Control of a Stationary Self-balancing Two-wheeled Vehicle

Student Name: Shen Xiaoitng
Department: Mechanical Engineering
E-mail Address: shen_xt2000@163.com
Supervisor: Ong Chongjin / Xiang Cheng

Abstract

This report is about the design and analysis about the linear state space model for the stationary two-wheeled vehicle. First, we build the system model and discuss the basic design specifications of transit response. Then we use the pole placement method to design the feedback model to stabilize the system and satisfy the specification of transit response. Different poles will affect the response of the system and we do analysis about it. Linear quadratic optimal control is another method for us to get the feedback gain and make the system stable. Matrix Q and R are key elements to design this system. We discuss the parameters in Q and R and the influence to the output of the system. Due to the limit of sensors for measuring all the state variables of the system, we build the observer to get the estimation of each state variable and analyze the estimation error of system. For the convenience of the control of each output signal, we can decouple the system and control each output by each input separately. Lastly, we do the servo control to make the output of the system follow the reference signal and analysis the range of the setting points. Through the steps above, different control strategies are explored to stabilize the vehicle to achieve its self-balance.

1 Contents

1 Introduction

There is a self-balance vehicle system with two wheels. It consists of three major parts including rider's center-of-gravity movement, a steering system(a front part) for steering, and a body(a rear part). The mechanical structure is given by the Figure 1.1 below.

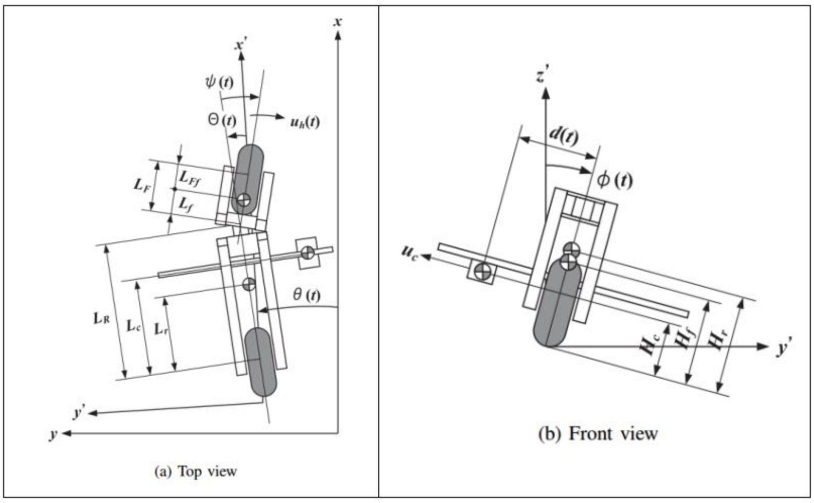


Figure 1.1 Two-wheeled vehicle structure model

For the dynamic model, there are some relate symbols are defined in Table 1.1.

Table 1.1 Definition of Symbols

M_f, M_r, M_c	Mass of each part
H_f, H_r, H_c	Vertical length from a floor to a center-of-gravity of each part
L_{Ff}, L_F	Horizontal length from a front wheel rotation axis to a center-of-gravity of part of front wheel and steering axis.
L_r, L_R	Horizontal length from a rear wheel rotation axis to a center-of-gravity of part of rear wheel and steering axis.
L_c	Horizontal length from a rear wheel rotation axis to a center-of-gravity of the cart system
J_x	Moment of inertia around center-of-gravity x axially
J_{fz}	Moment of inertia for part of front wheel z axially.
J_z	Moment of inertia for part of rear wheel that contains cart system z axially.
μ_x	Viscous coefficient around x axis.
μ_{fz}	Viscous coefficient for part of front wheel around z axis.
μ_z	Viscous coefficient for part of rear wheel that contains cart system around z axis.
μ_c	A viscosity coefficient of a movement direction of the cart system
Subscript f, r, c	Part of front wheel, rear wheel, and cart system respectively
$d(t)$, $\phi(t)$, $\psi(t)$	Cart position, handle angle and bike angle

An effective dynamic model has been built for this system

$$\dot{x} = Ax + Bu \tag{1.1}$$

$$y = Cx \tag{1.2}$$

where the state variable is

$$x = [d(t) \quad \varphi(t) \quad \psi(t) \quad \dot{d(t)} \quad \dot{\varphi(t)} \quad \dot{\psi(t)}] \tag{1.3}$$

According to my matriculation number, the parameter of $a = 3, b = 2, c = 5, d = 2$, then I can calculate the parameters which will be used in the project as follows:

Table 1.2 Physical parameters of the two-wheeled vehicle

Parameter	Value	Parameter	Value
$M_f[kg]$	2.39	$H_f[m]$	0.18
$M_r[kg]$	5.71	$H_r[m]$	0.161
$M_c[kg]$	1.74	$H_c[m]$	0.098
$L_{Ff}[m]$	0.05	$L_F[m]$	0.133
$L_r[m]$	0.128	$L_R[m]$	0.318
$L_c[m]$	0.259		
$J_x[kgm^2]$	0.53	$\mu_x[kgm^2/s]$	3.48
α	15.5	β	26.5
γ	11.2143	δ	60.5

1.1 Design specifications

Then transient response performance specifications for all the output y in state space model should satisfy the requirements as follows:

a) The overshoot is less than 10%

Considering the transient behavior of 2nd order system under unit step, the overshoot can be calculated by

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 10\% \tag{1.4}$$

We can solve $\zeta \geq 0.591$

b) The 2% settling time is less than 5 seconds

We can derive the parameter of ω_n by

$$t_s = \frac{4}{\zeta\omega_n} \leq 5 \tag{1.5}$$

$$\zeta\omega_n \geq 0.8 \tag{1.6}$$

1.2 Transient response of the system

To observe the transient response of the original system, we can give a step reference signal for the input channels with zero initial conditions to the open loop system. We can see the plant output channels of car position will go infinite, the handle angle will also go infinite, and the bike angle will go negative infinite in 10 seconds (Figure1.1). It is obvious that the open loop system does not satisfy the design criteria and it is not stable.

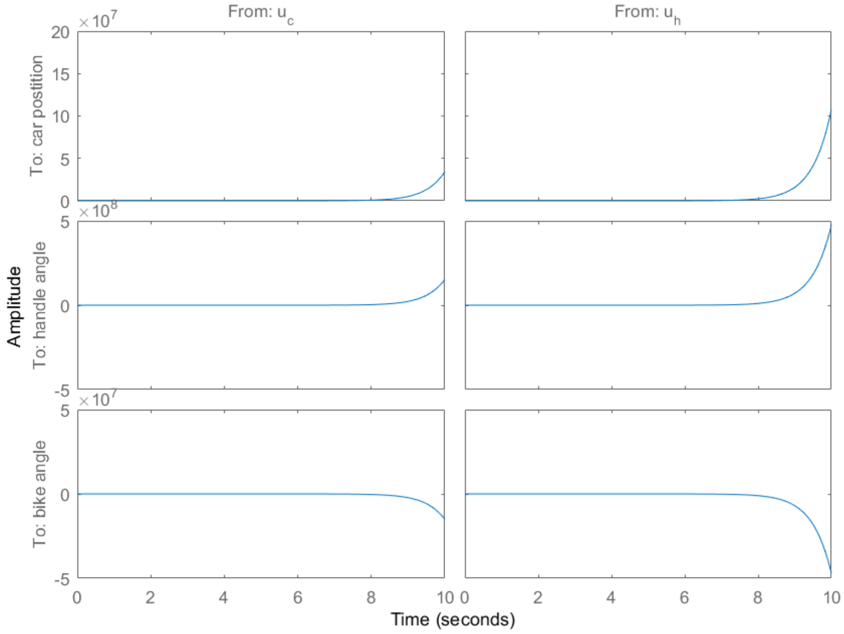


Figure 1.2 Step response for open-loop system

As a result, the feedback control is necessary to change the dynamic property of the system. Good close-loop poles are demanded for this system.

1.3 Response to non-zero initial state

The initial condition for the two-wheeled vehicle system is assumed to be $x_0 = [0.2 \quad -0.1 \quad 0.15 \quad -1 \quad 0.8 \quad 0]^T$. State response of the original system to non-zero state with zero external inputs. The response can be shown in Figure 1.2.

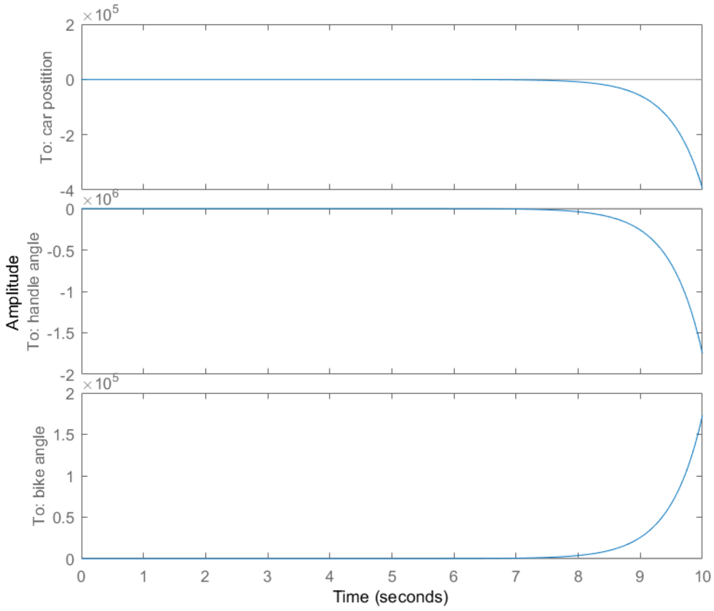


Figure 1.3 Response to non-zero initial state with zero external inputs

The response with a slight displacement of initial state, the car will go to no bond and the system is not stable.

2 Pole placement design

2.1 Controllability check

Consider the system

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 6.5 & -10 & -15.5 & 0 & 0 \\ -22.08 & 19.29 & 5.49 & -3.42 & -4.51 & 0.31 \\ 5 & -3.6 & 0 & 0 & 0 & -11.21 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 26.5 & 11.2 \\ 5.85 & -1.69 \\ 40 & 60.5 \end{bmatrix} u \tag{2.1}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} x \tag{2.2}$$

First, we should check the controllability of this system. We calculate the controllability matrix of this system is

$$W_C = [B \quad AB \quad A^2B \quad A^3B \quad A^4B \quad A^5B] \tag{2.3}$$

The rank of the controllability matrix is 6, which shows that the system is controllable. Then we can be arbitrarily assigned the closed-loop poles of the system.

2.2 Compute the feedback matrix K

We can choose 6 independent vectors from the 6×12 controllability matrix in the order from left to right and form the matrix C

$$\begin{aligned} Co &= \{b_1 \ Ab_1 \ A^2b_1 \ b_2 \ Ab_2 \ A^2b_2\} \\ &= \begin{bmatrix} 0 & 26.5 & -410.75 & 0 & 11.2 & -173.6 \\ 0 & 5.85 & -104.58 & 0 & -1.68 & -11.84 \\ 0 & 40 & -448.57 & 0 & 60.5 & -678.46 \\ 26.5 & -410.75 & 6.004 \times 10^3 & 11.2 & -173.6 & 2.07 \times 10^3 \\ 5.85 & -104.59 & 1.484 \times 10^3 & -1.68 & -11.84 & 488.04 \\ 40 & -448.57 & 5.142 \times 10^3 & 60.5 & -678.46 & 7.67 \times 10^3 \end{bmatrix} \end{aligned} \tag{2.4}$$

We take Co^{-1} as

$$Co^{-1} = \begin{bmatrix} {q_1}^T \\ {q_2}^T \\ {q_3}^T \\ {q_4}^T \\ {q_5}^T \\ {q_6}^T \end{bmatrix} \tag{2.5}$$

Then we can form matrix T

$$T = \begin{bmatrix} {q_3}^T \\ {q_3}^T A \\ {q_3}^T A^2 \\ {q_6}^T \\ {q_6}^T A \\ {q_6}^T A^2 \end{bmatrix} = \begin{bmatrix} 0.176 & -0.454 & -0.049 & 0.006 & -0.017 & -0.001 \\ 0379 & -0.291 & -0.160 & 0.137 & -0.375 & -0.036 \\ 8.106 & -6.219 & -3.429 & -0.458 & 1.400 & 0.124 \\ 0.061 & -0.161 & -0.016 & 0.001 & -0.003 & -3.1 \times 10^{-4} \\ 0.071 & -0.054 & -0.030 & 0.053 & -0.146 & -0.014 \\ 3.150 & -2.416 & -1.332 & -0.254 & 0.603 & 0.080 \end{bmatrix} \tag{2.6}$$

Through the result of matrix T, we can get $\bar{A} = TAT^{-1}, \bar{B} = TB$

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1.47 \times 10^3 & 791.50 & 27.56 & -4.12 \times 10^3 & -2.30 \times 10^3 & -127.22 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1.07 \times 10^3 & 6.59 & 21.45 & 3.0 \times 10^3 & 42.92 & -58.78 \end{bmatrix} \tag{2.7}$$

$$\bar{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.8}$$

(if $|\bar{A}_{ij}|$ or $|\bar{B}_{ij}|$ less than 0.001, than we consider it as 0)

