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

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

List of Tables

Acronyms

ARM Advanced RISC Machine

DAEC Dynamic Angle Error Compensation

MPU Microprocessor Unit

BLDC Brushless DC

GUI Graphical User Interface

NERMLAB New Earth Robotics Motor Lab

back-emf back electromotive force

RPM Rotations Per Minute

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 added to lecture components in universities [1]. This thesis will address the issues, like budget of laboratory hardware, class sizes, and space limitations, and try to assess the feasibility of utilizing low budget, smaller, and portable laboratory hardware for introductory control classes. The introduction of this thesis will be broken up into sections to address these issues individually. It's important to note, that while this thesis does recognize the importance of having laboratory components to control theory lectures, it will not be the main focus, rather importance will be given to addressing the feasibility of low budget portable hardware, more specifically the Motorlab, for control applications. This chapter serves more as a background to why lower budget, portable hardware is important to control classes, and in turn demonstrates a need for these low budget devices.

1.1 Hardware and Software Budget

1.2 Space Limitations

Chapter 2

Apparatus

Two pieces of apparatus were used to conduct the experiments in this thesis. This chapter will detail the purpose, design and recreation of the equipment. Section 2.1 will cover the new Motorlab, including the hardware implementation, design of components, and basic functionality. Section 2.1.3 will detail how a new type of position sensor works that is used for the position measurements of the Motorlab. Then, the older Motorlab will be discussed and compared to the new Motorlab in section 2.2.

2.1 New Motorlab

The new Motorlab is a reimplementation of older laboratory hardware created by Dr. Schinstock and Dr. White for Control of Mechanical Systems I at Kansas State University. The Motorlab allows users to connect the theoretical ideas of control theory with those in practice. (Maybe include applications of the motorlab and its use in the laboratory).

2.1.1 Hardware

The new Motorlab consists of several key pieces of hardware, namely a Microprocessor Unit (MPU), motor driver, and a Brushless DC (BLDC) motor. The main MPU of the Motorlab is the STM32 Nucleo, which allows Arduino attachment shields and other STM boards to be attached for added functionality. For the purposes of the Motorlab, a motor driver was required to drive a brushless DC motor, namely a RCTIMER GBM2804. An X-Nucleo-IHM07M1 (a three-phase brushless DC motor driver) was selected to be the primary driver for the Motorlab.

2.1.2 Position Sensor

The main purpose of the Motorlab is to conduct control laboratory experiments, as a result, feedback via sensor readings is necessary to do such control. The typical way to do position and speed control of mechanical systems and motors 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 Motorlab, 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 on the Motorlab consists of 14-bit on-axis magnetic rotary position sensor chip, specifically the AS5047D by AMS [1]. The position sensor chip provides high resolution absolute angle measurements through a full 360 degree range [2]. 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 [2].

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 2.1), angle measurements are calculate and

[1] AMS is an Austrian analog sensor and semi-conductor manufacturer

[2] These chips typically provide a maximum resolution of 2000 steps/revolution in decimal mode and 2048 steps/revolution in binary mode

transmitted through the chip [2]. The Motorlab uses the AS5047D chip primarily as a position and speed control system.

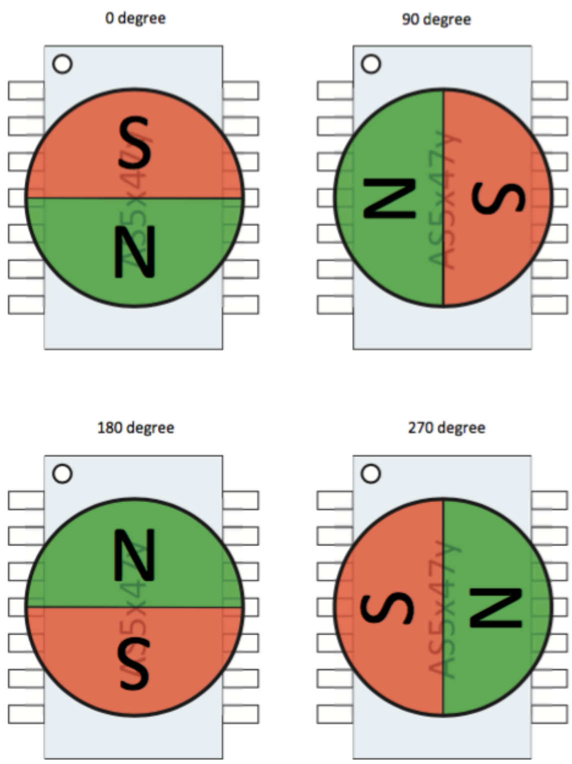


Figure 2.1: Magnet and AS5047D [2]

2.1.3 Motorlab Parts

Along with the hardware mentioned in section 2.1.1, three components were needed to be developed in order to bring the Motorlab 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 [3]. Both the spacer and magnet holder which can be seen in figure 2.2 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

[3]Diametrically opposed meaning the north and south poles of the magnet are in-plane as opposed to top/bottom poles. Reference figure 2.1 for further clarification

with the two 3D printed parts, a printed circuit board had to be designed by Eric Patterson of Kansas State University 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 [4]. 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.

