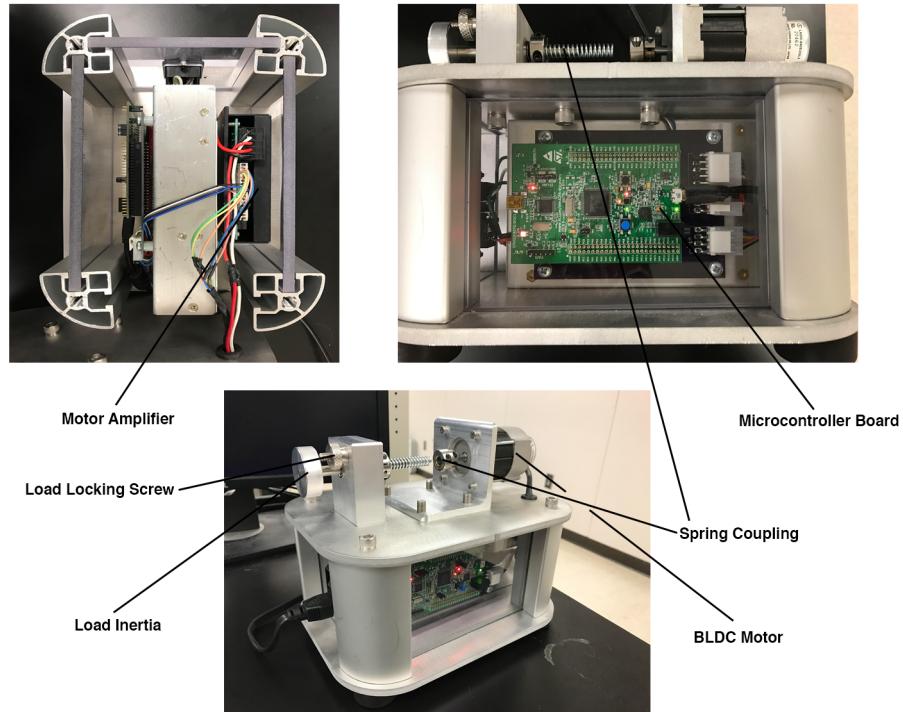


Figure 2.6: Motorlab

The Motorlab has been in service at Kansas State University for over 15 years, and as a result, is a time-tested piece of laboratory hardware that has proven to be reliable in providing quality data. Additionally, the Motorlab's hardware components were selected to ensure a very clear translation of laboratory results, allowing the Motorlab to have a much larger operating limit and bandwidth than students use in laboratory. As a result, the Motorlab represents to a good base model to compare the results of the NERMLAB

apparatus in this thesis too.

### 2.2.1 Motorlab Hardware

Various hardware make up the Motorlab, namely, a BLDC motor, BLDC servo amplifier by Copley Controls Corp., and a ST Discovery board <sup>5</sup>. Typically the Motorlabs run a cost of about 700 USD per lab station [4]; however, this estimate does not include things like manufacturing, design and labor, as these processes were carried out at Kansas State University. Appendix E, figure E.2 hosts the system parameters that make up the Motorlab.

### 2.2.2 Motorlab GUI

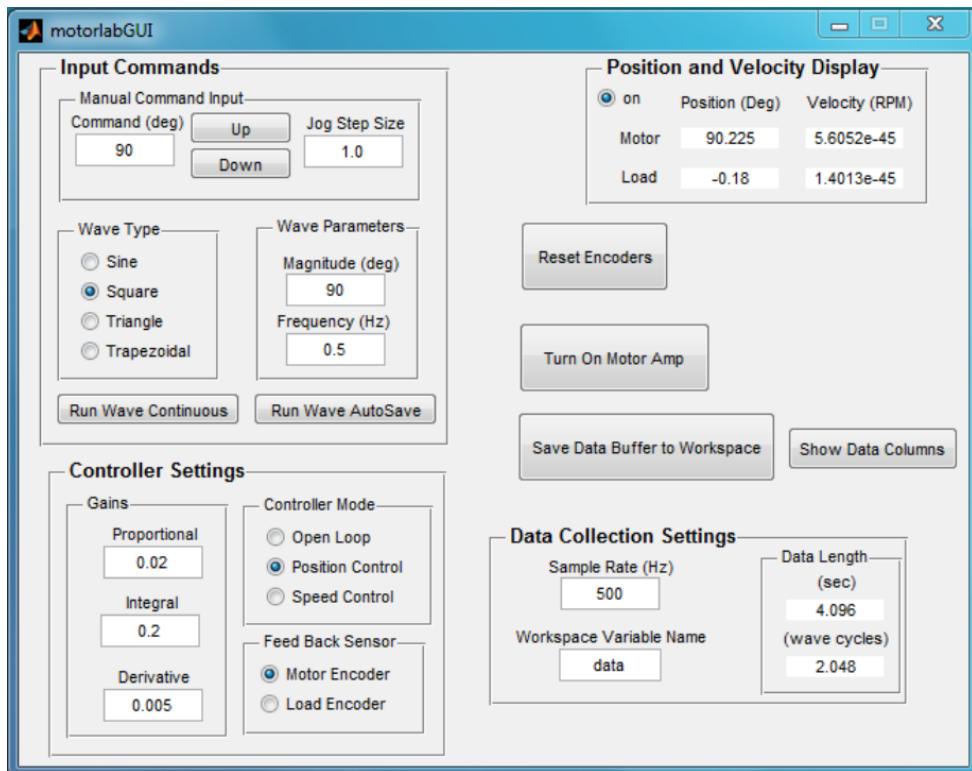


Figure 2.7: Motorlab GUI in MATLAB

The Motorlab interfaces with a Graphical User Interface (GUI) coded in MATLAB to allow users to run various laboratory experiments on the hardware. It allows the selection of various

---

<sup>5</sup>The ST Discovery Board is a 32-bit micro controller development board

wave types, frequency, controller gains, and sample rate that get sent to the Motorlab. After the parameters of the experiment are setup, the GUI can run the Motorlab, which in turn sends the experimental data to the workspace of MATLAB in the form of a matrix. The MATLAB GUI can be seen in figure [2.7](#).

# Chapter 3

## System Identification

Chapter 3 will be dedicated to developing the various parameters that make up the NERMLAB, such as the motor torque constant, back electromotive force ([back-emf](#)), inductance, and max voltage. Each section in Chapter 3 will detail the process of how the various parameters were measured, calculated, and experimentally determined. Nomenclature for various constants and parameters are detailed in table 3.1.

Table 3.1: Motor Parameters Nomenclature

Parameter	Description
V	Motor Voltage
$k_t$	Motor Torque Constant per Phase
$k_T$	Overall Motor Torque Constant
$K_e$	Line-Line Back Electromotive Force Constant
$K_{e,p}$	Back Electromotive Force Constant per Phase
J	Lumped Mass Moment of Inertia ( $3 \times J_w + J_r$ )
$J_w$	Washer Mass Moment of Inertia
$J_r$	Rotor Mass Moment of Inertia
$J_s$	Solidworks Approximated Rotor Mass Moment of Inertia
$J_m$	Mathematical Approximated Rotor Mass Moment of Inertia
L	Motor Inductance
R	Motor Phase Resistance
$R_{LL}$	Motor Line-Line Resistance
$\tau$	Time Constant
T	Motor Torque
$\omega_m$	Motor Speed

## 3.1 Motor Resistance

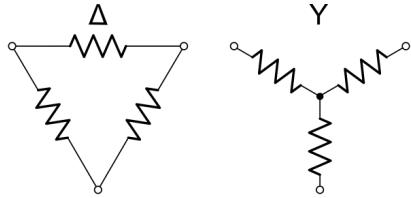


Figure 3.1: Motor Connection Configuration

BLDC motors are typically connected in two wiring configurations: WYE ( $Y$ ) or delta ( $\Delta$ ), which can be seen in figure 3.1. The RCTIMER GBM2804 utilizes the WYE ( $Y$ ) configuration and will be analyzed as such. Due to the wiring of WYE systems, the neutral connection is typically unavailable for measurement on most motors. As a result, it is common to measure resistance by a line-line reading; however, in terms of motor control, it is the phase resistance and not the line-line resistance that is of importance. Converting between the phase and line-line resistance is quite simple and can be done by dividing the line-line resistance by two (equation 3.1).

$$R = \frac{R_{LL}}{2} \quad (3.1)$$

### 3.1.1 Resistance Estimation

In order to gather a good estimate for the phase resistance of the RCTIMER GBM2804 motor, the resistance was measured line-line across all three phases. Each set of motor leads were hooked up to a digital multimeter, and the values were tabulated for each phase component in table 3.2. An average was then calculated between the different line-line resistances to get the overall resistance of the motor.

Table 3.2: Measured motor resistance

A-B	A-C	B-C
9.870 $\Omega$	9.900 $\Omega$	9.950 $\Omega$
<b>Average:</b>		9.91 $\Omega$

Using equation 3.1, it is then possible to find the overall phase resistance of the motor.

$$R = 4.955 \approx 5\Omega$$

## 3.2 Motor Torque Constant and Back EMF

The motor torque constant ( $k_t$ ) is a common parameter used in BLDC motors. It relates the armature current to the torque produced by a motor:  $T = k_t i$ . Many methods exist to determine the torque constant, including relating the motor velocity constant  $k_v$ , which is inversely related to the torque constant by  $k_T = \frac{1}{k_v}$ , or by measuring the line-line back-emf voltage per phase ( $K_{e,p}$ ).  $K_{e,p}$  is the peak value of the back-emf per angular velocity measured from line-neutral. However, since line-neutral is typically unavailable on most BLDC motors, the back-emf constant is often represented as a line measurement,  $K_e$ . The overall torque constant can then be related to the line measurement back-emf voltage for sinusoidal type outputs by equation 3.2 or for trapezoidal outputs by equation 3.3. [5].

$$k_T = \frac{\sqrt{3}}{2} K_e \quad (3.2)$$

$$k_T = K_e \quad (3.3)$$

Because  $K_e$  can be experimentally determined, it is possible to find the overall motor torque constant for a BLDC motor. One simply needs to measure the line-line sinusoidal or

trapezoidal back-emf voltage at various speeds to get a good estimate of  $K_e$ . With equation 3.2 or 3.3,  $k_T$  can then be determined.

