Solutions for ME31003 Lab Session 2

ZHANG Bin

November 18, 2023

Notice: You can download the codes for the solutions at `https://github.com/zbC137/ME31002_Lab_Session2/tree/master/solutions`.

Question 1: With the initial conditions $x(0) = 0$ and $\dot{x} = 0$, we obtain the Laplace transformation of the dynamic system:

$$ms^2X(s) + 4sX(s) + 8X(s) = 3U(s). \quad (1)$$

For $m = 2$, we can obtain the following transfer function of the system

$$G(s) = \frac{X(s)}{U(s)} = \frac{3}{2s^2 + 4s + 8}. \quad (2)$$

Now, we create a new file named "question1.m" and compute the partial fraction decomposition in Octave with the following code:

```
clc; clear;
close all;

% comment the next line if using Matlab
pkg load control;

% show the Laplace transformation
s = tf('s');
sys = 3/(2*s^2+4*s+8)

% compute the partial fraction decomposition
num = [3];
den = [2, 4, 8];
[r, p, k] = residue(num, den)
```

The results shown in the command prompt are:

```
Transfer function 'sys' from input 'u1' to output ...

              3
 y1:  ---------------
      2 s^2 + 4 s + 8
```

```

Continuous-time model.
r =

  -0.0000 - 0.4330i
  -0.0000 + 0.4330i

p =

  -1.0000 + 1.7321i
  -1.0000 - 1.7321i

k = [](0x0)
```

Therefore, we can obtain the partial fraction decomposition as:

$$X(s) = \frac{-0.4330\mathrm{i}}{s+1-1.7321\mathrm{i}} + \frac{0.4330\mathrm{i}}{s+1+1.7321\mathrm{i}}. \tag{3}$$

Question 2: Create a new file named"question2.m" in Octave and copy the following codes in it.

```
clc; clear;
close all;

% comment the next line if using Matlab
pkg load control;

% compute the transfer function
s = tf('s');

sys1 = 2/s;
sys2 = 2/(s+4);
sys4 = series(sys1, sys2);
H1 = 1;
sys5 = feedback(sys4, H1, -1);

G1 = 0.75;
sys3 = 1/s;
G2 = series(G1, sys5);
sys6 = series(G2, sys3);

H2 = 1;
G = feedback(sys6, H2, -1)

% compute the poles
pole(G)

```

```
% poles-zeros map
figure(1)
pzmap(G);

% root locus
figure(2)
rlocus(G)
```

Run the file. The results shown in the command prompt are

```
Transfer function 'G' from input 'u1' to output ...

                 3
 y1:  ---------------------
      s^3 + 4 s^2 + 4 s + 3

Continuous-time model.
ans =

  -3.0000 +      0i
  -0.5000 + 0.8660i
  -0.5000 - 0.8660i
```

Therefore, we obtain that the transfer function of the system is

$$G(s) = \frac{3}{s^3 + 4s^2 + 4s + 3}. \tag{4}$$

The poles-zeros map and the root locus of the system are shown in Figure 1.

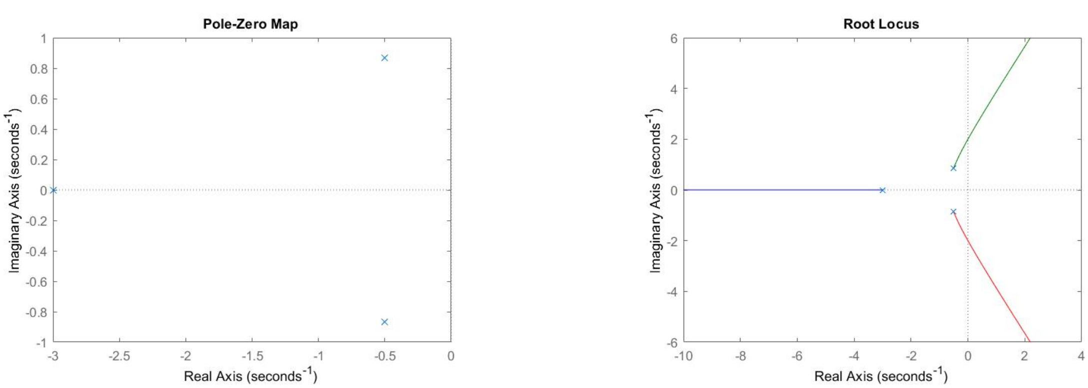

Figure 1: Simulation results.

