EXPERIMENT KW7: CONTROL SYSTEM SIMULATION USING MATLAB AND SIMULINK (OPEN-ENDED)

Related course: KIE3006 (Control System)

OBJECTIVES:
1. To represent systems in terms of transfer function or pole-zero-gain using MATLAB
2. To analyse systems using plots of zero-pole and time response
3. To simulate a control system in Simulink environment

EQUIPMENT:
PC with MATLAB and Simulink

INSTRUCTIONS:
1. Record all your results and observations in a pendrive
2. Follow the demonstrator's instructions throughout the experiment

REFERENCE(S):
Refer to the main references of KIE3006

TESTS:
Test 1: Control System in MATLAB
Test 2: Control System in Simulink

TEST 1: Control System in MATLAB

INTRODUCTION:

The dynamic characteristics of simple systems can be described by first order (RC and LC circuits) or second order (RLC circuit) differential equations. The characteristic equation for the second order RLC circuit can be expressed by:

$$\frac{d^2v}{dt^2} + 2\alpha\frac{dv}{dt} + \omega_0^2 v = 0$$

$$\alpha = \frac{R}{2L}\text{ (for series RLC)}, \alpha = \frac{1}{2RC}\text{ (for parallel RLC) and } \omega_0 = \frac{1}{\sqrt{LC}}\text{ (natural frequency)}$$

Applying the Laplace transformation, the obtained transfer function is:

$$H(s) = \frac{V_o(s)}{V_{in}(s)} = \frac{\omega_0^2}{s^2 + 2\alpha s + \omega_0^2}$$

Linear Time-Invariant Systems in MATLAB:

Control System Toolbox in MATLAB offers extensive tools to manipulate and analyze linear time-invariant (LTI) models. It supports both continuous and discrete-time systems. Systems can be single-input/single-output (SISO) or multiple-input/multiple-output (MIMO). You can specify LTI models as a transfer functions (TF) such as

$$P(s) = \frac{s+2}{s^2+s+10}$$

Note: All LTI models are represented as a ratio of polynomial functions

PROCEDURES:

1. Creating Transfer Function Models

Open MATLAB. Then, create a transfer function (TF) model by specifying numerator and denominator coefficients by typing the following lines on the MATLAB Command Window. After typing every line, press Enter.

```
>> num = [2 0];
>> den = [1 2 1];
>> sys = tf(num,den)
```

Copy the output that is displayed on the Command Window after you typed the above lines and paste it in a Word document.

2. Creating Zero-Pole-Gain Models

To create zero-pole-gain (ZPK) models for the TF in step 1, specify each of the three components in vector format by typing:

```
>> sys = zpk([0],[-1 -1],[2])
```

Copy the output that is displayed on the Command Window after you typed the above lines and paste it in a Word document.

Note: In “zpk([0],[-1 -1],[2]),” [0] is for the zero, [-1 -1] is for the pole and [2] is for the gain.

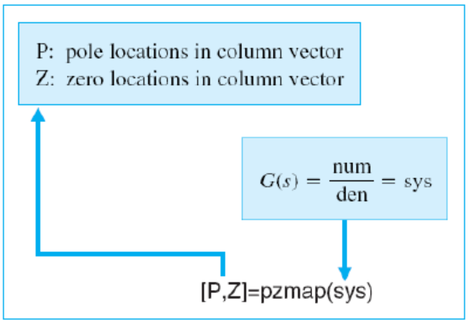
Type the following line for a more complicated ZPK model:

```
>> sys = zpk([2 0], [-1 -3 -.28],[.776])
```

Copy the output that is displayed on the Command Window after you typed the above lines and paste it in a Word document.

3. Plotting poles and zeros of a system

To compute pole-zero map of LTI models, the following codes are used in MATLAB:
pzmap(sys)
pzmap(sys1,sys2,...,sysN)
[p,z] = pzmap(sys)



Description:
pzmap(sys) plots the pole-zero map of the continuous or discrete-time LTI model system. For SISO systems, pzmap plots the transfer function of poles and zeros. The poles are plotted as x and the zeros are plotted as o.

pzmap(sys1,sys2,...,sysN) plots the pole-zero map of several LTI models on a single figure. The LTI models can have different numbers of inputs and outputs. If we write with left-hand arguments, **[p,z]= pzmap(sys)** returns the system poles and zeros in the column vectors p and z. No plot is drawn on the screen. You can use the functions **sgrid** or **zgrid** to plot lines of constant damping ratio and natural frequency in the s- or z- plane.

Plot the poles and zeros of a continuous-time system $H(s) = \frac{2s^2+5s+1}{s^2+2s+3}$ by typing the following lines in the MATLAB Command Window:

```
>> H = tf([2 5 1],[1 2 3])
>> figure(1)
>> pzmap(H)
>> sgrid
```

Print screen the results and paste in a Word document.

4. Simulation of linear systems for different input:

4.1 Impulse and step input

You can simulate the LTI systems to input like impulse, step and other standard input and see the plot of the response in the Figure window. MATLAB command **impulse** calculates the unit impulse response of the system, and **step** calculates the unit step response of the system. If we write without left-hand arguments, all commands plot the response on the screen.

To obtain an impulse response, type:

```
>> H = tf([2 5 1],[1 2 3]);
>> figure(2)
>> impulse(H)
```

To obtain a step response, type:

```
>> figure(3)
>> step(H)
```

Print screen the results for both impulse and step response and paste in a Word document.

4.2 Time-interval specification

To contain the response of the system, you can specify the time interval to simulate the system. For example, type:

```
>> t = 0:0.01:15;
>> figure(2)
>> impulse(H,t)
```

Or

```
>> t = 0:0.01:15;
>> figure(3)
>> step(H,t)
```

Print screen the results for both impulse and step response and paste in a Word document.

