Mini Project 1

ECE4530J - Decision Making in Smart Cities Summer 2022

* Name: Huang Yucheng ID: 519021910885

Part 1: Design your own dynamical system

Consider the dynamical system for longitudinal control:

$$\dot{v} = -\frac{\rho C_d}{2M_t}v^2 + \frac{T_e - R_g T_b}{M_t} - \frac{C_r mg}{M_t},$$
$$M_t = \frac{\left(mR^2 + I_w\right) R_g}{R}.$$

Please search on the internet for relevant materials, and make your best guess for all the parameters involved above, such as the air drag coefficient C_d, the rolling resistance coefficient C_r, etc. Hint: the decision variables in the system are T_e and T_b.

Answer:
In this problem, we need to clarify the following constant.

a. Air drag coefficient $C_d \approx 0.35$

b. Air density $\rho \approx 1.29$ kg/m^3

c. Radius of wheel $R \approx 0.3$ m

d. Moment of inertia of wheel $I_w \approx 1.5$

e. Rolling resistance coefficient $C_r \approx 0.02$

f. Mass of a car $m \approx 1500$ kg

g. Gear ratio $R_g \approx 0.25$

Part 2(a): Design a linear controller for the linearized system

Consider the uniform trajectory tracking problem with reference trajectory $\bar{x}[t] = \bar{v}t$.

1. Reformulate the system that you generated in Part 1, with the speed error being the state variable and the total torque T being the control input.
2. Linearize the above system at the origin, i.e., use Taylor expansion to linearize the RHS of the dynamical equation in the neighborhood of $v = \bar{v}$.
3. Design a linear controller that stabilizes the linearized system.
4. Plug in your controller to the system, and reformulate it with the absolute speed v being the state variable again.
5. Discretize the above system.

6. Consider the discretized system in the face of additive Gaussian noise. To be specific, a random noise $w[t]$ is added to the RHS of the system:

$$v[t+1] = v[t] + u[t]\delta + w[t]$$

Here $w[t]$ is a Gaussian random variable with zero mean. Please select the variance of $w[t]$ on your own.

Answer:

1. The state is essentially speed $v(t)$:

$$\dot{v} = -av^2 + T - c.$$

However, a more convenient representation is the tracking error

$$e(t) = \bar{v} - v(t).$$

Since $v = \bar{v} - e$, we have $\dot{v} = -\dot{e}$ and thus

$$\begin{aligned} -\dot{e} &= -a(\bar{v} - e)^2 + T - c \\ &= -ae^2 + 2a\bar{v}e + T - \left(a\bar{v}^2 + c\right). \end{aligned}$$

Finally,

$$\dot{e} = ae^2 - 2a\bar{v}e - T + \left(a\bar{v}^2 + c\right)$$

2. Dynamical equation

$$\dot{x} = ax^2 - 2a\bar{v}x - u + \left(a\bar{v}^2 + c\right).$$

Objective: find a policy $\mu : \mathbb{R} \to \mathbb{R}$ such that

$$\lim_{t\to\infty} x(t) = 0. \quad \text{(speed tracking)}$$

One way is to linearize the model in the neighborhood of the origin:

$$\dot{x} \approx -2a\bar{v}x - u + \left(a\bar{v}^2 + c\right).$$

Finally, we substitute the variables

$$\dot{e} \approx -2a\bar{v}e - T + \left(a\bar{v}^2 + c\right).$$

3. Total torque given by $T = \mu(e)$

$$\mu(e) = (1 - 2a\bar{v})e + \left(a\bar{v}^2 + c\right)$$

4. Plug the controller into the dynamic equation

$$\dot{e} \approx -2a\bar{v}e - T + \left(a\bar{v}^2 + c\right) \ :$$

we have

$$\begin{aligned} -\dot{e} &= -2a\bar{v}e - \left(ae^2 + (1 - 2a\bar{v})e + \left(a\bar{v}^2 + c\right)\right) \\ &+ \left(a\bar{v}^2 + c\right) = -ae^2 - e \end{aligned}$$

