Ministry of Education and Science of the Russian Federation
Peter the Great St. Petersburg State Polytechnic University
Institute of Computer Science and Control Systems

**Control Systems and Technology Department**

# Report for Laboratory No. 2

Dynamic Analysis of Current and Velocity Loops

Course: Mathematical Modeling and Simulation

Student Group: 13541/8

____________________ Christopher W. Blake

____________________ Susanne Garcia

____________________ Valentyn Sichkar

____________________ Christos Voulvoukelis

Professor

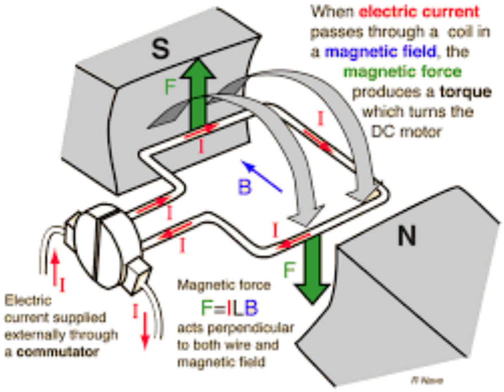
____________________ Rostov N.V

St. Petersburg
2016

# Contents

# Introduction

Lab 2 is about the simulation of a DC motor’s feedback loop via current and velocity regulation. A motor is selected from an online catalog, providing required design constant values. Additionally, design constants are defined to describe the power controller, current regulator, and current sensor. Using these design constants, various plots and critical values are calculated. Finally, using these plots and critical values, a summary of the motor’s feedback loops and their optimization is produced.



The analysis software for this lab is Matlab, specifically the “Control System Toolbox”. It will be used for producing Bode and step charts, which produce important result data such as percent overshoot and settling time. All code used for this analysis can be seen in Appendix 1.

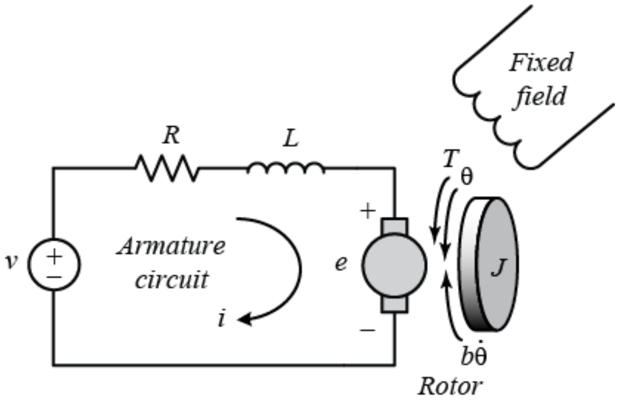
# Background

## DC Motor

The analytical transfer function for a DC motor’s angular speed is shown below. Using this transfer function, plots are produced and utilized in Matlab.

<u>Motor Constants</u>

Ra = Electrical Resistance
L = Electrical Inductance
$K_e$ = Electromotive force constant
$K_m$ = Motor torque constant
J = Moment of inertia of the rotor
$W_{DCM}$ = Angular motor speed



<u>Motor Defined Variables</u>

$$T_m = \frac{JR_a}{K_e K_m}$$

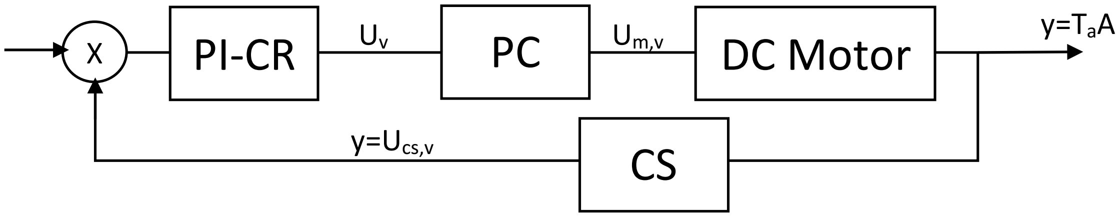
$$T_a = \frac{L}{R_a}$$

$$K_m = K_e$$

<u>Motor Transfer Function</u>

$$W_{DCM}(s) = \frac{{}^{1}/_{K_e}}{T_m T_a s^2 + T_m s + 1}$$

## Current Loop



### Feedback Loop Constants

Kpc = Power controller constant
Tpc = Power controller Inductance/resistance ratio
Kcs = Current sensor constant
Kcr = Current regulator constant
Tcr = Current regulator inductance/resistance ratio

