Discrete-time models- Lab 5

Zhu Chenhao

22320630

Consider a linear system of the form

$$\ddot{y} + a_1\dot{y} + a_0 y = b_0 u, \tag{1}$$

where $a_0, a_1, b_0 \in \mathbb{R}$, y - is output of the system and u - is control signal.

0. Initial Data

```
task.parameters
```

```
ans = 包含以下字段的 struct:

     T: 2.3000
    a0: 10
    a1: 20
    b0: 1
```

```
T = 2.3000;
a0 = 10;
a1 = 20;
b0 = 1;
```

1. State space model

```
A = [0 1;-a0 -a1];
B = [0; b0];
C = [1 0];
D = 0;
```

2. Eigenvalues

```
eig_value = eig(A)
```

eig_value = *2×1*

$$\begin{array}{r} -0.5132 \\ -19.4868 \end{array}$$

```
eig_value_continuous_1 = eig_value(2,1);
eig_value_continuous_2 = eig_value(1,1);
```

3. Discrete system

```
sys = ss(A, B, C, D);
Ad =  expm(A*T);
Bd = A \ (Ad - eye(size(A)))*B;
```

4. Eigenvalues of a discrete system

```
sysd = ss(Ad, Bd, C, D, T);
eig_value_d = eig(Ad);
eig_value_discrete_1 = eig_value_d(2,1);
eig_value_discrete_2 = eig_value_d(1,1);
```

5. Stepinfo to find the number of steps

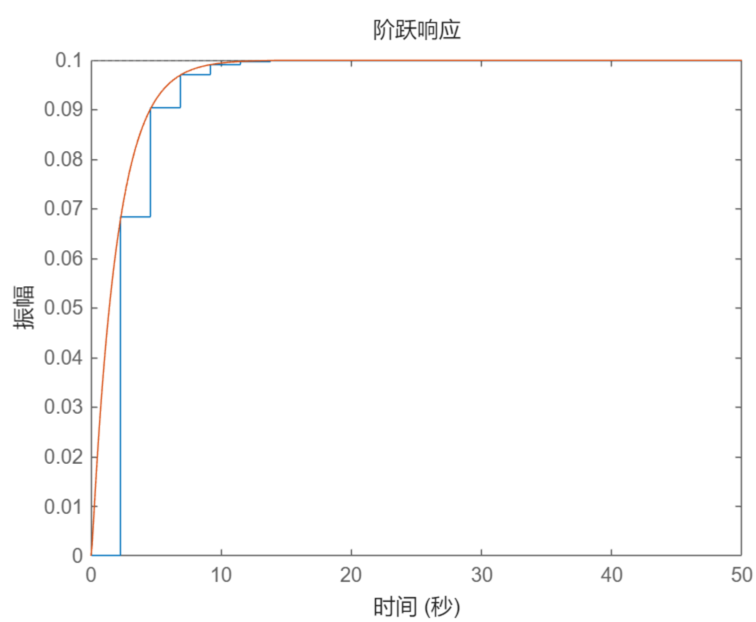
```
step_info = stepinfo(sysd);
disp(step_info);
```

```
         RiseTime: 2.3000
    TransientTime: 9.2000
     SettlingTime: 9.2000
      SettlingMin: 0.0903
      SettlingMax: 0.1000
        Overshoot: 2.2204e-14
       Undershoot: 0
             Peak: 0.1000
         PeakTime: 75.9000
```

```
transient_time = step_info.TransientTime;
num_steps_until_end_of_transient = transient_time / T
```