Since $v = \bar{v} - e$, we have $\dot{v} = -\dot{e}$ and thus $\dot{v} = -a(v - \bar{v})^2 - (v - \bar{v})$

5. Discretization: consider a small time increment δ,

$$\dot{x}(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t) \approx \frac{x(t+\delta) - x(t)}{\delta}.$$

And we have

$$\frac{v[t+1] - v[t]}{\delta} = -a(v[t] - \bar{v})^2 - (v[t] - \bar{v})$$

Finally

$$v[t+1] = v[t] - a\delta(v[t] - \bar{v})^2 - \delta(v[t] - \bar{v})$$

6.

$$v[t+1] = v[t] - a\delta(v[t] - \bar{v})^2 - \delta(v[t] - \bar{v}) + w[t]$$

Part 2(b): Directly design a controller for the nonlinear system

Consider the same uniform trajectory tracking problem with reference trajectory $\bar{x}[t] = \bar{v}t$.

1. Consider the reformulated system that you generated in Part $2a - 1$, directly design a controller for the nonlinear system. Hint: recall the method for HW2 P2(c).
2. Plug in your controller to the system, and reformulate it with the absolute speed v being the state variable again.
3. Discretize the above system.
4. Consider the discretized system in the face of additive Gaussian noise, which should follow the same distribution as $w[t]$ in Part2a-6.

Answer:

1. Since we have

$$\dot{e} = ae^2 - 2a\bar{v}e - T + \left(a\bar{v}^2 + c\right)$$

and $\dot{e} = -e$. The desired controller is given as follows:

$$T = ae^2 - 2a\bar{v}e + \left(a\bar{v}^2 + c\right) + e$$

2. Plug the controller into the dynamic equation, we have $\dot{e} = -e$. Since $v = \bar{v} - e$, we have $\dot{v} = -\dot{e}$ and thus $\dot{v} = \bar{v} - v$

3. Discretization: consider a small time increment δ,

$$\dot{x}(t) = \frac{\mathrm{d}}{\mathrm{d}t}x(t) \approx \frac{x(t+\delta) - x(t)}{\delta}.$$

And we have

$$\frac{v[t+1] - v[t]}{\delta} = \bar{v} - v[t]$$

Finally

$$v[t+1] = v[t] + \bar{v}\delta - v[t]\delta$$

4.

$$v[t+1] = v[t] + \bar{v}\delta - v[t]\delta + w[t]$$

Part 3: Simulate both controllers with respect to the reference trajectory

Consider the above uniform trajectory tracking problem with reference trajectory $\bar{x}[t] = \bar{v}t$.

a) Please simulate and plot the following three trajectory curves $s[t]$ v.s. t on the same figure:

1. Consider the linearized system with random noise in Part 2(a). Simulate the actual trajectory $s_1[t]$ with your linear controller applied.
2. Consider the nonlinear system with random noise in Part 2(b). Simulate the actual trajectory $s_2[t]$ with your controller applied.
3. Simulate the reference trajectory $s_{ref}[t]$.
   Hint: you may find Python function random.gauss 0 useful.

b) Compare the trajectory tracking performances.

Answer:

In this problem, I draw three figures for small-noise/large-noise/non-noise. We can easily see that in non-noise figure, the three lines drop coincide and become one line. And as for large noise, I choose numbers randomly from 0 to 3, and then we can find that they deviate from the reference line largely. And as for small noise, I choose numbers randomly from 0 to 0.4, and then we can find that they deviate from the reference line slightly.

When the noise is low, the final route is closer to the standard route. When there is a lot of noise, the final distance can vary greatly. Also, in many simulations I found that the linear and nonlinear controllers have little effect on the final result, sometimes the linear is closer to the reference, sometimes it is nonlinear, the important factor is the random normal distribution number .