### 3.2.1 Estimating the Back EMF Constant

In order to calculate the back-emf of the RCTIMER GBM2804 BLDC motor, an experiment had to be set up to measure the voltage generated by the motor. Three pieces of equipment were needed: an oscilloscope, the Motorlab, and a torque transmission shaft. The torque transmission shaft was a 3D printed part<sup>1</sup> that allowed the Motorlab to spin the RCTIMER GBM2804 at a constant speed to generate a back voltage. A line-line voltage (peak-peak) was then read from the leads of the RCTIMER GBM2804 by an oscilloscope<sup>2</sup>. The data collected is tabulated in table 3.3.

Table 3.3: Measured back voltage

Speed (RPM)	Speed $\omega_m$ (rad/s)	Peak-Peak Voltage (V)	Peak Voltage (V)
300	31.41	4.64	2.32
500	52.36	7.60	3.8
1000	104.72	15.1	7.55
1500	157.10	22.0	11.0
2000	209.44	30.0	15.0

There is a fairly linear relationship between the peak voltage and speed. Due to this fact,  $K_e$  can be approximated from the slope of  $\frac{V}{\omega_m}$ . The normal equation from the least-squares method was employed to find the best fit for the data in table 3.3. Two matrices were constructed from the data, namely  $\mathbf{V}$  and  $\boldsymbol{\omega}_m$ .

$$K_e = (\boldsymbol{\omega}_m \boldsymbol{\omega}_m^T)^{-1} \boldsymbol{\omega}_m \mathbf{V}^T \quad (3.4)$$

---

<sup>1</sup>Appendix B figure B.3

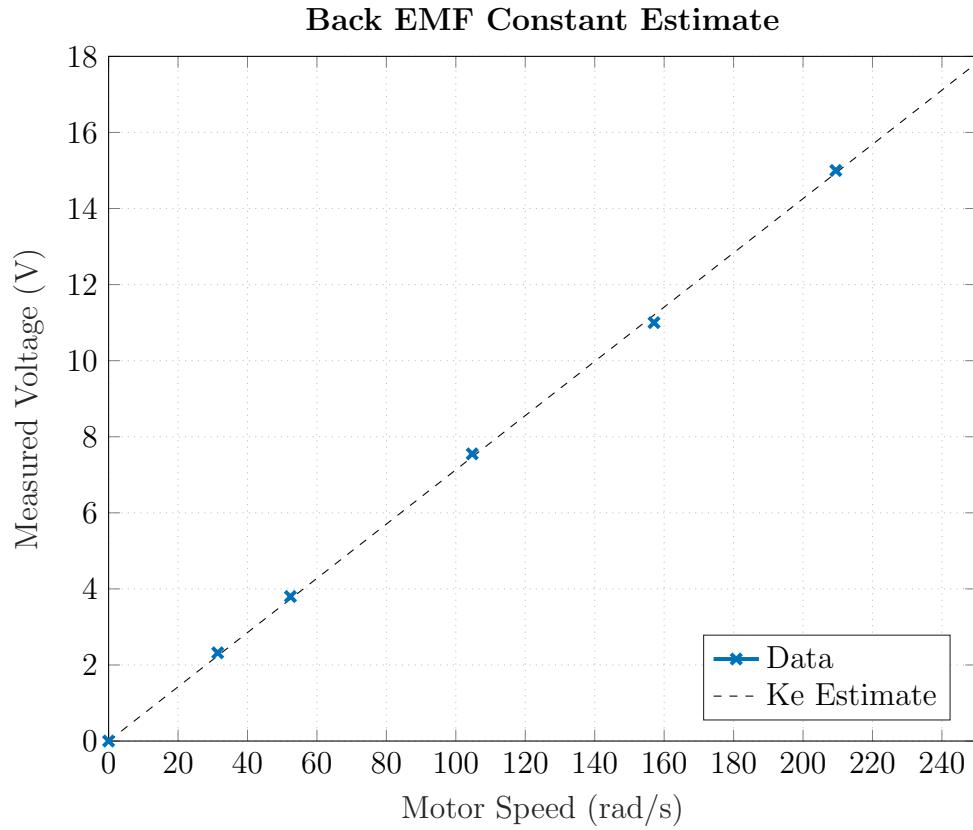
<sup>2</sup>Appendix A figure A.1

From equation 3.4, the back-emf constant was found to be:

$$K_e = 0.0713 \frac{V \cdot s}{rad}$$

To verify that  $K_e$  was the best fit to that data,  $K_e$  was plotted against the collected data in figure 3.2.

Figure 3.2: Measured Back EMF vs Speed



Since the relationship between  $k_T$  and  $K_e$  is known by equation 3.2 and 3.3,  $k_T$  can now be calculated.

$$k_T = 0.0617 \approx 0.06 \frac{N \cdot m}{A} [Sinusoidal]$$

$$k_T = 0.0713 \approx 0.07 \frac{N \cdot m}{A} [Trapezoidal]$$

### 3.3 Mass Moment of Inertia Estimation

Mass moment of inertia ( $J$ ) is the equivalent to mass in a rotational system (commonly referred to as angular mass). More formally, is it defined as  $J = \int r^2 dm$ , where  $r$  is the distance to a mass from an axis of rotation.

The angular mass of the NERMLAB will be determined in two ways: approximating  $J$  through software modeling and approximating  $J$  through mathematical formulation.

Multiple inertias will be referenced throughout this thesis, one being the base load inertia of the rotor, which is comprised of individual neodymium magnets held by an outer aluminum casing, and the second inertia being a simple steel washer. Most of section 3.3 will be devoted to developing the base load inertia as it has a more complicated geometry than the simple washer. Because of this, the washer will be only developed in the section 3.3.2. Table 3.4 tabulates the various parameters that are referenced in all mass moment of inertia calculations.

Table 3.4: Measured Inertia Parameters

Parameter	Value
Magnet Thickness	0.002 m
Rotor Outer Radius	0.0164 m
Rotor Mass	0.018 kg
Washer Mass	0.045 kg
Washer Inner Radius	0.0105 m
Washer Outer Radius	0.0253 m

#### 3.3.1 Software Modeling of Mass Moment of Inertia

Computer Aided Design ([CAD](#)) software was utilized to construct a 3-D model of the neodymium magnets and aluminum casing. CAD software like SolidWorks has the ability to determine complex mass moment of inertias via numerical methods. Knowing the

average density of neodymium ( $7.3 - 7.5 \frac{g}{cm^3}$ ) and aluminum ( $2.7 \frac{g}{cm^3}$ ), it is possible to numerically find an inertia ( $J_s$ ) of the rotor.

$$J_s = 4.0842 \times 10^{-6} \text{ kg} \cdot \text{m}^2$$

### 3.3.2 Mathematical Approximation of Mass Moment of Inertia

To simplify the mathematical analysis of the mass moment of inertia calculation of the angular mass of the NERMLAB, an engineering assumption will be made that the angular mass is a rotating ring mass. This assumption is valid for the particular motor used in this thesis, due to the fact that most of the mass is concentrated around the outside parameter of the motor. The outside ring mass of the motor contributes the most to the inertial load, so the mathematical formulation would result in the following equation:

$$J_z = mr^2 \quad (3.5)$$

Knowing the outer radius of the rotor, it is possible to find the mathematical approximation of the rotor's inertia using equation 3.5.

$$J_m = 4.8413 \times 10^{-6} \text{ kg} \cdot \text{m}^2$$

### Washer Inertia

Approximating the washer inertia is relatively straight forward as the geometry is simple to measure. As stated in subsection 3.3.2, the only dimensions that need measuring are the inner and outer radii, which are provided in table 3.4.

With the measured radii results and mass properties from table 3.4, the washer inertia can be easily approximated using equation 3.5.

$$J_w = 2.8804 \times 10^{-5} \approx 3.0 \times 10^{-5} \text{ kg} \cdot \text{m}^2$$

### 3.3.3 Lumped Mass Moment of Inertia

$$J_r = \frac{J_m + J_s}{2} \quad (3.6)$$

Having two approximations for the inertia of the rotor, it is now possible to calculate an average between the two. Using the results from subsections 3.3.2 and 3.3.1, and equation 3.6, the overall rotor inertia is found to be:

$$J_r = 4.4627 \times 10^{-6} \approx 4.5 \times 10^{-6} \text{ kg} \cdot \text{m}^2$$

For most of the experiments ran in this thesis, additional inertia is used to slow down the response of the system to allow a better visualization of what is happening when looking directly at the motor as it runs. For the total lumped inertia of the system, three steel washers are being used. Since all inertias in the system are rotating about the same axis, the total lumped inertia of the system is described by equation 3.7.

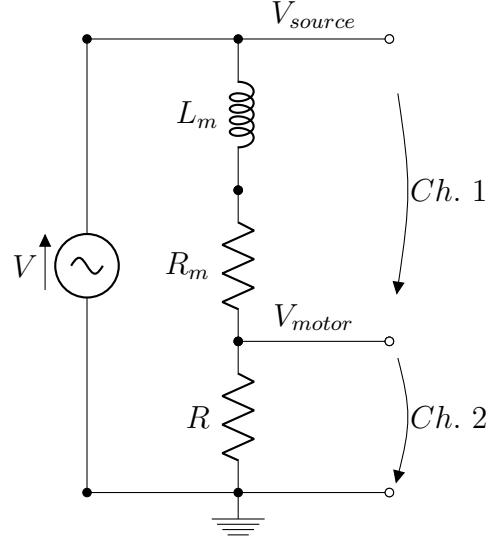
$$J = 3J_w + J_r \quad (3.7)$$

$$J = 9.45 \times 10^{-5} \text{ kg} \cdot \text{m}^2$$

## 3.4 Motor Inductance

The motor inductance is measured by building a simple circuit, which consists of a resistor in series with one of the motor's phase circuitry, as depicted in figure 3.3.

Figure 3.3: Inductance Measurement Circuit



A function generator is required for this experiment to generate a variable frequency sin wave across the external resistor and motor. Knowing that figure 3.3 is a simple voltage divider, it is possible to arrive at equation 3.8 using the relationship  $|V_{motor}| = \frac{1}{2}|V_{source}|$ , where  $f$  is the driving frequency of the sinusoid. Then, the voltages of two outputs are monitored on separate channels on an oscilloscope, and the sinusoidal frequency is increased until the motor output's voltage reaches one half of that of the source voltage.

$$L = \frac{R\sqrt{3}}{2\pi f} \quad (3.8)$$

### 3.4.1 Inductance Estimate

With the outlined procedure developed in the previous section (3.4), three resistors were measured, and the voltages across each junction were tabulated when an appropriate frequency was reached. Table 3.5 hosts the experimental results, along with the calculated average inductance that was found from equation 3.8.

