Solutions for ME567 Lab Session 1

ZHANG Bin

September 21, 2022

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:

$$s^2X(s) + 2sX(s) + 10X(s) = \frac{2}{s^3}. \tag{1}$$

Therefore, we have

$$X(s) = \frac{2}{s^5 + 2s^4 + 10s^3}. \tag{2}$$

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 = 2/(s^3*(s^2+2*s+10))

% compute the partial fraction decomposition
num = [2];
den = [1, 2, 10, 0, 0, 0];
[r, p, k] = residue(num, den)
```

The results shown in the command prompt are:

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

                 2
 y1:  --------------------
      s^5 + 2 s^4 + 10 s^3

```

```
Continuous-time model.
r =

   0.0060 - 0.0087i
   0.0060 + 0.0087i
  -0.0120 +      0i
  -0.0400 +      0i
   0.2000 +      0i

p =

  -1.0000 + 3.0000i
  -1.0000 - 3.0000i
        0 +      0i
        0 +      0i
        0 +      0i

k = [](0x0)
```

Therefore, we can obtain the partial fraction decomposition as:

$$X(s) = \frac{0.006 - 0.0087\mathrm{i}}{s+1-3\mathrm{i}} + \frac{0.006 + 0.0087\mathrm{i}}{s+1+3\mathrm{i}} + \frac{-0.012}{s} + \frac{-0.04}{s^2} + \frac{0.2}{s^3}. \tag{3}$$

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 = 1/s^2;

G2 = 50/(s+1);
H1 = 2/s;
sys1 = feedback(G2, H1, -1);
sys2 = series(G1, sys1);

sys3 = s-2;

sys4 = series(sys2, sys3);
H2 = 1;
G = feedback(sys4, H2, -1)
```

```
% compute the poles
pole(G)
```

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

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

              50 s^2 - 100 s
 y1:  ---------------------------
       s^4 + s^3 + 150 s^2 - 100 s

Continuous-time model.
ans =

   -0.8309 + 12.2642i
   -0.8309 - 12.2642i
    0.6618 +       0i
         0 +       0i
```

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

$$G(s) = \frac{50s - 100}{s^3 + s^2 + 150s - 100}. \tag{4}$$

Since there is a positive pole for the transfer function, the system is unstable.

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 = 16/(s^2+3*s+16);

%% 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/8);
[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 1.

Question 4: Denote $x_1 = x$ and $x_2 = \dot{x}$, we can rewrite the ODE as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{c}{m}x_2 - \dfrac{k}{m}x_1 + \dfrac{f}{m} \end{cases}. \tag{5}$$

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;
c = 2;
k = 1;

dx(1,:) = x(2);
dx(2,:) = -c/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

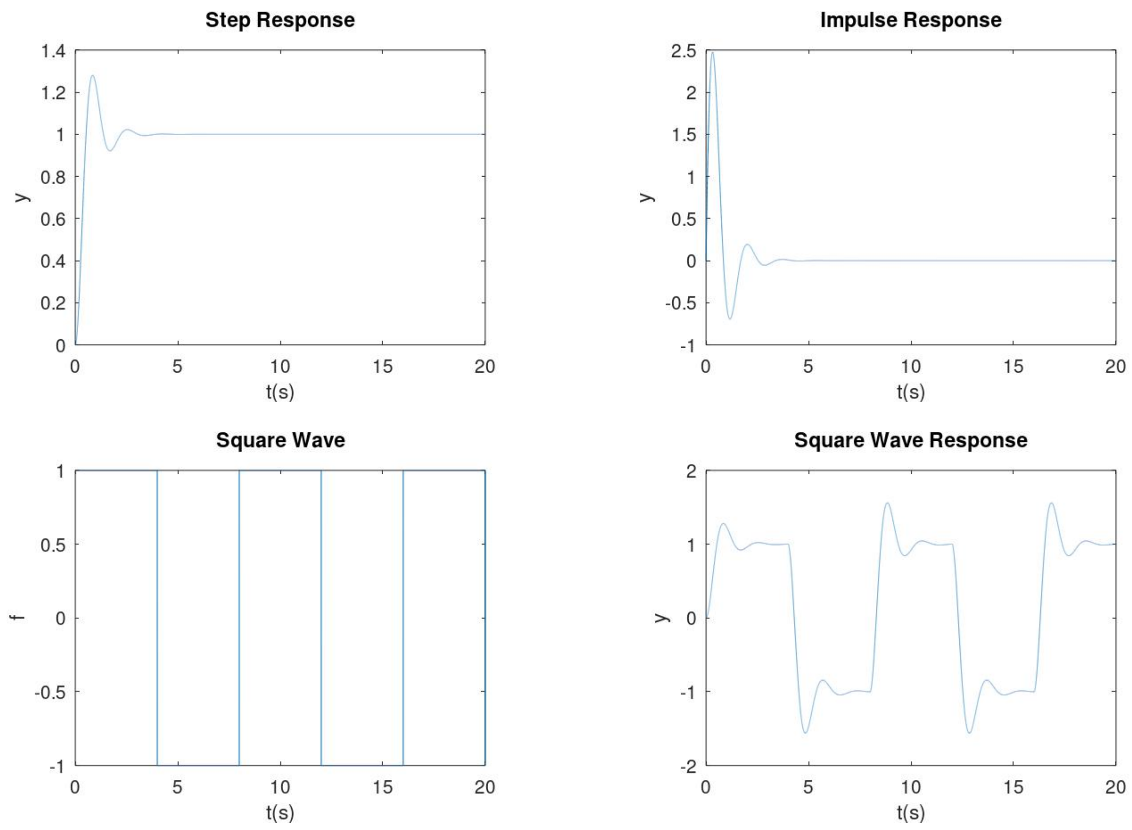

Figure 1: Simulation results.