Question 3: Create a new file named "question3.m" in Octave and copy the following codes in it.

```
clc; clear;
close all;

% comment the next line if you are using Matlab
pkg load signal;

t = 0:0.001:20;
%% transfer function
s = tf('s');
G = 5*(s+20)/(s*(s+4.5883)*(s^2+3.4118*s+16.346));
H = 6;
sys = feedback(G, H, -1)

%% step response
[y1, t] = step(sys, t);

figure(1);
plot(t, y1);
xlabel('t(s)'); ylabel('y');
title('Step Response');

%% impulse response
[y2, t] = impulse(sys, t);

figure(2)
plot(t, y2);
xlabel('t(s)'); ylabel('y');
title('Impulse Response');

%% square wave response
f = square(2*pi*t/5);
[y3, t] = lsim(sys, f, t);

figure(3)
plot(t, f);
xlabel('t(s)'); ylabel('f');
title('Square Wave');

figure(4)
plot(t, y3);
xlabel('t(s)'); ylabel('y');
title('Square Wave Response');
```

Run the file. The results shown in the command prompt are

```
Transfer function 'sys' from input 'u1' to output ...
```

```

                     5 s + 100
 y1:  ----------------------------------
      s^4 + 8 s^3 + 32 s^2 + 105 s + 600

Continuous-time model.
```

Therefore, we can obtain the transfer function

$$\frac{C(s)}{R(s)} = \frac{5s + 100}{s^4 + 8s^3 + 32s^2 + 105s + 600} \tag{5}$$

The response results are shown in Figure 2.