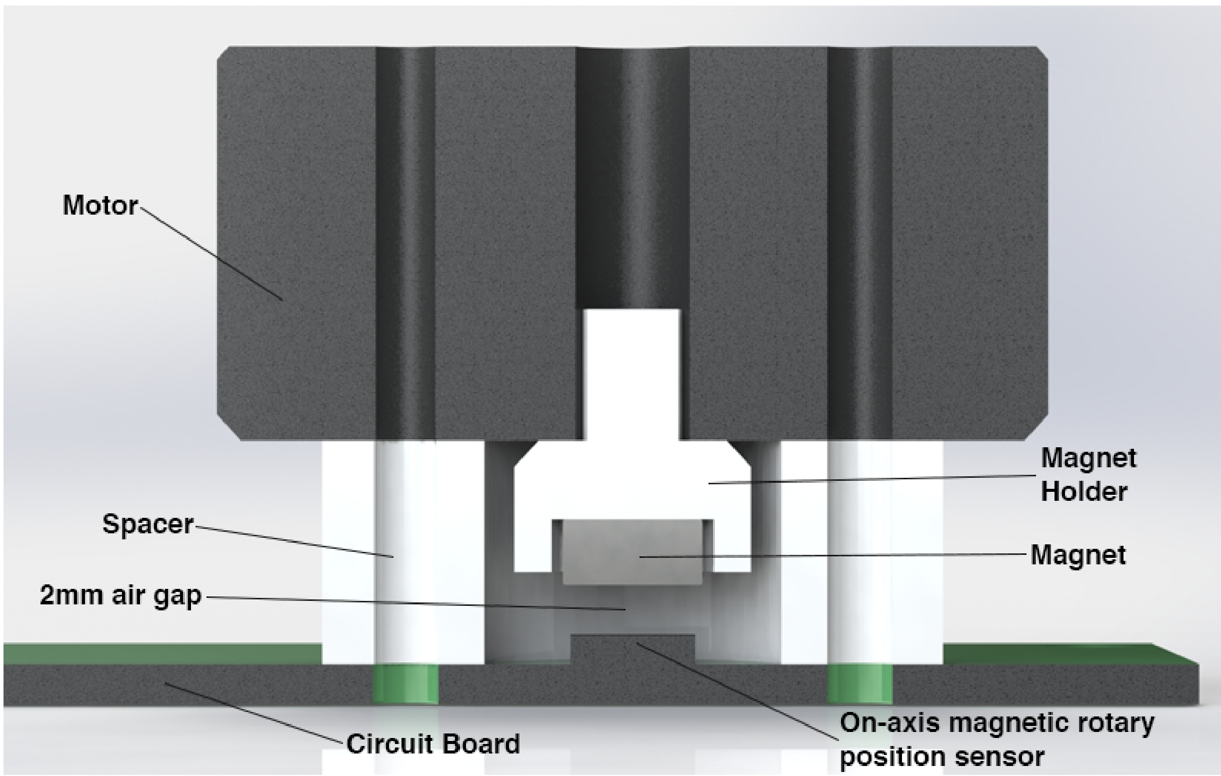


Figure 2.2: Section View of Motorlab Assembly

2.1.4 Motorlab Cost

Text here.

[4]Because of this variability in resolution, the magnet holder was printed in iterations, varying the diameter of the spline to insure a tight fit in the motor shaft

Table 2.1: 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
	TOTAL COST	**69.07 USD**