Table 3.5: Inductance Experiment

Resistance ( $\Omega$ )	$V_{source}$ (V)	$V_{motor}$ (V)	Frequency (kHz)	Inductance (H)
22	0.392	0.176	8.34	$7.27 \times 10^{-4}$
98.4	1.04	0.507	34.4	$7.88 \times 10^{-4}$
461	1.88	0.920	167.5	$7.58 \times 10^{-4}$
<b>Average:</b>				$7.58 \times 10^{-4}$

$$L = 7.58 \times 10^{-4} \text{ H}$$

## 3.5 Viscous Friction

Section 3.5 will not detail the process conducted to determine the viscous friction of the NERMLAB, as this will be discussed in chapter 4, where a laboratory has already been developed for students to estimate this coefficient. For the sake of completeness the result will be provided below.

$$b = 5.3 \times 10^{-4} \text{ N} \cdot \text{m} \cdot \text{s}$$

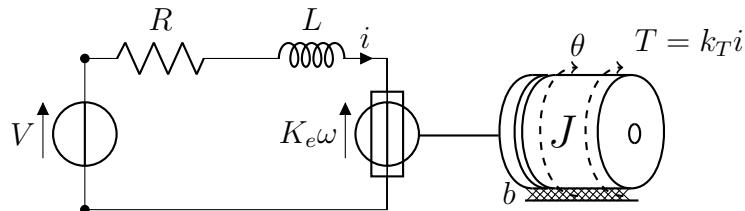
# Chapter 4

## Development of Mathematical Models for NERMLAB

Chapter 4 will develop various mathematical models for the NERMLAB and Motorlab. These models are used throughout this thesis to help compare, analysis, and develop effective control solutions for motor controllers. Rather than have each experiment develop its own model in the corresponding chapter, it will be done here to help simplify the content of each experiment. Since the NERMLAB and Motorlab use different input sources <sup>1</sup>, different mathematical models will be developed for each system seperately.

### 4.1 Electrical Dynamics

Figure 4.1: Electrical and Mechanical Diagram of NERMLAB



<sup>1</sup>Nermlab uses voltage control. Motorlab uses current control.

The electrical dynamics play an important role in motor control when voltage is being used as an input to the system. Using Kirchoff's voltage law it is possible to find the dynamics of figure 4.1.

$$\sum V = 0$$

$$V(t) = Ri(t) + L \frac{di}{dt} + K_e \omega = Ri(t) + K_e \omega$$

$$V(t) = Ri(t) + K_e \omega \quad (4.1)$$

Because the pole at  $-\frac{R}{L}$  ( $6596.3 \frac{\text{rad}}{\text{s}}$ ) is 10 times as large as the other dynamics of the system, it can be ignored in the analysis process. From equation 4.1, it is now possible to derive an equation for the output torque of the system in terms of supplied voltage.

$$T = k_T i(t) \Leftrightarrow i(t) = \frac{T(t)}{k_T} \quad (4.2)$$

Using equations 4.2 and 4.1, equation 4.3 and 4.4 can be derived, respectively.

$$T(t) = \frac{k_T}{R} (V(t) - K_e \omega) \quad (4.3)$$

$$i(t) = \frac{V(t) - K_e \omega}{R} \quad (4.4)$$

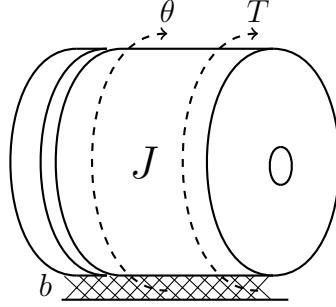
## 4.2 Combined Dynamics - Electrical and Mechanical

Section 4.2 will detail the model development for the various electro-mechanical models that are used throughout this thesis. Subsections 4.2.1 and 4.2.2 will derive mathematical models for position and speed systems, respectively, with different input sources. For the model derivations using current as an input source, the closed-loop current control system

is assumed to be much faster than the mechanical dynamics. As a result of this assumption only the mechanical dynamics and controller will be in the model development.

#### 4.2.1 Position Models

Figure 4.2: Position Model



The best way to start the formulation is to begin with a time domain differential equation of the mechanical system. Because the system is composed of only an angular mass and viscous friction (figure 4.2), a describing differential equation can be written as such.

$$T = k_T i(t) = b\dot{\theta}(t) + J\ddot{\theta}(t) \quad (4.5)$$

Taking the Laplace transform of equation 4.5:

$$k_T I(s) = (bs + Js^2)\theta(s) \quad (4.6)$$

The transfer function can then be developed for  $G_m$  from equation 4.6.

$$\frac{\theta(s)}{I(s)} = \frac{k_T}{Js^2 + bs} \quad (4.7)$$

#### Position Model - Voltage as Input

Equation 4.7 adequately describes the system for the Motorlab because the electrical dynamics are much faster than the mechanical. However in the case of the NERMLAB, voltage

control is used and the same can not be said, and as a result, a different model must be developed. Starting with the differential equations of the electrical dynamics and mechanical dynamics, the following equations are found.

$$V(t) = Ri(t) + L \frac{di}{dt} + K_e \dot{\theta}(t) \quad (4.8)$$

$$T(t) = J\ddot{\theta}(t) + b\dot{\theta}(t) \quad (4.9)$$

Taking the Laplace transform of equations 4.8 and 4.9.

$$V(s) = RI(s) + LS I(s) + K_e s \theta(s) \quad (4.10)$$

$$T(s) = (Js^2 + bs)\theta(s) \quad (4.11)$$

From section 4.1, the relationship between torque and current is known. Substituting equation 4.2 into equation 4.11, an equation for current is found.

$$I(s) = \frac{(Js^2 + bs)}{k_T} \theta(s) \quad (4.12)$$

Substituting equation 4.4 into equation 4.10, grouping and finding a common denominator in the process, yields equation 4.13.

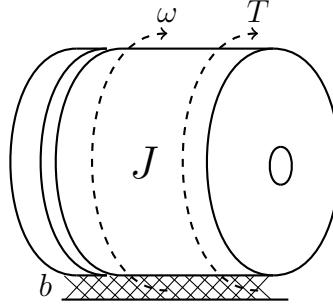
$$\frac{\theta(s)}{V(s)} = \frac{k_T}{(Ls + R)(Js^2 + bs) + K_e k_T s} \quad (4.13)$$

As stated in section 4.1, the electrical dynamics involving  $L$  are much larger than any other dynamics in the system, therefore  $L$  can be ignored, resulting in the final position equation in terms of voltage as an input source.

$$\frac{\theta(s)}{V(s)} = \frac{k_T}{RJs^2 + (Rb + K_e k_T)s} \quad (4.14)$$

## 4.2.2 Speed Models

Figure 4.3: NERMLAB Speed Model



The position models just developed in section 4.2.1 can be used to help derive models for speed. The process is a relatively straight forward one, with one simple substitution needed. Knowing the relationship between speed and position, it is possible to write the following equation.

$$\theta(s) = \frac{\omega(s)}{s} \quad (4.15)$$

Substituting equation 4.15 into 4.7 simply cancels the free integrator, leaving the final equation in terms of speed.

$$\frac{\omega(s)}{I(s)} = \frac{k_T}{Js + b} \quad (4.16)$$

### Speed Model - Voltage as Input

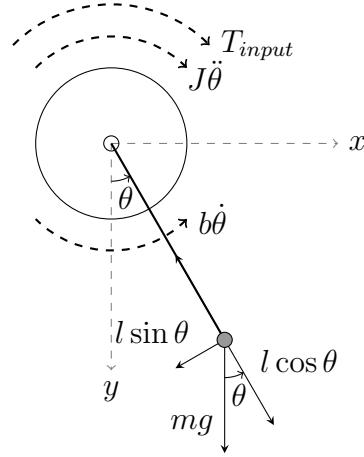
Just as in section 4.2.2, a simple substitution of equation 4.15 into 4.14 yields the final speed model in terms of voltage.

$$\frac{\omega(s)}{V(s)} = \frac{k_T}{RJs + (Rb + K_e k_T)} \quad (4.17)$$

### 4.2.3 Pendulum Model

The pendulum model is a mathematical formulation for NERMLAB that is used to help find the resonant frequency. It should be noted that the dynamic equations for this setup are non-linear in nature, and as a result, the small angle approximation ( $\sin(\theta) \approx \theta$ ) must be used in order to transform the time dependent system into the frequency domain. The small angle approximation is only valid for angles that are smaller than 0.244 radians, as the relative error between  $\sin(\theta)$  and  $\theta$  exceeds 1 percent at this value. This will be taken into account when running experiments on the NERMLAB system.

Figure 4.4: NERMLAB Pendulum Model



The best way to start the derivation is to find the describing differential equations for both the mechanical and electrical system. In the case of the pendulum setup, the electrical dynamics are simply the torque the motor provides the pendulum with, which was found in section 4.1, equation 4.3.

$$J\ddot{\theta}(t) = -mgl\theta(t) - b\dot{\theta}(t) + T_{input} \quad (4.18)$$

Substituting the electrical motor torque (eq. 4.3), into equation 4.18.

$$J\ddot{\theta}(t) = -mgl\theta(t) - b\dot{\theta}(t) + \frac{k_T}{R}(V(t) - K_e\dot{\theta}(t)) \quad (4.19)$$

$$J\ddot{\theta}(t) + mgl\theta(t) + b\dot{\theta}(t) + \frac{k_T K_e}{R}\dot{\theta}(t) = \frac{k_T}{R}V(t)$$

Taking the Laplace transform of the above equation yields:

$$(Js^2 + mgl + bs + \frac{k_T K_e}{R}s)\theta(s) = \frac{k_T}{R}V(s)$$

$$\frac{JRs^2 + mglR + bRs + k_T K_e s}{R}\theta(s) = \frac{k_T}{R}V(s)$$

$$\frac{\theta(s)}{V(s)} = \frac{k_T}{JRs^2 + (bR + k_T K_e)s + mglR} \quad (4.20)$$

Putting equation 4.20 into a standard second order form yields:

$$\frac{\theta(s)}{V(s)} = \frac{\frac{k_T}{JR}}{s^2 + (\frac{b}{J} + \frac{k_T K_e}{JR})s + \frac{mgl}{J}} \quad (4.21)$$

To further simplify the model, lumped coefficients for friction and the trailing term of equation 4.21 will be made since they are comprised of statically defined variables.