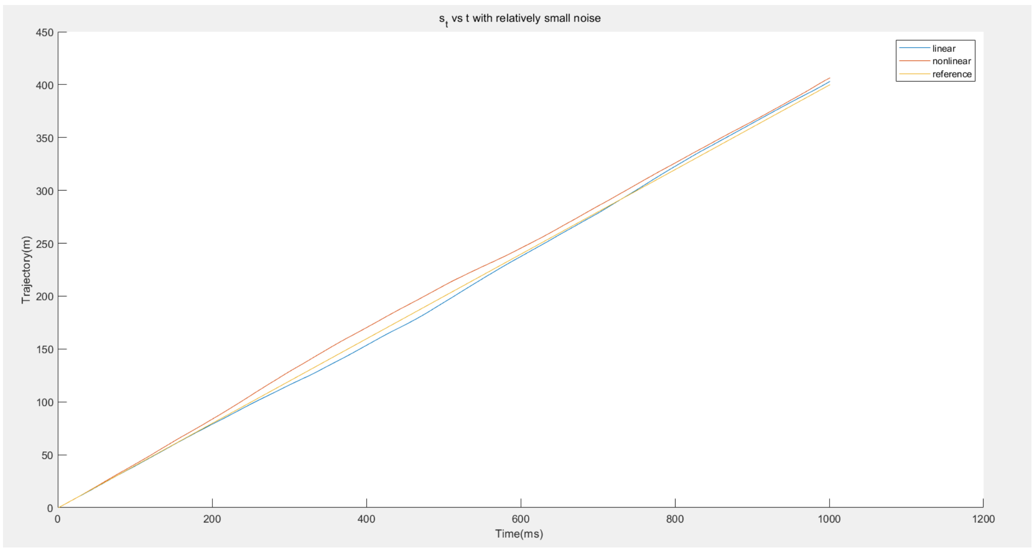


Figure 1: $s[t]$ vs t with noise

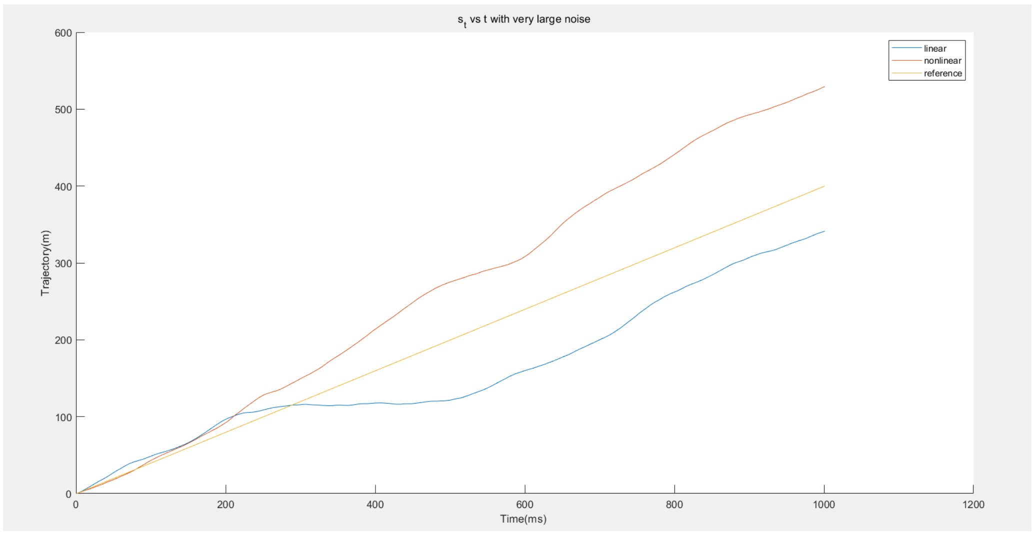


Figure 2: $s[t]$ vs t with very large noise