2.1.5 Motorlab GUI

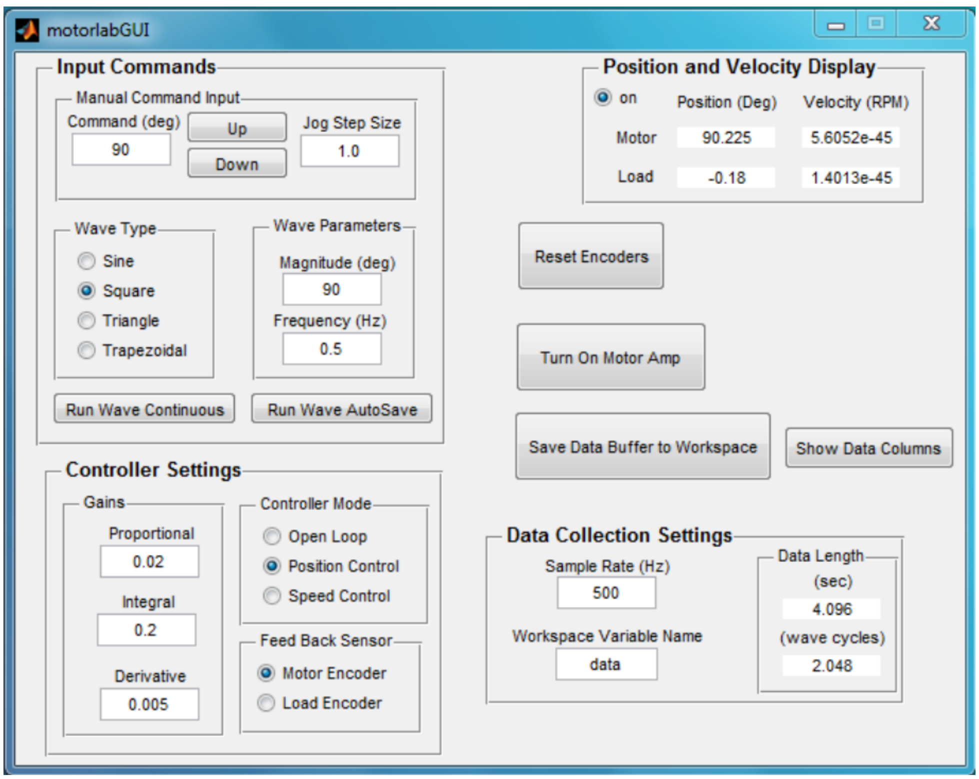


Figure 2.3: Motorlab GUI in MATLAB

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 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 2.3.

2.2 Motorlab

The Motorlab is a piece of laboratory equipment developed by Dr. Dale Schinstock and Dr. Warren N. White at Kansas State University. It has been in service at the university for over 15 years, and as a result is a time-tested piece of laboratory hardware that has proven to be reliable in terms of providing quality practical control theory application to students enrolled in the class Control of Mechanical Systems I. Because the Motorlab's hardware components were designed and created to insure a very clear translation of laboratory results, in addition to the Motorlab having a much larger operating limit and bandwidth than students use in laboratory practice, this represents to a good base model to compare the results of the apparatus in this thesis too.

Various hardware make up the Motorlab, namely, a high quality BLDC motor, BLDC servo amplifier by Copley Controls Corp., and a ST Discovery board [5]. Typically the Motorlabs run a cost of about 700 USD per lab station [3].

[5]STMicroelectronics is a Switzerland based micro-controller manufacturer

Chapter 3

Model Development

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 the table 3.1.

Table 3.1: Motor parameters

Parameter	**Description**
V	Motor Voltage
k_t	Motor Torque Constant per Phase
k_T	Overall Motor Torque Constant
K_e	Back Electromotive Force Constant per Phase
$K_{e,LL}$	Line-Line Back Electromotive Force Constant
J	Mass Moment of Inertia
L	Motor Inductance
R	Motor Phase Resistance
R_{LL}	Motor Line-Line Resistance
τ	Time Constant
T	Motor Torque
ω_m	Motor Speed

3.1 Motor Resistance

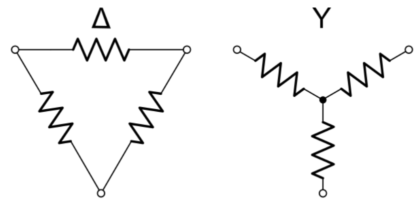


Figure 3.1: Motor Connection Configuration

BLDC motors are typically connected in two wiring configurations, WYE (Y) or delta (Δ) 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).

$$R = \frac{R_{LL}}{2} \tag{3.1}$$

3.1.1 Procedure and Results

In order to gather a good estimate for the phase resistance of the RCTIMER GBM2804 motor, the resistance was measured line-line across all 3 phases. Each set of motor leads were hooked up to a digital multimeter and the values were tabulated for each phase component in table 3.3. An average was then calculated between the different line-line resistances to get the overall resistance of the motor.

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

$$\boxed{R = 4.955 \approx 5\Omega}$$

Table 3.2: Measured motor resistance

A-B	**A-C**	**B-C**
9.870 Ω	9.900 Ω	9.950 Ω
	Average:	9.91 Ω

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). K_e 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, $K_{e,LL}$. 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. [4].

$$k_T = \frac{\sqrt{3}}{2} K_{e,LL} \tag{3.2}$$

$$k_T = K_{e,LL} \tag{3.3}$$

Because $K_{e,LL}$ 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 $K_{e,LL}$. With equation 3.2 or 3.3, k_T can then be determined.

3.2.1 Procedure and Results

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[1] 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[2]. The data collected is tabulated in table 3.3.

Table 3.3: Measured back voltage

Speed (RPM)	**Speed ω_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 $K_{e,LL}$ 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 $\boldsymbol{V}$ and $\boldsymbol{\omega_m}$.

$$K_{e,LL} = (\boldsymbol{\omega_m}\boldsymbol{\omega_m^T})^{-1}\boldsymbol{\omega_m}\boldsymbol{V^T} \tag{3.4}$$

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

$$K_{e,LL} = 0.0713 \quad \frac{V \cdot s}{rad}$$

To verify that $K_{e,LL}$ was the best fit to that data, $K_{e,LL}$ was plotted against the collected data in figure 3.2.