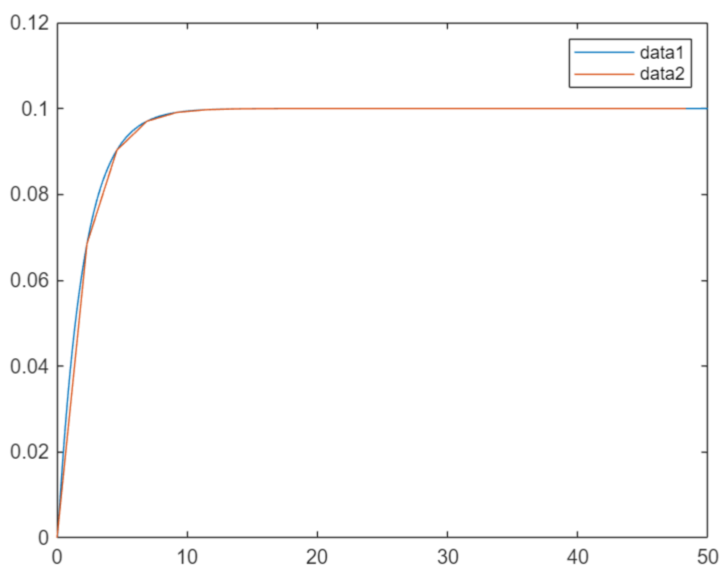
```
num_steps_until_end_of_transient = 4
```

```
step(sysd, sys)
```

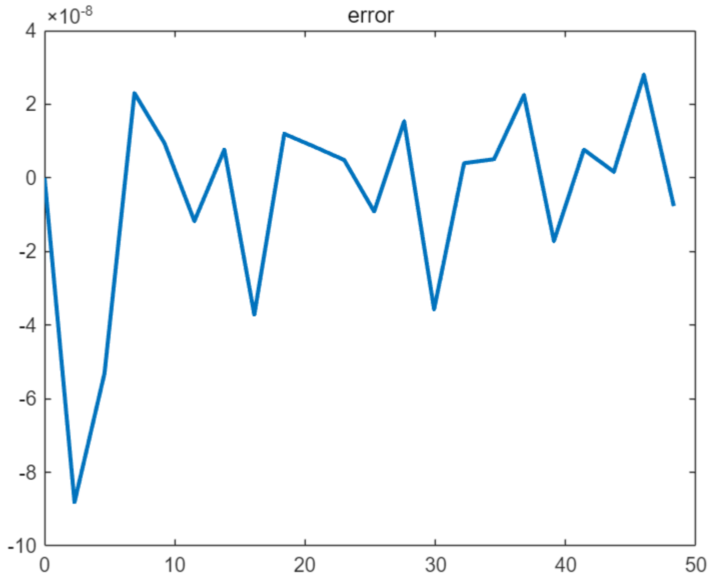


6. Using ode45 and input signal u(t) = 1

```
u = 1;
tspan = [0 50];
x0 = [0; 0];
sol = ode45(@(t,x) odefun(t,x,u,a0,a1,b0),tspan,x0);
plot(sol.x,sol.y(1,:))
hold on;
t_d = 0:T:50;
n = length(t_d);
x_d = zeros(2,n);
x_d(:, 1) = x0;
for k = 1:n-1
    x_d(:, k+1) = Ad*x_d(:, k)+Bd*u;
end
plot(t_d, x_d(1, :))
legend show;
hold off;
```