$$b_l = \frac{b}{J} + \frac{k_T K_e}{JR} \quad (4.22)$$

$$\omega_p = \frac{mgl}{J} \quad (4.23)$$

$$k_{T,l} = \frac{k_T}{JR} \quad (4.24)$$

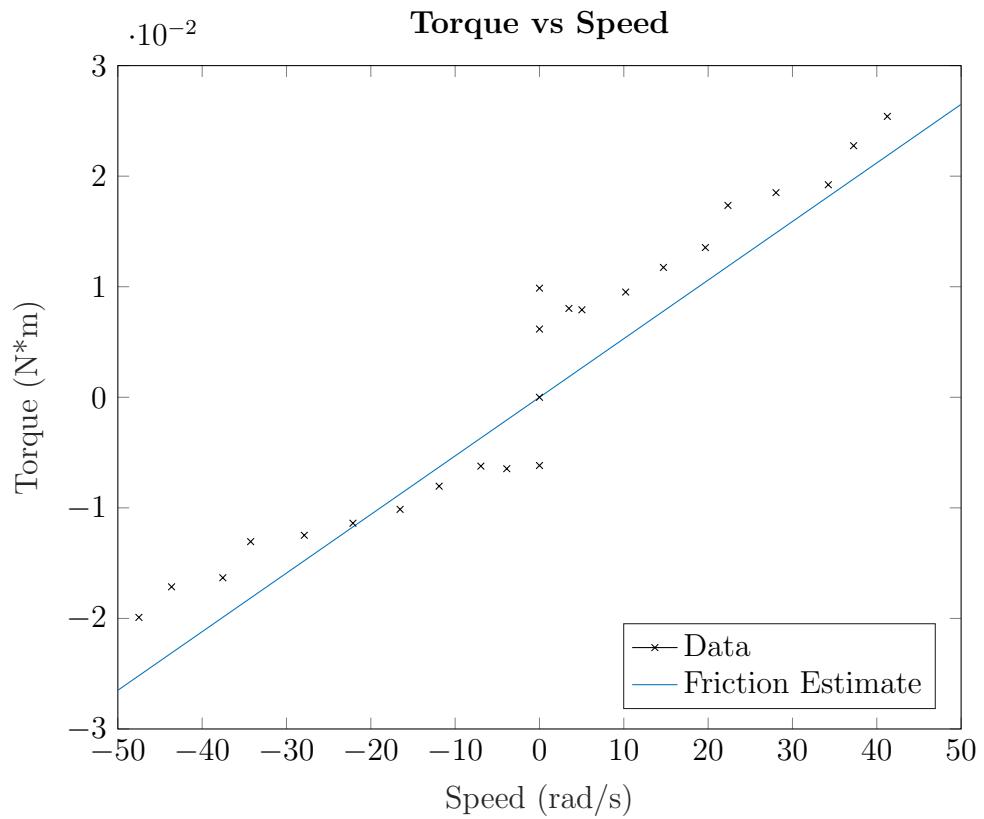
$$\frac{\theta(s)}{V(s)} = \frac{k_{T,l}}{s^2 + b_ls + \omega_p} \quad (4.25)$$

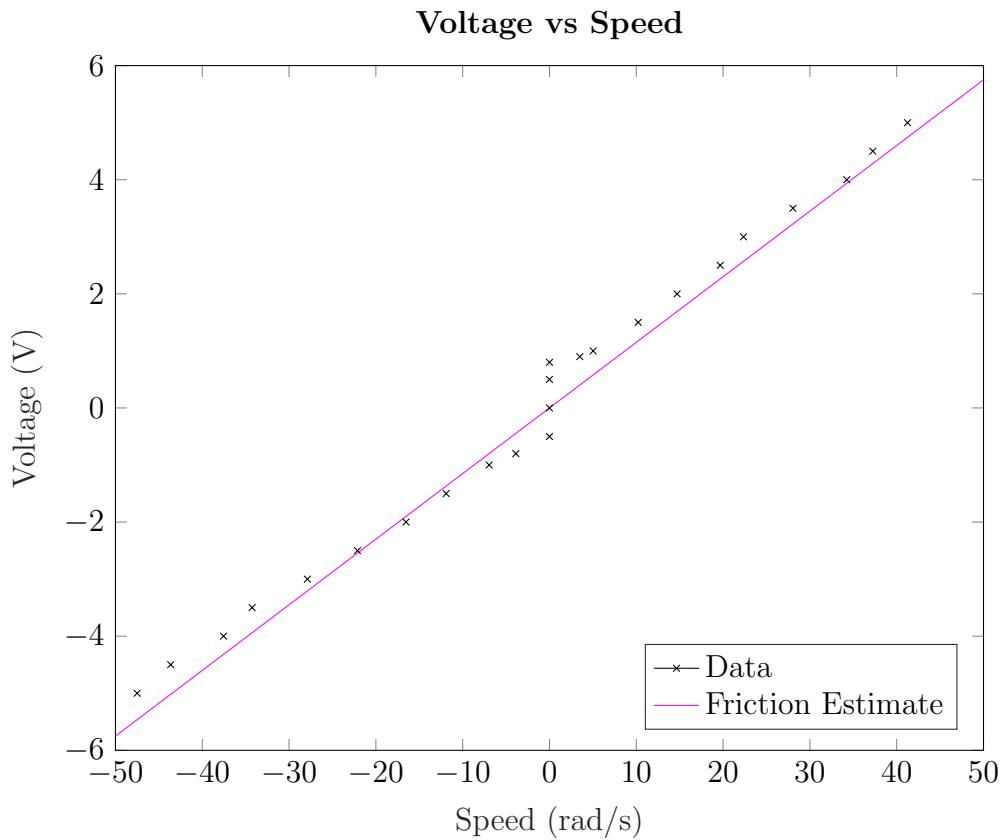
It should be noted that  $J$  is the pendulum's inertia and not the rotors in equation 4.25; likewise,  $l$  is the distance to the center of mass of the pendulum.

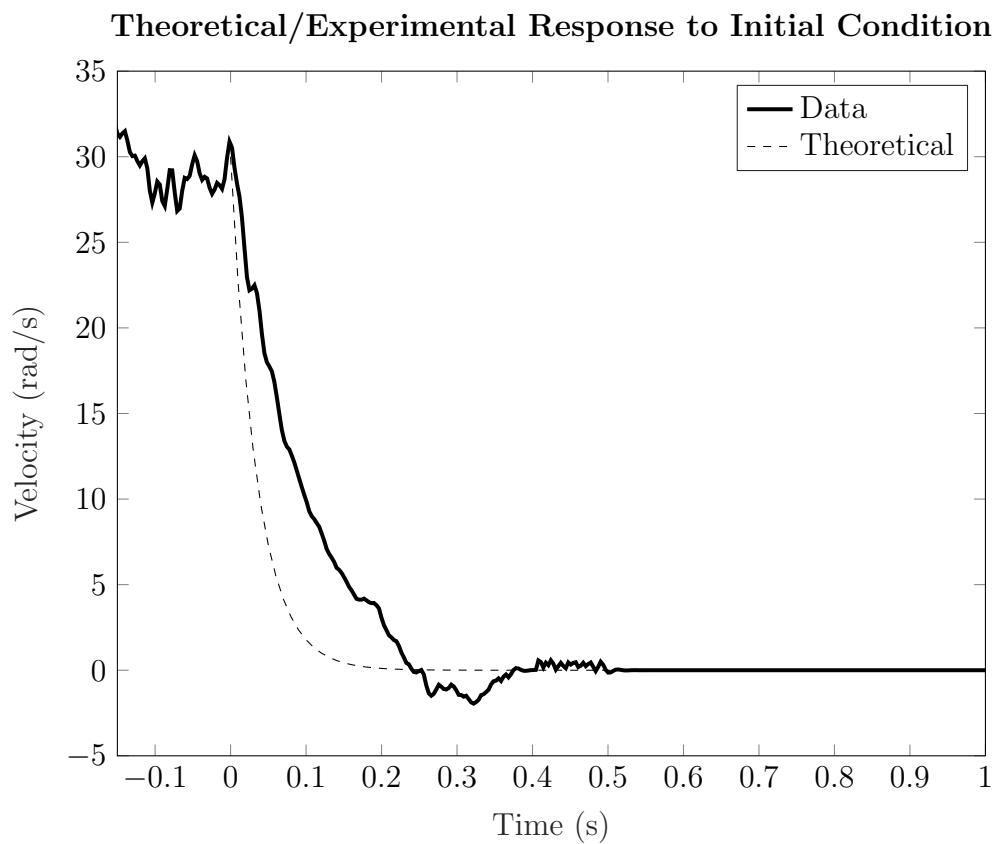
# Chapter 5

## Approximating Friction of a BLDC Motor

Friction is an important characteristic that must be considered in control theory, as it effects precise control of electro-mechanical systems, and as a result, needs inclusion in the mathematical analysis. In motor applications, there are various types of friction that come into play, such as: static, viscous, Stribeck, and Coulombic friction. However, for the sake of model simplicity, Stribeck and Coulombic friction are ignored, with focus geared towards static and viscous friction. A fit will be made between these two types of friction forces which will help develop an overall friction coefficient ( $b$ ) to be used in mathematical formulations and control analysis. Results from this chapter will then be compared against the Motor-lab. Sections 4.1 and 4.2 will cover the model development that will be used in subsequent subsections.







# Chapter 6

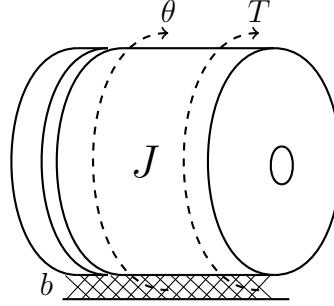
## Experiment One

Chapter 6 will experiment with a position control system on the NERMLAB. In this experiment a model of the closed-loop position control system will be developed and will be used to predict the response of the NERMLAB. The main purpose of this experiment is not to develop a better position control system but rather demonstrate the concept of changing pole locations and the characteristic of different response to a changing proportional gain. This will be done by looking frequency of oscillation and decay rate of the oscillations of the various responses. Results produced by the NERMLAB will then be compared against the Motorlab, which is the basis of comparison of all experiments in this thesis.

### 6.1 Mathematical Model of a Closed-Loop Position Control System

To simply the mathematical formulation the closed-loop current control system is assumed to be much faster than the mechanical dynamics. As a result of this assumption only the mechanical dynamics and controller will be in the model development. The best way to start the formulation is to begin with a time domain differential equation of the NERMLAB system. The NERMLAB is composed of only an angular mass and viscous friction in this experiment which can be seen in figure 6.1.