We design the feedback gain matrix for the controllable canonical form

$$\bar{K} = \begin{bmatrix} \bar{K}_{11} & \bar{K}_{12} & \bar{K}_{13} & \bar{K}_{14} & \bar{K}_{15} & \bar{K}_{16} \\ \bar{K}_{21} & \bar{K}_{22} & \bar{K}_{23} & \bar{K}_{24} & \bar{K}_{25} & \bar{K}_{26} \end{bmatrix} \tag{2.9}$$

Then

$\bar{A} - \bar{B}\bar{K} =$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1.47 \times 10^3 - \bar{K}_{11} & 791.50 - \bar{K}_{12} & 27.56 - \bar{K}_{13} & -4.12 \times 10^3 - \bar{K}_{14} & -2.30 \times 10^3 - \bar{K}_{15} & -127.22 - \bar{K}_{16} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1.07 \times 10^3 - \bar{K}_{21} & 6.59 - \bar{K}_{22} & 21.45 - \bar{K}_{23} & 3.0 \times 10^3 - \bar{K}_{24} & 42.92 - \bar{K}_{25} & -58.78 - \bar{K}_{26} \end{bmatrix} \tag{2.10}$$

Consider a standard second-order system, we regard $\zeta = 0.8,\ \omega_n = 1.25\ and$ its poles are calculated by $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$ are $\lambda_{1,2} = -1 \pm 0.75j$. This is a pair of dominant poles and we can locate extra poles other than the dominant ones to be 2-5 times faster than the dominant ones $\lambda_{3,4} = -2$ and $\lambda_{5,6} = -4$. We can get the

$$det(sI - A_d) = (s + 1 + 0.75j)(s + 1 - 0.75j)(s + 2)^2(s + 4)^2 \tag{2.11}$$

So we design the matrix

$A_d =$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0.36 \times 10^3 & 0.26 \times 10^3 & 39.2 & -1.01 \times 10^3 & -0.76 \times 10^3 & -126.5 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0.14 \times 10^3 & 101.5 & 16.9 & -0.38 \times 10^3 & -291.5 & -53.2 \end{bmatrix} \tag{2.12}$$

Compare the possible desired closed-loop matrix A_d and closed loop matrix $\bar{A} - \bar{B}\bar{K}$, than we can get the feedback gain matrix of canonical form

$$\bar{K} =$$

$$\begin{bmatrix} 1.11\times10^3 & 0.53\times10^3 & -11.7 & -3.11\times10^3 & -1.54\times10^3 & -0.7 \\ -1.21\times10^3 & -0.095\times10^3 & 4.5 & 3.38\times10^3 & 0.33\times10^3 & -5.5 \end{bmatrix} \quad (2.13)$$

Through the calculation of $K = \bar{K}T$, we can get the feedback gain matrix

$$K = \begin{bmatrix} 1.52 & -0.82 & -1.06 & -0.58 & -0.02 & 0.10 \\ -1.00 & 0.58 & 0.86 & 0.43 & -0.01 & -0.17 \end{bmatrix} \quad (2.14)$$

2.3 Response with pole placement

If we do the pole placement, we can get the step response of the system as Figure 2.1. The step response with zero initial conditions can satisfy the transient response performance as shown in Table 2.1.

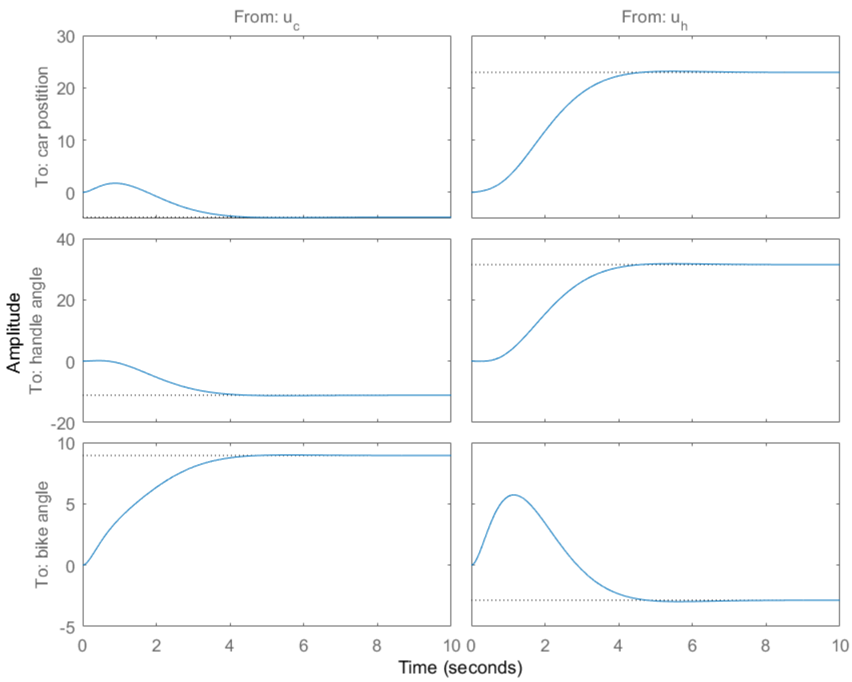


Figure 2.1 Step response of the system with proper pole placement

Table 2.1 The transient specifications of the system with pole placement

Input	Output	Settling time/sec	Overshoot/%
	Output1	4.2	1.63
u_c	Output2	4.13	1.02
	Output3	4.02	0.444
	Output1	4.14	0.903
u_t	Output2	4.14	0.965
	Output3	4.41	3.73

Then we can show all the six states response to non-zero initial state ($x_0 = [0.2 \quad -0.1 \quad 0.15 \quad -1 \quad 0.8 \quad 0]^T$) with zero external inputs, both the disturbance and set point can be assumed to be zero. We can see that the system can be stable after few seconds.

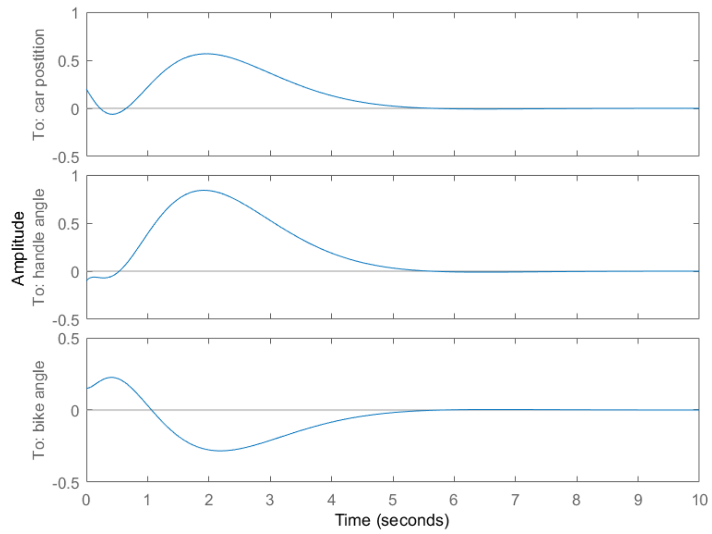


Figure 2.2 Response to non-zero initial state with zero external inputs under proper pole placement

2.4 The influence of different positions of poles

We choose the different dominant poles and we can get the transit response of different system. It is shown in Figure 2.3. The details of the response with different pols are listed in Table 2.1. We can see that the system with smaller dominant poles, it will have longer settling time and smaller overshoot, while the system with larger dominant poles will have shorter settling time and larger overshoot.

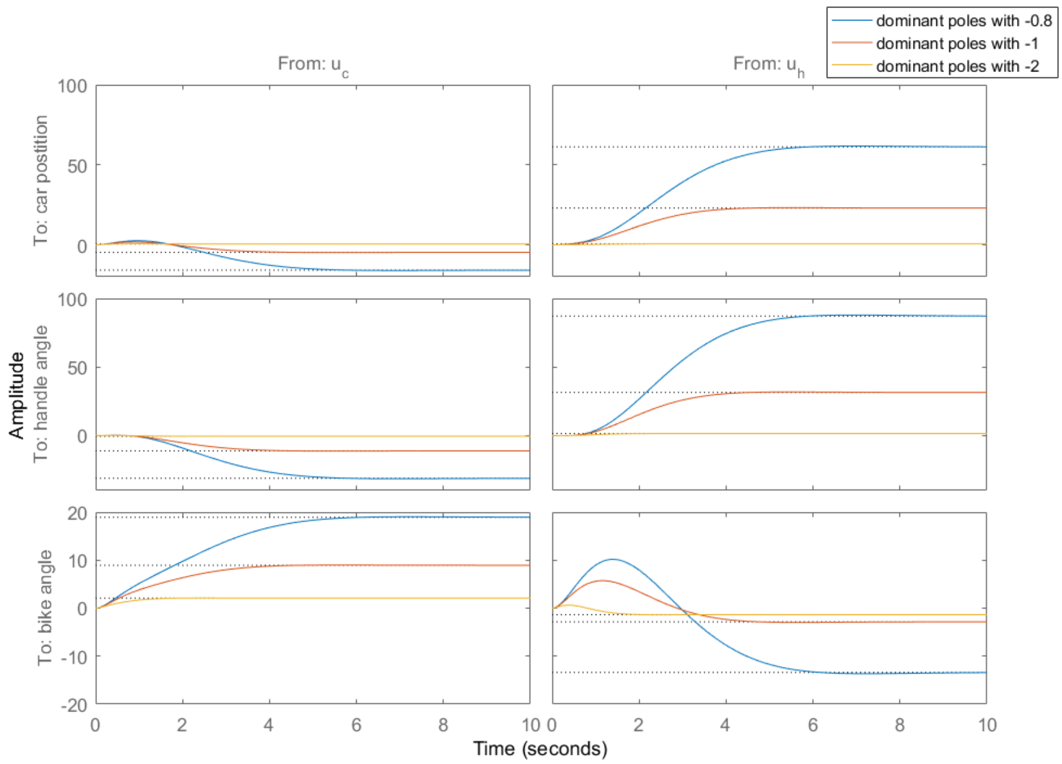


Figure 2.3 Step response of the system with different poles

Table 2.2 The comparison of step response with different poles

Parameters	Poles	Input	Output	Settling time/sec	Overshoot/%
$\zeta = 0.8$ $\omega_n = 1$	$\lambda_{1,2} = -0.8 \pm 0.6j$ $\lambda_{3,4} = -1.5$ $\lambda_{5,6} = -3$	u_c	Output1	5.33	1.21
			Output2	5.26	0.92
			Output3	5.29	0.50
		u_t	Output1	5.28	0.84
			Output2	5.28	0.88
			Output3	5.65	1.92
$\zeta = 0.8$ $\omega_n = 1.25$	$\lambda_{1,2} = -1 \pm 0.75j$ $\lambda_{3,4} = -2$ $\lambda_{5,6} = -4$	u_c	Output1	4.20	1.63
			Output2	4.13	1.02
			Output3	4.02	0.44
		u_t	Output1	4.14	0.90
			Output2	4.14	0.97
			Output3	4.41	3.73
$\zeta = 0.8$ $\omega_n = 2.5$	$\lambda_{1,2} = -2 \pm 1.5j$ $\lambda_{3,4} = -4$ $\lambda_{5,6} = -10$	u_c	Output1	2.04	19.8
			Output2	2.06	1.31
			Output3	1.71	2.11
		u_t	Output1	2.26	0.81
			Output2	2.06	1.02
			Output3	1.88	2.27

If we get the picture (Figure 2.4) of the response to non-zero initial state with zero external inputs with different poles, we can easily observe that if the system with larger dominant negative pole, it will have smaller oscillation and will converged to zero at a faster speed.

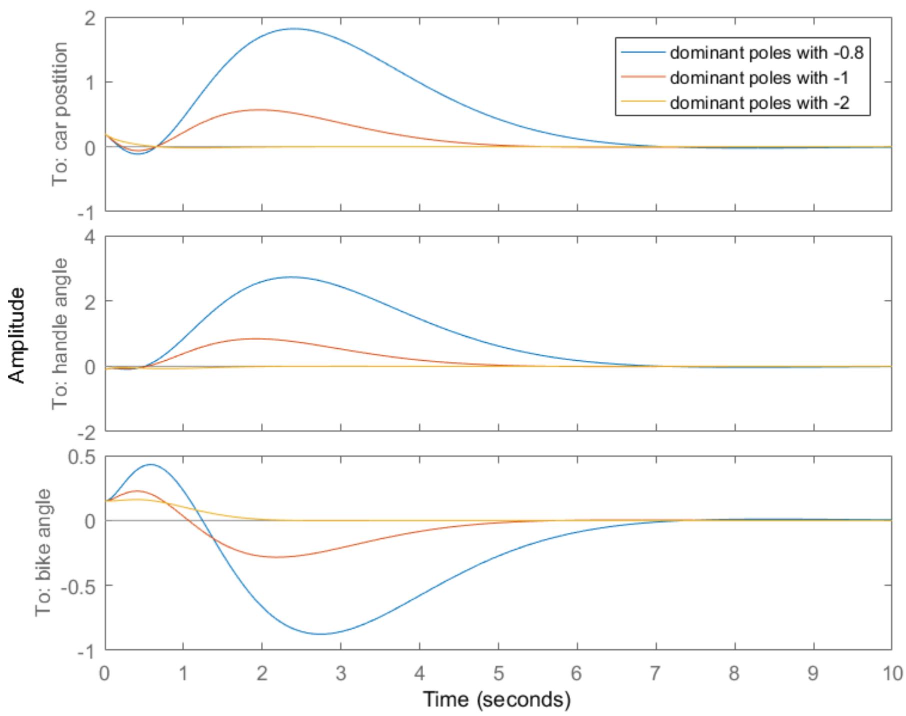


Figure 2.4 Response to non-zero initial state with zero external inputs with different poles