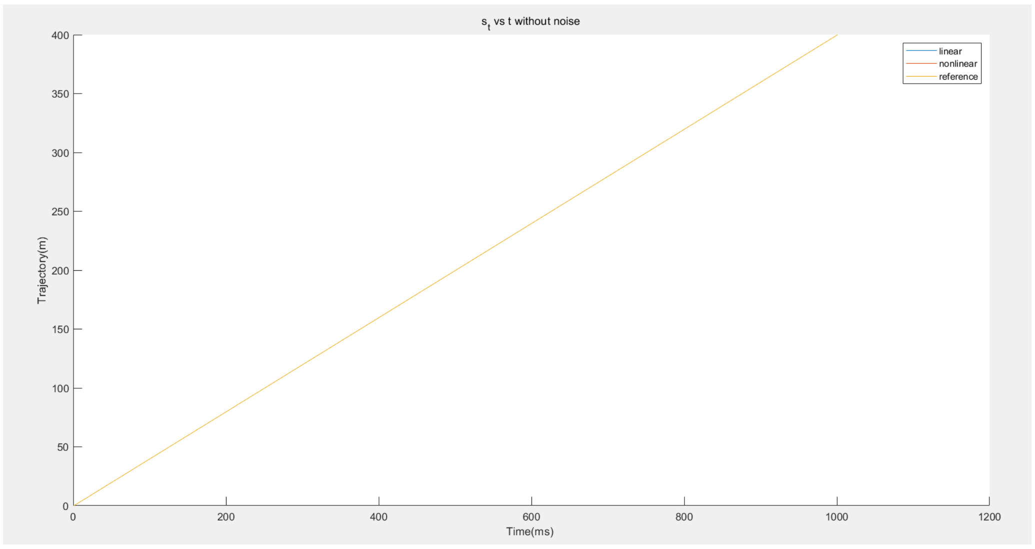
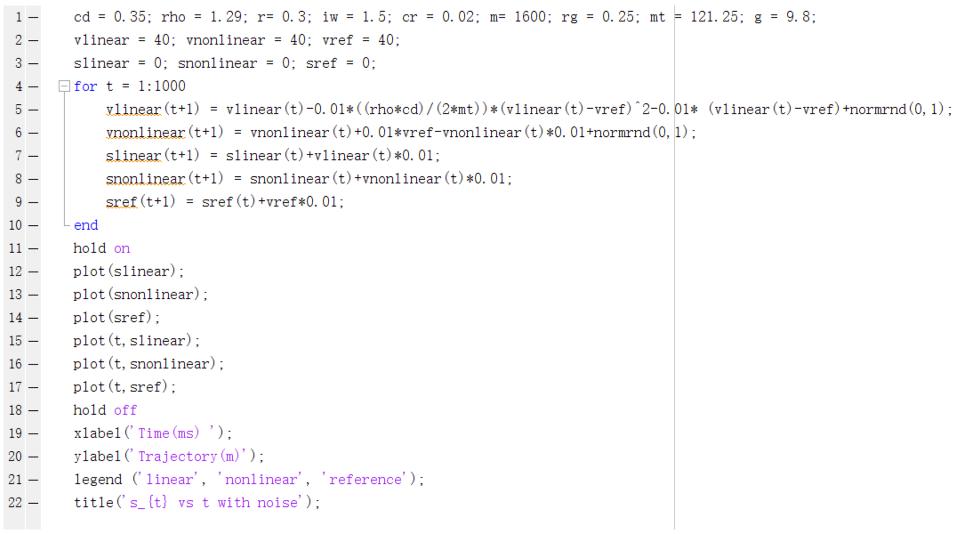
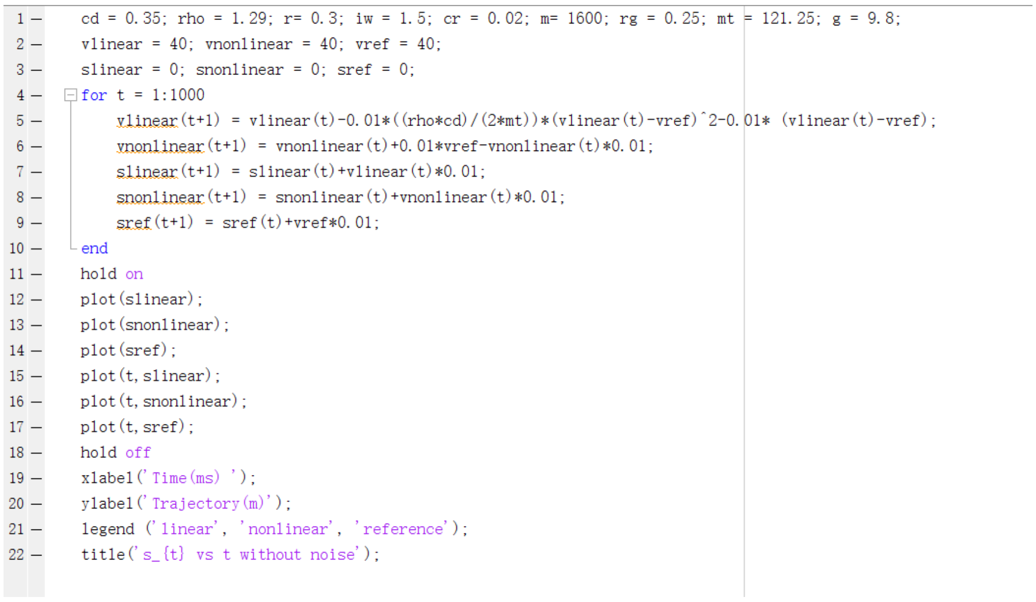