Figure 6.1: NERMLAB Model



$$T = k_T i(t) = b\dot{\theta}(t) + J\ddot{\theta}(t) \quad (6.1)$$

Taking the Laplace transform of equation 6.1:

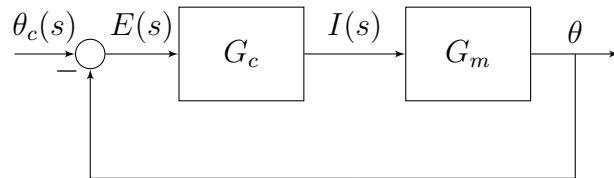
$$k_T I(s) = (bs + Js^2)\theta(s) \quad (6.2)$$

The transfer function can then be developed for  $G_m$  from equation 6.2.

$$\frac{\theta(s)}{I(s)} = \frac{k_T}{Js^2 + bs} \quad (6.3)$$

As for the controller transfer function, a proportional gain  $K_p$  is being used. To further develop the closed loop transfer function, the block diagram in figure 6.2 can be used. It is as simple as doing a block diagram reduction by merging the blocks in series and then performing a feedback calculation.

Figure 6.2: Closed Loop Control System



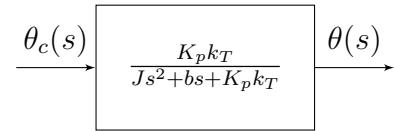
Reducing the blocks in series gives the result:

$$G = G_c G_m = \frac{K_p k_T}{J s^2 + b s} \quad (6.4)$$

The final reduction is performing a feedback calculation:

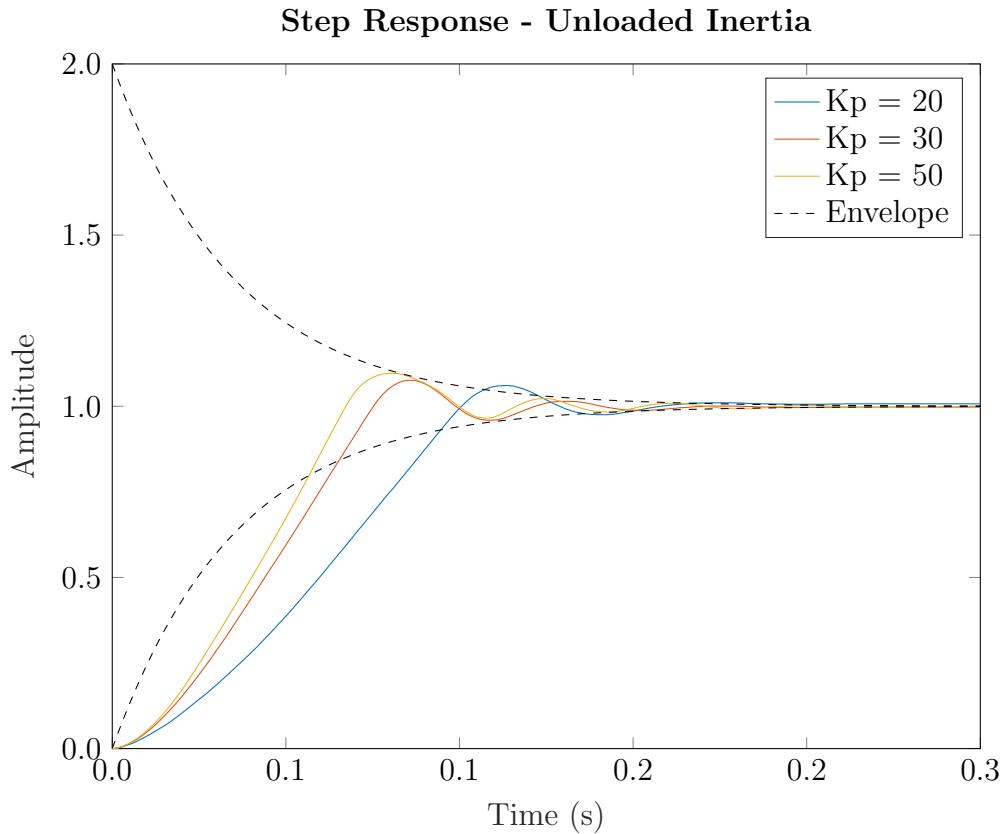
$$\frac{\theta(s)}{\theta_c(s)} = \frac{G}{1 + G} = \frac{K_p k_T}{J s^2 + b s + K_p k_T} \quad (6.5)$$

Figure 6.3: Block Diagram Reduction



## 6.2 Experimental Results

Figure 6.4: Base Load Inertia Step Responses



# Chapter 7

## Experiment Two

Chapter 7 will cover the concept of 'high frequency dynamics'. High frequency dynamics in this context are the faster dynamics in comparison to the mechanical models in the Motorlab and NERMLAB systems. In experiment 2 the higher frequency or faster dynamics are a low pass filter on the output speed from the nominal plant<sup>1</sup>.

Figure 7.1: Closed Loop Speed Control System

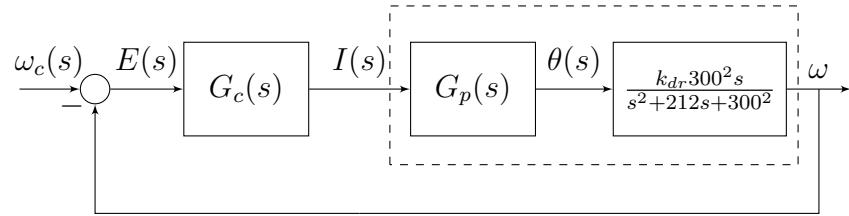
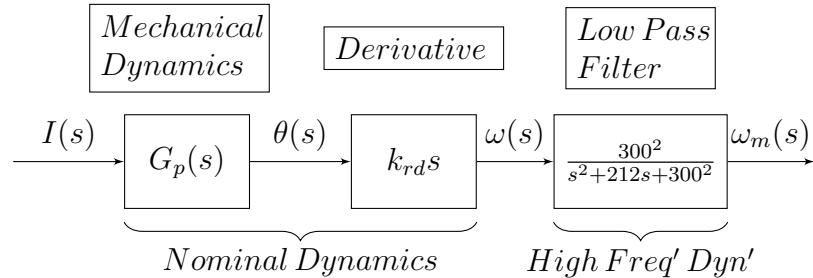


Figure 7.2: Open Loop System from Figure 7.1




---

<sup>1</sup>Nominal plant being the combination of the mechanical dynamics and derivative, as seen in figure 7.2

# Bibliography

- [1] R. M. Reck and R. S. Screenivas, “Developing a new affordable dc motor laboratory kit for an existing undergraduate controls course,” in *American Control Conference (ACC)*, (Chicago, IL), pp. 2801–2806, 2015.
- [2] R. M. Reck, “Experiential learning in control systems laboratories and engineering project management.”
- [3] AMS, *AS5047D 14-Bit On-Axis Magnetic Rotary Position Sensor with 11-Bit Decimal and Binary Incremental Pulse Count*. AMS, April 2016.
- [4] S. R. Smith, “Demonstrating introductory control systems concepts on inexpensive hardware,” Master’s thesis, Kansas State University, 2017.
- [5] J. R. Mevey, “Sensorless field oriented control of brushless permanent magnet synchronous motors,” Master’s thesis, Kansas State University, 2009.

## Appendix A

# Experiment Documentation

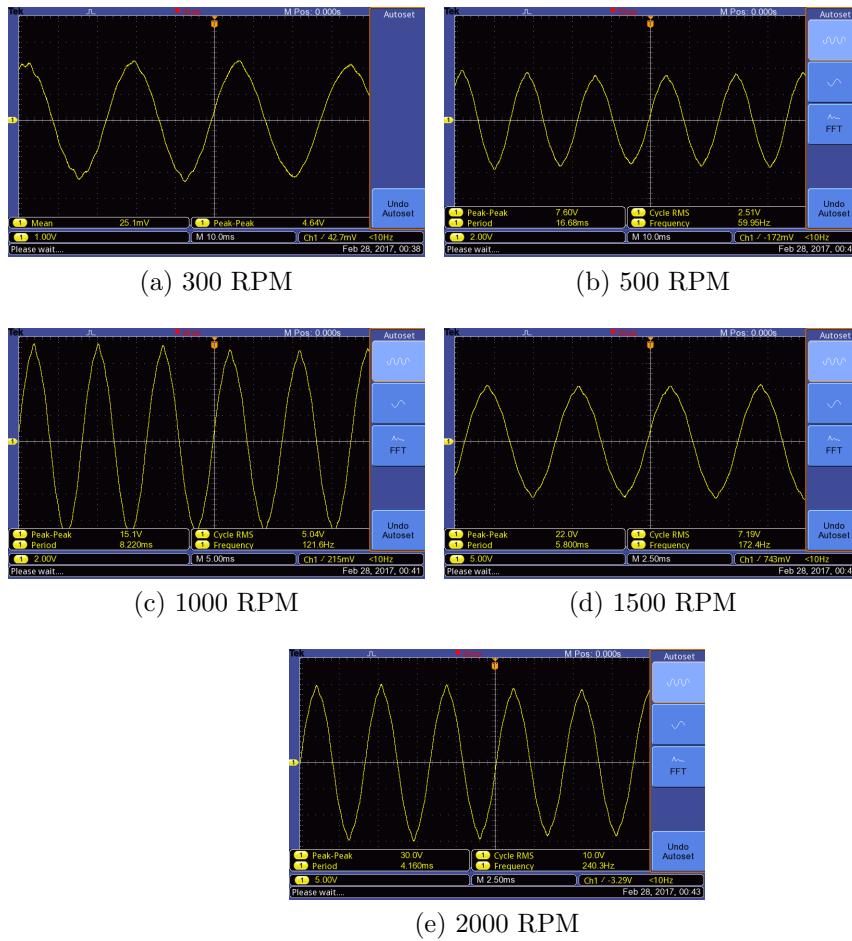


Figure A.1: Back emf at various motor speeds

## **Appendix B**

### **Part Drawings**

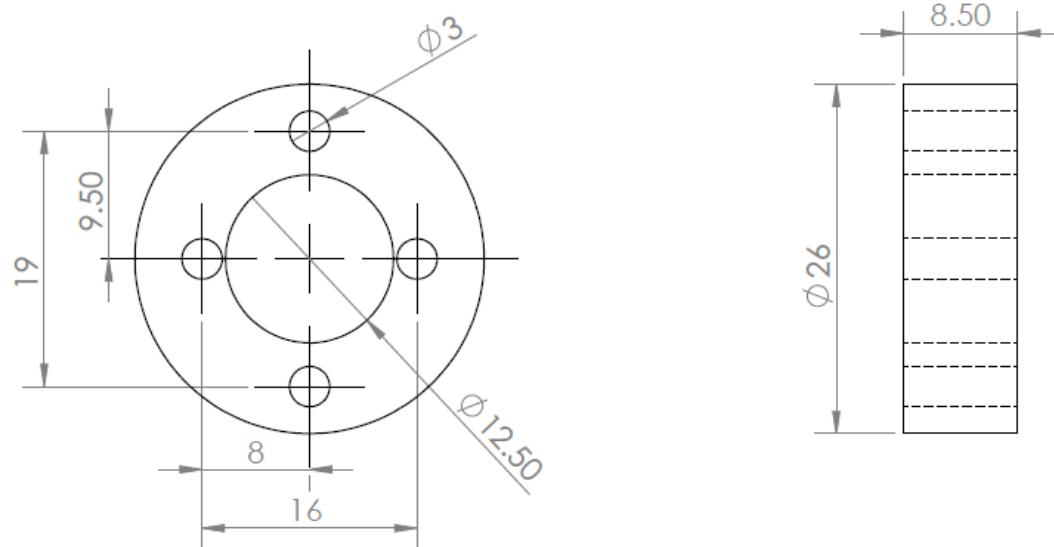
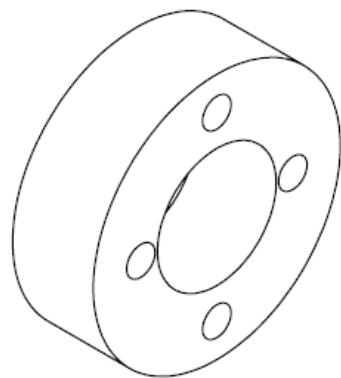