[1] Appendix B figure B.3
[2] Appendix A figure A.1

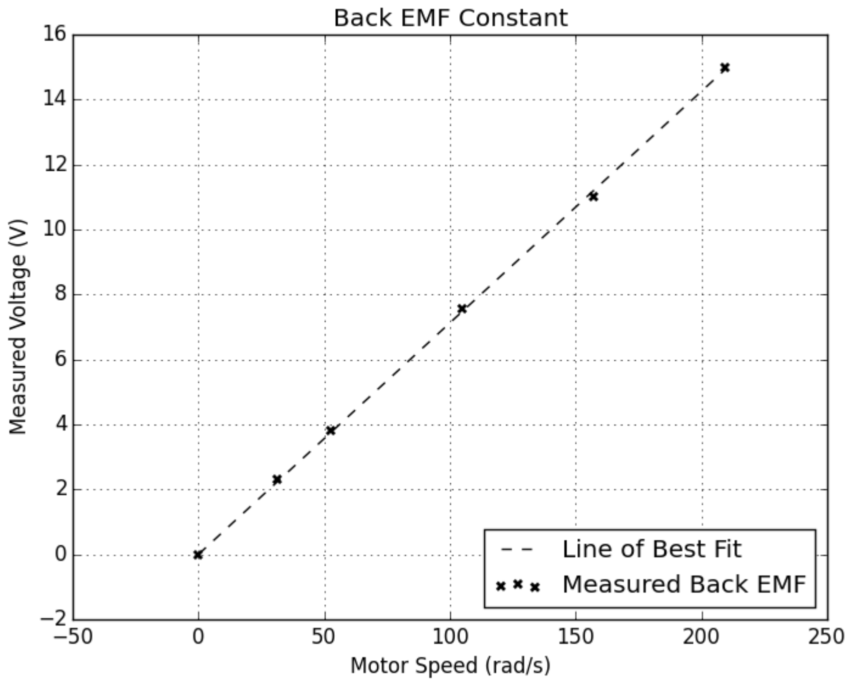


Figure 3.2: Measured Back EMF vs Speed

Since the relationship between k_T and $K_{e,LL}$ is known by equation 3.2 and 3.3, k_T can now be calculated.

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

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

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 some mass from an axis of rotation.

The angular mass of the NERMLAB will be determined in two ways: experimentally determining J through software modeling, and approximating J through mathematical formulation. For both setups the mass of the rotating inertia had to be measured.

3.3.1 Software Modeling of Mass Moment of Inertia

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:

$$J_z = \frac{m}{2}(r_1^2 + r_2^2) = mr_2^2(1 - t + \frac{t^2}{2})$$

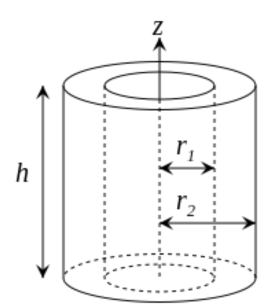


Figure 3.3: Thick Walled Cylinder (J)

3.4 Motor Inductance

$$L = R\tau$$

Chapter 4

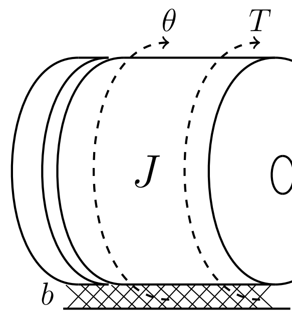
Experiment One

Chapter 4 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.

4.1 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 4.1.

Figure 4.1: NERMLAB Model



$$T = k_T i(t) = b\dot{\theta}(t) + J\ddot{\theta}(t) \tag{4.1}$$

Taking the Laplace transform of equation 4.1:

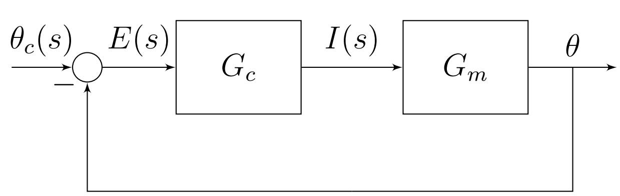
$$k_T I(s) = (bs + Js^2)\theta(s) \tag{4.2}$$

The transfer function can then be developed for G_m from equation 4.2.

$$\frac{\theta(s)}{I(s)} = \frac{k_T}{Js^2 + bs} \tag{4.3}$$

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 4.2 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 4.2: Closed Loop Control System



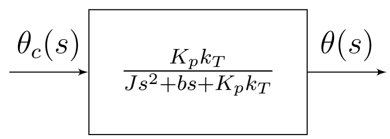
Reducing the blocks in series gives the result:

$$G = G_c G_m = \frac{K_p k_T}{Js^2 + bs} \tag{4.4}$$

The final reduction is performing a feedback calculation:

$$\frac{\theta(s)}{\theta_c(s)} = \frac{G}{1+G} = \frac{K_p k_T}{Js^2 + bs + K_p k_T} \tag{4.5}$$

Figure 4.3: Block Diagram Reduction



Chapter 5

Experiment Two

Chapter 5 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[1].