3 Quadratic optimal control

3.1 Weighting matrices Q and R selection

The weighting matrices Q and R are symmetric. The matrix R is positive definite, while the matrix Q is semi-positive definite. Let

$$Q = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.1}$$

$$R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{3.2}$$

3.2 Compute the feedback matrix K

The key to do LQR is to compute the matrix P in Riccati equation and get the feedback gain. First, we form the 12×12 matrix

$$\Gamma = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \tag{3.3}$$

We can find 6 stable eigenvalues λ_i of matrix Γ and their corresponding eigenvectors $\begin{pmatrix} v_i \\ \mu_i \end{pmatrix}, i = 1,2,\dots,6.$

$$\lambda_1 = -14.85, v_1 = \begin{bmatrix} 0.06 \\ 0.01 \\ -0.02 \\ -0.92 \\ -0.17 \\ 0.35 \end{bmatrix}, \mu_1 = \begin{bmatrix} 0.02 \\ 0.01 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.4}$$

$$\lambda_2 = -9.84 + 5.31i, v_2 = \begin{bmatrix} -0.02 - 0.02i \\ 0 \\ -0.07 - 0.04i \\ 0.31 + 0.12i \\ 0.07 + 0.08i \\ 0.93 \end{bmatrix}, \mu_2 = \begin{bmatrix} -0.01i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.5}$$

$$\lambda_3 = -9.84 - 5.31i, v_3 = \begin{bmatrix} -0.02 + 0.02i \\ 0 \\ -0.07 + 0.04i \\ 0.31 - 0.12i \\ 0.07 - 0.08i \\ 0.93 \end{bmatrix}, \mu_3 = \begin{bmatrix} 0.01i \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.6}$$

$$\lambda_4 = -6.98, v_4 = \begin{bmatrix} 0.02 \\ -0.13 \\ -0.03 \\ -0.14 \\ 0.92 \\ -0.21 \end{bmatrix}, \mu_4 = \begin{bmatrix} 0.08 \\ -0.24 \\ -0.02 \\ 0 \\ -0.02 \\ 0 \end{bmatrix} \tag{3.7}$$

$$\lambda_5 = -3.77 + 2.27i, v_5 = \begin{bmatrix} -0.09 - 0.03i \\ -0.10 - 0.63i \\ -0.03 + 0.12i \\ 0.42 - 0.09i \\ 0.54 \\ -0.17 - 0.50i \end{bmatrix}, \mu_5 = \begin{bmatrix} -0.22 + 0.08i \\ -0.03 - 0.39i \\ 0.06 - 0.03i \\ 0.01i \\ -0.04i \\ 0 \end{bmatrix} \tag{3.8}$$

$$\lambda_6 = -3.77 - 2.27i, v_5 = \begin{bmatrix} -0.09 + 0.03i \\ -0.10 + 0.63i \\ -0.03 + 0.12i \\ 0.42 + 0.09i \\ 0.54 \\ -0.17 + 0.50i \end{bmatrix}, \mu_5 = \begin{bmatrix} -0.22 - 0.08i \\ -0.03 + 0.39i \\ 0.06 + 0.03i \\ -0.01i \\ 0.04i \\ 0 \end{bmatrix} \tag{3.9}$$

Then we can compute the $6 \times 6\ matrix$ P by $P = \mu_i {v_i}^{-1}$. The 2×6 feedback matrix K can be computed by the $K = R^{-1}B^TP$

$$K = \begin{bmatrix} 6.59 & -3.72 & -1.28 & 0.47 & -0.60 & -0.10 \\ -15.95 & 17.02 & 7.33 & -1.32 & 2.50 & 0.52 \end{bmatrix} \tag{3.10}$$

Then we can get the step response of the system (Figure 3.1). The overshoot of the system can be less than 10% and the settling time can be less than 5 seconds.

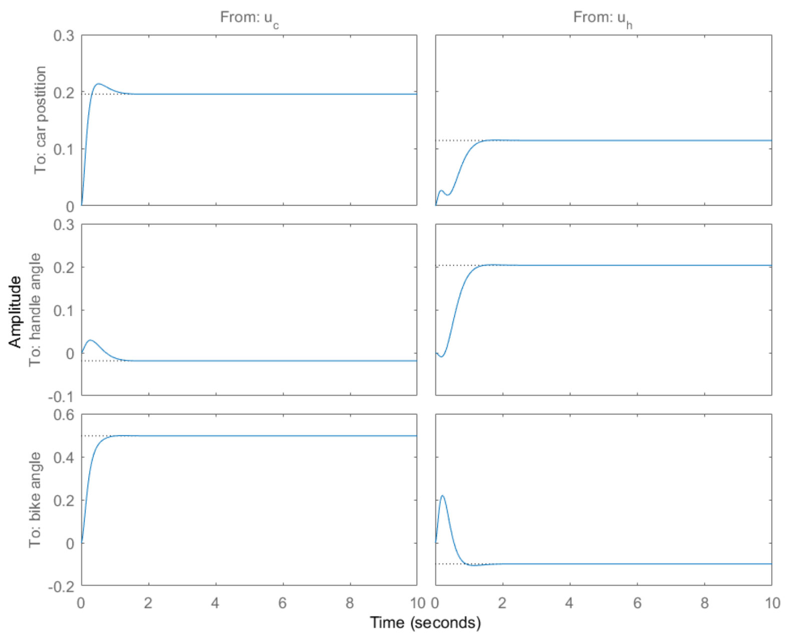


Figure 3.1 Step response of the system with LQR

Table 3.1 The transient specifications of the system with LQR

Input	**Output**	**Settling time/sec**	**Overshoot/%**
u_c	Output1	1.08	9.17
	Output2	1.34	0.93

	Output3	0.743	0.272
	Output1	1.35	0.459
u_t	Output2	1.26	0.457
	Output3	1.28	7.37

The state response of non-zero initial state with zero external inputs are shown in Figure 3.2 and it is clear that the state will converge to zero in 2 seconds.

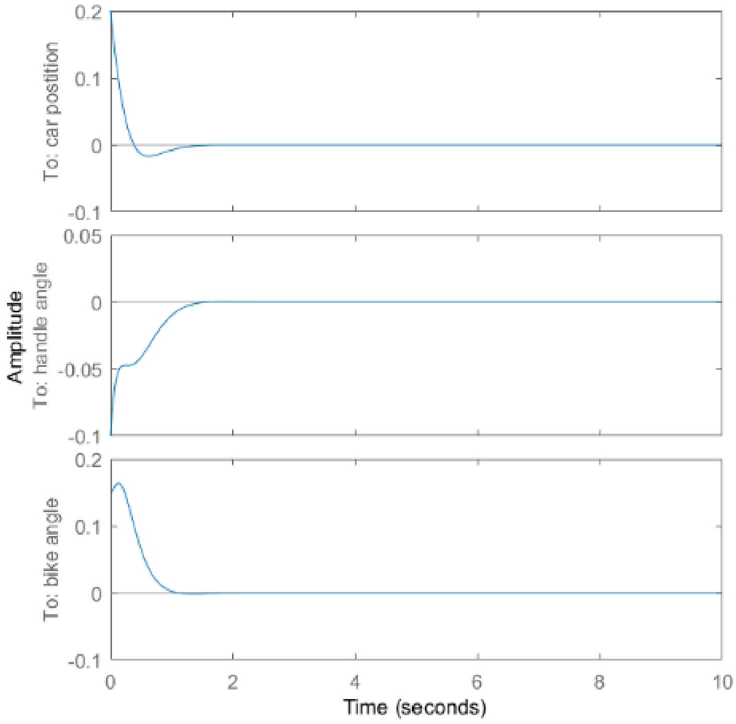


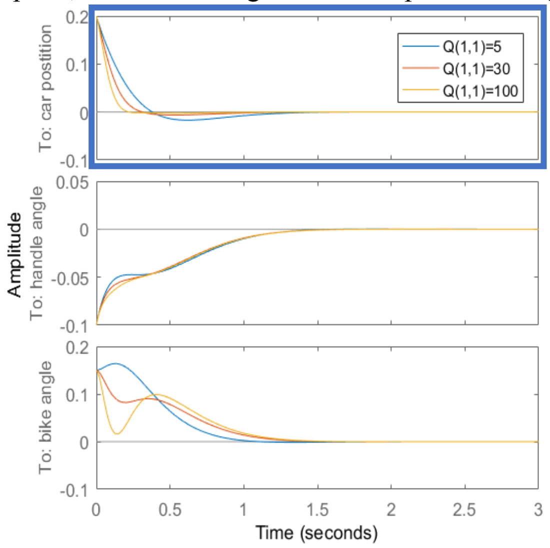
Figure 3.2 Response to non-zero initial state with zero external inputs under LQR

3.3 The effects of weightings Q and R on system performance

There are 6 states in this system, and the matrix Q and R are diagonal and we just tune the parameter in the diagonal line to observe the performance of the system.

We change the parameter of Q(1,1) from 5 to 30 to 100 and keep other parameters fixed, and we can see that the larger Q(1,1), the faster converge to zero of the first state (Figure 3.3) and has not the same influence on other states. We change the parameter of Q(2,2) from 10 to 30 to 100 and we can also observe that the second state will converge to zero faster with larger parameter of Q(2,2) (Figure 3.4). We change the parameter of Q(3,3) from 1 to 10 to 100, and we can see that the third state will also converge to zero at a higher speed with larger parameter. Also, it can also see that the parameter will influence other state response, and we should tune the parameters simultaneously to get the desired response.

From Figure 3.6 and Figure 3.7, we can see that the change of the parameter of matrix R will have an effect to the output. If we change the first parameter R(1,1), the response will change induced by input 1, and if we change the second parameter R(2,2), the response will change induced by input2.



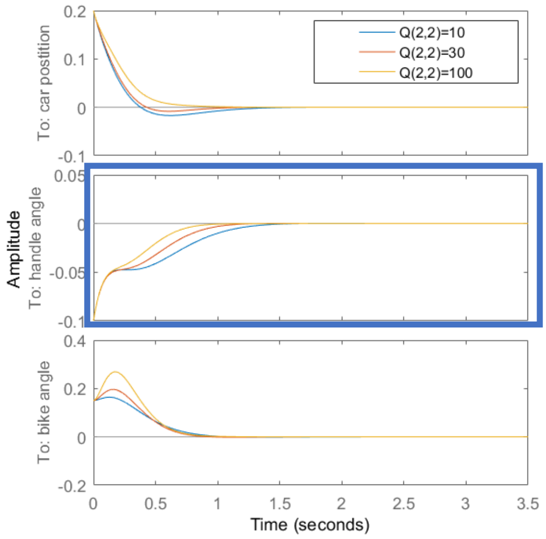


Figure 3.3 Response to non-zero initial state with zero external inputs with different Q(1,1)

Figure 3.4 Response to non-zero initial state with zero external inputs with different Q(2,2)

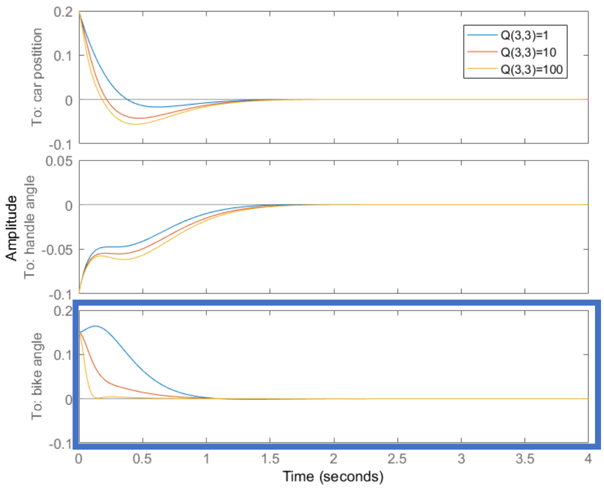


Figure 3.5 Response to non-zero initial state with zero external inputs with different Q(3,3)

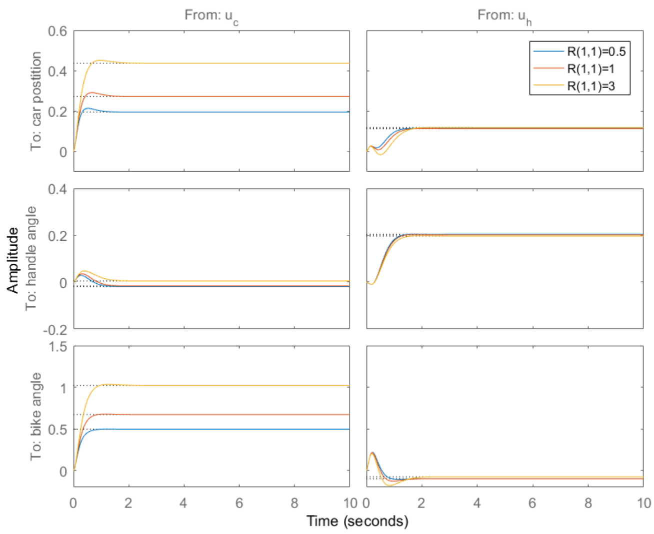


Figure 3.6 Step response with different R(1,1)