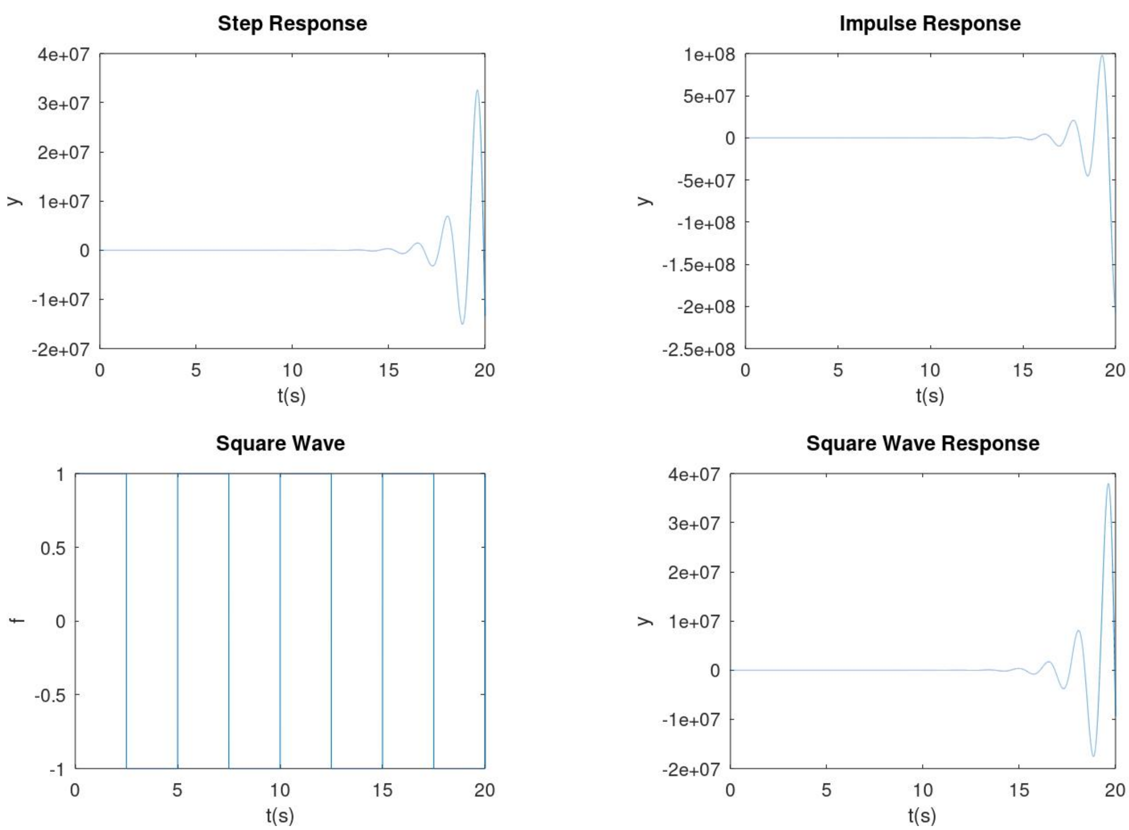

Figure 2: Simulation results.

Question 4: We can derive that the dynamic function of the system is in the following form:

$$m\ddot{x} + (b_1 + b_2)\dot{x} + kx = f \tag{6}$$

Denote $x_1 = x$ and $x_2 = \dot{x}$, we can rewrite the ODE as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{b_1 + b_2}{m}x_2 - \dfrac{k}{m}x_1 + \dfrac{f}{m} \end{cases} . \tag{7}$$

Then, create a new file named “dynamics_q4.m” in Octave and we can represent the ODE as a Octave function by the following codes.

```
function dx = dynamics_q4(t, x, f)

m = 1;
b1 = 3;
b2 = 2;
k = 1;

dx(1,:) = x(2);
dx(2,:) = -(b1+b2)/m*x(2)-k/m*x(1)+f/m;

end
```

Now, create another new file named “question4.m” in Octave and copy the following codes in it to build the simulation.

```
% time span
t = 0:0.001:40;

% initial conditions
x0 = [0, 0];
f = 1.5;

% simulation
options = odeset(’RelTol’, 1e-6, ’AbsTol’, 1e-6);
[t, x] = ode45(@(t, x)dynamics_q4(t, x, f), t, x0, options);

% plotting
figure(1)
plot(t, x(:, 1));
xlabel(’t(s)’); ylabel(’x’);
```

Run the simulation, then we can get the response of x shown in Figure 3.

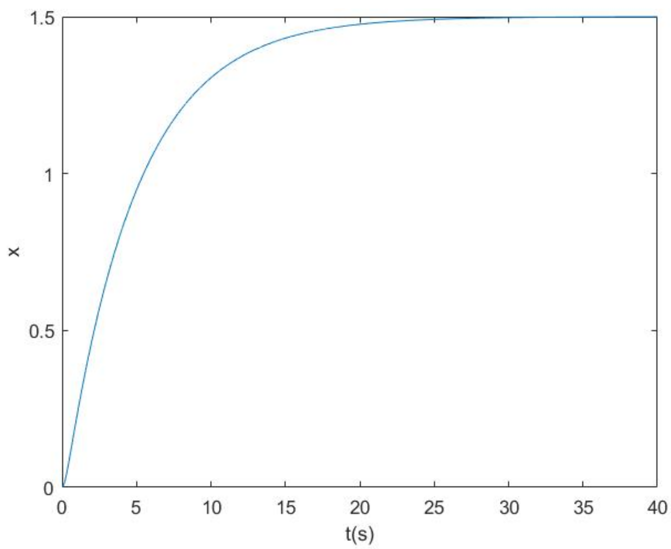

Figure 3: The response of x under a constant force $f = 1$.