Figure 5.1: Closed Loop Speed Control System

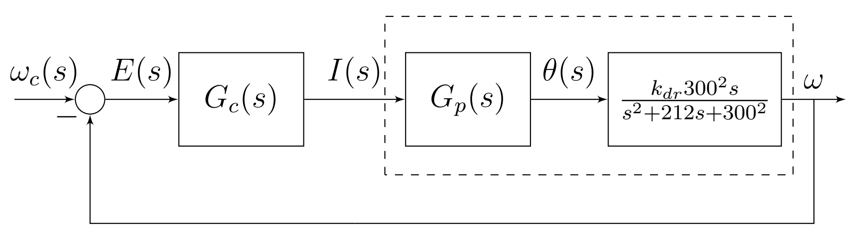
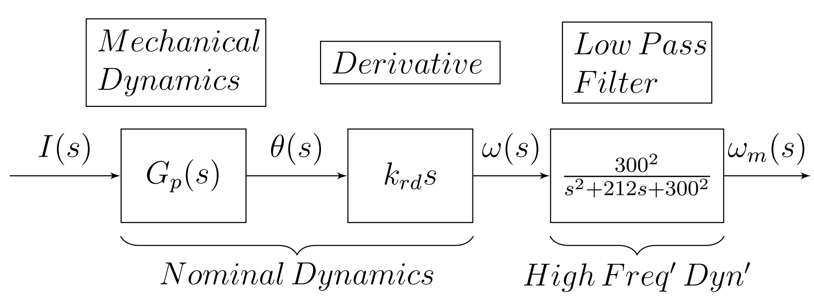


Figure 5.2: Open Loop System from Figure 5.1



[1]Nominal plant being the combination of the mechanical dynamics and derivative, as seen in figure 5.2

Chapter 6

Experiment Three

Chapter 7

Experiment Four

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.

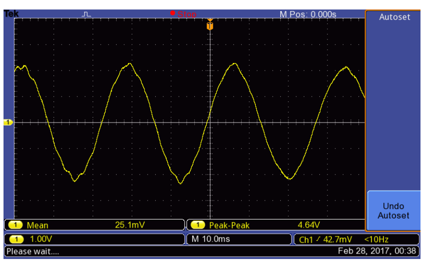
[2] AMS, *AS5047D 14-Bit On-Axis Magnetic Rotary Position Sensor with 11-Bit Decimal and Binary Incremental Pulse Count*. AMS, April 2016.

[3] S. R. Smith, "Demonstrating introductory control systems concepts on inexpensive hardware," Master's thesis, Kansas State University, 2017.

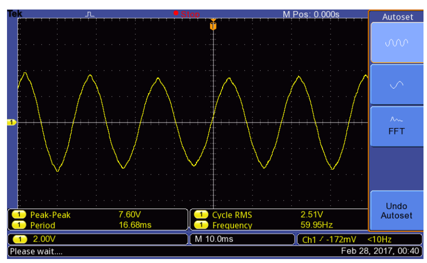
[4] J. R. Mevey, "Sensorless field oriented control of brushless permanent magnet synchronous motors," Master's thesis, Kansas State University, 2009.