Figure 3: $s[t]$ vs t without noise



```
cd = 0.35; rho = 1.29; r= 0.3; iw = 1.5; cr = 0.02; m= 1600; rg = 0.25; mt = 121.25; g = 9.8;
vlinear = 40; vnonlinear = 40; vref = 40;
slinear = 0; snonlinear = 0; sref = 0;
for t = 1:1000
    vlinear(t+1) = vlinear(t)-0.01*((rho*cd)/(2*mt))*(vlinear(t)-vref)^2-0.01* (vlinear(t)-vref)+normrnd(0,1);
    vnonlinear(t+1) = vnonlinear(t)+0.01*vref-vnonlinear(t)*0.01+normrnd(0,1);
    slinear(t+1) = slinear(t)+vlinear(t)*0.01;
    snonlinear(t+1) = snonlinear(t)+vnonlinear(t)*0.01;
    sref(t+1) = sref(t)+vref*0.01;
end
hold on
plot(slinear);
plot(snonlinear);
plot(sref);
plot(t,slinear);
plot(t,snonlinear);
plot(t,sref);
hold off
xlabel('Time(ms) ');
ylabel('Trajectory(m)');
legend ('linear', 'nonlinear', 'reference');
title('s_{t} vs t with noise');
```

Figure 4: Code for Figure 1,2



```
cd = 0.35; rho = 1.29; r= 0.3; iw = 1.5; cr = 0.02; m= 1600; rg = 0.25; mt = 121.25; g = 9.8;
vlinear = 40; vnonlinear = 40; vref = 40;
slinear = 0; snonlinear = 0; sref = 0;
for t = 1:1000
    vlinear(t+1) = vlinear(t)-0.01*((rho*cd)/(2*mt))*(vlinear(t)-vref)^2-0.01* (vlinear(t)-vref);
    vnonlinear(t+1) = vnonlinear(t)+0.01*vref-vnonlinear(t)*0.01;
    slinear(t+1) = slinear(t)+vlinear(t)*0.01;
    snonlinear(t+1) = snonlinear(t)+vnonlinear(t)*0.01;
    sref(t+1) = sref(t)+vref*0.01;
end
hold on
plot(slinear);
plot(snonlinear);
plot(sref);
plot(t,slinear);
plot(t,snonlinear);
plot(t,sref);
hold off
xlabel('Time(ms) ');
ylabel('Trajectory(m)');
legend ('linear', 'nonlinear', 'reference');
title('s_{t} vs t without noise');
```

Figure 5: Code for Figure 3