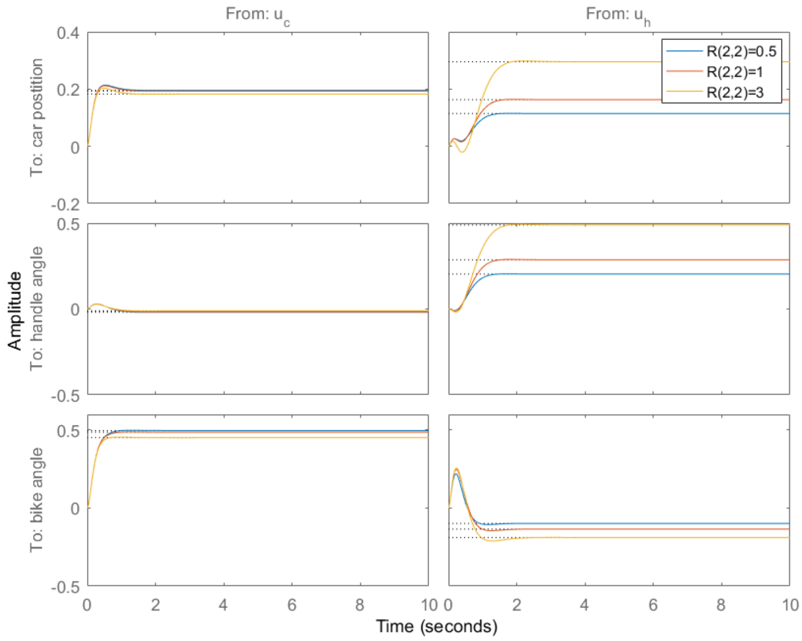


Figure 3.7 Step response with different R(2,2)

4 State Estimation

4.1 Observability check

First we check $\{A,C\}$ for observability. We can get the 18×6 dimension observability matrix and the rank of the matrix is full column. The system is observable.

$$W_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4.1}$$

4.2 Full-order observer design

Since the dimension of the output y is 3, we can construct the full order observer of the system. First we design the estimator

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y - \hat{y}] \tag{4.2}$$

The feedback control law of the original system will be

$$u = -K\,\hat{x}(t) \tag{4.3}$$

We can design the resultant observed-based LQR control system in the SIMULINK as Figure 4.1.

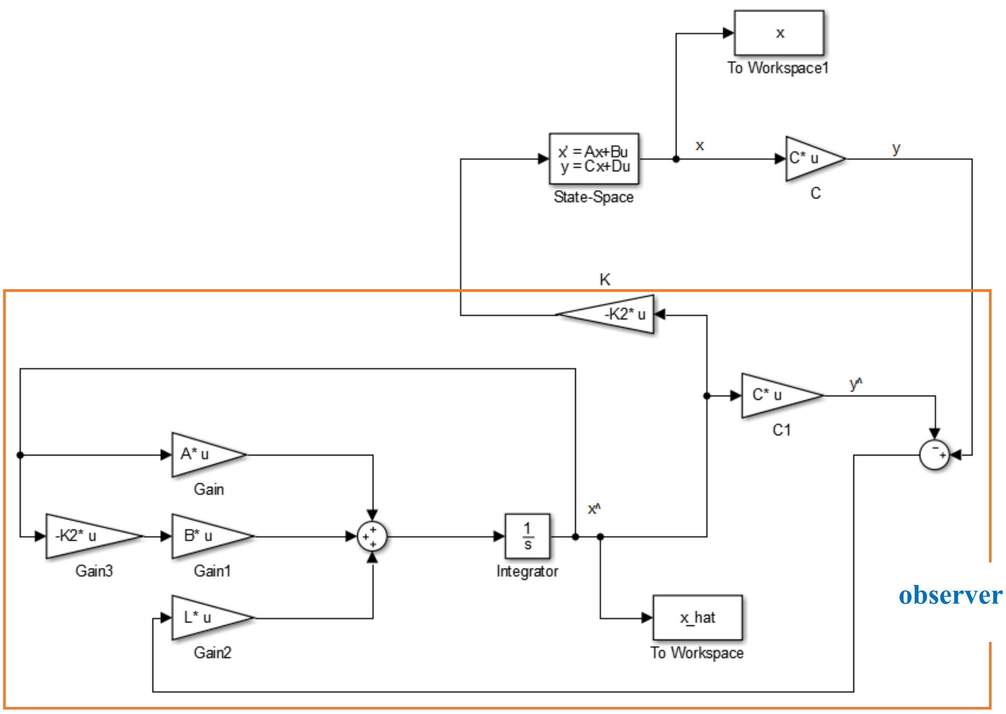


Figure 4.1 Resultant observer-based LQR control system

4.3 Selection of the poles of the observer.

For the observer problem, its error characteristic polynomial is

$$\begin{aligned} &det[sI-(A-LC)] = \\ &det[sI-(A-LC)]^T = det[sI-(A^T-C^TL^T)] \\ &\qquad = det[sI - (\tilde{A} - \tilde{B}\tilde{K})] \end{aligned} \tag{4.4}$$

Where $A^T = \tilde{A},\ C^T = \tilde{B},\ L^T = \tilde{K}$. Therefor we should find $\tilde{K}$ to make $(\tilde{A} - \tilde{B}\tilde{K})$ stable. Then we use the pole placement method to find the feedback gain $\tilde{K}$.

Considering the response speed of the observer, we set the poles are 2-5 times of the closed-loop poles. Then we select the poles of observer are $\lambda_{1,2} = -3 \pm 0.75j,\ \lambda_{3,4} = -6,\ \lambda_{5,6} = -12$. Then we can compute 6×3 matrix of the feedback gain $\tilde{K}$ by pole placement method the same as question 1. We take the transpose of $\tilde{K}$ to get the 6×3 matrix L.

$$L = \begin{bmatrix} 0.0732 & -1.8828 & -0.5485 \\ -1.0616 & 3.9648 & -0.0655 \\ -0.2230 & -0.1043 & 6.7408 \\ 39.4281 & 19.9457 & -7.9092 \\ -8.1723 & 21.4228 & 6.63 \\ 4.7398 & -3.4304 & -4.1259 \end{bmatrix} \tag{4.4}$$

The observer is based on question 2 LQR control system, then we still use the feedback matrix K computed by question 2 and build the model in Simulink as Figure4.1.

4.4 Results analysis

We can get the six state variables and their estimations through the simulation. It is clear that the error of each error of state variable goes to zero with in about 2 seconds.

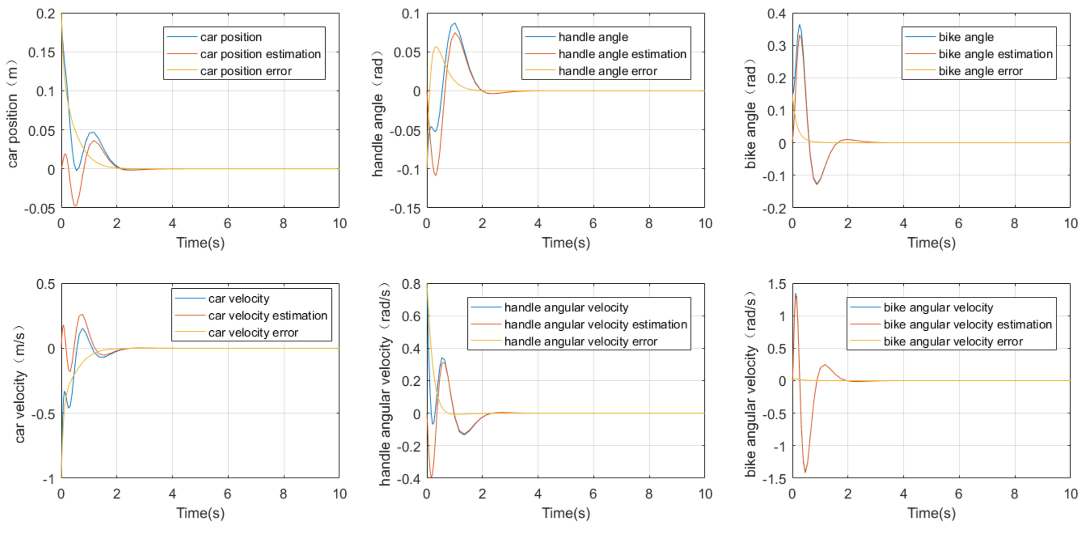


Figure 4.2 Performance of state observer and error signals

(poles with $\lambda_{1,2} = -3 \pm 0.75j,\ \lambda_{3,4} = -6,\ \lambda_{5,6} = -12$)

If we set the poles same as the original system, the error between will become larger and the speed to converge to zero will be slow.

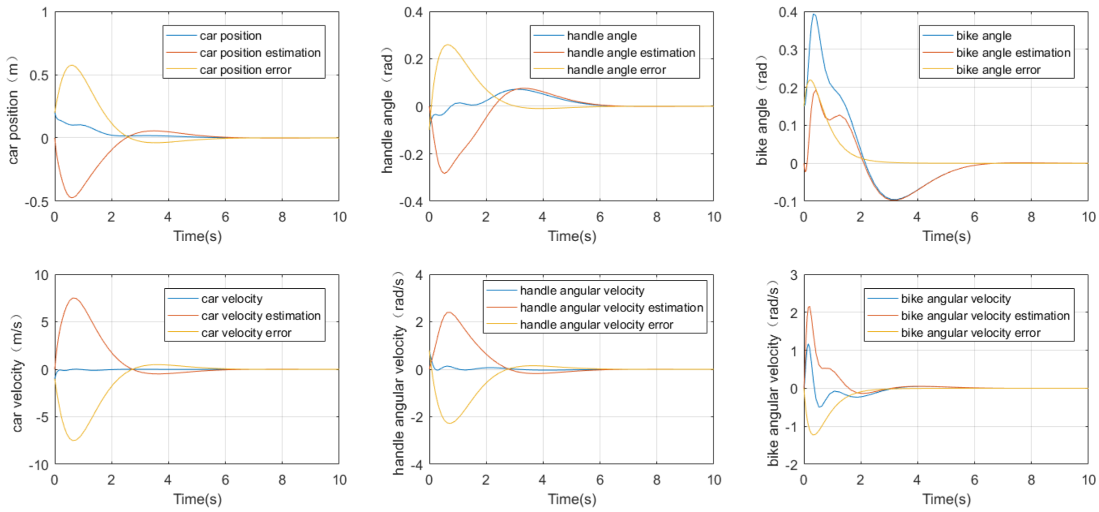


Figure 4.3 Performance of state observer and error signals

(poles with $\lambda_{1,2} = -1 \pm 0.75j,\ \lambda_{3,4} = -2,\ \lambda_{5,6} = -4$)

If we set the poles much larger than the original system, we can find that the estimation error will go to zero at a faster rate which are shown in Figure 4.4 and Figure 4.5.

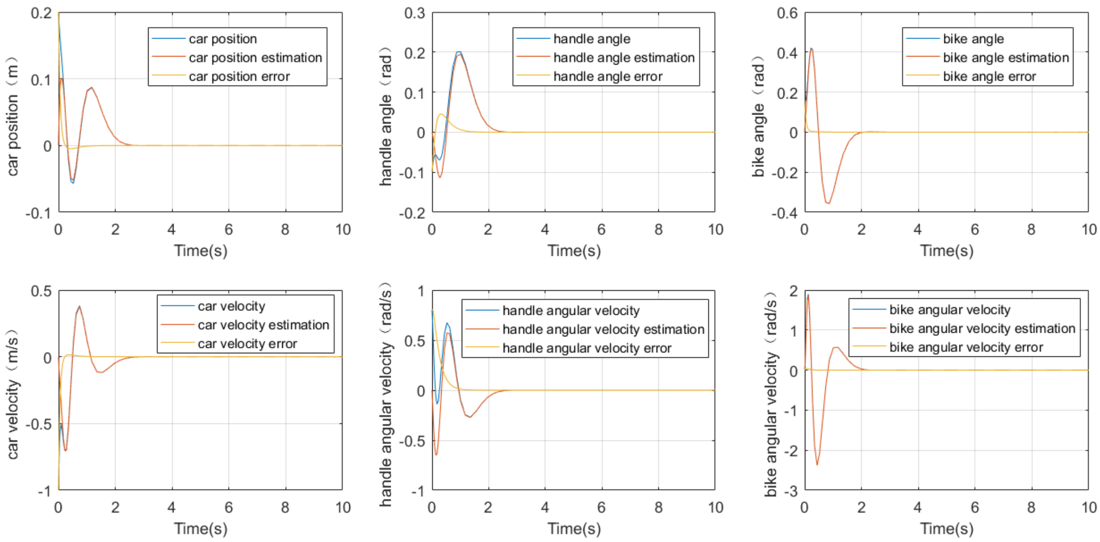


Figure 4.4 Performance of state observer and error signals

(poles with $\lambda_{1,2} = -5 \pm 0.75j,\ \lambda_{3,4} = -10,\ \lambda_{5,6} = -20$)

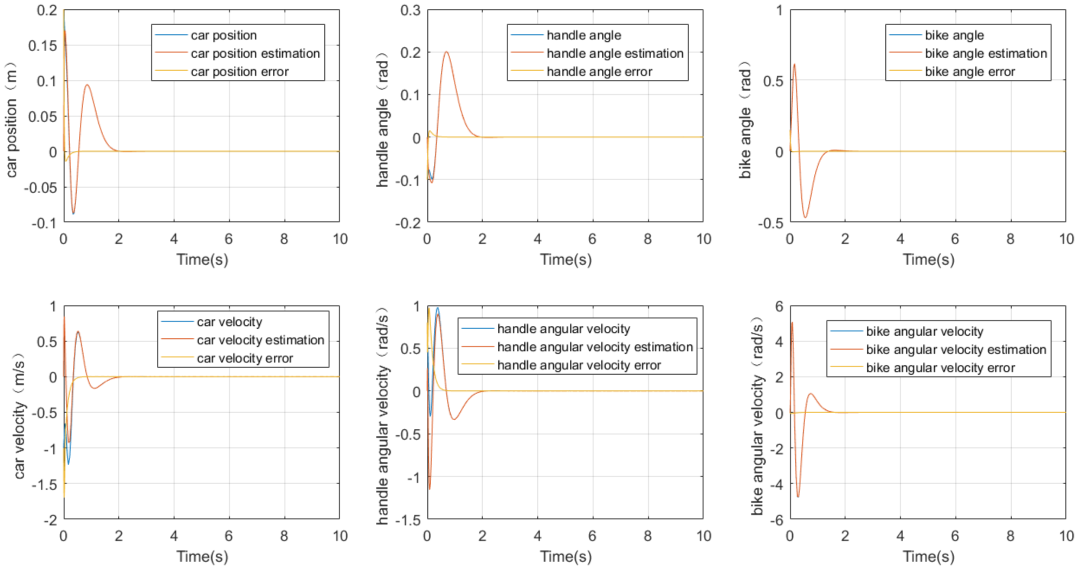


Figure 4.5 Performance of state observer and error signals

(poles with $\lambda_{1,2} = -10 \pm 0.75j$, $\lambda_{3,4} = -50$, $\lambda_{5,6} = -80$)

The response speed of an observer is faster than the original close-loop system, as we can set the pole be 2 to 5 times larger than the poles of original system. Then the estimation error will converge to zero quickly.

