MECE-743 Digital Controls Project

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

**Lab3: Arduino PI control of a RC circuit**

**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 obtain the transfer function of the RC circuit system. The order of the transfer function system is 1 since the RC circuit with only one 39k ohm resistance and one capacitance is used. The system ID is used to make theoretical simulation of the system. Root locus is plotted and gain value is determined. The PI controller is designed in this lab.

**INTRODUCTION**

Arduino is used to drive RC circuit system, and collect data from the output voltage across capacitance. The sampling time of .1 secs is used for discrete time domain. The square wave voltage 0 to 2 volts are set up. The voltage command is supplied within 0 to 5 volts. The output voltage of capacitance is measured. From the output data, system ID is used to find the transfer function of the system. The root locus of the system with and without the compensator is plotted. From root locus, the gain value K can be obtained. For the complicated systems, the location of the roots and K value can be optimized with root locus method.

**ANALYSIS**

System ID is the main method used in this lab to verify with theory R values. Using the values of RC circuit, the TF is:

for continuous time domain

for discrete time domain

**Zeyar Win**

Department of Mechanical Engineering

Rochester Institute of Technology

Rochester, NY 14623

Email. zxw6805@rit.edu

With the system ID, the TF is

for continuous

for discrete

The closed loop transfer function for proportional controller is derived as below.

where D(z) =k

From the CLTF, the stability region bound values of k can found using the characteristic equation.

For z =1 case,

For z=-1 case,

k = 11.2743

Therefore, the system is stable when

-1.3361< k < 11.2743

The gain Kp is determined to be 2.5 because the Arduino cannot provide output more than 5 Volts. (2\*2.5 = 5)

The closed loop transfer function for the PI controller is determined as below.

When a =.7881, b = .1586, Kp =2.5 and T=0.1

From the root locus, the gain Ki is determined to be 8.83.

**SIMULATION**

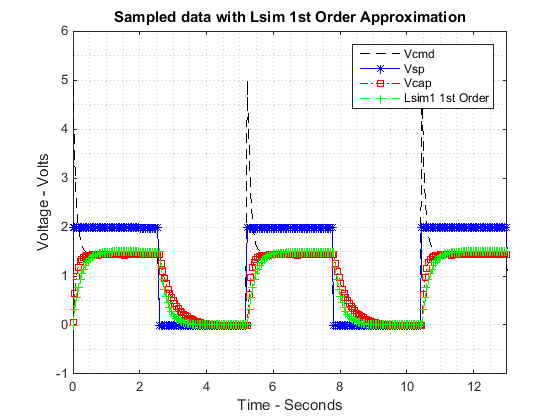
First, the data obtained is plotted against the theoretical data and Lsim 1st order MatLab simulation to check the obtained data is right to continue the lab process. The figure can be seen as in figure 1. 

Figure 1. Simulation of RC circuit

The root locus for the RC circuit system is plotted in figure 1.



Figure 2. Root locus plot of RC circuit

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. The theoretical simulation waveform is similar to the Vcap simulation. The theoretical waveform simulation act as validation data set in this lab.

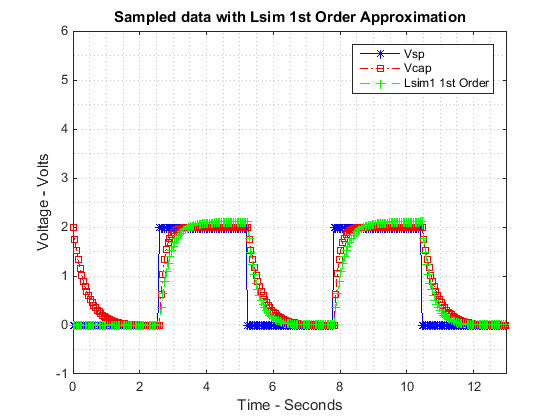


Figure 3. The simulation of RC circuit

With Kp value 2.5, the root locus is plotted. From root locus, Ki is determined to be 8.83.