### Component Transfer Functions

$$W_{pi-cr}(s) = K_{cr} + \frac{K_{cr}/T_{cr}}{s}$$

$$W_{pc} = \frac{K_{pc}}{T_{pc}s + 1}$$

$$W_{DCM}(s) = \frac{1/R_a}{T_A s + 1}$$

$$PI\_CR(s) = K_{cr}\frac{T_{cr}S + 1}{T_{cr}S}$$

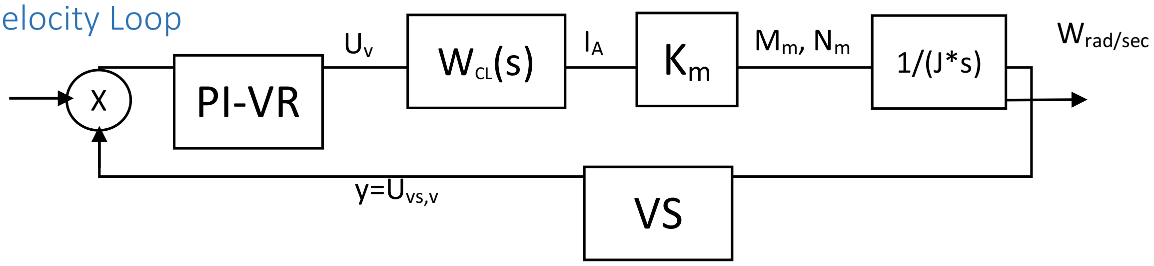
### System Transfer Functions

$$Plant\_CL(s) = W_{pc}(s) * W_{DCM}(s) * K_{cs}$$

$$CL\_Open(s) = PI\_CR(s) * Plant\_CL(s)$$

CL_Closed = ??? //This was made using the feedback function in Matlab.

## Velocity Loop



### Feedback Loop Constants

Kpc = Power controller constant
Tpc = Power controller Inductance/resistance ratio
Kvs = Current sensor constant
Kvr = Current regulator constant
Tvr = Current regulator inductance/resistance ratio

### Component Transfer Functions

$$W_{PI-VR}(s) = K_{vr} + \frac{K_{vr}}{T_{vr}S}$$

$$W_{CL}(s) = \frac{1/K_{cs}}{T_a S + 1}$$

### System Transfer Functions

$$Plant\_VL(s) = \frac{1/K_{cs}}{2T_{pc}s + 1} * K_m * \frac{1}{Js + 1} * K_{vs}$$

$$PIVR(s) = K_{VF} * \frac{T_{VR}s + 1}{T_{VR}}$$

$$VL\_Open(s) = PI\_VR * Plant\_VL$$

VL_Closed = ??? //This was made with the matlab function feedback.

# Results

## Motor Details

A random motor was chosen with the following characteristics. This motor has been analyzed via the Bode and Step functions to determine an optimization point for the current regulator ($K_{cr}$) or velocity regulator ($K_{vr}$).

