EE 547 (PMP) Lab 1

Wednesday, January 7, 2015

Professor Linda Bushnell, UW EE

Demo Problem 1 (Linearization around an equilibrium point)

(a) Consider a system of nonlinear differential equations:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2^2 + x_2 \cos x_1 \\ (1 + x_1)x_1 + (1 + x_2)x_2 + x_1 \sin x_2 \end{bmatrix} \tag{1}$$

Given equilibrium point $x^{eq} = \begin{bmatrix} x_1^{eq} \\ x_2^{eq} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, please linearize (1) and find the Jacobian matrix:

$$A = \left(\frac{\partial f(x)}{\partial x}\right)\Big|_{x=x^{eq}} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} \end{bmatrix}\Bigg|_{x=x^{eq}}$$

(b) Now a linearized system is available around equilibrium point. Let B and D be zero matrices and output y = x, please use MATLAB functions to present system responses with respect to different inputs, for example, impulse and step inputs.

Individual Problem 1 (Linearization around an equilibrium point)

(a) Consider a system of nonlinear differential equations:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \end{bmatrix} = \begin{bmatrix} x_2 \\ -3x_2 - x_2^3 + (1 + x_1 + x_1^2)u \end{bmatrix} \tag{2}$$

Given equilibrium point $x^{eq} = u^{eq} = 0$, please linearize (2) and find the Jacobian matrices:

$$A = \left(\frac{\partial f(x,u)}{\partial x}\right)\Big|_{x=x^{eq},u=u^{eq}} = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial x_1} & \frac{\partial f_1(x,u)}{\partial x_2} \\ \frac{\partial f_2(x,u)}{\partial x_1} & \frac{\partial f_2(x,u)}{\partial x_2} \end{bmatrix}\Bigg|_{x=x^{eq},u=u^{eq}}$$

$$B = \left(\frac{\partial f(x,u)}{\partial u}\right)\Big|_{x=x^{eq},u=u^{eq}} = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial u} \\ \frac{\partial f_2(x,u)}{\partial u} \end{bmatrix}\Bigg|_{x=x^{eq},u=u^{eq}}$$

(b) Let D be a zero matrix and output y = x. Use MATLAB function to evaluate the outputs of system with respect to step and impulse response.

Solution to Demo Problem 1:

(a) To find Jacobian matrix A, we have to find the partial derivative and calculate them at the equilibrium point:

$$\frac{\partial f_1(x)}{\partial x_1} = 2x_1 + x_2 \sin x_1 |_{x_1=x_2=0} = 0$$
$$\frac{\partial f_1(x)}{\partial x_2} = 2x_2 + \cos x_1 |_{x_1=x_2=0} = 1$$
$$\frac{\partial f_2(x)}{\partial x_1} = 1 + 2x_1 + \sin x_2 |_{x_1=x_2=0} = 1$$
$$\frac{\partial f_2(x)}{\partial x_2} = 1 + 2x_2 + x_1 \cos x_2 |_{x_1=x_2=0} = 1$$

The matrix A is therefore derived and the linearized system can be written as below.

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(b) The code working flow is listed as below:

i. Define parameters

ii. Define A B C D matrices

   1. Assuming the outputs are the same as state variables, then matrix C is an identity matrix.

iii. Define inputs

iv. Apply state space modeling functions

   1. **ss1 = ss(A,B,C,D)**
      To build *ss class* ss1 in MATLAB.
   2. **ZeroInputResponse = initial(ss1,xini,tspan)**
      Evaluate unforced response of ss1 with given initial condition and time span.
   3. **[ImpulseInputResponse,t1] = impulse(ss1)**
      Evaluate impulse response of ss1 and return corresponding evaluation time.
   4. **[StepInputResponse,t2] = step(ss1)**
      Evaluate tep response of ss1 and return corresponding evaluation time.

