MECE-743 Digital Controls Project

April 15-29, 2016, Rochester, NY, USA

**Lab4: Arduino Lead/Lag control of a DC motor system**

**Joseph Jennings**

Department of Mechanical Engineering

Rochester Institute of Technology

Rochester, NY 14623

Email. jej7600@rit.edu

**ABSTRACT**

Arduino is utilized in this lab to create the compensator lead/lag system of the motor. The open loop and closed loop are used to run the motor and RPM is recorded. The Set Point is RPM, Command is volts and Tachometer Feedback signals are RPMs.

**INTRODUCTION**

Arduino is used to drive motor drive system, and collect data from the output of the motor. The sampling time of .05 secs is used for discrete time domain. The square wave voltage 0 to 1,2,3,4 and 5 volts are set up. The voltage command is supplied as the voltages. The output data of motor is measured. From the output data, the required lead lag compensator of the system is determined using theoretical calculations. The simulation of the system with and without the compensator is plotted. From root locus, the gain value K can be obtained. When the gain value changes, it changes the closed loop poles’ locations. Depending on the changes in root locus determine how well the system can be stable and controlled well.

**ANALYSIS**

System ID is the main method used in this lab to get K\_tack values. Using the values of data, the TF is:

**Zeyar Win**

Department of Mechanical Engineering

Rochester Institute of Technology

Rochester, NY 14623

Email. zxw6805@rit.edu

K tack value is .0172

The closed loop transfer function is

When z = 1,

G(z) = .1285/ (1 -.835) = .7545

The steady state error is

Ess = 1/( 1+K(.7545)) = .1

K = .9/.0755 = 11.9277

For the steady state error less than 10%, k must be greater than 11.9277.

W\_w1 value is 16.8 rad/s and W\_w0 value is 1.68 rad/s .W\_wp is .8684.

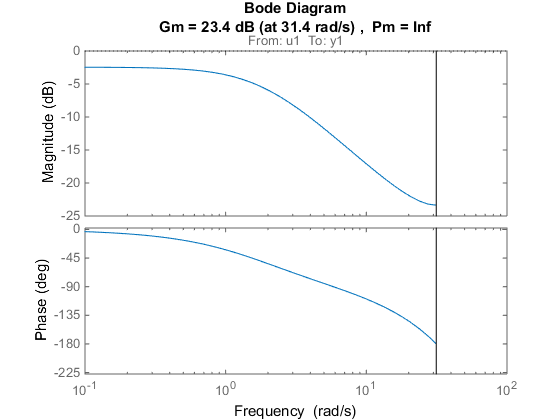


Figure 1.design of controller using this bode plot

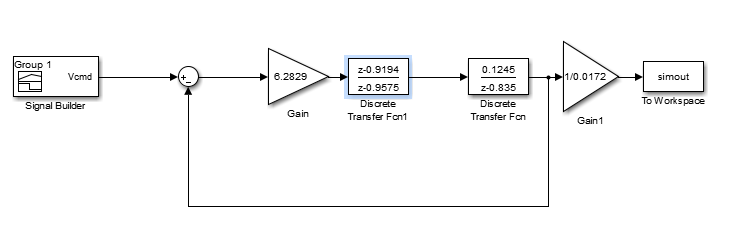
The compensator transfer function is 

The z domain is converted from w.

Kp value is 6.2879. Z0 is .9184 and Zp is .9575.

D(z) is 6.2829(z-.8184)/(z-.8575)

The design can be seen below.



**SIMULATION**

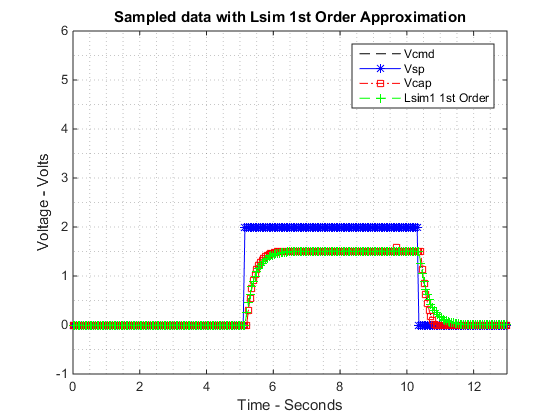
First, the data obtained is plotted against the theoretical data and Lsim 1st order MatLab simulation to check the obtained data is right to continue the lab process. The figure can be seen as in figure 1. 

Figure 2. Simulation of motor system

Simulation and real output data can be seen in figure 2.

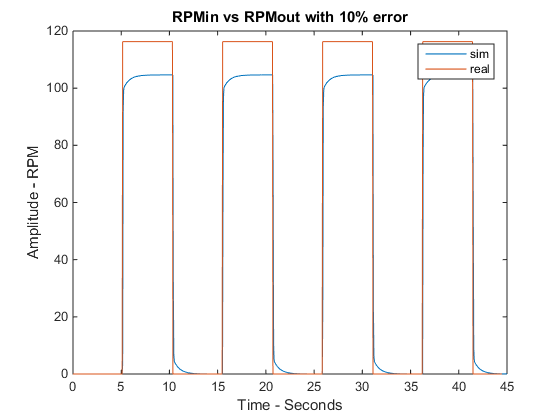


Figure 3. Simulation of RPM in matlab

The output data is collected from Arduino with 0 to 2 Volts’ square wave input set point. The model is simulated in the MatLab. The data plot obtained directly from Arduino is used to compare it with MatLab model as in figure 3.