### Chosen Motor Design Constants

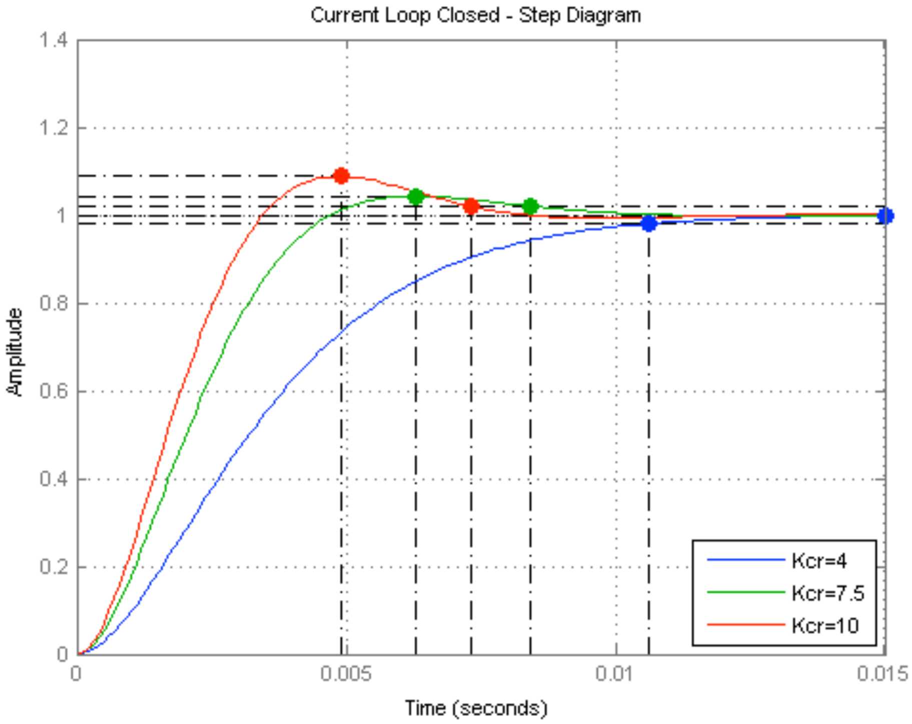
| | | | |
|---|---|---|---|
| Ra | = 15 | Ohms | = Electrical Resistance |
| L | = 0.15 | Henrys | = Electrical Inductance |
| $K_e$ | = 0.1 | V/(rad/sec) | = Electromotive force constant |
| $K_m$ | = $K_e$ | Nm / P^2 | = Motor torque constant |
| J | = 0.12E-5 | Kg/m^2 | = Moment of inertia of the rotor |
| $W_{DCM}$ | = (output) | rad/sec | = Angular motor speed |

## Current Loop

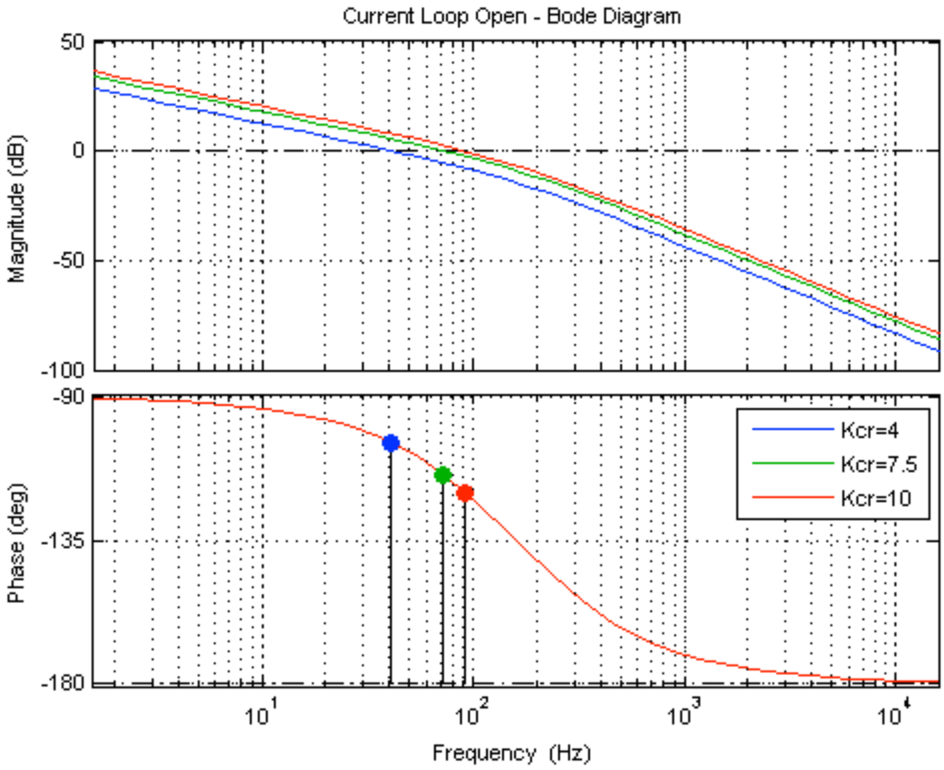
The current regulator constant (Kcr) was modified from 4 to 10 with intervals of 1. This can be seen in the legend of each chart.

