Practice # 2

Linear Quadratic Regulator.

Instructing Professor : Madina Sinetova
Author: Xu Miao

May 8, 2022

Contents

1 Introduction

§1.1 Student information & & variant

- ⋆ Name: Xu Miao
- ⋆ ITMO Number: 293687
- ⋆ HDU Number: 19322103
- ⋆ Variant: $k = 7$

The variant data used in this practice is shown below

Table 1.1: Parameters of the system

Parameter	values
k	7
Block mass m	$3kg$
Viscous damping coefficient c	1.5
Initial conditions$x(0)$	$\begin{bmatrix} \pi \\ -2 \end{bmatrix}$

§1.2 Notations

Table 1.2: Notation used in this report

Symbol	Definition
K	feedback controller
Q, R	parameter matrices
$x(0)$	initial condition
P	auxiliary matrix
J	secondary target function

- ⋆ Other notations instructions will be given in the text.

2 Solution of problem

§2.1 Part1.Build a simulation modelcalculation & & calculate coefficients for 3 cases

§2.1.i Theory

Linear Quadratic Regulator

The optimal control problem is called linear secondary problems | if the system is linear, and performance indicators are status variables and control variables. LQR (Linear Quadratic Regulator), a linear secondary regulator, is a common solution method for solving linear secondary problems. LQR, its object is a linear system given in the form of state space in modern control theory, and the target function is an object state and the secondary function of the input. The LQR optimal design refers to the designed state feedback controller K to make the secondary target function J take a minimum, and K is uniquely determined by the right matrix Q and R, so the choice of Q and R is particularly important. The LQR theory is a state space design method that developed the earliest and most mature in modern control theory. Particularly valuable is that the LQR can obtain the optimal control law of the state linear feedback, which is easy to constitute a closed loop optimal control.
Consider a linear system in state-space form:

$$\dot{x} = Ax + Bu,$$

where $x(t) \in R^n, u(t) \in R^m$. The initial condition is $x(0)$. We assume here that all the states are measurable and seek to find a state-variable feedback (SVFB) control

$$u = -Kx + v$$

algebraic Riccati equation (ARE)

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{2.1}$$

The design procedure for finding the LQR feedback K
The design procedure for finding the LQR feedback K is:

- Select design parameter matrices Q and R
- Solve the algebraic Riccati equation for P
- Find the SVFB using $K = R^{-1}B^T P$

There are very | good numerical procedures for solving the ARE. The MATLAB routine that performs this is named lqr(A, B, Q, R).

§2.1.ii Mathematical model

Mathematical model of Simple mass/damper model

Consider a mass/damper system in a Fig. 1. The block has the positive location l and positive velocity v. You can exert force F on this block. There's some viscous damping coefficient between the block and the surface c.

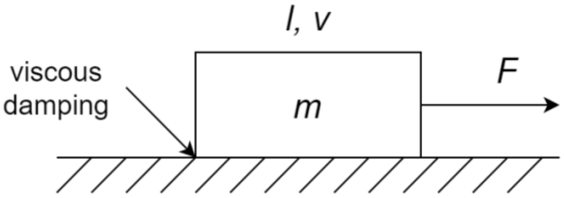

Fig. 2.1. Simple mass/damper model

According to the Newton's second law equations

$$m\ddot{l}(t) = F(t) - c\dot{l}(t) \tag{2.2}$$

And state-space representation

$$x(t) = \begin{bmatrix} l(t) \\ v(t) \end{bmatrix}, u(t) = F(t) \tag{2.3}$$

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{c}{m} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) \tag{2.4}$$

§2.1.iii Simulink Model

Use (2.3), (2.4) to build a model in simulink as follows:

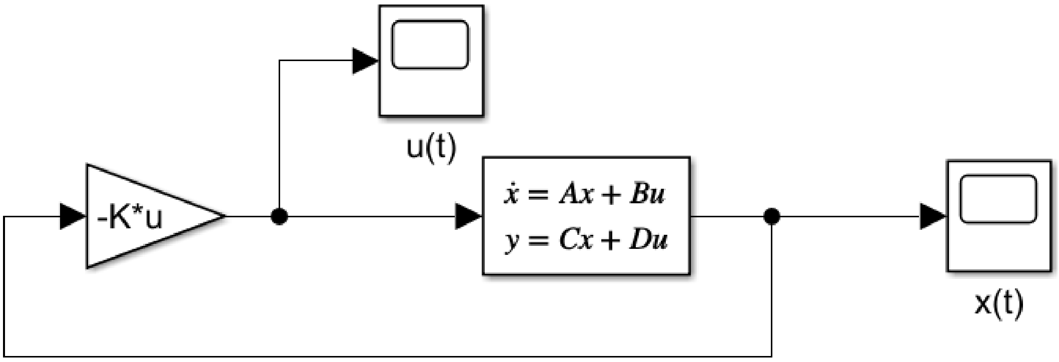

Fig. 2.2. simulink model

§2.1.iv calculation K - coefficients for 3 cases;