Figure B.1: Standoff for NERMLAB [mm]

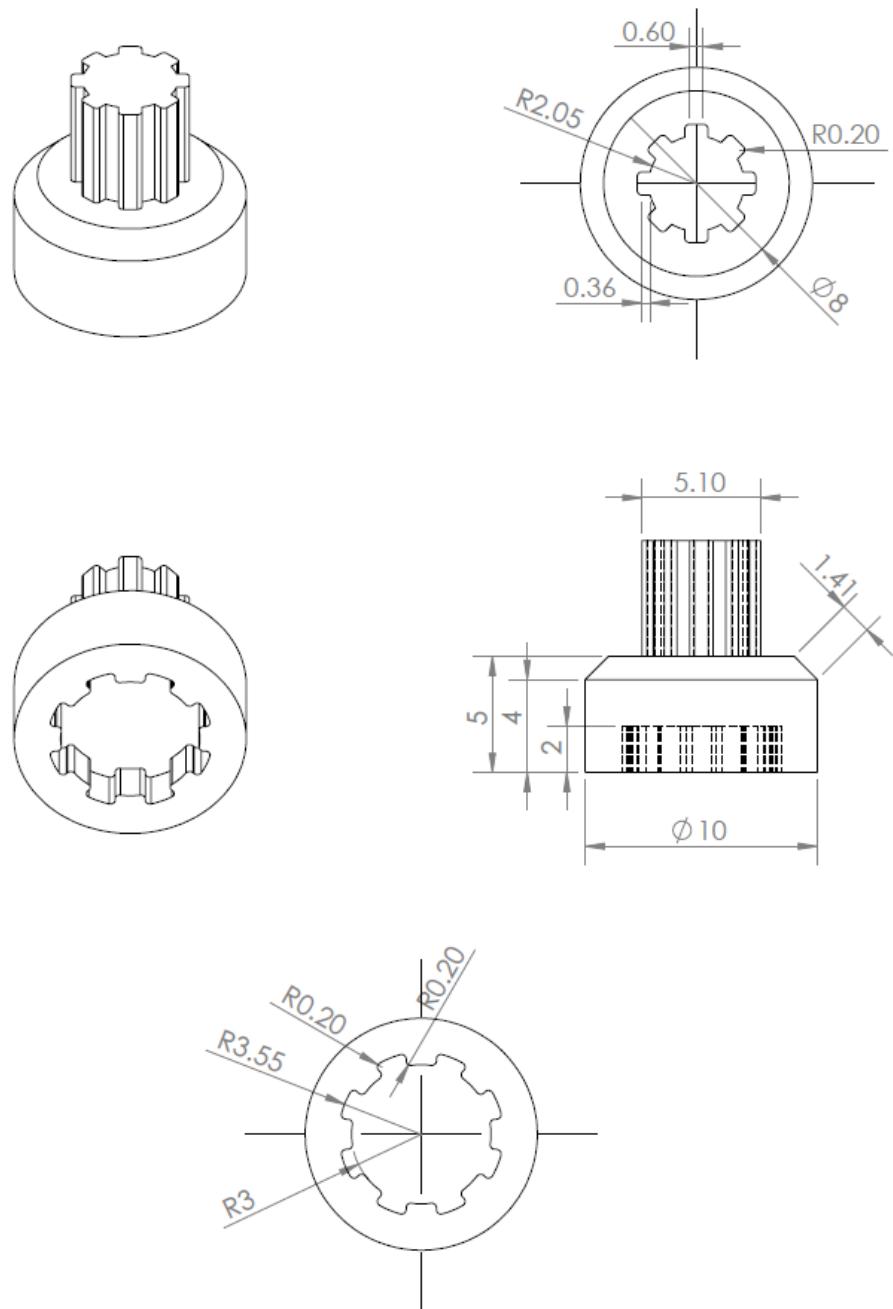


Figure B.2: Magnet holder for NERMLAB [mm]

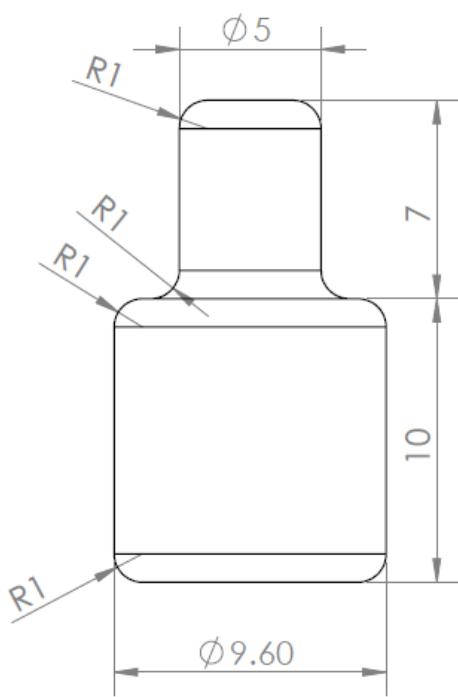
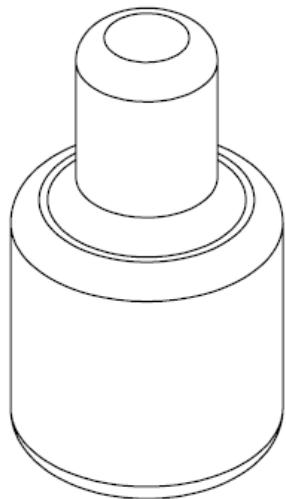


Figure B.3: Torque Transmission Shaft for NERMLAB [mm]

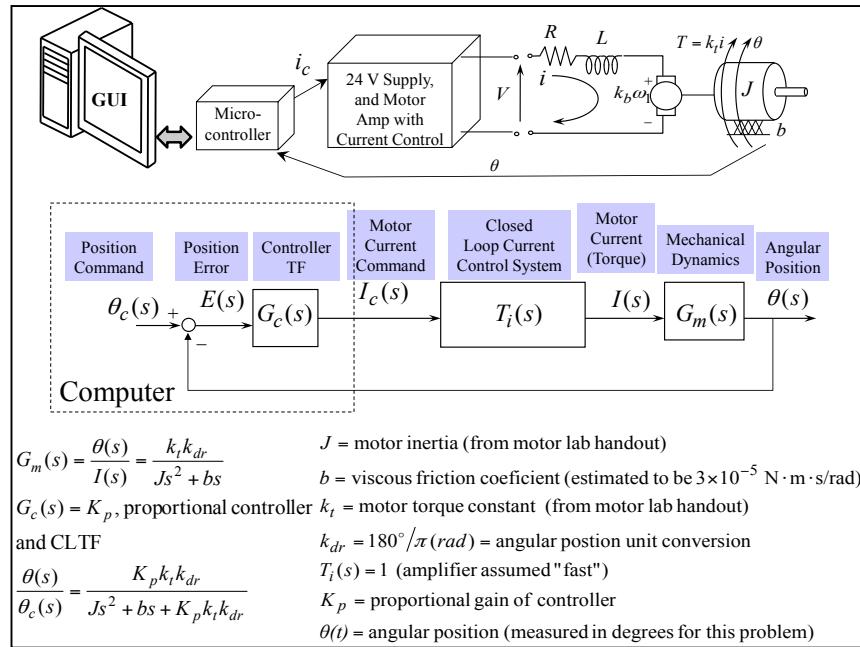
## **Appendix C**

### **Laboratory Procedures**

Appendix C includes the laboratory procedures that students carry out in Control of Mechanical Systems at Kansas State University.

## Laboratory #5

In this lab you are to experiment with the position control system of the "Motorlab" apparatus. Also, you are to use a model of the closed-loop position control system to predict the response. You will compare the theoretical step response with the actual response obtained experimentally from the Motorlab. You will compare the responses for three different proportional controller gains. You should also make connections between pole locations and characteristics of the response such as the frequency of oscillation and the decay rate of the oscillations.



### Work To Be Done Prior To Lab

- Assuming the transfer function of the closed loop current control system,  $T_i(s)$ , is one obtain a symbolic representation of the CLTF  $\theta(s)/\theta_c(s)$ .
- From a) write an equation for the closed-loop poles of the system.
- From b) determine an equation for  $K_p$  where the response of the system becomes oscillatory (i.e. where the poles become complex rather than real).
- Plug in the numbers and determine the value of  $K_p$  for part c).
- Plug the numbers and the following three gains into your equation for part b) to find the oscillation frequency, and time constant for the decay rate of the oscillations, for each gain.  $K_p = 0.01, 0.001, 0.0001 (\text{units?})$

### Obtaining Data From The Motorlab

In this lab you are using a position control system. Therefore you should run the Motorlab control program in position control mode. For this part of the lab you need to collect experimental data for the step response of the closed-loop system for the three different proportional gains given above. You will have to change the gains and you will have to play with the sample frequency and wave frequency to obtain appropriate data that shows the entire step response. Use the following wave magnitudes for the responses and save the data into the matrix name given. Hint: you should have at least 3 seconds of data on the positive portion of the square wave.

PAY ATTENTION TO THE ORDER!