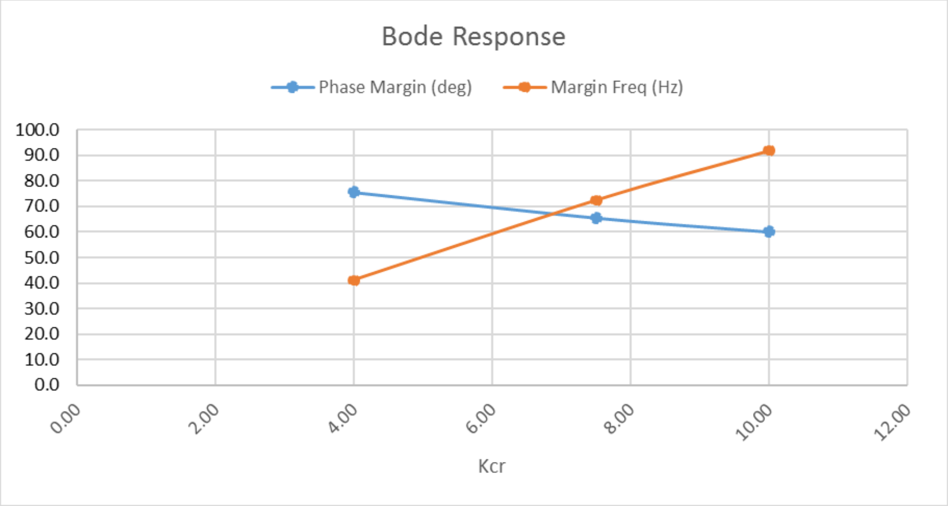
Feedback Loop Constants

Kpc = 3 = Power controller constant
Tpc = 0.001 = Power controller Inductance/resistance ratio
Kcs = 3.33 = Current sensor constant
Kcr = (input) = Current regulator constant
Tcr = $T_a$ = Current regulator inductance/resistance ratio



| Kcr | Peak Response | % Overshoot | Settling Time (sec) | Settling Time(millisec) |
|---|---|---|---|---|
| 4 | 0.998 | 0 | 0.0106 | 10.6 |
| 8 | 1.04 | 4.31 | 0.0063 | 6.3 |
| 10 | 1.09 | 8.75 | 0.0078 | 7.76 |





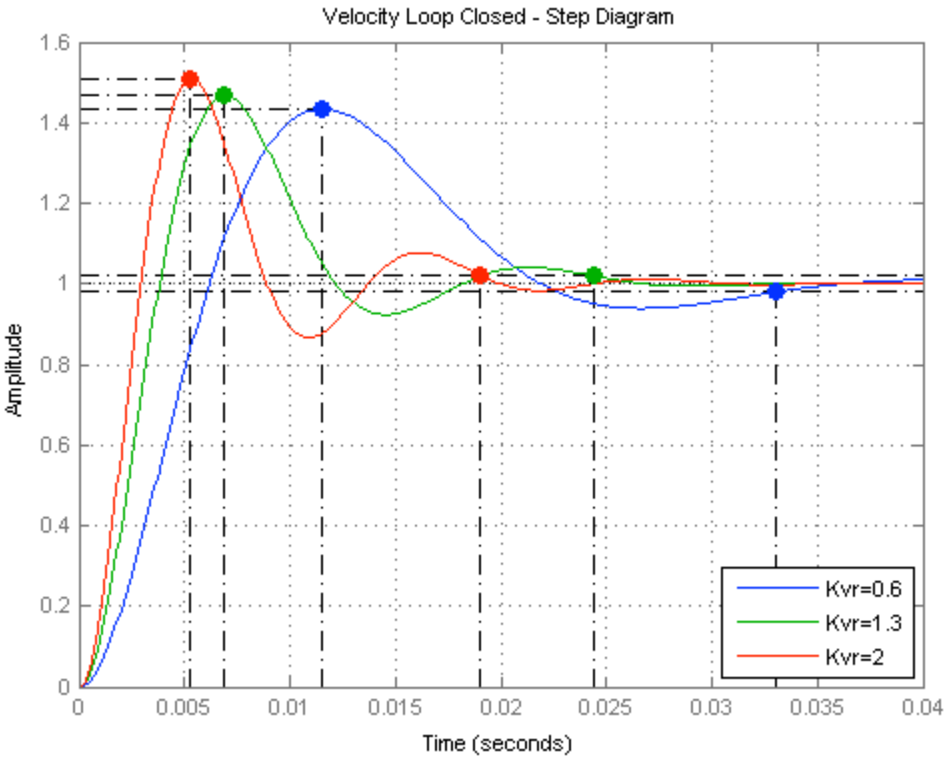
| Kcr | Phase Margin (deg) | Margin Freq (Hz) |
|---|---|---|
| 4.00 | 75.5 | 41.1 |
| 7.50 | 65.5 | 72.4 |
| 10.00 | 60.0 | 91.8 |