5 Decouple System

5.1 Compute the relative degree

We are only interested in the two outputs $d(t)$ and $\psi(t)$, a new output matrix is

$$C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.1}$$

We can get $c_1{}^T = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$ and $c_2{}^T = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]$.
It is readily checked that

$$c_1{}^T A^0 B = c_1{}^T B = [0 \quad 0] \tag{5.2}$$

$$c_1{}^T A^1 B = c_1{}^T AB = [26.5 \quad 11.2] \neq 0, \ \sigma_1 = 2 \tag{5.3}$$

$$c_2{}^T A^0 B = c_2{}^T B = [0 \quad 0] \tag{5.4}$$

$$c_2{}^T A^1 B = c_2{}^T AB = [40 \quad 60.5] \neq 0, \ \sigma_1 = 2 \tag{5.5}$$

5.2 Compute the matrix $\boldsymbol{B^*}$ and $\boldsymbol{C^*}$

So that we can compute B^* and it is non-singular

$$B^* = \begin{bmatrix} c_1{}^T AB \\ c_2{}^T AB \end{bmatrix} = \begin{bmatrix} 26.5 & 11.2 \\ 40 & 60.5 \end{bmatrix} \tag{5.6}$$

We suppose that the same system is required to be decoupled to give

$$H(s) = diag[(s+1)^{-2}, (s+1)^{-2}] \tag{5.7}$$

As we already have B^*, we can compute C^*since $\phi_f = s^2 + 2s + 1$

$$C^* = \begin{bmatrix} c_1{}^T \phi_f(A) \\ c_2{}^T \phi_f(A) \end{bmatrix} = \begin{bmatrix} c_1{}^T (A^2 + 2A + I) \\ c_2{}^T (A^2 + 2A + I) \end{bmatrix} \begin{bmatrix} 1 & 6.5 & -10 & -13.5 & 0 & 0 \\ 5 & -3.6 & 1 & 0 & 0 & -9.2 \end{bmatrix} \tag{5.8}$$

5.3 Build the decoupling close-loop system

Thus decoupling system can be achieved with

$$u(t) = -Kx(t) + Fr = -(B^*)^{-1} C^* x(t) + (B^*)^{-1} r(t) \tag{5.9}$$

which result in the close-loop system

$$\dot{x} = (A - BK)x + BFr \tag{5.10}$$

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & -2 & 0 & 0 \\ -22 & 16.58 & 9.24 & 1.5 & -4.51 & -0.57 \\ 0 & 0 & -1 & 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0.36 & 0 \\ 0 & 1 \end{bmatrix} r \tag{5.11}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} x \tag{5.12}$$

The transfer function matrix may easily be shown to be

$$H(s) = C_2 (SI - (A - BK))^{-1} BF \tag{5.13}$$

$$H(s) = \begin{bmatrix} \frac{1}{(s+1)^2} & 0 \\ 0 & \frac{1}{(s+1)^2} \end{bmatrix} \tag{5.14}$$

Then we can get the step response with zero initial state of fig and the initial response with respect to x0 of the decoupled system.

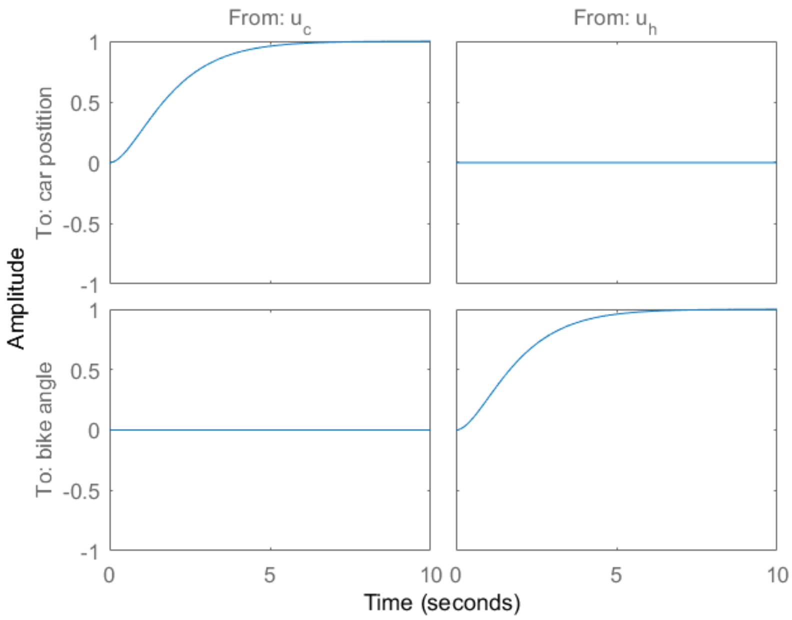


Figure 5.1 Step response of decouple system

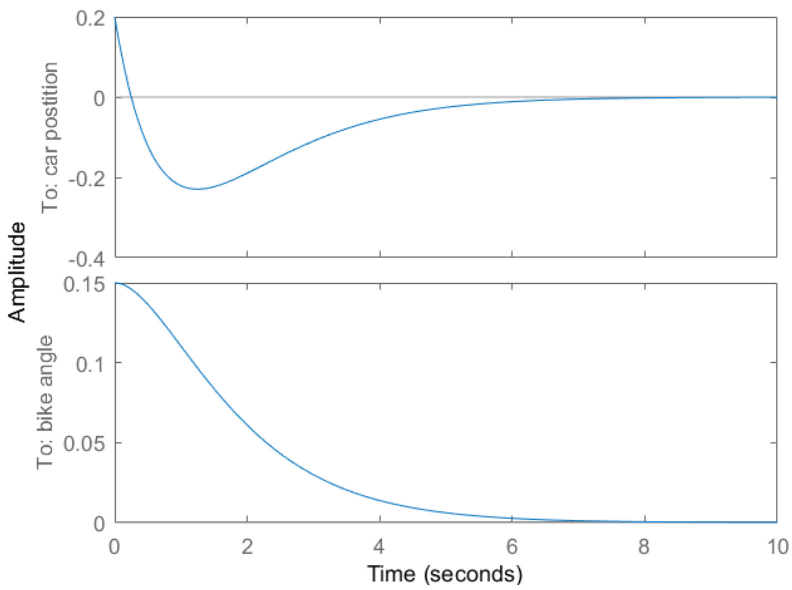


Figure 5.2 Zero input response with non-zero initial state

6 Servo Control

6.1 Integral system

The system with state space model shown below, and the disturbance of the system is a step input present at the plant.

$$\dot{x} = Ax + Bu + B_w w \tag{6.1}$$

$$y = Cx \tag{6.2}$$

The operating set point for the three outputs is

$$y_{sp} = -\frac{1}{10} CA^{-1}B \begin{bmatrix} -0.45 \\ -0.1 \end{bmatrix} = CA^{-1}B \begin{bmatrix} 0.045 \\ 0.01 \end{bmatrix} \tag{6.3}$$

Then we can get the error vector defined as

$$\dot{v}(t) = e(t) = y_{sp} - y \tag{6.4}$$

We put x and v together to form a new dynamic system

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A & O \\ -C & O \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} B \\ O \end{bmatrix} u + \begin{bmatrix} B_w \\ O \end{bmatrix} w + \begin{bmatrix} O \\ I \end{bmatrix} y_{sp} \tag{6.5}$$

$$y = \bar{C} \begin{bmatrix} x \\ v \end{bmatrix} \tag{6.6}$$

$$\bar{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 6.5 & -10 & -15.5 & 0 & 0 & 0 & \dots & 0 \\ -22.08 & 19.29 & 5.49 & -3.42 & -4.51 & 0.31 & 0 & \dots & 0 \\ 5 & -3.6 & 0 & 0 & 0 & -11.21 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{6.7}$$

$$\bar{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 26.5 & 11.2 \\ 5.85 & -1.69 \\ 40 & 60.5 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.8}$$

$$\bar{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.9}$$

6.2 Check the controllability

The controllability matrix is

$$Q_c = \begin{bmatrix} B & AB & A^2B & \dots \\ 0 & -CB & -CAB & \dots \end{bmatrix} = \begin{bmatrix} A & B \\ -C & 0 \end{bmatrix} \begin{bmatrix} 0 & B & AB & \dots \\ I & 0 & 0 & \dots \end{bmatrix} \tag{6.10}$$

The rank of the 9×8 matrix Q_c can be calculated by the matrix $\begin{bmatrix} A & B \\ -C & 0 \end{bmatrix}$, and we get its rank is 8, which is equal to the 2 inputs plus 6 state variables. As a result, the augmented system is controllable and can be stabilized by the state feedback control law

$$u = -[K_1 \quad K_2]\begin{bmatrix} \hat{x} \\ v \end{bmatrix} \tag{6.11}$$

6.3 Construct the feedback system

The resultant feedback system is

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & O \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} - \begin{bmatrix} B \\ O \end{bmatrix}[K_1 \quad K_2]\begin{bmatrix} \hat{x} \\ v \end{bmatrix} + \begin{bmatrix} B_w \\ O \end{bmatrix} w + \begin{bmatrix} O \\ I \end{bmatrix} y_{sp}$$

$$y = [C \quad O]\begin{bmatrix} x \\ v \end{bmatrix}$$

Due to the three cheap sensors for measuring the output, we can build an observer to get the estimation of the six states of the system to feedback to the original system. The system is built as figure 6.1.

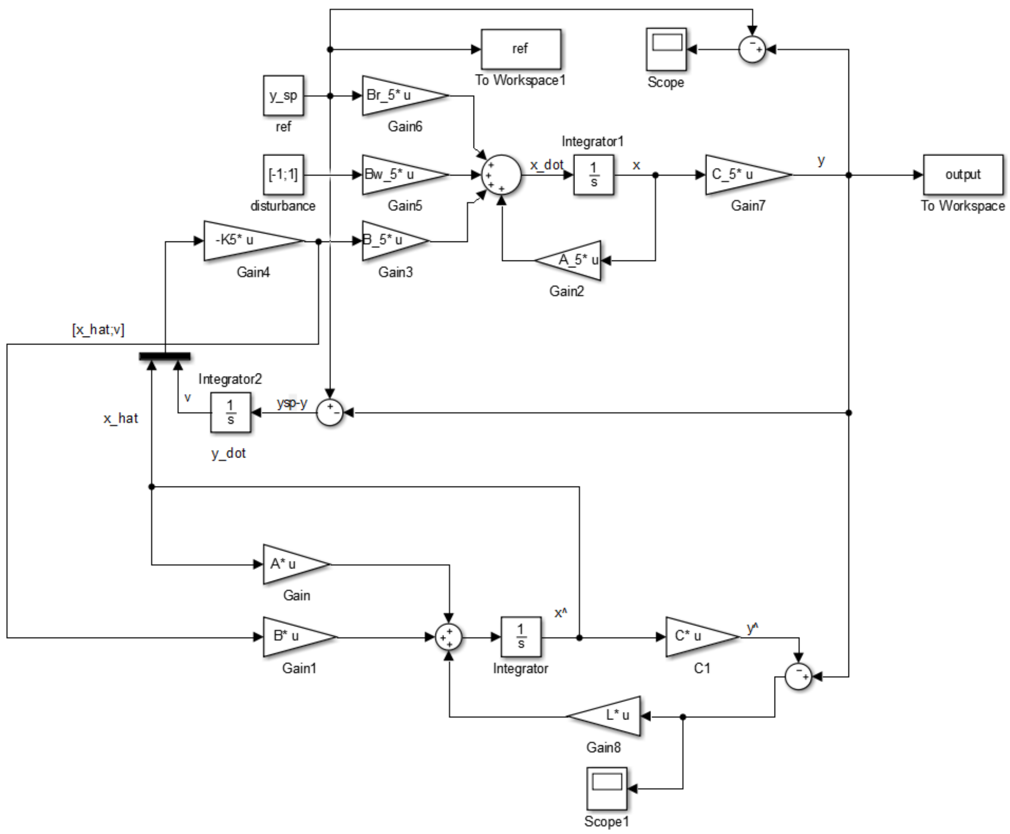


Figure 6.1 Servo control system

We apply the LQR method to solve the feedback gain matrix K with matrix Q and R as follows

$$Q = diag[5,10,1,0,0,0,0,0,0]\,, \tag{3.1}$$

$$R = I(2) \tag{3.2}$$

The feedback matrix K are calculated

$$K = \begin{bmatrix} 13.56 & -11.75 & -4.38 & 1.42 & -1.68 & 0.11 & -0.09 & 0.39 & -0.92 \\ -17.22 & 17.49 & 8.78 & -1.39 & 2.56 & 1.17 & -0.65 & -0.72 & -0.24 \end{bmatrix} \quad (3.10)$$

Then we can get the output signals and control signals as shown in Figure 6.2 and Figure 6.3. The system can track the setpoint in several seconds.