QUESTIONS (answer these during the lab session):

Q1. Consider the transfer function G(s) as follows. Using MATLAB, plot its pole-zero map.

$$G(s) = \frac{4.5s^2 + 1}{s^3 + 3s^2 + 3s + 7}$$

Q2. (a) Obtain the unit impulse response for the following system:

$$H(s) = \frac{2}{s^2 + 0.2s + 1}$$

(b) Obtain the unit step response for the following system:

$$H(s) = \frac{2}{s^2 + 0.2s + 1}$$

(c) Explain why the results in (a) and (b) are not the same.

Q3. A system has a transfer function:

$$\frac{X(s)}{R(s)} = \frac{(15/z)(s + z)}{s^2 + 3s + 15}$$

Plot the response of the system when *R*(*s*) is a unit impulse when (a) *z* = 3, (b) *z* = 6, (c) *z* = 9 and (c) *z* = 12. Repeat when *R*(*s*) is a unit step.

TEST 2: Control System in Simulink

INTRODUCTION:

Simulink is a graphical tool that can simulate feedback control systems. For most of the systems we will encounter, only a small number of Simulink's component library will be used. In particular, the components you should be familiar with are:

Library	Components
"Continuous"	• Integrator – integrates a signal • State-Space – used to add a system block in state-space form • Transfer Fcn – used to add a system block in transfer function form
"Math Operations"	• Gain – a constant gain • Sum – used to add two or more signals • Trigonometric Function – used to place non-linear trigonometric elements
"Signal Routing"	• Mux – used to multiplex signals together in order to plot several on one graph
"Sinks"	• Scope – used for viewing system output • To workspace – used to transfer a signal to MATLAB
"Sources"	• Ramp – generates a ramp signal • Sine Wave – generates a sinusoid • Step – generates a unit step signal

1. 'Gain'
To modify a 'Gain' component, double-click on the component. The 'Gain' parameter can be either a number or an expression. For example, '9.8/2.5', '0.45' or '1'.

2. 'Sum'
The 'Sum' component is equivalent to a summing junction in a block diagram. To add more input nodes or change the sign of an input node, double-click on the 'Sum' component and modify the text in the 'List of Signs' parameter. To make the sum have a '+' and a '-' node, change the 'List of Signs' parameter to '|+−'. To add a third summing input node, change the text to '| + ++'.

3. 'To Workspace'
The 'To Workspace' component sends data to the MATLAB workspace where you can plot it or process it as you wish. To set this up, double-click on the component and change the 'Save format' parameter to 'Array'. Also, set the 'Variable name' parameter to something descriptive, such as 'yout'. Simulink also automatically exports the time in the MATLAB variable 'tout'. This allows you to plot the output in MATLAB using the standard 'plot' command, and also allows you to nicely label the plots.

4. 'Transfer Fcn'
To modify a 'Transfer Fcn' component, double-click on the block. The 'Numerator' and 'Denominator' parameters are the coefficients of the polynomials of the numerator and denominator of the transfer function. The coefficients are in order of decreasing power. E.g., if 'Numerator' is set to '[1 2 3]', the numerator of the transfer function is $s^2 + 2s + 3$.

PROCEDURES:

1. Open simulink by typing **simulink** on the MATLAB Command Window. Once simulink has loaded, create a new model by going to File, New → Model. Next, begin placing components on the empty window to create the following system. Set the Gain to K.

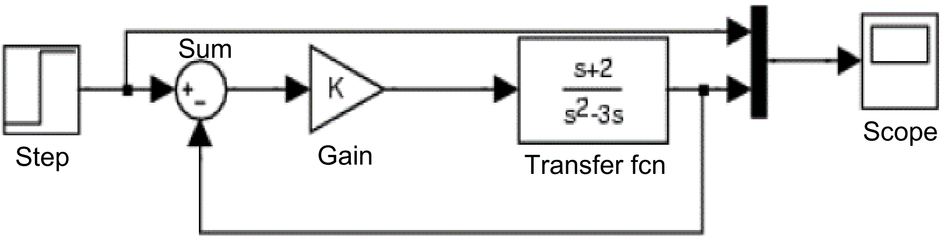


Figure 2

Notes:
- To search a component, type the component name in the search box. E.g. type **Transfer fcn** to search for Transfer function component.
- To place a component, drag it from the component browser to the model space. To make a connection, hold down **CTRL** and click on the arrows on each block that you want to connect. To connect multiple lines to a single block, hold down **CTRL** and click on the line already attached to the block and then make the second connection.

2. Replace the variable "K" with "1" in the constant gain controller. Change the simulation duration from 10 to 30. Then, run the simulation by clicking the triangle-shaped "play" icon in the toolbar and see what happens. Double-click on the scope to see the output of the simulation. Is the output stable? Increase the gain to 2 and re-run the simulation. Continue increasing the gain to 10 and observe the results.

3. Replace the ‘Gain’ with the ‘PID Controller’ block. Configure the PID parameters (Proportional, Integral and Derivative) by trial-and-error method. Observe the effect of changing the PID parameters on system performances (e.g., rise time, settling time, overshoot).

Open-ended tasks (do this task during laboratory session):

Q4. Find the value of K that gives the system response as “stable.” K can be an integer or a decimal number. Then, determine the pole locations using this K value.

Q5. Find the value of K that gives the system response as “oscillatory.” Then, determine the pole locations using this K value.

Q6. Find the value of K that gives the system response as “unstable.” Then, determine the pole locations using this K value.

Q7. What values of PID parameters provide the best performance based on your criteria? Explain how these parameters affect the system response.

Print screen the result for each case above and paste in a Word document.
.

END OF EXPERIMENT