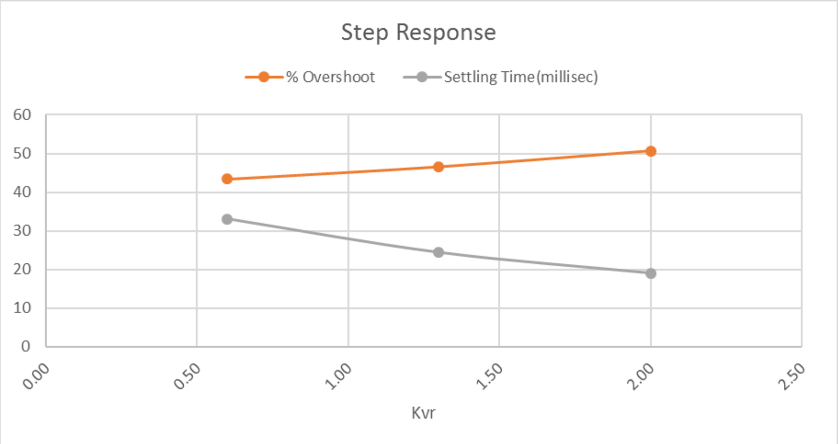
## Velocity Loop

The velocity regulator constant (Kvr) was modified from 0.6 to 2 with intervals of 0.2. This can be seen in the legend of each chart.

