

Solutions for ME567 Lab Session 1

ZHANG Bin

October 29, 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:

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

Therefore, we have

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

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

```
1 clc; clear;
2 close all;
3
4 % comment the next line if using Matlab
5 pkg load control;
6
7 % show the Laplace transformation
8 s = tf('s');
9 sys = 2/(s^3*(s^2+2*s+10))
10
11 % compute the partial fraction decomposition
12 num = [2];
13 den = [1, 2, 10, 0, 0, 0];
14 [r, p, k] = residue(num, den)
```

The results shown in the command prompt are:

```
1 Transfer function 'sys' from input 'u1' to output ...
2
3          2
4 y1:  -----
5      s^5 + 2 s^4 + 10 s^3
6
```

```

7 Continuous-time model.
8 r =
9
10     0.0060 - 0.0087i
11     0.0060 + 0.0087i
12    -0.0120 +      0i
13    -0.0400 +      0i
14     0.2000 +      0i
15
16 p =
17
18    -1.0000 + 3.0000i
19    -1.0000 - 3.0000i
20         0 +      0i
21         0 +      0i
22         0 +      0i
23
24 k = [] (0x0)

```

Therefore, we can obtain the partial fraction decomposition as:

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

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

```

1 clc; clear;
2 close all;
3
4 % comment the next line if using Matlab
5 pkg load control;
6
7 % compute the transfer function
8 s = tf('s');
9
10 G1 = 1/s^2;
11
12 G2 = 50/(s+1);
13 H1 = 2/s;
14 sys1 = feedback(G2, H1, -1);
15 sys2 = series(G1, sys1);
16
17 sys3 = s-2;
18
19 sys4 = series(sys2, sys3);
20 H2 = 1;
21 G = feedback(sys4, H2, -1)

```

```

22
23 % compute the poles
24 pole(G)

```

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

```

1 Transfer function 'G' from input 'u1' to output ...
2
3          50 s^2 - 100 s
4 y1:  -----
5      s^4 + s^3 + 150 s^2 - 100 s
6
7 Continuous-time model.
8 ans =
9
10      -0.8309 + 12.2642i
11      -0.8309 - 12.2642i
12      0.6618 +          0i
13      0 +          0i

```

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

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

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.

```

1 clc; clear;
2 close all;
3
4 % comment the next line if you are using Matlab
5 pkg load signal;
6
7 t = 0:0.001:20;
8 %% transfer function
9 s = tf('s');
10 G = 16/(s^2+3*s+16);
11
12 %% step response
13 [y1, t] = step(G, t);
14
15 figure(1);
16 plot(t, y1);
17 xlabel('t(s)'); ylabel('y');
18 title('Step Response');

```

```

19
20 %% impulse response
21 [y2, t] = impulse(G, t);
22
23 figure(2)
24 plot(t, y2);
25 xlabel('t(s)'); ylabel('y');
26 title('Impulse_Response');
27
28 %% square wave response
29 f = square(2*pi*t/8);
30 [y3, t] = lsim(G, f, t);
31
32 figure(3)
33 plot(t, f);
34 xlabel('t(s)'); ylabel('f');
35 title('Square_Wave');
36
37 figure(4)
38 plot(t, y3);
39 xlabel('t(s)'); ylabel('y');
40 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:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{c}{m}x_2 - \frac{k}{m}x_1 + \frac{f}{m} \end{cases} \quad (5)$$

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.

