Solutions for ME31002 Lab Session 2

ZHANG Bin

April 8, 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:

$$ms^2X(s) + 2sX(s) + 10X(s) = 5U(s). \tag{1}$$

For $m = 4$, we can obtain the following transfer function of the system

$$G(s) = \frac{X(s)}{U(s)} = \frac{5}{4s^2 + 2s + 10}. \tag{2}$$

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 = 5/(4*s^2+2*s+10)

% compute the partial fraction decomposition
num = [5];
den = [4, 2, 10];
[r, p, k] = residue(num, den)
```

The results shown in the command prompt are:

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

               5
 y1:  ----------------
      4 s^2 + 2 s + 10
```

```

Continuous-time model.
r =

    0.0000 - 0.4003i
    0.0000 + 0.4003i

p =

   -0.2500 + 1.5612i
   -0.2500 - 1.5612i

k = [](0x0)
```

Therefore, we can obtain the partial fraction decomposition as:

$$X(s) = \frac{-0.4003\mathrm{i}}{s + 0.25 - 1.5612\mathrm{i}} + \frac{0.4003\mathrm{i}}{s + 0.25 + 1.5612\mathrm{i}}. \tag{3}$$

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');

G1 = 2/((s+1)*(s+8));
H1 = 0.2;
sys1 = feedback(G1, H1, -1);

G2 = 4;
sys2 = series(G2, sys1);

G3 = 1/s;
sys3 = series(sys2, G3);

H2 = 1;
G = feedback(sys3, 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 ...

                   8
 y1:  -----------------------
      s^3 + 9 s^2 + 8.4 s + 8

Continuous-time model.
ans =

  -8.0833 + 0.0000i
  -0.4584 + 0.8829i
  -0.4584 - 0.8829i
```

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

$$G(s) = \frac{8}{s^3 + 9s^2 + 8.4s + 8}. \tag{4}$$

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

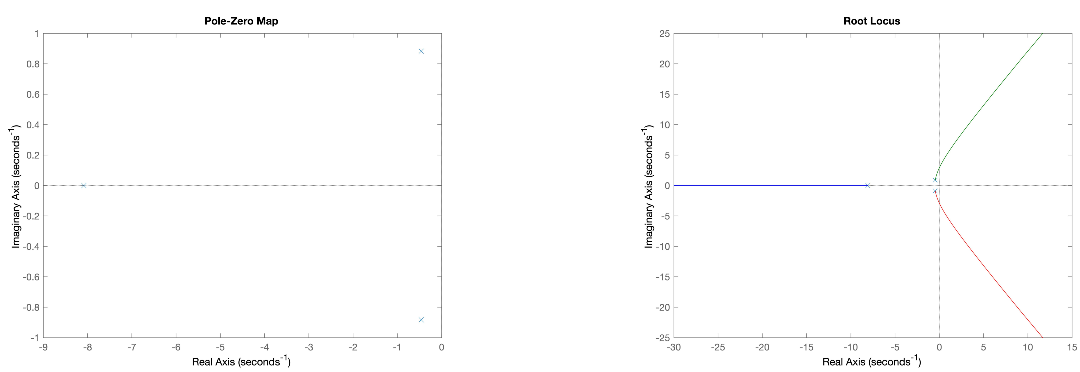

Figure 1: Simulation results.

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 = 8*(s+2)/((s+3.3)*(2*s+7));

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

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

%% impulse response
[y2, t] = impulse(G, 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(G, 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, then you can get the results shown in Figure 2.

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

$$m\ddot{x} + (b_1 + b_2)\dot{x} + kx = f \quad (5)$$

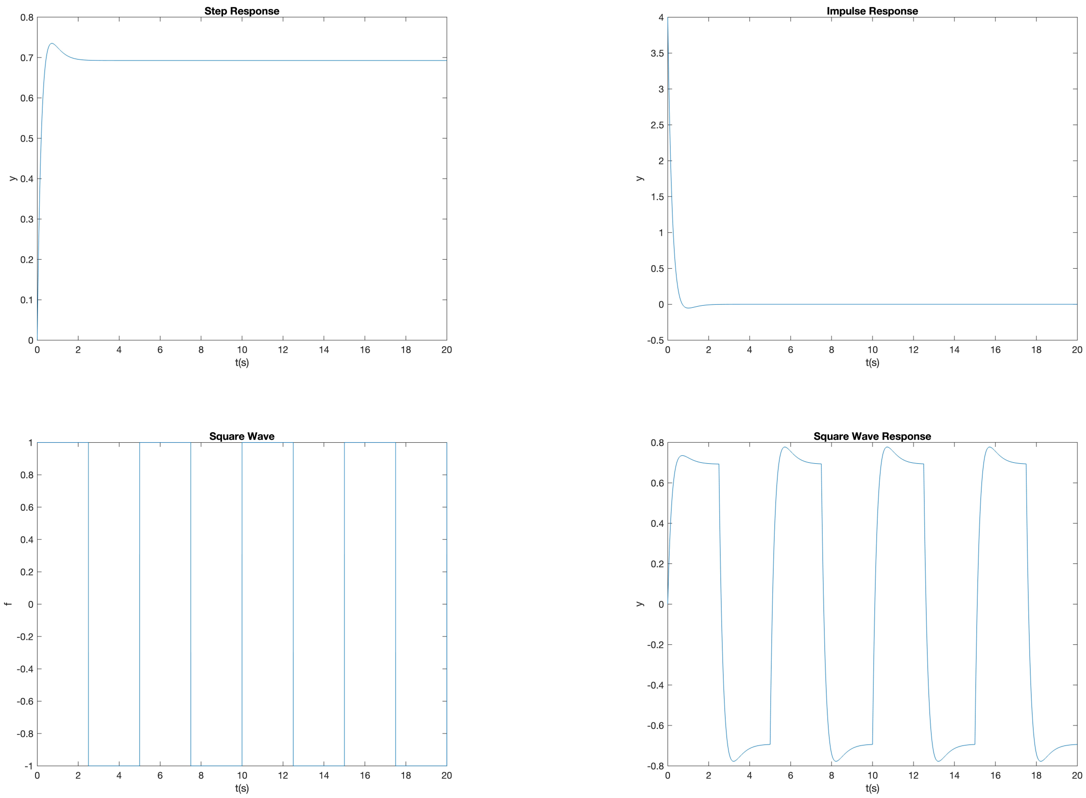

Figure 2: Simulation results.

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

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{b_1 + b_2}{m} x_2 - \dfrac{k}{m} x_1 + \dfrac{f}{m} \end{cases} . \tag{6}$$

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
```