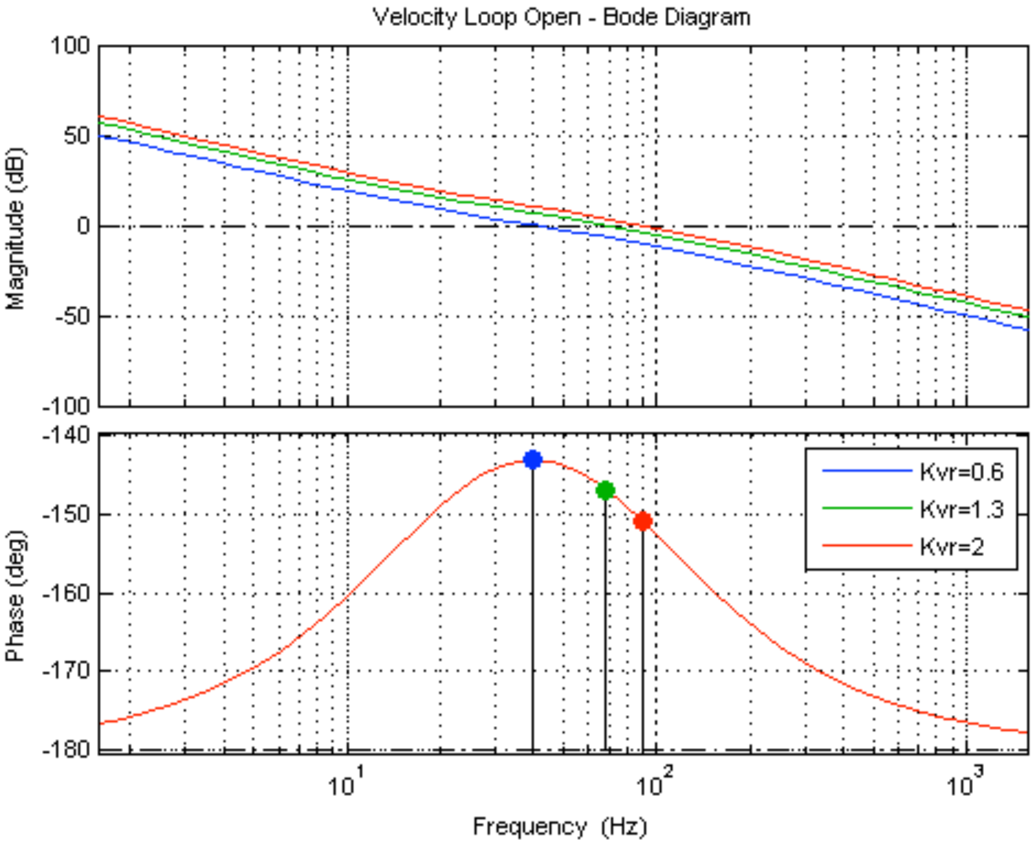
Feedback Loop Constants

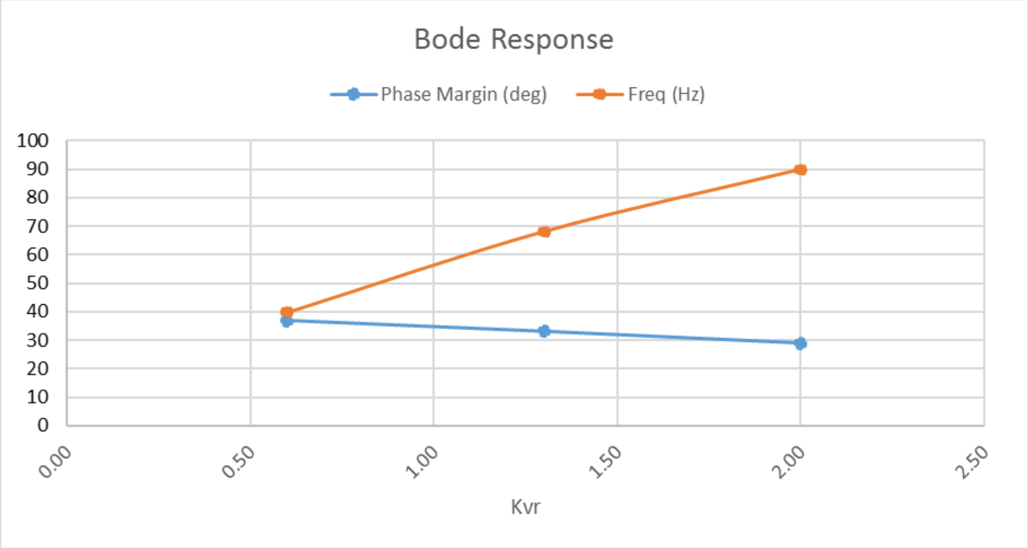
Kpc = 3 = Power controller constant
Tpc = 0.001 = Power controller Inductance/resistance ratio
Kvs = 3.33 = Velocity sensor constant
Kvr = (input) = Velocity regulator constant
Tvr = 8*$T_{pc}$ = Velocity regulator inductance/resistance ratio





| Kvr | Peak Response | % Overshoot | Settling Time (sec) | Settling Time(millisec) |
|---|---|---|---|---|
| 0.60 | 1.43 | 43.4 | 0.0331 | 33.1 |
| 1.30 | 1.47 | 46.6 | 0.0244 | 24.4 |
| 2.00 | 1.51 | 50.7 | 0.0190 | 19 |





| Kvr | Phase Margin (deg) | Freq (Hz) |
|---|---|---|
| 0.60 | 36.9 | 39.8 |
| 1.30 | 33.1 | 68.2 |
| 2.00 | 29.0 | 90.0 |

## Conclusion

### Current Loop

Utilizing the Bode and Step diagrams for varying Kcr, it can be shown that the optimal value for Kcr is 7.5. This Kcr value produces the idea overshoot of 4.3%, has a settling time of 0.0063 seconds, and has a phase margin of 65.5 deg.