Gain, $K_p$ (units?)	Magnitude of Square Wave (degrees)	Name MATLAB workspace matrix for data.
0.01	200	data3
0.001	1000	data2
0.0001	10000	data1

Immediately after importing the data to the workspace, plot the data using the “mlposplots(data1)” command inside of the MATLAB command window. Check the appropriateness of your sample frequency and wave frequency. Also, use the data cursors to measure the period of oscillation for the table below.

### **Obtaining the required plots and data**

By completing the given m-file code you should generate the required plots for this lab. You should also fill in the table below. Some of the data for this table is generated in the m-file. Other data can be found with the data cursors available in the plots generated with “mlposplots.m”.

Gain, $K_p$ (units?)	Theoretical CLTF poles, $-\zeta\omega_n \pm j\omega_d$ (rad/s)	Theoretical Period of Oscillations, $2\pi/\omega_d$ (seconds)	Measured Period of Oscillations, $T$ (seconds)	Theoretical Time Constant of Envelope, $1/\zeta\omega_n$ (seconds)	Measured Time Constant of Envelope, $\tau$ (estimate one for all three gains) (seconds)
0.01					
0.001					
0.0001					

### **Things to turn in**

- You should have three different plots (with axis labels including units, titles, and legends): 1) simulated unit step response for all three gains, 2) experimental normalized step responses for all three gains, and 3) simulated and experimental response for  $K_p=0.01$ .
- The completed table.
- Hand development of parts a) thru e).
- The answers to the fill in the blanks below (bold and underlined).

### **Fill in the blanks (Turn in by end of lab)**

1. In the theoretical model, as the proportional gain is increased beyond the value where the closed loop response becomes oscillatory, the damped frequency of oscillation \_\_\_\_\_ and the time constant for the envelope of the oscillations \_\_\_\_\_. This captures the behavior of the actual system pretty well, although the envelope does change a little. This might be explained by the nonlinear friction and saturations.
2. As we increase the proportional controller gain beyond 0.001 some aspects of the controller get better while others get much worse. If we try to turn the shaft with our fingers the higher gain system deflects much \_\_\_\_\_ than the lower gain (try it). This indicates \_\_\_\_\_ disturbance rejection. However, the damping of oscillations in the step response becomes much \_\_\_\_\_. This indicates the system is nearly unstable. This is one reason we often add “dynamics” to the controller rather than just the proportional gain which has no integrals or \_\_\_\_\_.
3. If we keep turning the proportional gain up the system actually becomes \_\_\_\_\_ (try it). The theoretical model we used doesn’t predict this. There are always more dynamics out there at higher frequency that we haven’t modeled (we’ll look at some in the next lab). For example, by assuming the current controller in the amplifier had a TF of 1, we assumed that it responds \_\_\_\_\_ fast.
4. Using mlposplots to plot the data in the "data1" matrix we see in the fourth plot, which compares the \_\_\_\_\_ with the \_\_\_\_\_ command, that early in the response the current does not actually track the commanded current. As we simulated in the previous lab real systems sometimes have saturations that can affect the response. Looking at the other plots we can see in plot number \_\_\_\_\_ that the \_\_\_\_\_ seems to saturate during this period, as can be seen by it reaching a high value and staying constant at that value for a short period. We asked our instructor (do this ☺) and they explained that this is actually due to the limited voltage of the power supply and the \_\_\_\_\_ constant of the motor. The motor actually generates a voltage as it spins that is proportional to the \_\_\_\_\_.

### **Starting m-file code**

```
% lab5.m file
% Requires that the square-wave-response data files
% have been imported into data1, data2, and data3.

kt = ???; % N-m/A
J=???; % kg-m^2 or N-m-s^2/rad
b= ???; % N-m-s/rad
kdr=???; % deg/rad

Gm=tf(????);

kp=0.0001;
Gol=kp*Gm;
T1=feedback(Gol,1);
[th1,t1]=step(T1);
[p1,z1]=pzmap(T1)

kp=0.001;
Gol=kp*Gm;
T2=feedback(Gol,1);
[th2,t2]=step(T2);
[p2,z2]=pzmap(T2)

kp=0.01;
Gol=kp*Gm;
T3=feedback(Gol,1);
[th3,t3]=step(T3);
[p3,z3]=pzmap(T3)

dt1=data1(:,1); %extract the time column of the data matrix
dth1=data1(:,3); %extract the first angle column of the data matrix
dth1=dth1/10000; %scale the response to a unit step response

dt2=data2(:,1); %extract the time column of the data matrix
dth2=data2(:,3); %extract the first angle column of the data matrix
dth2=dth2/2000; %scale the response to a unit step response

dt3=data3(:,1); %extract the time column of the data matrix
dth3=data3(:,3); %extract the first angle column of the data matrix
dth3=dth3/200; %scale the response to a unit step response

figure(1); %Theoretical for all three gains
plot(????)

figure(2) %Experimental for all three gains
plot(????)

figure(3) %Experimental and Theoretical for Kp=0.01
plot(????)
```

## Laboratory #6

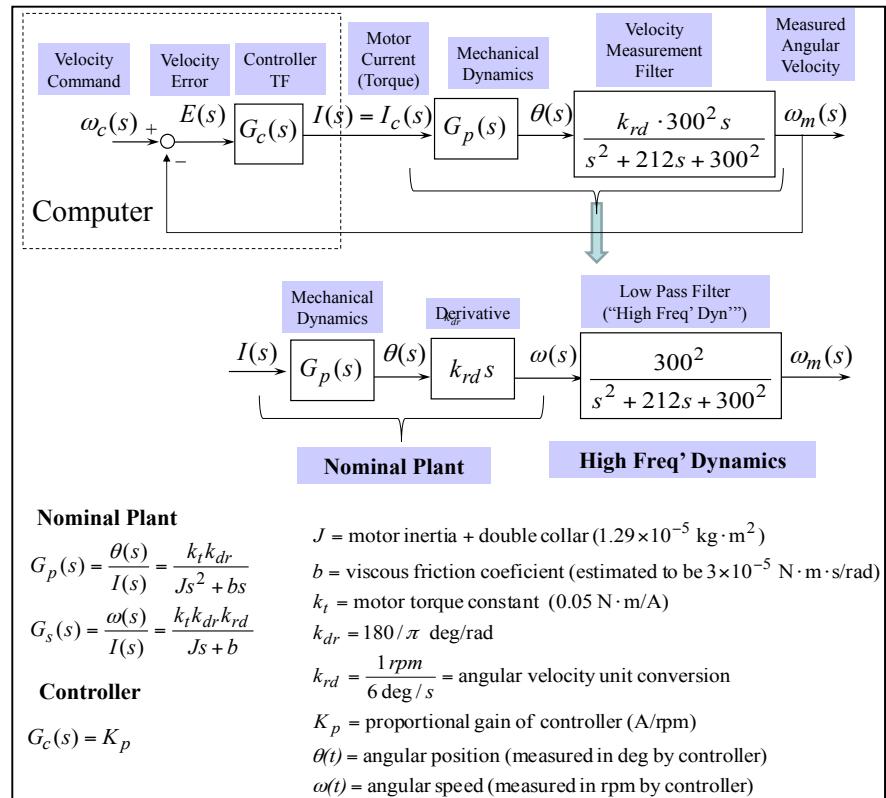
### Introduction

In this lab we will use the velocity control system in the Motorlab to look at the concept of "higher frequency dynamics." This lab should illustrate there are always some higher frequency dynamics that will affect you if you "turn up the gains" too much. We can ignore them to a point, but they are there. Often we do not have a good model for them, or even know for sure what is causing them, but they are there.

We have a rule of thumb: *We can ignore open loop poles and zeros when they are more than 10 times larger (in terms of magnitude, which is the distance from the origin of the s plane) than the closed loop poles that result from ignoring them.* You should note it refers to the effect of open loop poles on the closed loop system. This is typical in control system design. We are usually trying make predictions or calculations for the closed loop system using open loop models.

### Collecting Data

You should collect data for at least three gains, listed in the table below. Use a sample rate of 1000 Hz for all the data. For the first two gains you should collect a step response. For the last gain we want to capture the unstable growth of the response.



Gain, $K_p$ (units?)	Square Wave (rpm)	Name of MATLAB workspace matrix for data
0.0008	1000	data1
0.0016	1000	data2
0.008	Use Special Procedure below	data3

Table 1: Information for Acquiring Data

### Special Procedure:

1. Turn off the amplifier.
2. Set the gain to 0.008.
3. Change the rpm to 50 rpm in the manual command window.
4. Turn on the amplifier. Then wait one second and save the data to the workspace.
5. Use mlspeedplots to plot the data and zoom in on the exponential growth in both speed and current plots.

## Laboratory #6

### Things to Turn In

- Include the two plots the measured and calculated step response when for  $K_p = 0.0008$  and  $K_p = 0.0016$ . These should include separate responses, one measured and the two models of the closed loop system.
- Include a documented copy of your MATLAB code (complete the comments).
- Include the narrative below with the blanks filled in with **bold underlined** answers. Hint: to find the open loop poles of the two systems use “damp (Gs)” and “damp (Gs\*???)” in MATLAB.

The “damp” command in MATLAB prints out the poles of a TF in both Cartesian form and polar form. The Cartesian form gives the real and A parts, while the polar form gives the damping ratio and the magnitude. The angle in polar form is directly related to the B.

The nominal system model is the one without the higher frequency dynamics (i.e. without the low pass filter on speed). The nominal C TF has one pole and the nominal closed loop TF has D pole(s). With the higher frequency dynamics the open loop TF has E poles and the closed loop TF has F poles. Closing the loop G change the number of poles. It changes the H of the poles (i.e. they move in the s-plane).

One of the open loop poles in the system with the higher frequency dynamics has the same value as the nominal open loop pole. This pole is much I in magnitude than the other two poles. The magnitude of the other open loop poles in the model with the high frequency dynamics is J rad/s, which is their natural frequency. Therefore according to our rule of thumb, when the K loop poles of the nominal system approach a magnitude of L rad/s we might expect the accuracy of the nominal model to be questionable.

At a gain of  $K_p=0.0008$  M the magnitude of the closed loop pole of the nominal model is N rad/s and is close to the pole at O rad/s in the closed loop model with the higher frequency dynamics. At this gain we are very close to our rule of thumb, and although there are slight differences, the step responses from the two models and from the actual system all look very similar. The other two closed loop poles in the model with the filter have complex values of P rad/s, which have a magnitude of Q rad/s. This magnitude is much larger than the magnitude of the real pole and therefore these poles R affect the response much.