Appendix A

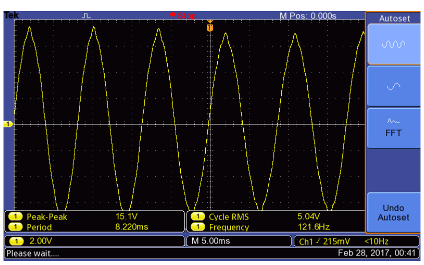
Experiment Documentation



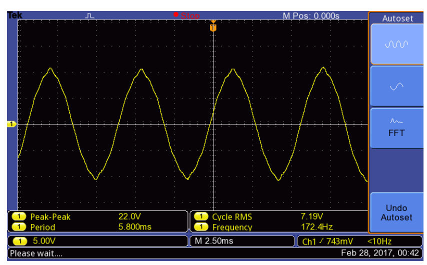
(a) 300 RPM



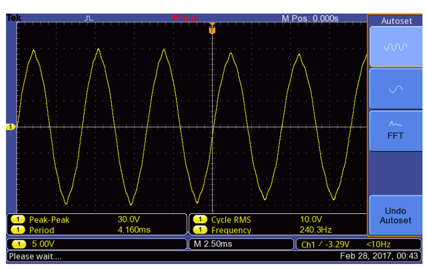
(b) 500 RPM



(c) 1000 RPM



(d) 1500 RPM



(e) 2000 RPM

Figure A.1: Back emf at various motor speeds

Appendix B

Part Drawings

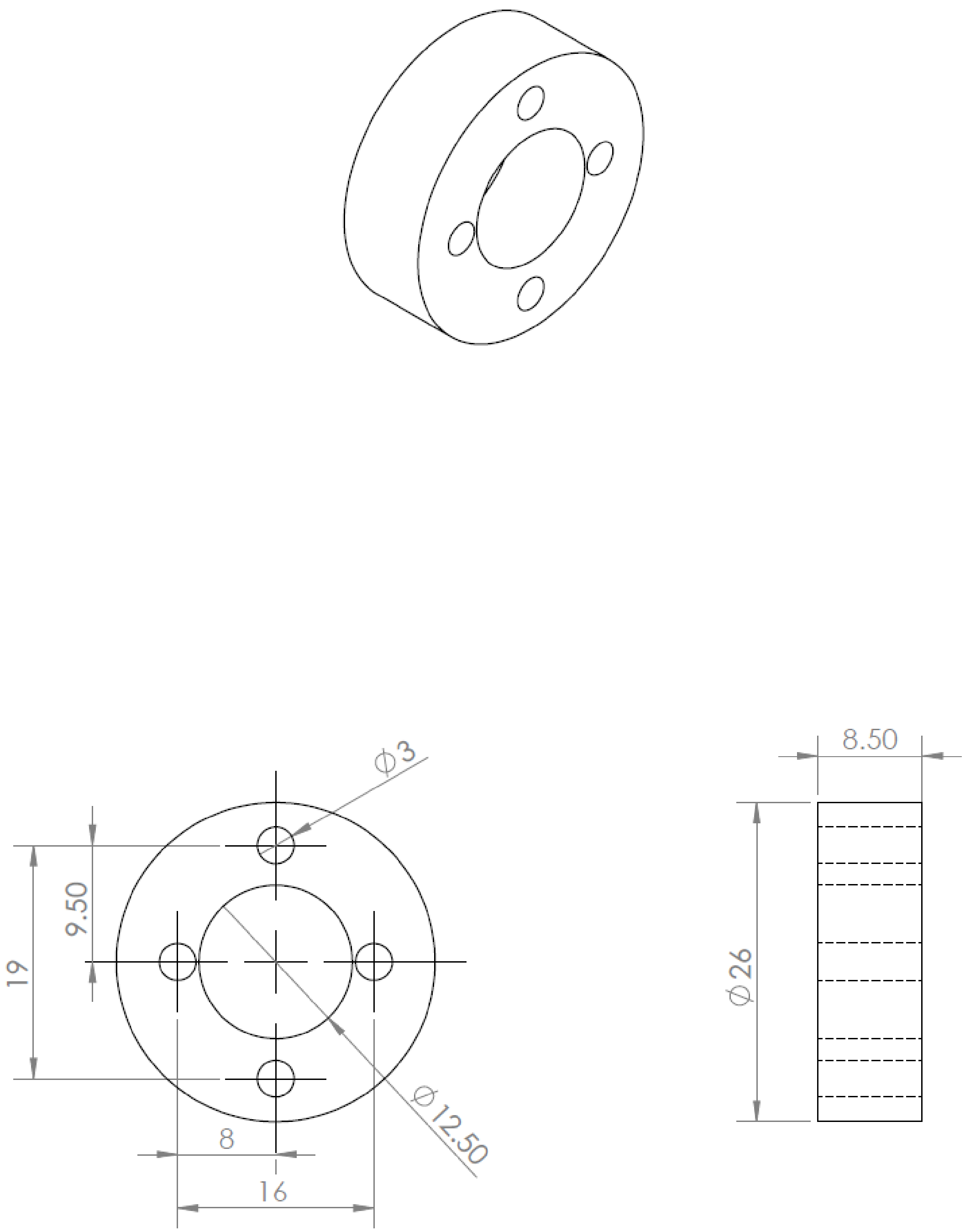


Figure B.1: Standoff for NERMLAB [mm]