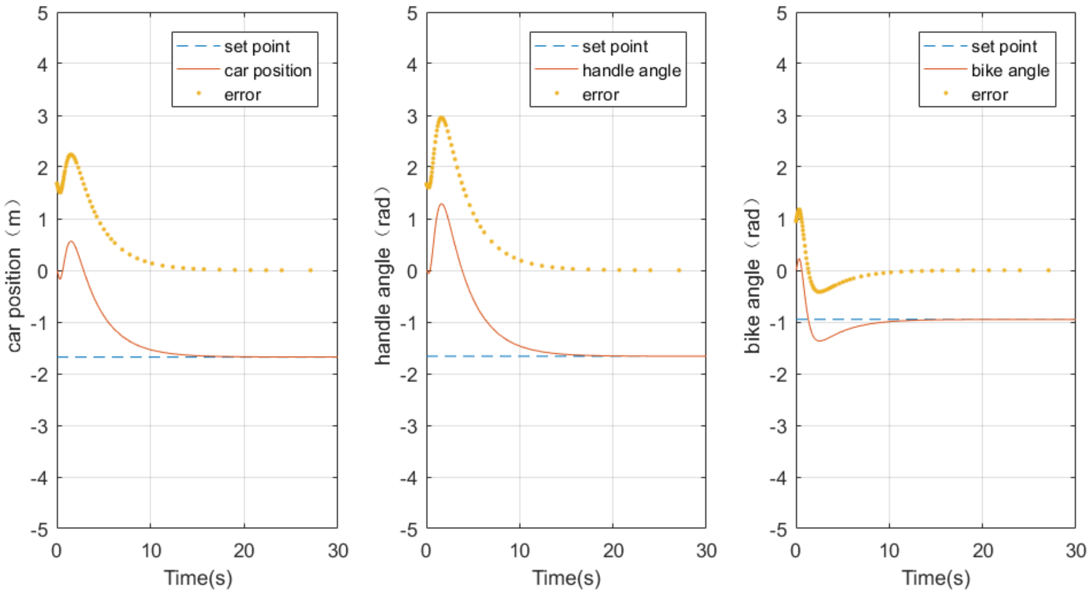


Figure 6.2 The response of the observer-based servo controller

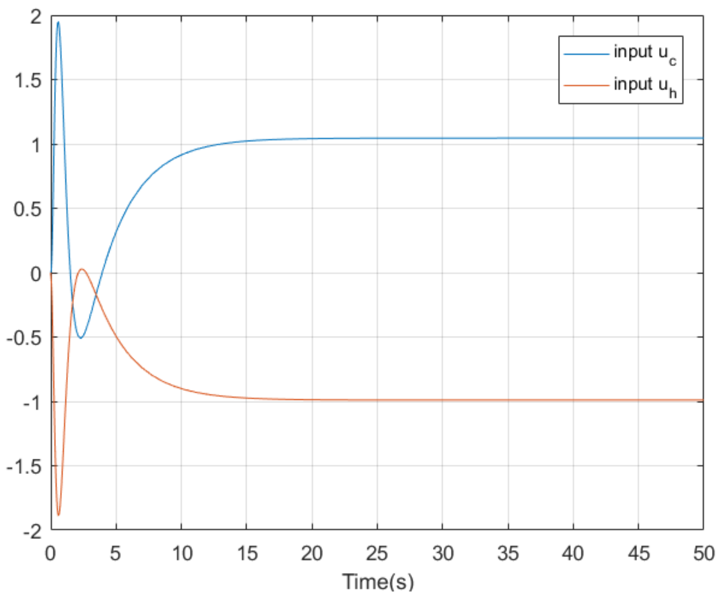


Figure 6.3 The control input of the observer-based servo controller

We can also get the step response and zero-input response of this system as figure 6.4 and figure 6.5. We can see the system is stable and. Due to the high order system, the selection of matrix Q and R are limited to make sure the system is controllable, the overshoot and settling time of the system may be larger than expected.

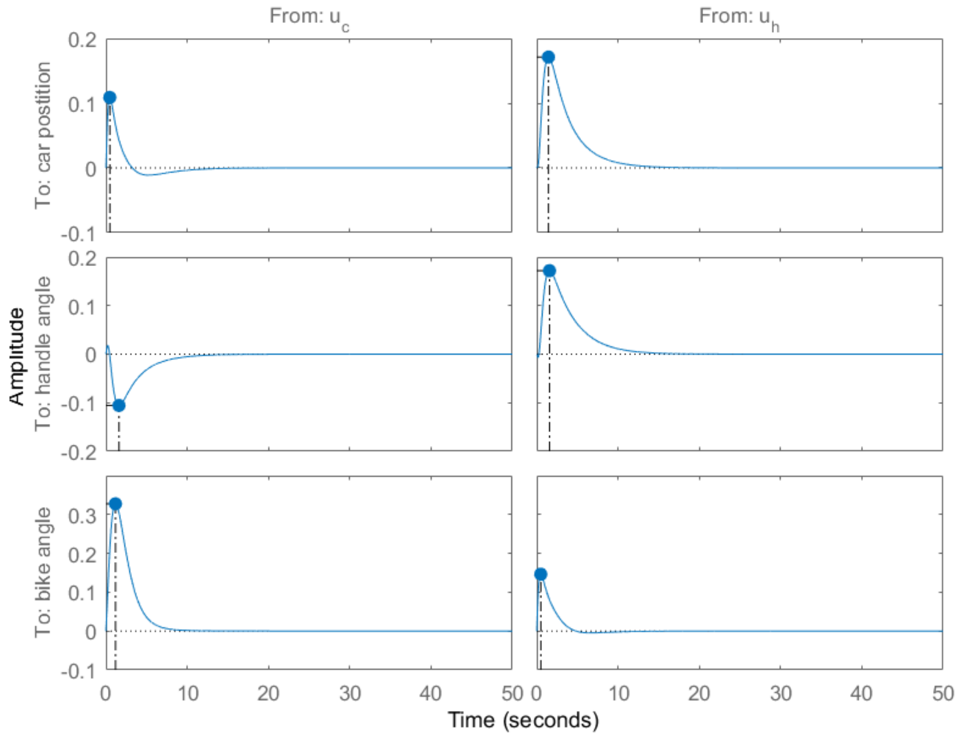


Figure 6.4 The step response of the observer-based servo controller

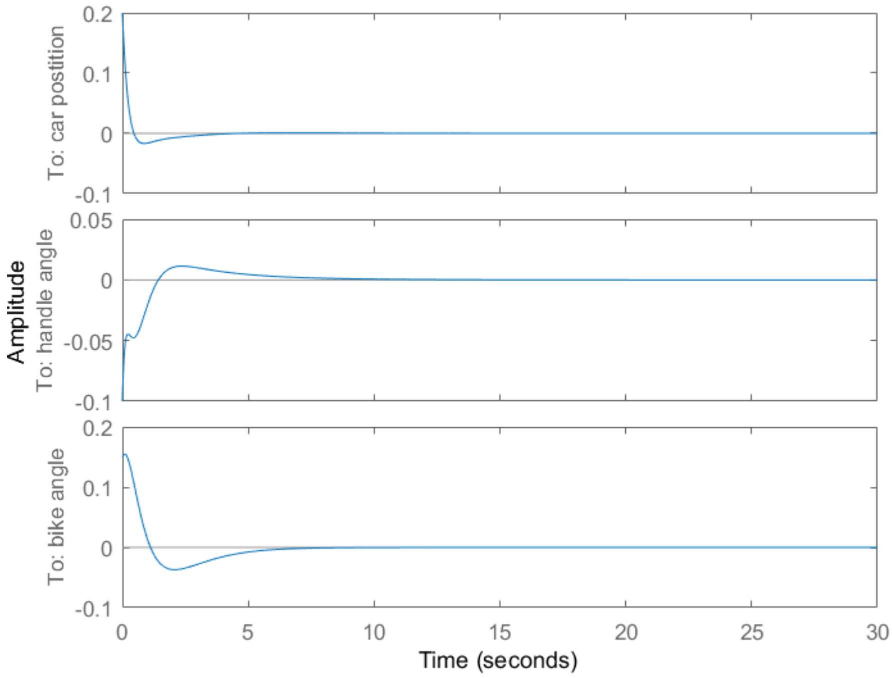


Figure 6.5 The zero input of the observer-based servo controller

7 Set-point tracking analysis

7.1 Theory analysis

Through the model of the system (6.5), if we want to track the setting point when time goes to infinity, then the output will be constant which means

$$\dot{v} = y_{sp} - y = -C * x + y_{sp} = 0 \tag{7.1}$$

$$y_{sp} = C * x \tag{7.2}$$

The first three dimension$[d(t) \quad \varphi(t) \quad \psi(t)]$ of the state variables will be constant (tracking the setting point) and the last three dimension$[\dot{d(t)} \quad \dot{\varphi(t)} \quad \dot{\psi(t)}]$ will also be zero. The equation of $\dot{x} = Ax + Bu$ will be zero when time goes to infinity which means

$$x = -A^{-1}B\,u \tag{7.3}$$

Then we can take the equation (7.3) into the equation (7.2) and get the

$$y_{sp} = -CA^{-1}B\,u \tag{7.4}$$

7.2 Tests for proof

So we can take an small input number which y_{sp} have to satisfy the transfer matrix $-CA^{-1}B$ with input. If we set the point randomly which means that we didn't satisfy the equation (7.1) like figure 7.1 and 7.2, we can not do setpoint tracking when time goes to infinity.

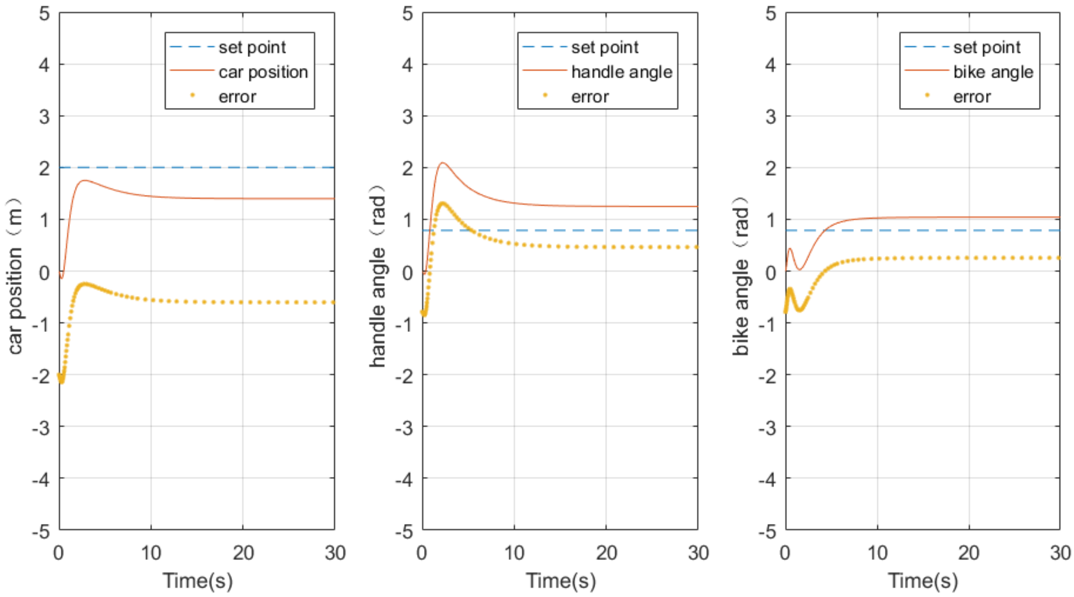


Figure 7.1 Set-point tracking ($y_{sp} = [2 \quad \frac{\pi}{4} \quad \frac{\pi}{4}]$)

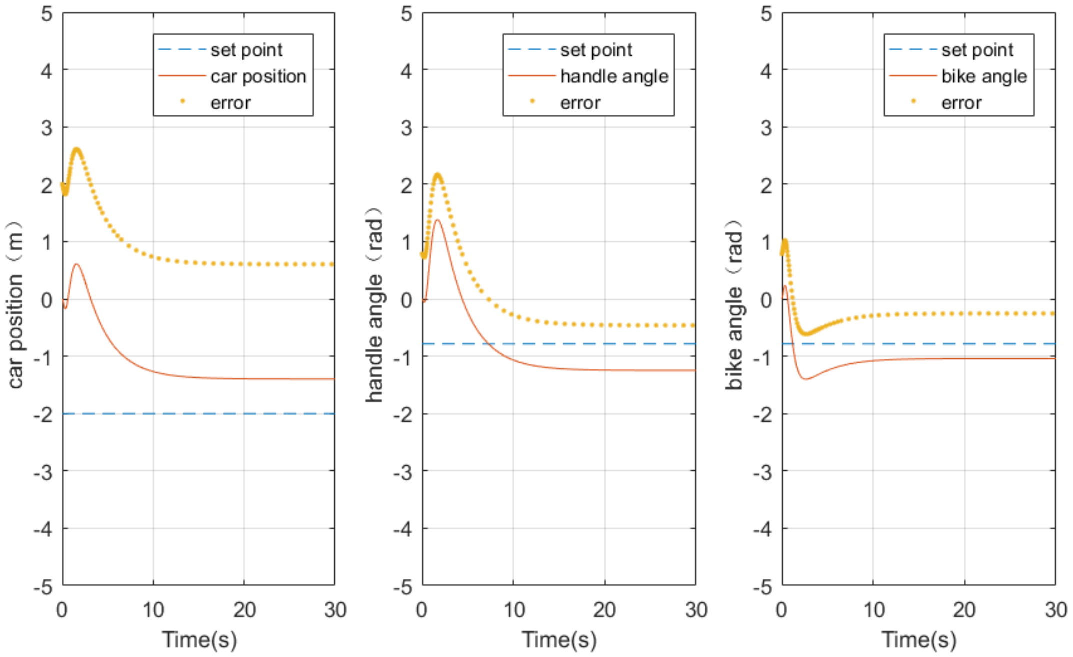


Figure 7.2 Set-point tracking ($y_{sp} = \begin{bmatrix} -2 & -\frac{\pi}{4} & -\frac{\pi}{4} \end{bmatrix}$)

If we set another number of y_{sp} which satisfy the equation (7.4), we can make the output track the setting point as expected (Figure 7.3 and Figure 7.4).