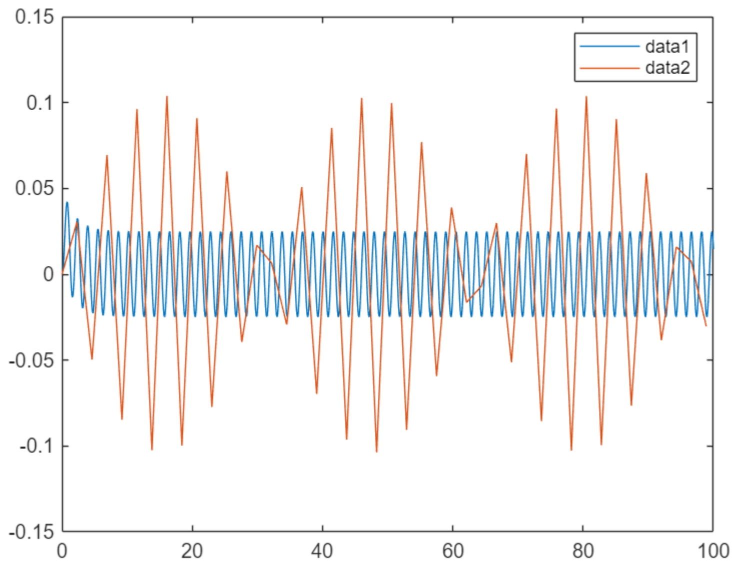



```
y_kt = x_d(1, :);
for k = 1:n
    y_t(:,k) = deval(sol, t_d(k));
end
plot(t_d, y_t(1, :) - y_kt, 'LineWidth', 2);
title("error")
```