Figure 4. The root locus of PI controller

The data obtained from voltage across capacitance is used as the output data. The theoretical system with close loop damping ratio of zeta being 1 is simulated in figure 5. The data directly obtained from the Arduino is plotted as well.



Figure 5. The simulation of RC with zeta value 1

**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='lab3data.xlsx';

filename1='Datastep10.xlsx';

% Reading Excel file to gather data to be interpreted

lab3data=xlsread(filename);

lab3q10=xlsread(filename1);

Data3=[lab3data(:,1),lab3data(:,2),lab3data(:,3),lab3data(:,4)];

Data3q10=[lab3q10(:,1),lab3q10(:,2),lab3q10(:,3),lab3q10(:,4)];

t2=Data3q10(:,1);

Vsp2=Data3q10(:,2);

Vcmd2=Data3q10(:,3);

Vcap2=Data3q10(:,4);

t1=Data3(:,1);

Vsp1= Data3(:,2);

Vcmd1=Data3(:,3);

Vcap1=Data3(:,4);

% Arx Data for question 10

Dat1=iddata(Vcap2,Vsp2,0.1);

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

tf2=tf(sysTF10)

ys10=lsim(tf2,Vsp2);

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

Dat=iddata(Vcap1,Vsp1,0.1);

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

tf1=tf(sysTF)

ys=lsim(tf1,Vsp1);

close all;

% Question 5 Plot

figure;

rlocus(tf1)

% Question 7

open\_system('step7.slx');

sim('step7');

figure;

plot(t1,Vcmd1,'--k',t1,Vsp1,'-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])

% plot for Question 7

figure;

plot(t2,Vsp2,'-b\*',t2,Vcap2,'-.rs',t2,ys10,'--g+');

xlabel('Time - Seconds');

ylabel('Voltage - Volts');

grid minor;

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

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

axis([0 13 -1 6])

% Plot for Question 9

figure;

kp=2.5;

T=0.1;

a=0.7881;

b=0.1586;

num2=[(T\*b) 0];

den2=[1 (kp\*b-1-a) (-kp\*b+a)];

sys4=tf(num2,den2,T)

rlocus(sys4)

% Question 10

figure;

open\_system('step10lab3');

sim('step10lab3');

plot(tout,step10sysid,'-g\*',tout,step10theo,'-.rs',tout,Vsp10,'-b\*',tout,Vcmd10,'--k');

axis([0 13 -1 6]);

title('Simulated Compared to Theoretical Model, Step 10')

xlabel('Time - Seconds')

ylabel('Amplitude - Volts');

legend('Vcap','Theoretical','Vsp','Vcmd','Location','Best');

grid minor

sysTF10 =

Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

A(z) = 1 - 0.835 z^-1

B(z) = 0.174 z^-1

Sample time: 0.1 seconds

Parameterization:

Polynomial orders: na=1 nb=1 nk=1

Number of free coefficients: 2

Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using ARX on time domain data "Dat1".

Fit to estimation data: 95.69% (prediction focus)

FPE: 0.001473, MSE: 0.00145

tf2 =

From input "u1" to output "y1":

0.174 z^-1

--------------

1 - 0.835 z^-1

Sample time: 0.1 seconds

Discrete-time transfer function.

sysTF =

Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)

A(z) = 1 - 0.7881 z^-1

B(z) = 0.1586 z^-1

Sample time: 0.1 seconds

Parameterization:

Polynomial orders: na=1 nb=1 nk=1

Number of free coefficients: 2

Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using ARX on time domain data "Dat".

Fit to estimation data: 92.56% (prediction focus)

FPE: 0.002219, MSE: 0.002185

tf1 =

From input "u1" to output "y1":

0.1586 z^-1

---------------

1 - 0.7881 z^-1

Sample time: 0.1 seconds

Discrete-time transfer function.

sys4 =

0.01586 z

----------------------

z^2 - 1.392 z + 0.3916

Sample time: 0.1 seconds

Discrete-time transfer function.

Published with MATLAB® R2014b