The feedback controller coefficients$K$ of the three control schemes are calculated in matlab as shown in the following table.

Table 2.1: Control Scheme Coefficients

Number	Control Plan	Q	R	K
1	Cheap control	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	[0.01]	$\begin{bmatrix} 10 & 12.5495 \end{bmatrix}$
2	Expensive control	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	[100]	$\begin{bmatrix} 0.1 & 0.6874 \end{bmatrix}$
3	Only penalize the velocity state	$\begin{bmatrix} 0.001 & 0 \\ 0 & 10 \end{bmatrix}$	[1]	$\begin{bmatrix} 0.0316 & 3.0937 \end{bmatrix}$

§2.2 Part2.Simulation

§2.2.i 1) Cheap control:

Simulation Data:

$$\begin{cases} Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ R = [0.01] \\ K = \begin{bmatrix} 10 & 12.5495 \end{bmatrix} \end{cases}$$

Simulation Result:

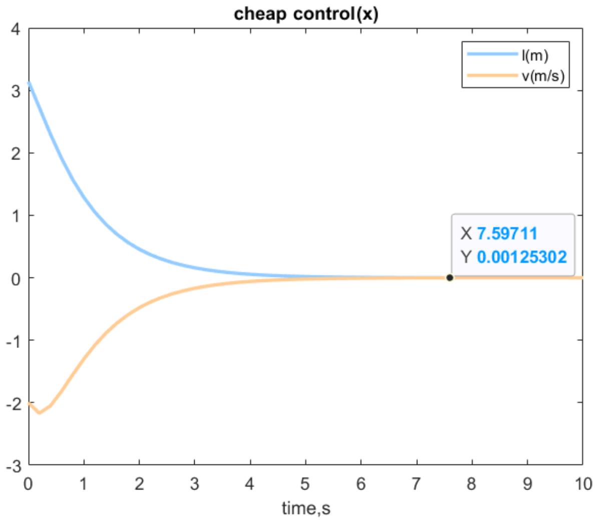

Fig. 2.3. Simulation Result1: $x(t)$

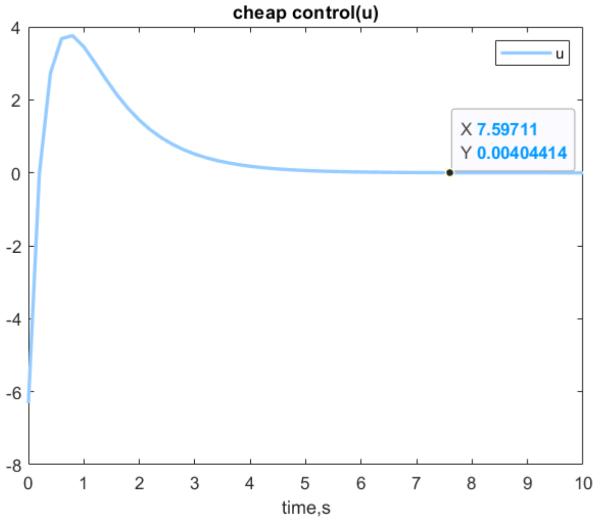

Fig. 2.4. Simulation Result2: $u(t)$

§2.2.ii 2) Expensive control:

Simulation Data:

reduce the variance of the noise by one order:

$$\begin{cases} Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ R = [100] \\ K = \begin{bmatrix} 0.1 & 0.6874 \end{bmatrix} \end{cases}$$

Simulation Result:

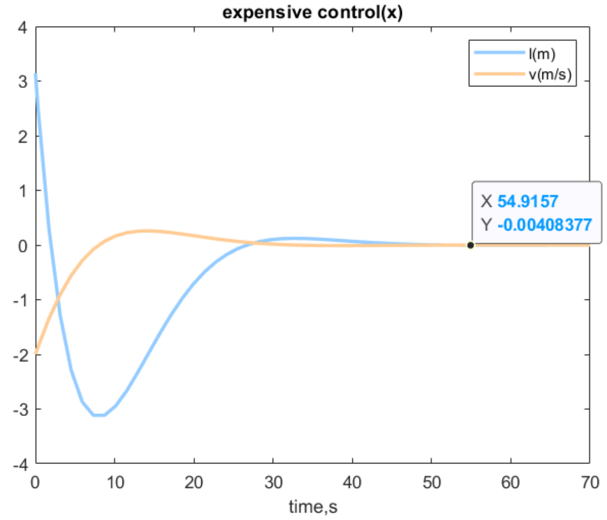

Fig. 2.5. Simulation Result1: $x(t)$

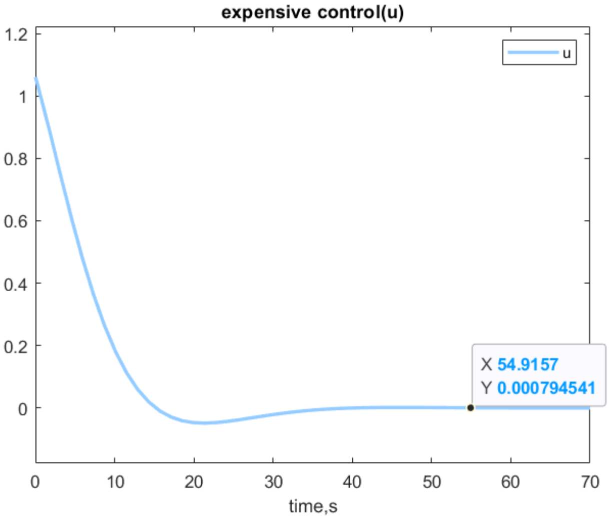

Fig. 2.6. Simulation Result2: $u(t)$

§2.2.iii 3) Only penalize the velocity state

Simulation Data:

$$\begin{cases} Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 10 \end{bmatrix} \\ R = [1] \\ K = \begin{bmatrix} 0.0316 & 3.0937 \end{bmatrix} \end{cases}$$

Simulation Result:

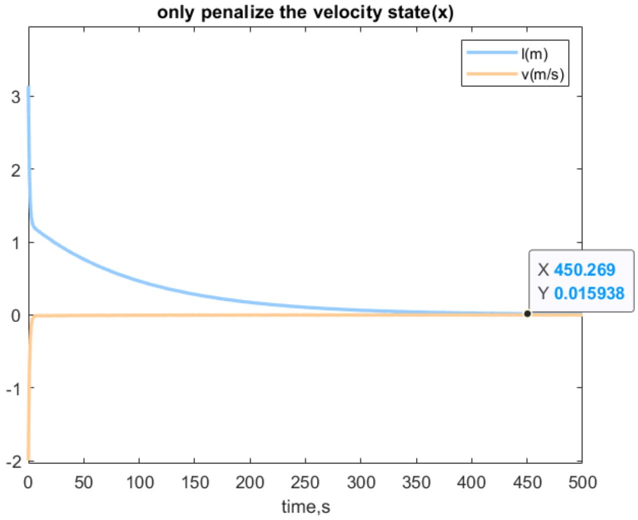

Fig. 2.7. Simulation Result1: $x(t)$

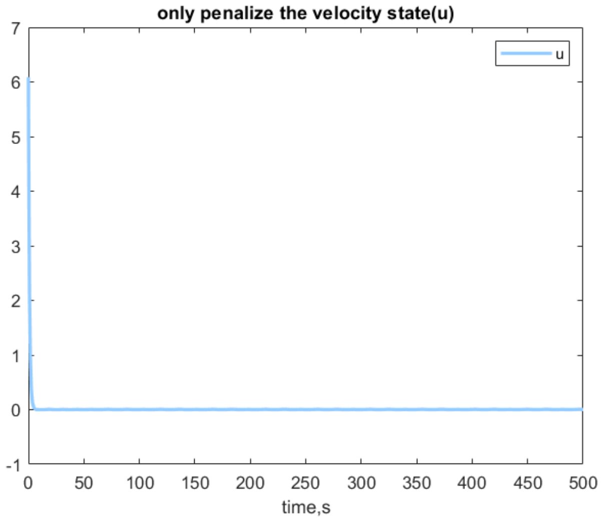

Fig. 2.8. Simulation Result2: $u(t)$

§2.3 Conclusion

From the simulation results, it can be seen that the time for each control scheme to make the system reach the steady state (0) is shown in the following table:

Table 2.2: Simulation results

Number	Control Plan	control time
1	Cheap control	≈ 7.6
2	Expensive control	≈ 54.92
3	Only penalize the velocity state	≈ 450.27

- ⋆ Since the plant is linear and the PI is quadratic, the problem of determining the SVFB K to minimize J is called the Linear Quadratic Regulator (LQR). The word 'regulator' refers to the fact that the function of this feedback is to regulate the states to zero. This is in contrast to tracker problems, where the objective is to make the output follow a prescribed (usually nonzero) reference command.
- ⋆ The two matrices $Q_{(n\times n)}$ and $R_{(m\times m)}$ are selected by the design engineer. Depending on how these design parameters are selected, the closed-loop system will exhibit a different response. Generally speaking, selecting Q large means that, to keep J small, the state $x(t)$ must be smaller. On the other hand, selecting R large means that the control input $u(t)$ must be smaller to keep J small.
- ⋆ The three control schemes in the experiment make the control time for the system to reach steady state: Cheap control < Expensive control < Only penalize the velocity state

I

Appendix

A Complete source code

LQR simulation

```
%%
% case □1cheap control
clear all
Q = [1 0;0 1];R = [0.01];
m = 3;c = 0.1;x_0 = [pi;-2];
A = [0 1;0 -c/m];
B = [0 ;1/m];C=[1 0;0 1];D = [0 ;0];
[K,P] = lqr(A,B,Q,R);
K
sim(’pra2.slx’)
time = ans.tout;
x = ans.X(:,2);
dx = ans.X(:,3);
u = ans.u(:,2);
figure(1)
plot(time,x,’Color’,[0.58,0.8,1],’LineWidth’,2);
hold on
plot(time,dx,’Color’,[1,0.8,0.58],’LineWidth’,2);
hold on
legend(’l(m)’,’v(m/s)’);
xlabel(’time,s’);
title(’cheap control(x)’);
figure(2)
plot(time,u,’Color’,[0.58,0.8,1],’LineWidth’,2);
hold on
legend(’u’);
xlabel(’time,s’);
title(’cheap control(u)’);
%%
% case □2expensive control
clear all
Q = [1 0;0 1];R = [100];
m = 3;c = 0.1;x_0 = [pi;-2];
A = [0 1;0 -c/m];
B = [0 ;1/m];C = [1 0;0 1];D=[0;0];
[K,P] = lqr(A,B,Q,R);
K
sim(’pra2.slx’)
time = ans.tout;
x = ans.X(:,2);
dx = ans.X(:,3);
u = ans.u(:,2);
figure(1)
plot(time,x,’Color’,[0.58,0.8,1],’LineWidth’,2);
hold on
```

```
        plot(time,dx,'Color',[1,0.8,0.58],'LineWidth',2);
        hold on
        legend('l(m)','v(m/s)');
        xlabel('time,s');
        title('expensive control(x)');
        figure(2)
        plot(time,u,'Color',[0.58,0.8,1],'LineWidth',2);
        hold on
        legend('u');
        xlabel('time,s');
        title('expensive control(u)');
        %%
        % case ▯3only penalize the velocity state
        clear all
        Q = [0.001 0;0 10];R = [1];
        m = 3;c = 0.1;x_0 = [pi;-2];
        A = [0 1;0 -c/m];
        B = [0 ;1/m];C = [1 0;0 1];D = [0 ; 0];
        [K,P] = lqr(A,B,Q,R);
        K
        sim('pra2.slx')
        time = ans.tout;
        x = ans.X(:,2);
        dx = ans.X(:,3);
        u = ans.u(:,2);
        figure(1)
        plot(time,x,'Color',[0.58,0.8,1],'LineWidth',2);
        hold on
        plot(time,dx,'Color',[1,0.8,0.58],'LineWidth',2);
        hold on
        legend('l(m)','v(m/s)');
        xlabel('time,s');
        title('only penalize the velocity state(x)');
        figure(2)
        plot(time,u,'Color',[0.58,0.8,1],'LineWidth',2);
        hold on
        legend('u');
        xlabel('time,s');
        title('only penalize the velocity state(u)');
```