

## EE 547 (PMP) Lab 2

Wednesday, January 14, 2015

Professor Linda Bushnell, UW EE

**Demo Problem 1** Consider the following state-space representation of a system:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\end{aligned}\tag{1}$$

- (a) Create an m-file script to define the system in MATLAB.
- (b) Use MATLAB function to convert the state-space representation to transfer function.
- (c) Use MATLAB function to convert the transfer function in previous step back into state space.
- (d) Response to the Step Inputs:
  - a. Create a Simulink model, **Lab2**, of the given system by using the Simulink Blocks shown in Figure 1.
  - b. Add a set of commands to the m-file script to simulate the response of the Simulink model. Please plot the input and outputs as a function of time.



Figure 1 Simulink blocks to be used in demo problem. Blocks **Clock** and **Step** can be found in folder **Sources**, block **To Workspace** in folder **Sinks**, and block **Transfer Fcn** in folder **Continuous**.

- (e) Response to the Sine Inputs:
  - a. Add second Simulink model into **Lab2** of the given system by using the Simulink Blocks shown in Figure 2.
  - b. Add a set of commands to the m-file script to simulate the response of the Simulink model. Please plot the input and outputs as a function of time.



Figure 2 Simulink blocks to be used in demo problem. Blocks **Clock** and **Sine** can be found in folder **Sources**, block **To Workspace** in folder **Sinks**, and block **Transfer Fcn** in folder **Continuous**.

(f) Response to the Ramp Inputs:

- Add third Simulink model into **Lab2** of the given system by using the Simulink Blocks shown in Figure 3.
- Add a set of commands to the m-file script to simulate the response of the Simulink model. Please plot the input and outputs as a function of time.

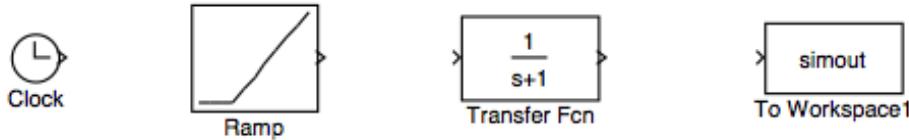


Figure 3 Simulink blocks to be used in demo problem. Blocks **Clock** and **Ramp** can be found in folder **Sources**, block **To Workspace** in folder **Sinks**, and block **Transfer Fcn** in folder **Continuous**.

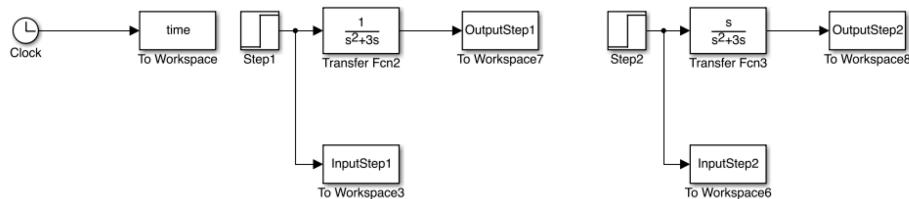
**Individual Problem 1** Consider the following state-space representation of a system:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -2 & 4 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ -4 \end{bmatrix} u \\ y &= [3 \quad 10] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 2u \end{aligned} \tag{2}$$

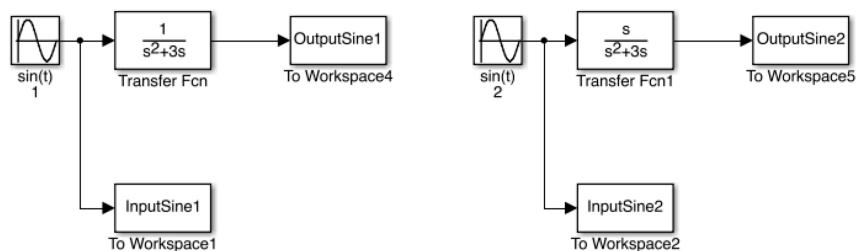
- Please build up state-space representation of this system in MATLAB.
- Use MATLAB function to convert state space into transfer function.
- Build the Simulink model with transfer function in previous step.
- Plot the output response of the system with respect to a sine input which is defined as  $u(t) = \sin(100t)$ .