v. Output data or figures

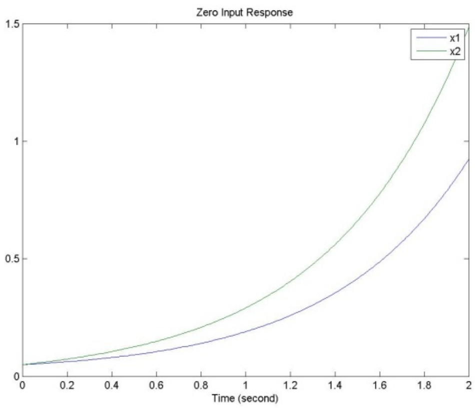

Figure 1 Unforced response of the system

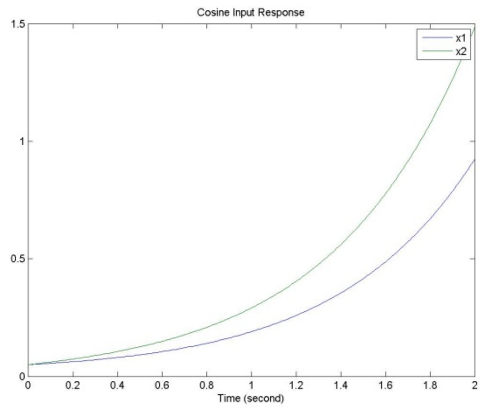

Figure 2 Response of the system with cosine input

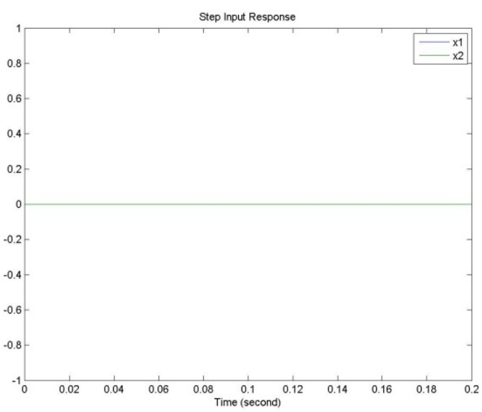

Figure 3 Response of the system with step input

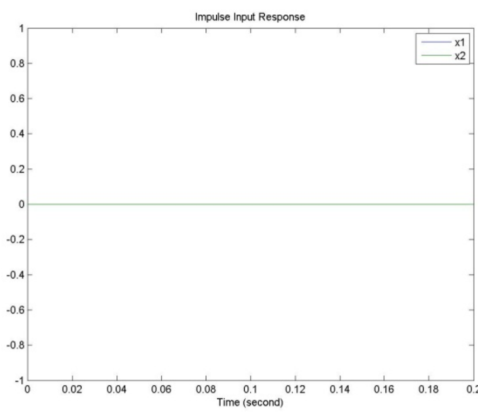

Figure 4 Response of the system with impulse input

Solution to Individual Problem:

(a) To find Jacobian matrices A and B, we have to find the partial derivative and calculate them at the equilibrium point:

$$\frac{\partial f_1(x)}{\partial x_1} = 0$$
$$\frac{\partial f_1(x)}{\partial x_2} = 1$$
$$\frac{\partial f_2(x)}{\partial x_1} = u|_{x_1=x_2=u=0} = 0$$
$$\frac{\partial f_2(x)}{\partial x_2} = -3-3x_2^2|_{x_1=x_2=0} = -3$$
$$\frac{\partial f_1(x)}{\partial u} = 0$$
$$\frac{\partial f_2(x)}{\partial u} = 1 + x_1 + x_1^2|_{x_1=x_2=0} = 1$$

The matrices A and B are therefore derived and the linearized system can be written as below.

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

(b) The code working flow is listed as below:

i. Define parameters

ii. Define A B C D matrices

   1. Assuming the outputs are the same as state variables, then matrix C is therefore an identity matrix.

iii. Define inputs

iv. Apply state space modeling functions

   1. **ss1 = ss(A,B,C,D)**
      To build *ss class* ss1 in MATLAB.
   2. **[StepInputResponse,t2] = step(ss1)**
      Evaluate tep response of ss1 and return corresponding evaluation time.
   3. **[ImpulseInputResponse,t1] = impulse(ss1)**
      Evaluate impulse response of ss1 and return corresponding evaluation ti