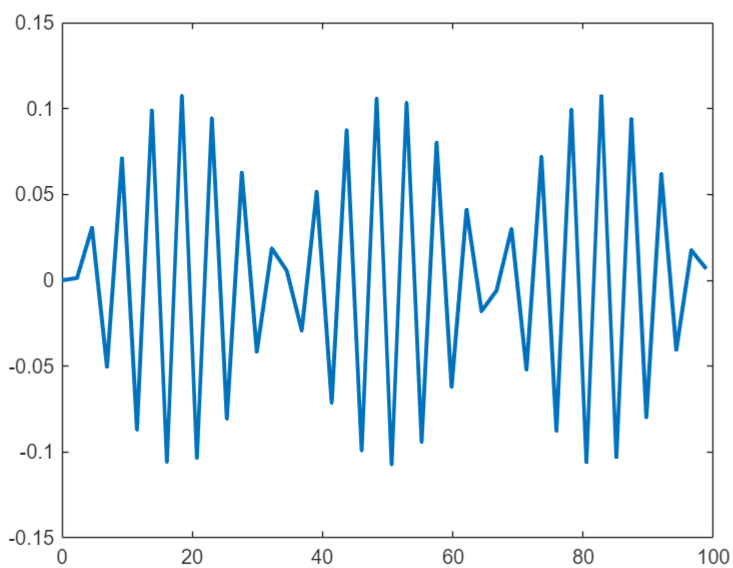


7. Input signal u(t) = 2*sin(4*t)

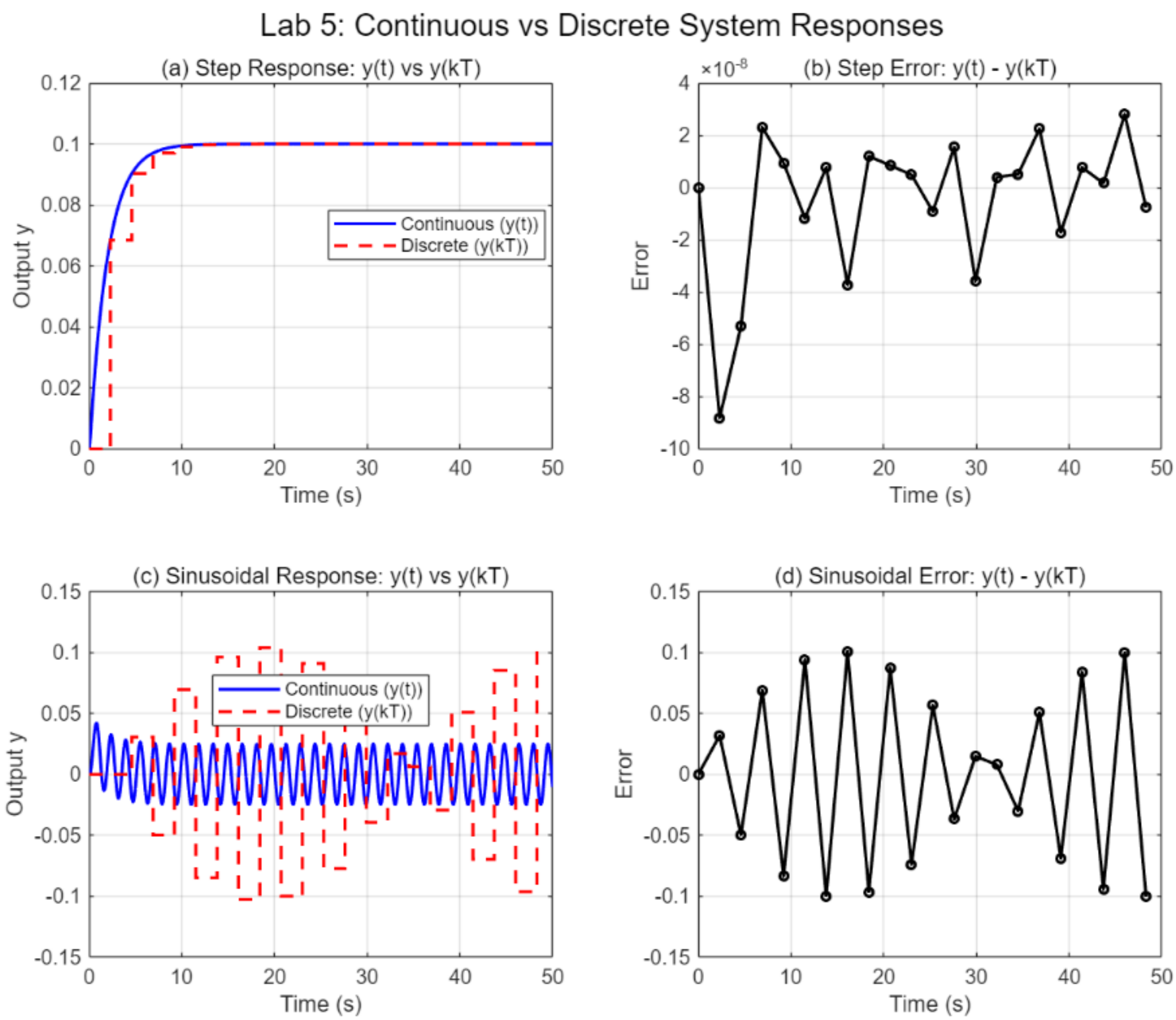
```
tspan = [0 100];
x0 = [0; 0];
sol = ode45(@(t,x) odefun2(t,x,a0,a1,b0), tspan, x0);
plot(sol.x, sol.y(1,:))
hold on;
t_d2 = 0:T:100;
n = length(t_d2);
x_d = zeros(2, n);
x_d(:, 1) = x0;
u_d = zeros(1,n);
for k = 1:n-1
    u_d(1, k) = 2*sin(4*k*T);
    x_d(:, k+1) = Ad*x_d(:, k) + Bd * u_d(1, k);
end
plot(t_d2, x_d(1,:))
legend show;
hold off;
```




```
y2_kt = x_d(1, :);
for k = 1:n
    y2_t(:, k) = deval(sol, t_d2(k));
end
plot(t_d2, y2_t(1, :) - y2_kt, 'LineWidth', 2);
```



8. Plot the graphs



Because my sampling time Ts is relatively large, the discrete data does not fit well for sinusoidal response.

Answers

```
ans = 包含以下字段的 struct:

eig_value_continuous_1: -19.4868
 eig_value_continuous_2: -0.5132
eig_value_discrete_1: 1.7347e-18
   eig_value_discrete_2: 0.3072
     steps_transition_process: 4


You have 37 attempts left
ans = 包含以下字段的 struct:

eig_value_continuous_1: 0.5000
eig_value_continuous_2: 0.5000
eig_value_discrete_1: 0.5000
eig_value_discrete_2: 0.5000
steps_transition_process: 1
total_score: 3
```