### Solution to Demo Problem 1:

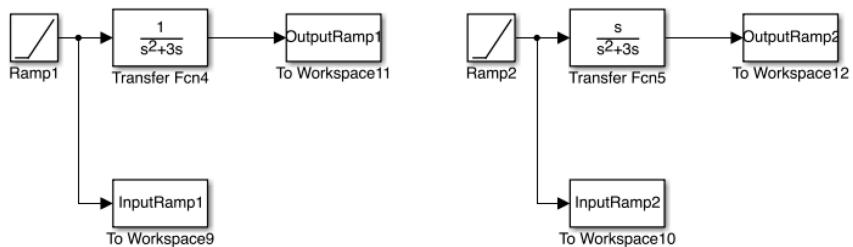
- (a) This lab is based on MATLAB implementation.
  - i. Define parameters
  - ii. Define A B C D matrices
- (b) Convert state space to transfer function
  - i. **[num,den] = ss2tf(A,B,C,D,1);**  
num is to return the coefficients of numerator.  
den is to return the coefficients of denominator.  
“1” inside ss2tf means conversion with respect to 1<sup>st</sup> input.
- (c) Convert transfer function into state space matrices.
  - i. **[AA, BB, CC, DD] = tf2ss(num , den);**  
Please note the order of state variables may be different.
  - ii. **tf1 = tf(num(1,:),den);**  
Transfer function corresponding to first output.
- (d) Build up Simulink model with Step input.



- (e) Repeat building up Simulink model with Sine input.



- (f) Repeat building up Simulink model with Ramp input.



(g) Output figures from Simulink results.

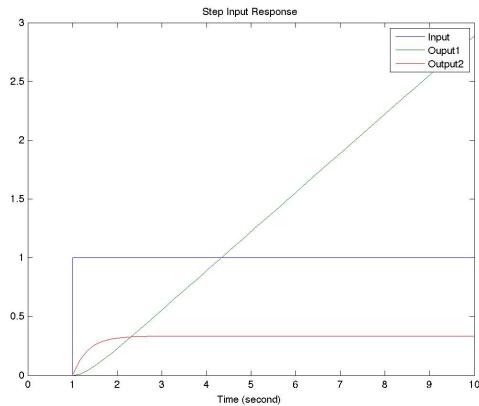


Figure 4 Step Input Responses of the system

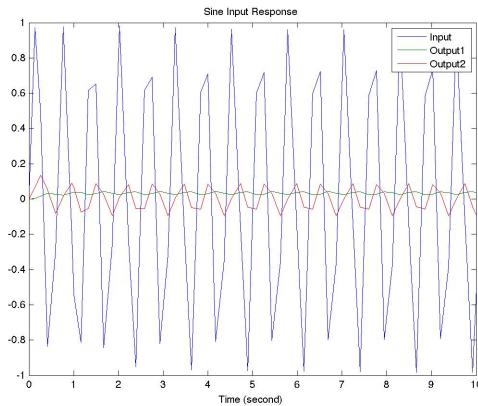


Figure 5 Sine Input Responses of the system

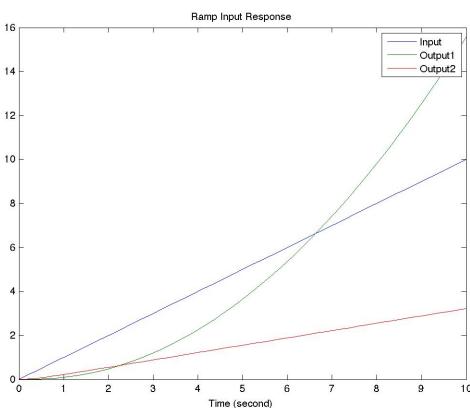


Figure 6 Ramp Input Responses of the system

### Solution to Individual Problem 1:

- (a) Since the example code is structured, we can simply re-type A, B, C and D matrices in specific section as below:

```
%% Define A, B, C, D matrices after linearization
A = [-2 4 ; 0 5];
B = [2;-4];
C = [3 10];
D = -2;
```

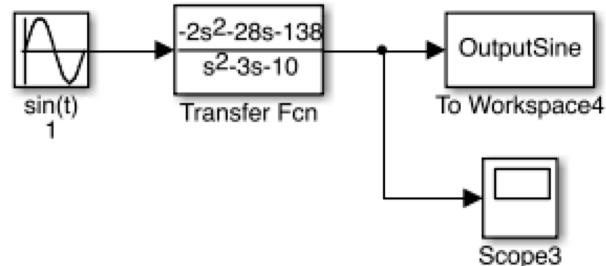
- (b) Convert state space to transfer function

- [num,den] = ss2tf(A,B,C,D);**  
num is to return the coefficients of numerator.

dem is to return the coefficients of denominator.