v. Output data or figures

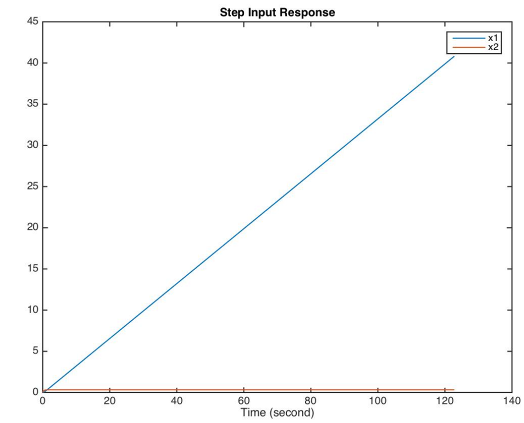

Figure 5 Response of the system with step input

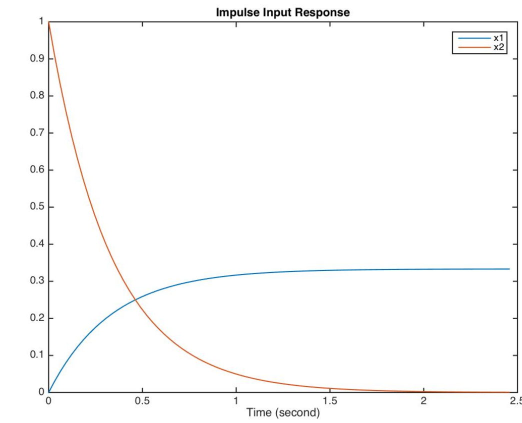

Figure 6 Response of the system with impulse input

MATLAB Code:

Demo

```
%% EE547 Linear Systems Theory Lab 1
%% Developed by HRLin

%% Clean variables and close figures.
close all;
clear all;

%% Define parameters
tspan = 0:0.01:5;
xini = [0.05;0.05];

%% Define A, B, C, D matrices after linearization
A = [0 1 ; 1 1];
B = [0; 0];
C = eye(2);
D = [0 ; 0];

%% Define input
u = cos(tspan);

%% StateSpace Modeling

% Zero input response
ss1 = ss(A,B,C,D);
ZeroInputResponse = initial(ss1,xini,tspan);

% Impulse and step responses
[ImpulseInputResponse,t1] = impulse(ss1);
[StepInputResponse,t2] = step(ss1);

% Response with cosine input
LinearResponse = lsim(ss1,u,tspan,xini);

%% Output figures
figure(1)
plot(tspan, ZeroInputResponse);
title('Zero Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'ZeroInputResponse.jpeg'

figure(2)
plot(t1, ImpulseInputResponse);
title('Impulse Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'ImpulseInputResponse.jpeg'

figure(3)
plot(t2, StepInputResponse);
title('Step Input Response');
xlabel ('Time (second)');
legend('x1','x2');
```

```
print -djpeg 'StepInputResponse.jpeg'

figure(4)
plot(tspan, LinearResponse);
title('Cosine Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'LinearResponse.jpeg'
```

Individual

```
%% EE547 Linear Systems Theory Lab 1
%% Developed by HRLin

%% Clean variables and close figures.
close all;
clear all;

%% Define parameters
tspan = 0:0.01:5;
xini = [0.05;0.05];

%% Define A, B, C, D matrices after linearization
A = [0 1 ; 0 -3];
B = [0; 1];
C = eye(2);
D = [0 ; 0];

%% Define input
u = cos(tspan);

%% StateSpace Modeling

% Zero input response
ss1 = ss(A,B,C,D);
ZeroInputResponse = initial(ss1,xini,tspan);

% Impulse and step responses
[ImpulseInputResponse,t1] = impulse(ss1);
[StepInputResponse,t2] = step(ss1);

% Response with cosine input
LinearResponse = lsim(ss1,u,tspan,xini);

%% Output figures
figure(1)
plot(tspan, ZeroInputResponse);
title('Zero Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'ZeroInputResponse.jpeg'

figure(2)
plot(t1, ImpulseInputResponse);
title('Impulse Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'ImpulseInputResponse.jpeg'

figure(3)
plot(t2, StepInputResponse);
title('Step Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'StepInputResponse.jpeg'
```

```
figure(4)
plot(tspan, LinearResponse);
title('Cosine Input Response');
xlabel ('Time (second)');
legend('x1','x2');
print -djpeg 'LinearResponse.jpeg'
```