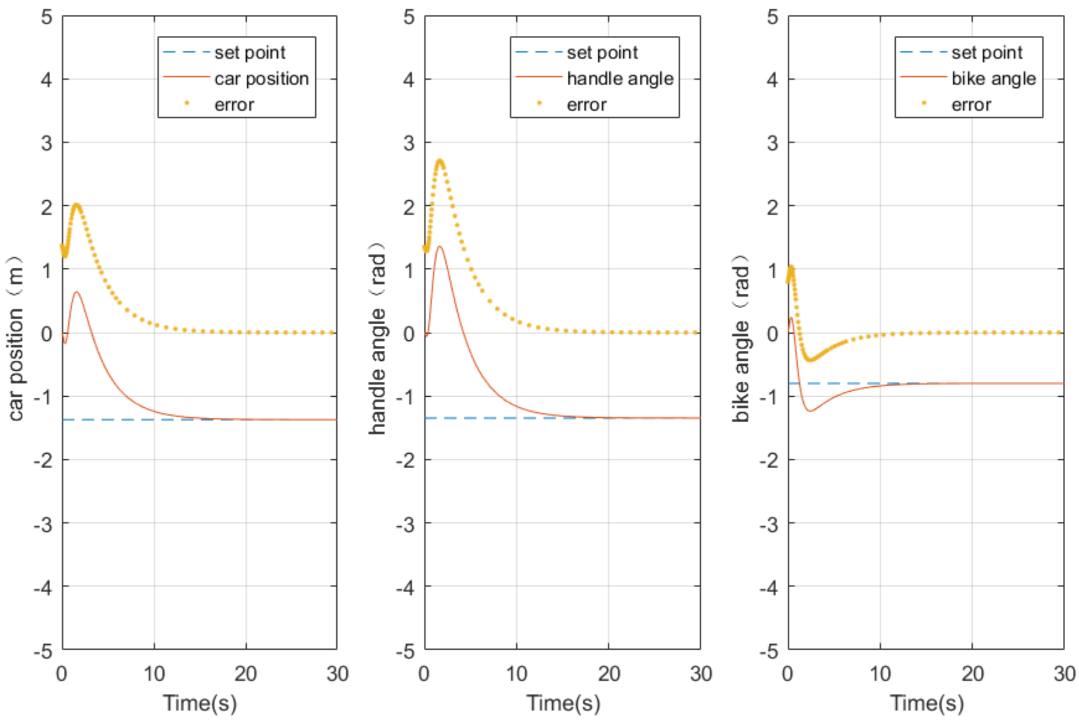


Figure 7.3 Set-point tracking ($y_{sp} = 0.02 * (-CA^{-1}B) * \mathrm{I}(2)$)

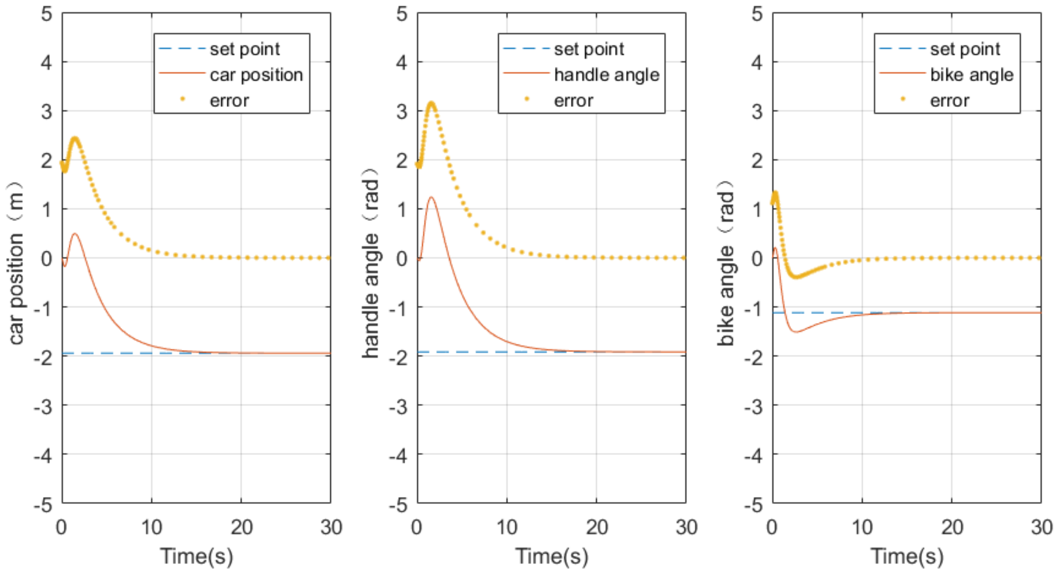


Figure 7.4 Set-point tracking ($y_{sp} = 0.02 * (-CA^{-1}B) * [0.2;\ 0.1]$)

Conclusion

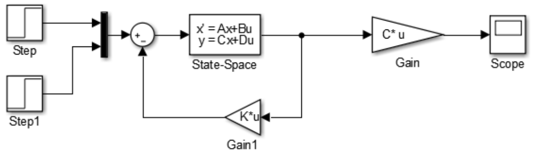
In the design process, I tried to apply the theory in a real problem and through the simulation in MATLAB, I can design the feedback properly and stabilize the system. Pole placement method and LQR method are commonly used to design the close-loop system to make it stable. The poles setting and Q, R setting analysis indicates the effect of parameters and we can design the proper system under specifications of the real condition. The observer design is an important part in the control of the system. Limited sensor measurement will limit the feedback states of the system. We can design the system with state estimation and get each state variable to feedback to the system. In the real case, the output may be random and we can select what we care about and design the decoupled system. Each input can control its corresponding output and analyze them separately which will bring a lot of convenience to the engineer. In control theory, a setpoint is the desired or target value of the output of the system. Departure of such a variable from its setpoint is one basis for error-controlled regulation using negative feedback for automatic control. The range of setpoint is also limited by the system and we did some analysis and proof. Through all the steps above, we finish the basic design and control of a linear system.

Appendices

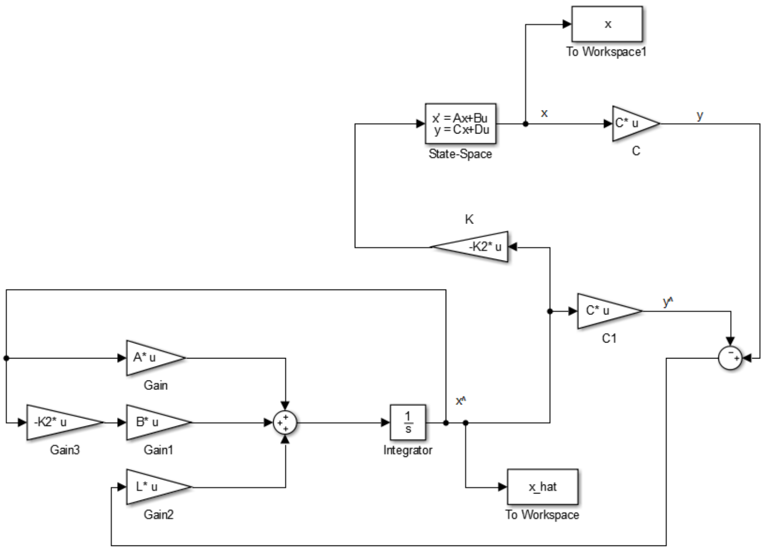
List of Figures

List of Tables

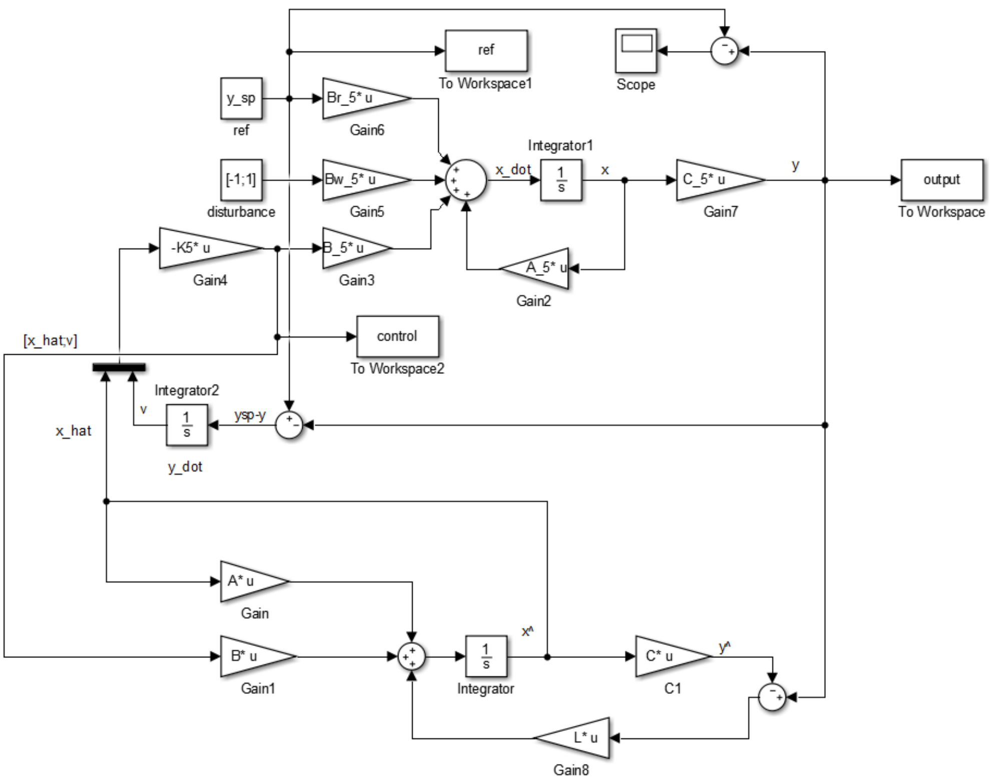
Simulink Model



The model of pole placement and LQR



State estimation model



Observer-based servo controller

MATLAB Code

```
clc
clear
set(0,'defaultfigurecolor','w')
%% initial condition
a = 3;
b = 2;
c = 5;
d = 2;

%parameters
Mf = 2.14 + c/20; %mass of front(kg)
Mr = 5.91 - b/10; %mass of body(rear part)(kg)
Mc = 1.74; %mass of rider's center-of-gravity (kg)
Hf = 0.18; %vertical length from floor to front part(m)
Hr = 0.161; %vertical length from floor to body(m)
Hc = 0.098; %vertical length from floor to rider's center-of-gravity(m)
LFf = 0.05; %Horizontal length from a front wheel rotation axis to a center-of-gravity of
part of front wheel
LF = 0.133; %Horizontal length from a front wheel rotation axis to a center-of-gravity of
part of steering axis
Lr = 0.128; %Horizontal length from a rear wheel rotation axis to a center-of-gravity of
part of rear wheel
LR = 0.308 + (a - d)/100; %Horizontal length from a rear wheel rotation axis to a center-of-
gravity of part of steering axis
Lc = 0.259; %Horizontal length from a rear wheel rotation axis to a center-of-gravity of the
cart system
Jx = 0.5 + (c - d)/100; %Moment of inertia around center-of-gravity x axially
ux = 3.33 - b/20 + a * c/60; %Viscous coefficient around x axis

alpha = 15.5 - a/3 + b/2;
beta = 27.5 - d/2;
gama = 11.5 + (a - c)/(b + d + 3);
delta = 60 + (a - b)*c/10;
g = 9.8;

%coefficient of matrix
den = Mf * Hf^2 + Mr * Hr^2 + Mc * Hc^2 + Jx;
a_51 = -(Mc * g)/den;
a_52 = (Mf * Hf + Mr * Hr + Mc * Hc)*g/den;
a_53 = (Mr * Lr * LF + Mc * Lc * LF + Mf * LFf * LR)*g/((LR + LF)*den);
a_54 = -(Mc * Hc * alpha)/den;
a_55 = - ux/den;
```

```
a_56 = Mf * Hf * LFf * gama/den;
b_51 = Mc * Hc * beta/den;
b_52 = - Mf * Hf * LFf * delta/den;

%matrix A, B, C
A = [0, 0, 0, 1, 0, 0;
    0, 0, 0, 0, 1, 0;
    0, 0, 0, 0, 0, 1;
    0, 6.5, -10, -alpha, 0, 0;
    a_51, a_52, a_53, a_54, a_55, a_56;
    5, -3.6, 0, 0, 0, -gama];

 B = [0, 0;
     0, 0;
     0, 0;
     beta, 11.2;
     b_51, b_52;
     40, delta];

 C = [1, 0, 0, 0, 0, 0;
     0, 1, 0, 0, 0, 0;
     0, 0, 1, 0, 0, 0];

 D = zeros(3,2);

states = {'d' 'phi' 'psi' 'd_dot' 'phi_dot' 'psi_dot'};
inputs = {'u_c' 'u_h'};
outputs = {'car postition'; 'handle angle'; 'bike angle'};
outputs2 = {'car postition';  'bike angle'};

 % initial state
 x0 = [0.2; -0.1; 0.15; -1; 0.8; 0];
 t = 0:0.01:10;
```