the simulation.

```
% time span
t = 0:0.001:20;

% 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('Linear velocity');
```

Run the simulation, then we can get the response of x shown in Figure 2.

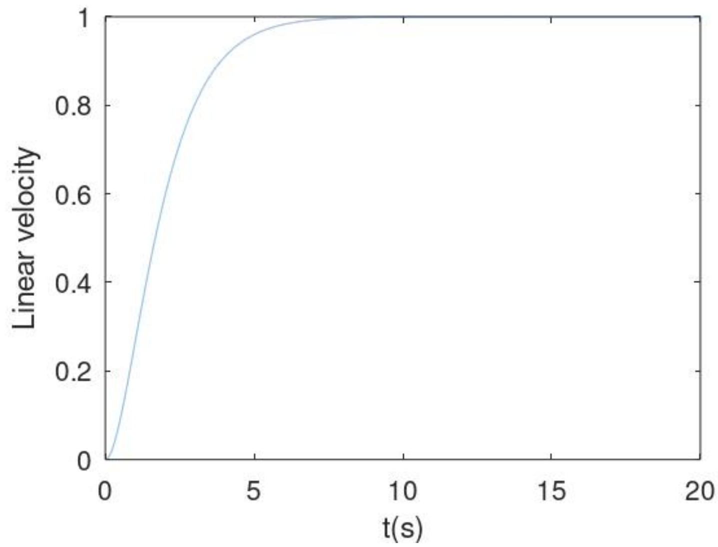

Figure 2: The response of x under a constant force $f = 1$.

Question 5: Create a file named "question5.m" in Octave and copy the following codes in it to build the simulation.

```
% Incremental PID Controller
clear all;
close all;

% comment this line if you use matlab.
pkg load control

ts=0.001;
sys=tf(400,[1,50,0]);
dsys=c2d(sys,ts,'z');
[num,den]=tfdata(dsys,'v');
n = size(num, 2);
num = num(n-1:n)

u_1=0.0;u_2=0.0;u_3=0.0;
y_1=0;y_2=0;y_3=0;

x=[0,0,0]';

error_1=0;
error_2=0;
for k=1:1:1000
   time(k)=k*ts;

   yd(k)=1.0;
   kp=8;
```

```
    ki=0.1;
    kd=10;

    du(k)=kp*x(1)+kd*x(2)+ki*x(3);
    u(k)=u_1+du(k);

    if u(k)>=10
       u(k)=10;
    end
    if u(k)<=-10
       u(k)=-10;
    end
    y(k)=-den(2)*y_1-den(3)*y_2+num(1)*u_1+num(2)*u_2;

    error=yd(k)-y(k);
    u_3=u_2;u_2=u_1;u_1=u(k);
    y_3=y_2;y_2=y_1;y_1=y(k);

    x(1)=error-error_1;               %Calculating P
    x(2)=error-2*error_1+error_2;     %Calculating D
    x(3)=error;                       %Calculating I

    error_2=error_1;
    error_1=error;
end
figure(1);
plot(time,yd,'r',time,y,'k','linewidth',2);
xlabel('time(s)');ylabel('yd,y');
legend('Ideal position signal','Position tracking');
```

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

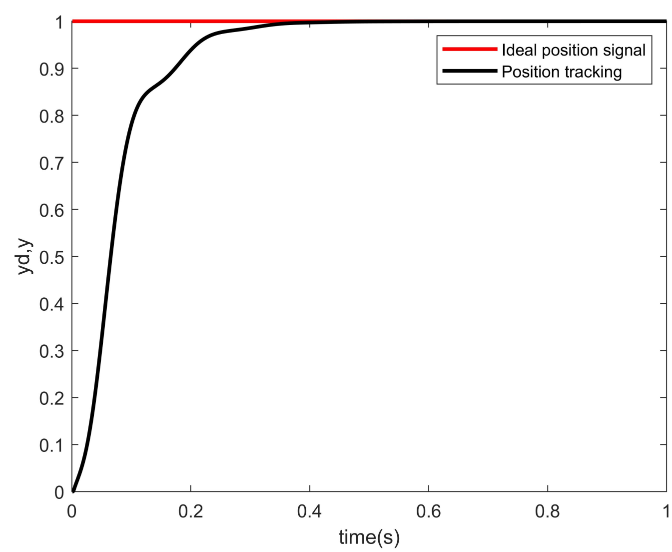

Figure 3: The response of $y(k)$ under a incremental PID controller.