At a gain of 0.0016 S the magnitude of the closed loop pole of the nominal model is T rad/s, so it is U than the magnitude predicted by our rule of thumb where we will start to see significant differences between the two models. At this gain the real closed loop pole of the higher order model has a value of -68.8 rad/s, so we might expect similar behavior from the two models. However, the other two closed loop poles in the model with the filter have complex values of V rad/s, which have a magnitude of W rad/s. At this gain the magnitude of the real pole and the complex poles are closer together than for the lower gain and we begin to see the effects of the complex poles with some X in the responses of the higher order model and in the actual system.

At a gain of 0.008 Y the closed loop pole of the nominal model is at Z rad/s, which predicts a fast, first order, stable response. However the higher order model and the actual system are AA, as predicted by the positive real parts of the two closed loop poles at BB rad/s. Although the oscillations in the actual response from the Motorlab do not grow to infinity, they do begin grow and then reach a CC cycle after a few oscillations. It can be seen in plot of the motor current that it saturates at DD Amps.

A. \_\_\_\_\_

B. \_\_\_\_\_

C. \_\_\_\_\_

D. \_\_\_\_\_

E. \_\_\_\_\_

F. \_\_\_\_\_

G. does/does not

H. \_\_\_\_\_

I. \_\_\_\_\_

J. \_\_\_\_\_

K. \_\_\_\_\_

L. \_\_\_\_\_

M. (units) \_\_\_\_\_

N. \_\_\_\_\_

O. \_\_\_\_\_

P.    +/-    j

Q. \_\_\_\_\_

R. do/do not

S. (units) \_\_\_\_\_

T. \_\_\_\_\_

U. larger/smaller

V.    +/-    j

W. \_\_\_\_\_

X. \_\_\_\_\_

Y. (units) \_\_\_\_\_

Z. \_\_\_\_\_

AA. \_\_\_\_\_

BB.    +/-    j

CC. \_\_\_\_\_

DD. \_\_\_\_\_

## Laboratory #6

```
% High Frequency Dynamics Lab

kt = 0.05; % N-m/A
J = 1.29e-5; % kg-m^2 or N-m-s^2/rad
b = 3e-5; % N-m-s/rad
kdr = 180/pi; % deg/rad
krd = 1/6; % rpm/(deg/s)

Gs = tf([kt*kdr*krd],[J b]);

wn = ; % Low pass filter in velocity measurement
zwn = ;
Ghf = tf(wn^2,[1 2*zwn wn^2]);

% WHEREVER YOU SEE ???? IN THE COMMENTS YOU NEED TO COMPLETE
kp=[ 0.0008 0.0016 0.008]; % array of kp gains used
for i=1:length(kp) % cycle through the gains
    Tnominal(i) = feedback(kp(i)*Gs,1); % ????? TF for nominal model
    Thf(i) = feedback(kp(i)*Ghf*Gs,1); % CL TF for higher order model
    display(kp(i)); % display the current kp value
    damp(Tnominal(i)) % show the poles in polar form
    damp(Thf(i)) % ""
    [p1,z1]=pzmap(Tnominal(i)); % CL loop poles and zeros of ????? model
    [p2,z2]=pzmap(Thf(i)); % CL loop poles and zeros of ????? model
    p1= sort(p1); % order the poles small to *****
    p2 = sort(p2); % ""
    pnominal(:,i) = p1; % add poles for this gain to list
    phf(:,i) = p2; % ""
end

figure(1); % Plot the closed loop poles for each of the gains
hold on;
for i=1:length(kp)
    plot(real(pnominal(:,i)),imag(pnominal(:,i)), '+'); % Nominal model
    plot(real(phf(:,i)),imag(phf(:,i)), 'x'); % Higher order model
end
hold off;
axis equal;
s=sprintf('Poles of nominal CL system, + \n');
s=[s sprintf('and higher order CL system, x \n')];
s=[s sprintf('for three gains \n')];
title(s);
axis equal;

tfinal = .3;
[speedN,timeN]=step(1000*Tnominal(1), tfinal); % step response of nominal model
[speedHF,timeHF]=step(1000*Thf(1), tfinal); % "" of higher order model
speed= data1(:,5); %extract the first speed column of the data matrix
time=data1(:,1); %extract the time column of the data matrix

figure(2);
plot(timeN,speedN,timeHF,speedHF,time,speed); % step response plot kp=0.0008
title('Plot of CL step response for Kp=0.0008');
legend('model with nominal dynamics','model with hi freq dynamics','actual');
xlabel('time (sec)'); ylabel('speed (rpm)');

tfinal = .15;
[speedN,timeN]=step(1000*Tnominal(2), tfinal); % step response of nominal model
[speedHF,timeHF]=step(1000*Thf(2), tfinal); % "" of higher order model
speed= data2(:,5); %extract the first speed column of the data matrix
time=data2(:,1); %extract the time column of the data matrix

figure(3);
plot(timeN,speedN,timeHF,speedHF,time,speed); % step response plot kp=0.0016
title('Plot of CL step response for Kp=0.0016');
legend('model with nominal dynamics','model with hi freq dynamics','actual');
xlabel('time (sec)'); ylabel('speed (rpm)');
```

## **Appendix D**

## **Code**

Figure D.1: Encoder Noise Experiment

MATLAB

```
%{
AUTHOR: Derek Black
TOPIC: Noise Characterization
DESCRIPTION: Show that sensor noise frequency is proportional to rotor
speed
%}

%% Extract data
% Time data
time1 = noise1(:,1);
time2 = noise2(:,1);
time3 = noise3(:,1);
time4 = noise4(:,1);
time5 = noise5(:,1);
time6 = noise6(:,1);

% Speed data - Scaled
speed1 = noise1(:,5)/mean(noise1(:,5));
speed2 = noise2(:,5)/mean(noise2(:,5));
speed3 = noise3(:,5)/mean(noise3(:,5));
speed4 = noise4(:,5)/mean(noise4(:,5));
speed5 = noise5(:,5)/mean(noise5(:,5));
speed6 = noise6(:,5)/mean(noise6(:,5));

% Calculated Frequency
f = [4.386 5.55 8.1967 10.8696 13.51 18.5185];
% Speed
s = [mean(noise1(:,5)) mean(noise2(:,5)) mean(noise3(:,5))...
      mean(noise4(:,5)) mean(noise5(:,5)) mean(noise6(:,5))];

% Fit data
x = [0 70];
b = inv(s*s')*s*f';
y = b*x;

%% Comparison figure
figure(1)
subplot(2,1,1)
plot(time1,speed1,'k—')
hold on;
plot(time3,speed3,'k:', 'linewidth', 1.2);
hold on;
plot(time6,speed6,'k');
xlim([0 0.5]);
ylim([0.9 1.14]);
xlabel('Time_(Sec)');
ylabel('Normalized_Speed_(rad/s)');
title('Sensor_Noise_Frequency');
legend('12_rad/s','32_rad/s','63_rad/s');
grid on;

subplot(2,1,2)
plot(x,y,'k—',s,f,'x', 'MarkerSize', 5.3);
xlabel('Frequency_(Hz)');
ylabel('Average_Speed_(rad/s)');
legend('Proportionality','Data','Location','southeast');
grid on;
```

Figure D.2: Back-EMF Plotter

Python 2.7

```
,,  
Derek Black 2017 (c)  
Kansas State University  
  
This file plots data collected for peak-peak voltages of a brushless DC motor.  
This plot is used to estimate the back-emf constant ( $K_e$ ).  
,,  
  
# Imports  
import matplotlib.pyplot as plt  
import math  
import numpy as np  
from numpy.linalg import inv  
  
# Collected Data  
Speed_RPM = [0.0, 300.0, 500.0, 1000.0, 1500.0, 2000.0]      # RPM  
Voltage = [0.0, 4.64, 7.60, 15.1, 22.0, 30.0]                # Volts (Peak-Peak)  
  
# Convert Speed to Appropriate Units  
Speed_RAD = [(2.0*math.pi/60.0)*x for x in Speed_RPM]        # RPM -> Rad/s  
Voltage = [(x/2.0) for x in Voltage]  
  
# Data to Numpy Matricies  
Speed = np.matrix(Speed_RAD)  
Voltage = np.matrix(Voltage)  
  
# Find Line of Best Fit - Least-Squares Normal Equation  
b = inv(Speed*np.transpose(Speed))*Speed*np.transpose(Voltage)  
flattened = [val for sublist in b.tolist() for val in sublist]  
b = flattened[0]  
print b  
print Speed_RAD  
  
# Create plottable data to show best fit  
y = [b*x for x in Speed_RAD]  
  
# Plot Data  
plt.figure(1)  
plt.scatter(Speed, Voltage, color='black', marker='x', linewidth=2, label='Measured_Back_EMF')  
plt.plot(Speed_RAD, y, '--', color='black', label='Line_of_Best_Fit')  
plt.grid(True)  
plt.xlabel('Motor_Speed_(rad/s)')  
plt.ylabel('Measured_Voltage_(V)')  
plt.legend(loc='lower_right')  
plt.title('Back_EMF_Constant')  
plt.show()
```

# Appendix E

## System Specifications

Table E.1: NERMLAB Parameters

Parameter	Description	Value	Units
$J$	Lumped Inertia $(3J_w + J_r)$	$9.45 \times 10^{-5}$	$kg \cdot m^2$
$J_w$	Washer Inertia	$3.0 \times 10^{-5}$	$kg \cdot m^2$
$J_r$	Rotor Inertia	$4.5 \times 10^{-6}$	$kg \cdot m^2$
$b$	Viscous Friction	$5.3 \times 10^{-4}$	$\frac{N \cdot m \cdot s}{rad}$
$k_T$	Motor Torque Constant	0.06	$\frac{N \cdot m}{A}$
$L$	Motor Inductance	$7.58 \times 10^{-4}$	$H$
$R$	Motor Phase Resistance	5	$\Omega$
$K_e$	Back-EMF Constant	0.07	$\frac{V \cdot s}{rad}$

Table E.2: Motorlab Parameters

Parameter	Description	Value	Units
Inertia	Single Shaft Collar	15	$g \cdot cm^2$
Inertia	Double Shaft Collar	19	$g \cdot cm^2$
Inertia	Rotor	110	$g \cdot cm^2$
$b$	Viscous Friction	$3 \times 10^{-5}$	$\frac{N \cdot m \cdot s}{rad}$
$k_T$	Motor Torque Constant	5	$\frac{N \cdot cm}{A}$
$L$	Motor Inductance	4.4	$mH$
$R$	Motor Phase Resistance	1.18	$\Omega$
$K_e$	Back-EMF Constant	5.2	$\frac{V}{krpm}$