"1" inside ss2tf means conversion with respect to 1<sup>st</sup> input.

(c) Simulink model is demonstrated as below:



(d) The output from the scope is attached as below.



Figure 7 Sine Input Responses of the system.

**MATLAB Code for Demo Problem:**

```
%% EE547 Lab 2: State Space and Transfer Function
%% Developer : HRLin
%%
close all;
clear all;

%% Define parameters
xini = [0.05; 0.05];
tspan = 0:0.01:100;

%% Define A, B, C, D matrices of Lab 1 individual problem.
% Note: Remember we used state-space in lab1 ?
A = [0 1 ; 0 -3];
B = [0 ; 1];
C = [1 0 ; 0 1];      % eye(2)
D = [0 ; 0];           % zeros(2,1)

%% Define input
%u = [cos(tspan),sin(tspan)];

%% Convert state space to transfer function.
[num,den] = ss2tf(A,B,C,D,1);
% note: input is 1x1 vector; output is 2x1 vector.

%% Convert transfer function to state space.
[AA,BB,CC,DD] = tf2ss(num , den);           % reformed A B C D matrices.
% note: the reformed state variable order may not be the same as original.

%% Formulation of transfer functions and relative zeros and poles
tf1 = tf(num(1,:),den) % transfer function corresponding to first output.
tf2 = tf(num(2,:),den) % transfer function corresponding to second output.

zero1 = zero(tf1)
zero2 = zero(tf2)

pole1 = pole(tf1)
pole2 = pole(tf2)

%% Simulate Simulink Model
sim('EE547_Lab2Simulink_Winter2015_demo',10);    % 10 defines simulation time

%% Output figures
figure(1)
plot(time,InputStep1,time,OutputStep1,time,OutputStep2);
title('Step Input Response');
xlabel ('Time (second)');
legend('Input','Output1','Output2');
print -djpeg 'StepInputResponse.jpeg'

figure(2)
plot(time,InputSine1,time,OutputSine1,time,OutputSine2);
title('Sine Input Response');
```

```
xlabel ('Time (second)');
legend('Input','Output1','Output2');
print -djpeg 'SineInputResponse.jpeg'

figure(3)
plot(time,InputRamp1,time,OutputRamp1,time,OutputRamp2);
title('Ramp Input Response');
xlabel ('Time (second)');
legend('Input','Output1','Output2');
print -djpeg 'RampInputResponse.jpeg'
```

### MATLAB Code for Individual Problem:

```
%% EE547 Lab 2: State Space and Transfer Function
%% Developer : HRLin
%%
close all;
clear all;

%% Define parameters
xini = [0.05; 0.05];
tspan = 0:0.01:100;

%% Define A, B, C, D matrices of Lab 1 individual problem.
% Note: Remember we used state-space in lab1 ?
A = [-2 4 ; 0 5];
B = [2 ; -4];
C = [3 10];      % eye(2)
D = [-2];         % zeros(2,1)

%% Define input
%u = [cos(tspan),sin(tspan)];

%% Convert state space to transfer function.
[num,den] = ss2tf(A,B,C,D,1);
% note: input is 1x1 vector; output is 2x1 vector.

%% Convert transfer function to state space.
[AA,BB,CC,DD] = tf2ss(num , den);           % reformed A B C D matrices.
% note: the reformed state variable order may not be the same as original.

%% Formulation of transfer functions and relative zeros and poles
tf1 = tf(num(1,:),den) % transfer function corresponding to first output.

zero1 = zero(tf1)
pole1 = pole(tf1)

%% Simulate Simulink Model
sim('EE547_Lab2Simulink_Winter2015_IND',10);    % 10 defines simulation time

%% Output figures
% figure(1)
% plot(time,InputStep1,time,OutputStep1);
% title('Step Input Response');
% xlabel ('Time (second)');
% legend('Input','Output1','Output2');
% print -djpeg 'StepInputResponse.jpeg'

figure(2)
plot(time,InputSine1,time,OutputSine1);
title('Sine Input Response');
xlabel ('Time (second)');
legend('Input','Output1','Output2');
print -djpeg 'SineInputResponse.jpeg'

% figure(3)
% plot(time,InputRamp1,time,OutputRamp1);
```

```
% title('Ramp Input Response');
% xlabel ('Time (second)');
% legend('Input','Output1','Output2');
% print -djpeg 'RampInputResponse.jpeg'
```