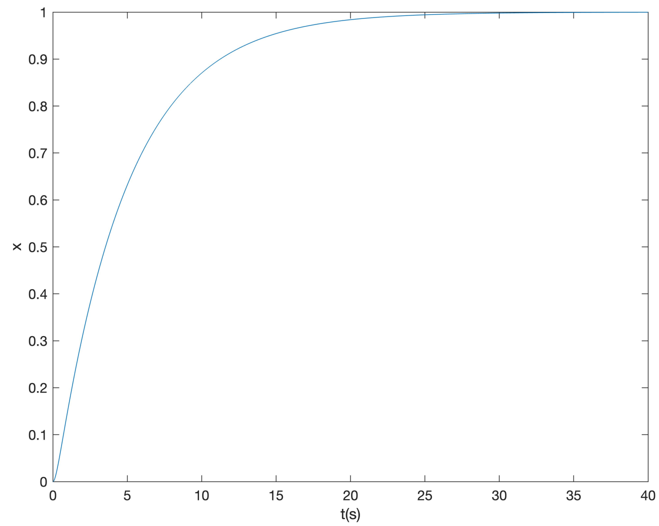

Figure 3: The response of x under a constant force $f = 1$.

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;

% 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.

Question 5: We design a P controller as

$$u = 0.5e \tag{7}$$

and use the transfer function method to do the simulation. Note that you can change the gain of the controller as long as the tracking error converges to zero.

Create a file named "question5.m" in Octave and copy the following codes in it to build the simulation.

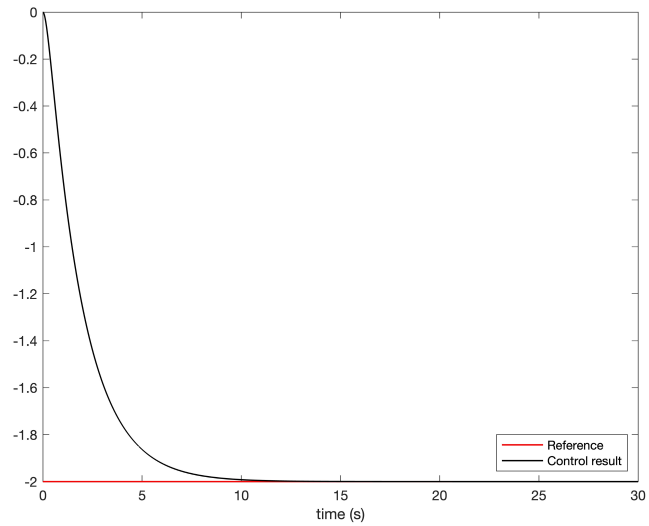

Figure 4: The response of x under a proportional controller.

```
close all;
clear;

% System function
G = tf(5,[1, 5, 0]);
% PID controller
G_c = tf([0, 0.5, 0.0], [1, 0]);

sys1 = series(G, G_c);
% Closed-loop system
sys2 = feedback(sys1, 1)

t = 0:0.01:30;
r = ones(size(t))*(-2);
[y, t] = lsim(sys2, r, t);

plot(t, r, 'Color', 'r', 'LineWidth', 1)
hold on;
plot(t, y, 'Color', 'k', 'LineWidth', 1)
xlabel('time (s)')
legend('Reference', 'Control result', 'Location', 'southeast')
```

Run the simulation, then we can get the response of x shown in Figure 4.