### Velocity Loop

Utilizing the Bode and Step diagrams for varying Kvr, it can be shown that the optimal value for Kvr is 0.6. This Kcr value produces the idea overshoot of 43.4%, has a settling time of 0.0331 seconds, and has a phase margin of 36.9 deg.

# Appendix 1 – Matlab Code

```
%Lab2(Dynamic Analysis of Current and Velocity Loops)
%Using the control systems toolbox
clear all; close all; clc;

% Nomenclature
% pi-cr = PI Current Regulator
% pc = Power Controller
% cr = Current Regulator
% cs = Current Sensor

%% Motor Properties
Um_max = 30;     % Max applied voltage          Units: volts
Wmax = 300;      % Max velocity                 Units: rad/sec
Ra=15;           % Electrical resistance        Units: ohms
La= 0.15;        % Electrical inductance        Units: henry
J=0.12E-5;       % Inertia                      Units: Kg*m^2

Ta = La/Ra;      % Inductance Resistance Ratio
Ke=Um_max/Wmax; % Electromotive force constant  Units: volts / (rad/sec)
Km=Ke;           % Torque constant              Units: Nm / (Work)^2

%% Feedback Control Loop Properties
% Power Controller
Kpc=3;
Tpc=0.001;

% Current Sensor
Kcs=3.33;

% Current Regulator
Kcr=25;
Tcr=Ta;

%% Current Loop
% Create transfer functions for changing Kcr
i = 0;
for Kcr=[4.0, 7.5, 10]
    i = i + 1;
    %Store Transfer Functions
    Plant_CL= tf(Kpc, [Tpc 1]) * tf(1/Ra, [Ta 1]) *Kcs;
    PI_CR = Kcr*tf([Tcr, 1], [Tcr, 0]);
    CL_Open(i) = PI_CR*Plant_CL;
    CL_Closed(i) = feedback(CL_Open(i), 1);

    %Store legend entries
    theLegend(i) = {['Kcr=' num2str(Kcr)]};
end
L = i;

%Plot: Open Loop
f1 = figure(1); hold;
movegui(f1,'northeast');
P = bodeoptions;
P.FreqUnits = 'Hz';
for i = 1:L
     bode(CL_Open(i), {10,100000}, P); grid on
end
title('Current Loop Open - Bode Diagram');
```

```
legend(theLegend);

%Plot: Closed Loop
f2 = figure(2); hold;
movegui(f2,'southeast');
for i = 1:L
     step(CL_Closed(i), 0:0.0001:0.015); grid on
end
title('Current Loop Closed - Step Diagram');
legend(theLegend);
set(legend, 'location', 'southeast');

%% Velocity Loop
Tvr=8*Tpc;
Kvs=10/600;
Kvr=1;

%Create transfer functions for changing Kcr
i = 0;
for Kvr=[0.6, 1.3, 2.0]
    i = i + 1;
    %Store Transfer Functions
    Plant_VL= tf(1/Kcs, [2*Tpc 1]) * Km * tf(1, [J 0]) * Kvs;
    PI_VR = Kvr*tf([Tvr, 1], [Tvr, 0]);
    VL_Open(i) = PI_VR*Plant_VL;
    VL_Closed(i) = feedback(VL_Open(i), 1);

    %Store legend entries
    theLegend(i) = {['Kvr=' num2str(Kvr)]};

end
L = i;

%Plot: Open Loop
f1 = figure(3); hold;
movegui(f1,'northeast');
P = bodeoptions;
P.FreqUnits = 'Hz';
for i = 1:L
     bode(VL_Open(i), {10,10000}, P); grid on
end
title('Velocity Loop Open - Bode Diagram');
legend(theLegend);

%Plot: Closed Loop
f2 = figure(4); hold;
movegui(f2,'southeast');
for i = 1:L
     step(VL_Closed(i), 0:0.0001:0.04); grid on
end
title('Velocity Loop Closed - Step Diagram');
legend(theLegend);
set(legend, 'location', 'southeast');
```