```

1 function dx = dynamics_q4(t, x, f)
2
3 m = 1;
4 c = 2;
5 k = 1;
6
7 dx(1,:) = x(2);
8 dx(2,:) = -c/m*x(2) - k/m*x(1) + f/m;
9
10 end

```

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

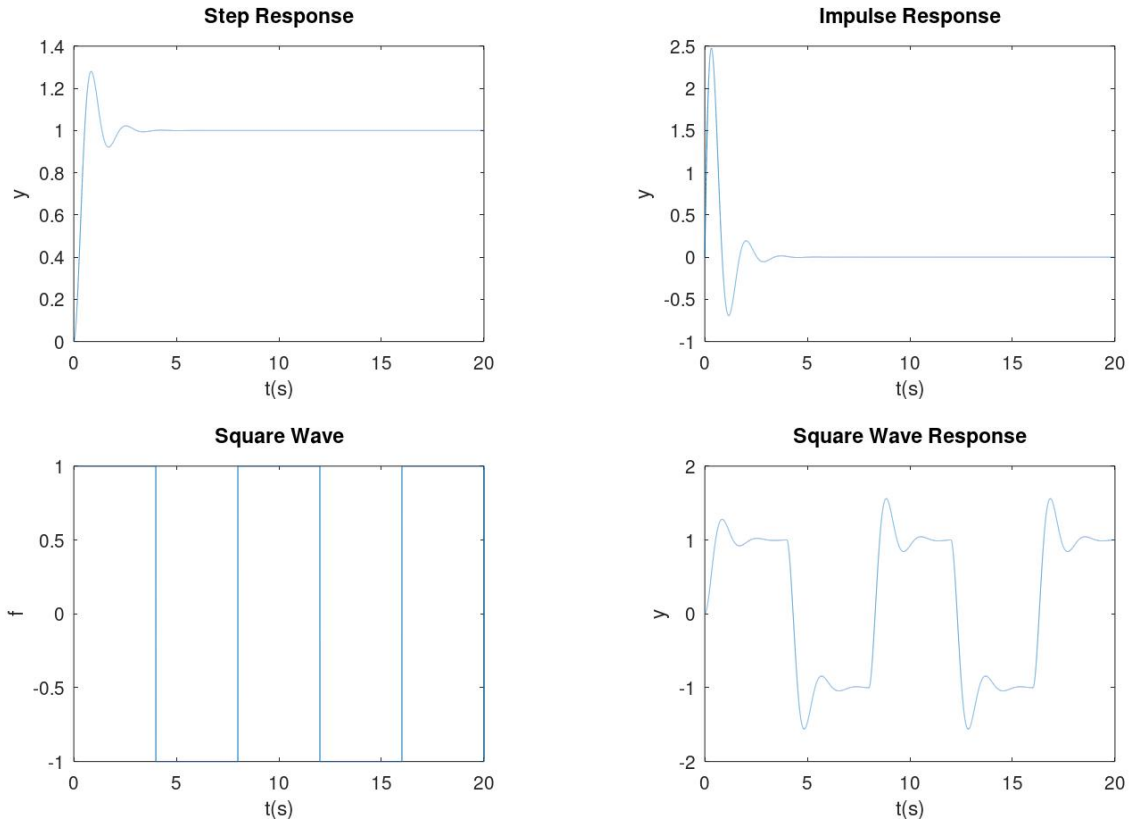


Figure 1: Simulation results.

the simulation.