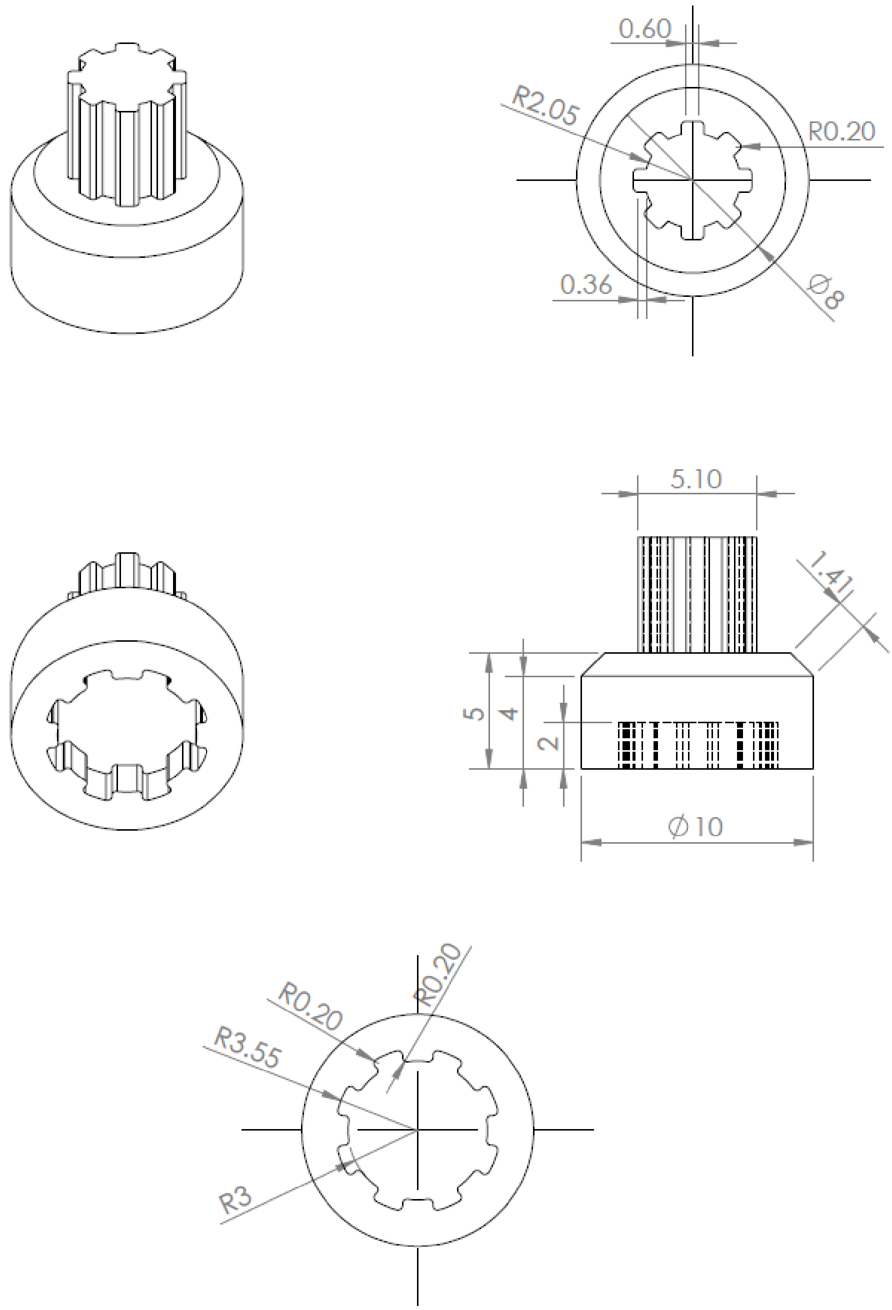


Figure B.2: Magnet holder for NERMLAB [mm]

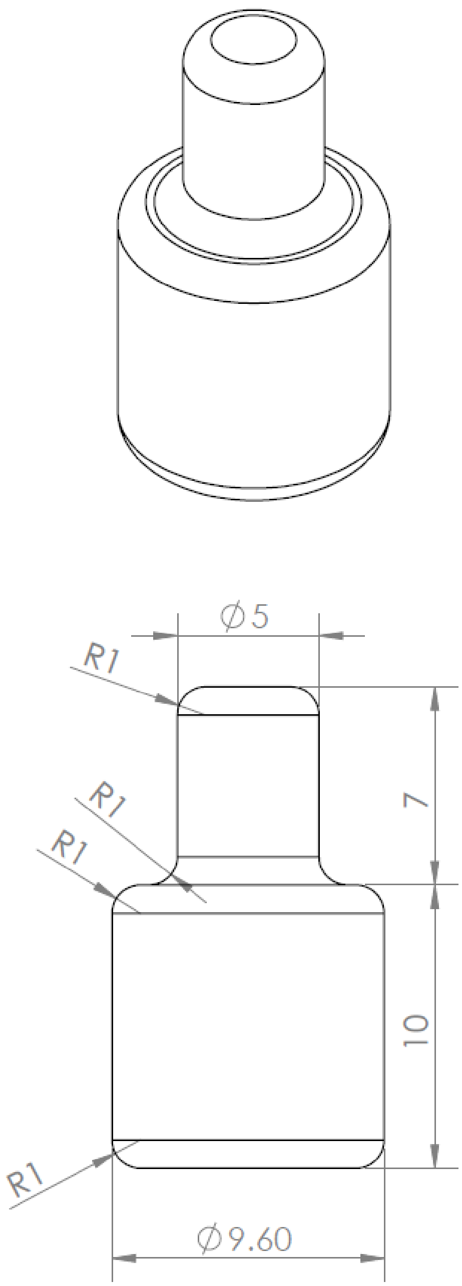


Figure B.3: Torque Transmission Shaft for NERMLAB [mm]

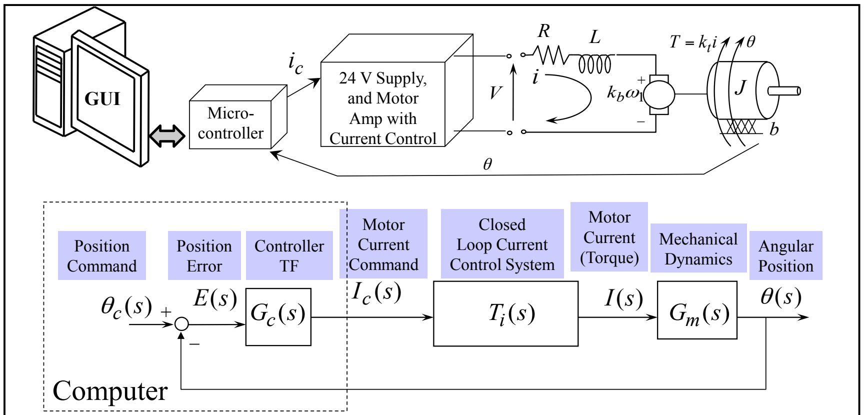
Appendix C

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.