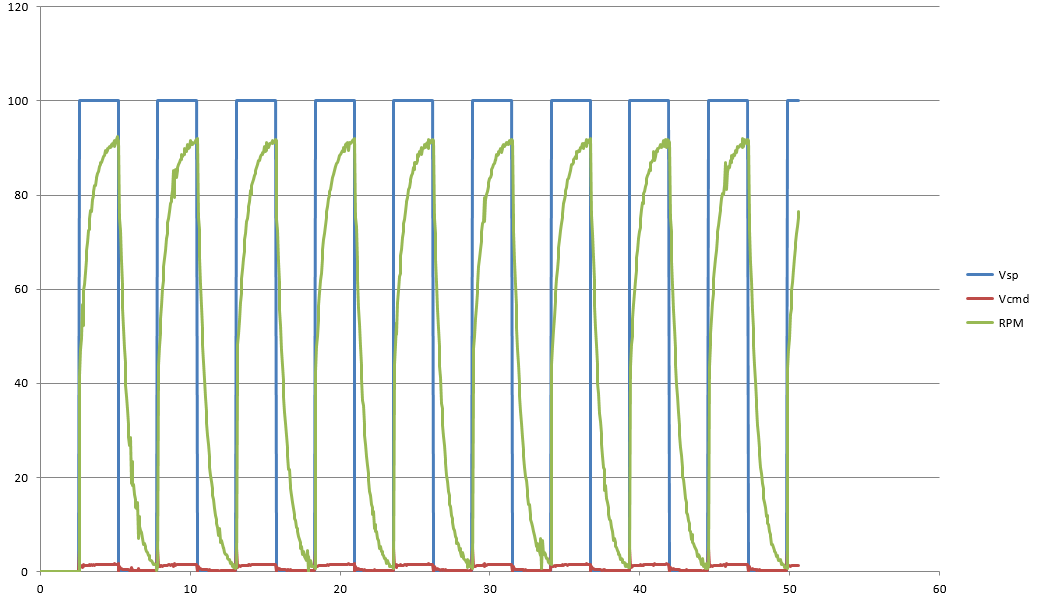


Figure 4. The Simulation of system with compensator 0 to 100RPM

The output data don’t match well with the simulation since there is noise in motor drive system. There is error when doing continuous system and more error is added when discrete system is used.

Figure 5. The Simulation of system with compensator 100 to 200 RPM

**CONCLUSION**

If there is a saturation, some of the output data will be missing. The controller doesn’t saturate since the voltage supplied is only between 0 to 5 volts. The controller will start saturation when the Kp is greater than 2.5. The validation data simulation with K within stability region being similar to original output data confirms that the transfer function of system ID method is correct. Overall, the lab is completed successfully using Arduino and MatLab simulation.

clear

clc

close all

filename='lab4\_0to2V\_86RPM.xlsx';

filename1='lab4\_0to4V\_174RPM.xlsx';

% Reading Excel file to gather data to be interpreted

lab3data=xlsread(filename);

lab3q10=xlsread(filename1);

Data3=[lab3data(:,1),lab3data(:,2),lab3data(:,3)];

Data3q10=[lab3q10(:,1),lab3q10(:,2),lab3q10(:,3)];

t2=Data3q10(:,1);

Vcmd2=Data3q10(:,2);

Vcap2=Data3q10(:,3);

t1=Data3(:,1);

Vcmd1= Data3(:,2);

Vcap1=Data3(:,3);

% Arx Data for question 10

Dat1=iddata(Vcap2,Vcmd2,0.1);

sysTF10=arx(Dat1,[1 1 1]);

tf2=tf(sysTF10)

ys10=lsim(tf2,Vcmd2);

% Arx Data for sys ID for G(z) to be used in CLTF

Dat=iddata(Vcap1,Vcmd1,0.1);

sysTF = arx(Dat,[1 1 1]);

tf1=tf(sysTF)

ys=lsim(tf1,Vcmd1);

close all;

sim('lab4');

figure;

plot(t1,Vcap1,'--k',t1,Vcmd1,'-b\*',t1,Vcap1,'-.rs',t1,ys,'--g+');

xlabel('Time - Seconds');

ylabel('Voltage - Volts');

grid minor;

legend('Vcmd','Vsp','Vcap','Lsim1 1st Order');

title('Sampled data with Lsim 1st Order Approximation');

axis([0 13 -1 6])

figure;

plot(tout,simout,t1,(1/0.0172)\*Vcmd1)

title('RPMin vs RPMout with 10% error');

xlabel('Time - Seconds');

ylabel('Amplitude - RPM');