```

1 % time span
2 t = 0:0.001:20;
3
4 % initial conditions
5 x0 = [0, 0];
6 f = 1;
7
8 % simulation
9 options = odeset('RelTol', 1e-6, 'AbsTol', 1e-6);
10 [t, x] = ode45(@(t, x)dynamics_q4(t, x, f), t, x0, options);
11
12 % plotting
13 figure(1)
14 plot(t, x(:, 1));
15 xlabel('t(s)'); ylabel('Linear velocity');

```

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

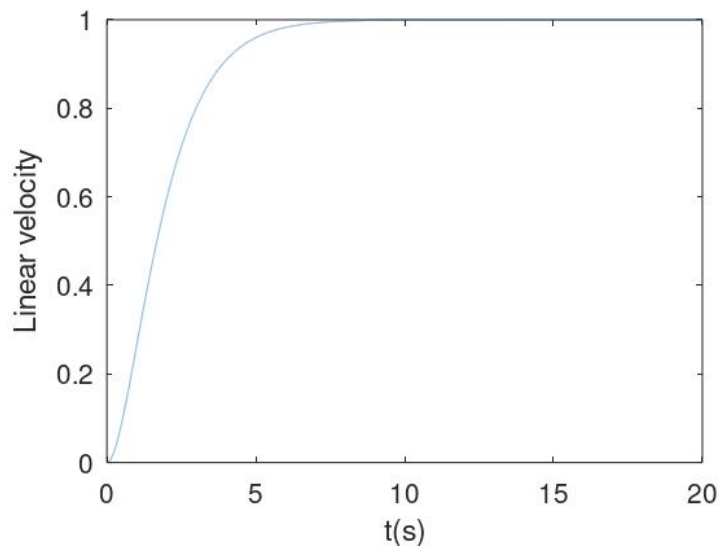


Figure 2: The response of x under a constant force $f = 1$.

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

```

1 % Incremental PID Controller
2 clear all;
3 close all;
4
5 % comment this line if you use matlab.
6 pkg load control
7
8 ts=0.001;
9 num = [0.1967, 0.1935]*1e-3;
10 den = [1, -1.9512, 0.9512];
11
12 u_1=0.0;u_2=0.0;u_3=0.0;
13 y_1=0;y_2=0;y_3=0;
14
15 x=[0,0,0]';
16
17 error_1=0;
18 error_2=0;
19 for k=1:1:1000
20     time(k)=k*ts;
21
22     yd(k)=1.0;
23     kp=8;
24     ki=0.1;
25     kd=10;
26

```

```

27     du(k)=kp*x(1)+kd*x(2)+ki*x(3);
28     u(k)=u_1+du(k);
29
30     if u(k)>=10
31         u(k)=10;
32     end
33     if u(k)<=-10
34         u(k)=-10;
35     end
36     y(k)=-den(2)*y_1-den(3)*y_2+num(1)*u_1+num(2)*u_2;
37
38     error=yd(k)-y(k);
39     u_3=u_2;u_2=u_1;u_1=u(k);
40     y_3=y_2;y_2=y_1;y_1=y(k);
41
42     x(1)=error-error_1;           %Calculating P
43     x(2)=error-2*error_1+error_2; %Calculating D
44     x(3)=error;                  %Calculating I
45
46     error_2=error_1;
47     error_1=error;
48 end
49 figure(1);
50 plot(time,yd,'r',time,y,'k','linewidth',2);
51 xlabel('time(s)');ylabel('yd,y');
52 legend('Ideal_position_signal','Position_tracking');

```

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

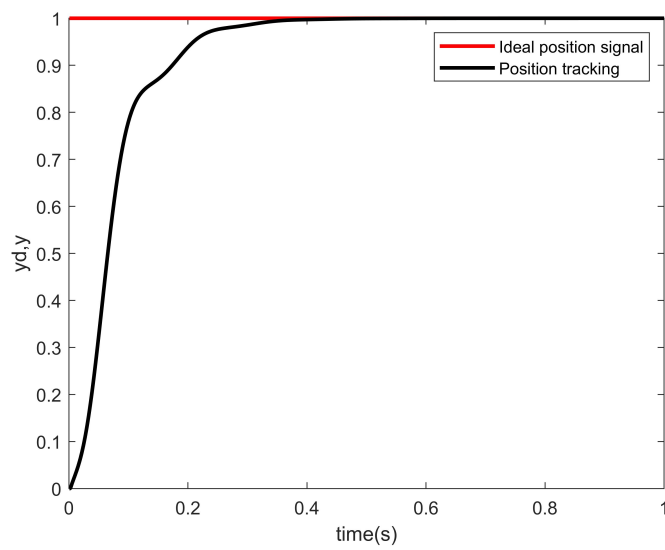


Figure 3: The response of $y(k)$ under a incremental PID controller.