$G_m(s) = \dfrac{\theta(s)}{I(s)} = \dfrac{k_t k_{dr}}{Js^2 + bs}$

$G_c(s) = K_p$, proportional controller

and CLTF

$\dfrac{\theta(s)}{\theta_c(s)} = \dfrac{K_p k_t k_{dr}}{Js^2 + bs + K_p k_t k_{dr}}$

J = motor inertia (from motor lab handout)

b = viscous friction coeficient (estimated to be 3×10^{-5} N $\cdot$ m $\cdot$ s/rad)

k_t = motor torque constant (from motor lab handout)

$k_{dr} = 180^\circ / \pi\,(rad)$ = angular postion unit conversion

$T_i(s) = 1$ (amplifier assumed "fast")

K_p = proportional gain of controller

$\theta(t)$ = angular position (measured in degrees for this problem)

Work To Be Done Prior To Lab

a) Assuming the transfer function of the closed loop current control system, $T_i(s)$, is one obtain a symbolic representation of the CLTF $\theta(s)/\theta_c(s)$.

b) From a) write an equation for the closed-loop poles of the system.

c) 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).

d) Plug in the numbers and determine the value of K_p for part c).

e) 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. $K_p = 0.01, 0.001, 0.0001(units?)$

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(data*i*)" 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, $-\varsigma\omega_n \pm j\omega_d$ (rad/s)	Theoretical Period of Oscillations, $2\pi/\omega_d$ (seconds)	Measured Period of Oscillations, T (seconds)	Theoretical Time Constant of Envelope, $1/\varsigma\omega_n$ (seconds)	Measured Time Constant of Envelope, τ (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 Kp=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

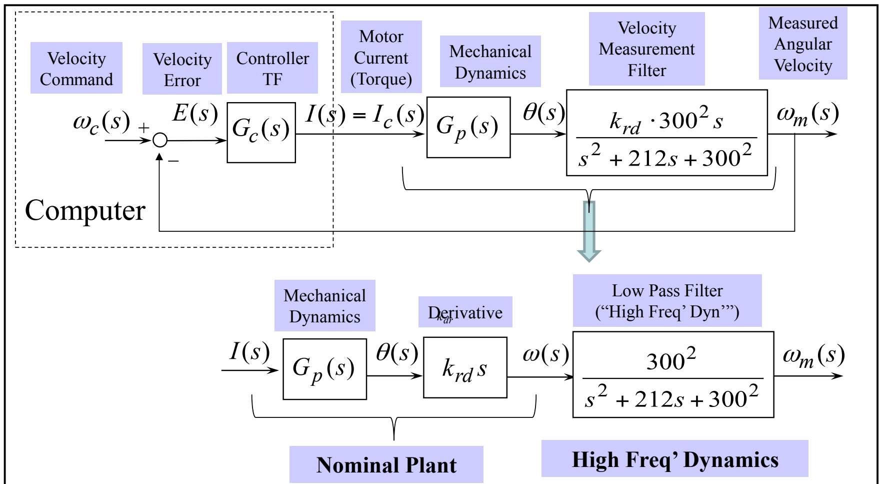
```
% lab5.m file
% Requires that the square-wave-response data files
% have been imported into data1, data2, and data3.

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 dyanamics." 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.



Nominal Plant

$$G_p(s) = \frac{\theta(s)}{I(s)} = \frac{k_t k_{dr}}{Js^2 + bs}$$

$$G_s(s) = \frac{\omega(s)}{I(s)} = \frac{k_t k_{dr} k_{rd}}{Js + b}$$

Controller

$$G_c(s) = K_p$$

J = motor inertia + double collar (1.29×10^{-5} kg·m^2)

b = viscous friction coeficient (estimated to be 3×10^{-5} N·m·s/rad)

k_t = motor torque constant (0.05 N·m/A)

$k_{dr} = 180/\pi$ deg/rad

$k_{rd} = \frac{1\,rpm}{6\,\text{deg}/s}$ = angular velocity unit conversion

K_p = proportional gain of controller (A/rpm)

θ(t) = angular position (measured in deg by controller)

ω(t) = angular speed (measured in rpm by controller)

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 .**

A. ____________________
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).

C. ____________________
D. ____________________
E. ____________________
F. ____________________
G. does/does not
H. ____________________

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.

I. ____________________
J. ____________________
K. ____________________
L. ____________________

At a gain of Kp=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.

M. (units)
N. ____________________
O. ____________________
P. ________ +/- ________ j
Q. ____________________
R. do/do not

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 **<u>-68.8</u>** 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.

S. (units)
T. ____________________
U. larger/smaller
V. ________ +/- ________ j
W. ____________________
X. ____________________

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.

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]);

% WHEREEVER 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: Back-EMF Plotter

```
'''
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 (Ke).

'''

# 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()
```