%% state space to transform function

```
sf = ss(A, B, C, D,'statename',states,'inputname',inputs,'outputname',outputs);
G = tf(sf);
%  step(sf ,t)
%  title('Step response of Open-Loop system')



%  lsim(sf,zeros(2,1001),t,x0)
%  title('response of non-zero initial state with zero initial inputs')
```

```
 %% check the controbaility
 con_matrix = ctrb(A, B);
 ro = rank(con_matrix);
```

%% pole set step

```
Co = zeros(size(A));
Co(:,1) = con_matrix(:,1);
Co(:,2) = con_matrix(:,3);
Co(:,3) = con_matrix(:,5);
Co(:,4) = con_matrix(:,2);
Co(:,5) = con_matrix(:,4);
Co(:,6) = con_matrix(:,6);
Co_inv  = inv(Co);

T(1,:) = Co_inv(3,:);
T(2,:) = Co_inv(3,:) * A;
T(3,:) = Co_inv(3,:) * A * A;
T(4,:) = Co_inv(6,:);
T(5,:) = Co_inv(6,:) * A;
T(6,:) = Co_inv(6,:) * A * A;
T_inv = inv(T);

format short
A_bar = T * A * T_inv;

format short
B_bar = T * B;

for i = 1:6
    for j = 1:6
        if(abs(A_bar(i,j)) < 0.001) A_bar(i,j) = 0;
        end
    end
end

for i = 1:6
    for j = 1:2
        if(abs(B_bar(i,j)) < 0.001) B_bar(i,j) = 0;
        end
    end
end

 %% pole set
%  syms s
```

```
 P1 = [-1+0.75i, -1-0.75i, -2, -2, -4, -4];%ideal poles
 P2 = [-0.8+0.6i, -0.8-0.6i, -1.5, -1.5, -3, -3];
 P3 = [-2+1.5i, -2-1.5i, -4, -4, -10, -10];
 K1 = place(A, B, P1);
%  K_bar = K1 * T_inv
%  A_d = A_bar - B_bar * K_bar
 K2 = place(A, B, P2);
 K3 = place(A, B, P3);
 sys_cl = ss(A-B*K1,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
 sys_cl2 = ss(A-B*K2,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
 sys_cl3 = ss(A-B*K3,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);

% lsim(sys_cl2,zeros(2,1001),t,x0)
% hold on
% lsim(sys_cl,zeros(2,1001),t,x0)
% hold on
% lsim(sys_cl3,zeros(2,1001),t,x0)

%  step(sys_cl2, t)
%  hold on
%  step(sys_cl, t);
%  hold on
%  step(sys_cl3, t)
%  legend('dominant poles with -0.8','dominant poles with -1','dominant poles with -2')
```

%% LQR

```
 Q = diag([5 10 1 0 0 0]);

 Q2 = diag([30 10 1 0 0 0]);
%  Q2 = diag([5 30 1 0 0 0]);
%  Q2 = diag([5 10 10 0 0 0]);

 Q3 = diag([100 10 1 0 0 0]);
%  Q3 = diag([5 100 1 0 0 0]);
% Q3 = diag([5 10 100 0 0 0]);

R = [0.5, 0;
     0, 0.5];

% R2 = [1, 0;0, 0.5];
% R2 = [0.5, 0;0, 1];

% R3 = [3, 0;0, 0.5];
```

```
% R3 = [0.5, 0;0, 3];

%LQR step
Tao_11 = A;
Tao_12 = (-1) * B * inv(R) * B';
Tao_21 = (-1) * Q;
Tao_22 = (-1) * A';
Tao =[Tao_11, Tao_12;
     Tao_21, Tao_22];
format short
[e_vector,e_value] = eig(Tao);

 K2 = lqr(A, B, Q, R);
 K22 = lqr(A, B, Q2, R);
% K22 = lqr(A, B, Q, R2);
 K23 = lqr(A, B, Q3, R);
% K23 = lqr(A, B, Q, R3);
 sys_c2 = ss(A-B*K2,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
 sys_c22 = ss(A-B*K22,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
 sys_c23 = ss(A-B*K23,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);

% step(sys_c2, t);

% lsim(sys_c2,zeros(2,1001),t,x0)
% step(sys_c2, t)
% hold on
% lsim(sys_c22,zeros(2,1001),t,x0)
% step(sys_c22, t)
% hold on
% lsim(sys_c23,zeros(2,1001),t,x0)
% step(sys_c23, t)

% title('response of non-zero initial state with zero initial inputs')


%  legend('Q(1,1)=5','Q(1,1)=30','Q(1,1)=100')
%  legend('Q(2,2)=10','Q(2,2)=30','Q(2,2)=100')
%  legend('Q(3,3)=1','Q(3,3)=10','Q(3,3)=100')
%  legend('Q(4,4)=0','Q(4,4)=10','Q(4,4)=100')
%  legend('Q(5,5)=0','Q(5,5)=10','Q(5,5)=30')
%  legend('Q(6,6)=0','Q(6,6)=10','Q(6,6)=30')
%  legend('R(1,1)=0.5','R(1,1)=1','R(1,1)=3')
%  legend('R(2,2)=0.5','R(2,2)=1','R(2,2)=3')
```

%% Decouple system

```
syms s
s_I = s*eye(6);
C2 = [1, 0, 0, 0, 0, 0;
     0, 0, 1, 0, 0, 0];
c1_TAB = [1, 0, 0, 0, 0, 0] *A*B;
c2_TAB = [0, 0, 1, 0, 0, 0] *A*B;
c1_TA2 = [1, 0, 0, 0, 0, 0] *(A*A + 2*A + eye(6));
c2_TA2 = [0, 0, 1, 0, 0, 0] *(A*A + 2*A + eye(6));
B_star = [c1_TAB;c2_TAB];
C_star = [c1_TA2;c2_TA2];
K4 = inv(B_star) * C_star;
F = inv(B_star);
% A_prim = A - B*K4;
% B_prim = B * F;
% simplify the coefficient of matrix B_prim and A_prim
B_prim = [0, 0;
    0, 0;
    0, 0;
    1, 0;
    0.36, 0;
    0, 1];

A_prim = [0, 0, 0, 1, 0, 0;
       0, 0, 0, 0, 1, 0;
       0, 0, 0, 0, 0, 1;
       -1, 0, 0, -2, 0, 0;
       -22, 16.6, 9.2, 1.5, -4.5, -0.6;
       0, 0, -1, 0, 0, -2];
H = C2 * inv(s_I - A_prim)* B_prim;

sys4 = ss(A_prim, B_prim, C2,
zeros(2,2),'statename',states,'inputname',inputs,'outputname',outputs2);
% step(sys4, t);
% lsim(sys4,zeros(2,1001),t,x0)
```

%% Servo control

```
% y_sp =0.1 * C * inv(A)* B * [-0.5+(a-b)/20; 0.1+(b-c)/(a+d+10)];
y_sp = -0.2 * C * inv(A)* B *[0.2;0.1];
% y_sp = [2;pi/4;pi/4];
% y_sp = [-2;-pi/4;-pi/4];

%controllability check
con_matrix2 = [A, B; -C, zeros(3,2)];
```

```
ro2 = rank(con_matrix2);

%% observer design
%check the observability
Vo = obsv(A,C);
P_obse = [-3+0.75i, -3-0.75i, -6, -6, -12, -12];%ideal
% P_obse = [-1+0.75i, -1-0.75i, -2, -2, -4, -4];
% P_obse = [-5+0.75i, -5-0.75i, -10, -10, -20, -20];
% P_obse = [-10+0.75i, -10-0.75i, -50, -50, -80, -80];
Lt = place(A',C',P_obse);
L = Lt';

% evalin('base','sim(''que3.slx'')')
%
% subplot(2,3,1)
% plot(x.time,x.data(:,1),x_hat.time, x_hat.data(:,1),x.time,x.data(:,1)-x_hat.data(:,1))
% legend('car position','car position estimation','car position error')
% ylabel('car position£¨m£©');
% xlabel('Time(s)');
% grid on
%
% subplot(2,3,2)
% plot(x.time,x.data(:,2),x_hat.time, x_hat.data(:,2),x.time,x.data(:,2)-x_hat.data(:,2))
% legend('handle angle','handle angle estimation','handle angle error')
% ylabel('handle angle£¨rad£©');
% xlabel('Time(s)');
% grid on
%
% subplot(2,3,3)
% plot(x.time,x.data(:,3),x_hat.time, x_hat.data(:,3),x.time,x.data(:,3)-x_hat.data(:,3))
% legend('bike angle','bike angle estimation','bike angle error')
% ylabel('bike angle£¨rad£©');
% xlabel('Time(s)');
% grid on
%
% subplot(2,3,4)
% plot(x.time,x.data(:,4),x_hat.time, x_hat.data(:,4),x.time,x.data(:,4)-x_hat.data(:,4))
% legend('car velocity','car velocity estimation','car velocity error')
% ylabel('car velocity£¨m/s£©');
% xlabel('Time(s)');
% grid on

% subplot(2,3,5)
% plot(x.time,x.data(:,5),x_hat.time, x_hat.data(:,5),x.time,x.data(:,5)-x_hat.data(:,5))
```

```
% legend('handle angular velocity','handle angular velocity estimation','handle angular velocity error')
% ylabel('handle angular velocity£¨rad/s£©');
% xlabel('Time(s)');
% grid on
%
% subplot(2,3,6)
% plot(x.time,x.data(:,6),x_hat.time, x_hat.data(:,6),x.time,x.data(:,6)-x_hat.data(:,6))
% legend('bike angular velocity','bike angular velocity estimation','bike angular velocity error')
% ylabel('bike angular velocity£¨rad/s£©');
% xlabel('Time(s)');
% grid on
```

%% Setpoint tracking

```
%stat space model
A_5 = [A,zeros(6,3);
      -C, zeros(3,3)];
B_5 = [B;zeros(3,2)];
C_5 = [C,zeros(3,3)];
D_5 = zeros(3,2);
Bw_5 = [B;zeros(3,2)];
Br_5 = [zeros(6,3);eye(3)];
Q_5 = eye(9);
Q_5(1,1) =5;
Q_5(2,2) = 10;
R_5 = eye(2);
K5 = lqr(A_5, B_5, Q_5, R_5);
% t2 = 0:0.01:30;
% x02 = [0.2; -0.1; 0.15; -1; 0.8; 0;0;0;0];
% sys_c5 = ss(A_5-B_5*K5,B_5,C_5,D_5,'inputname',inputs,'outputname',outputs);
% step(sys_c5, t2);
% lsim(sys_c5,zeros(2,3001),t2,x02)

for i=1:6
   K5_obse(:,i) = K5(:,i);
end

evalin('base','sim(''qus5.slx'')')
subplot(1,3,1)
plot(ref.time,ref.data(:,1),'--',output.time,output.data(:,1),output.time,output.data(:,1)-y_sp(1,1),'.')
legend('set point','car position','error')
axis([0 30 -5 5])
```

```
grid on
ylabel('car position£¨m£©');
xlabel('Time(s)');

subplot(1,3,2)
plot(ref.time,ref.data(:,2),'--',output.time,output.data(:,2),output.time,output.data(:,2)-
y_sp(2,1),'.')
legend('set point','handle angle','error')
axis([0 30 -5 5])
grid on
ylabel('handle angle£¨rad£©');
xlabel('Time(s)');

subplot(1,3,3)
plot(ref.time,ref.data(:,3),'--',output.time,output.data(:,3),output.time,output.data(:,3)-
y_sp(3,1),'.')
legend('set point','bike angle','error')
axis([0 30 -5 5])
grid on
ylabel('bike angle£¨rad£©');
xlabel('Time(s)');
%
% plot(control.time,control.data(:,1),control.time,control.data(:,2))
% legend('input u_c','input u_h')
% xlabel('Time(s)');
% grid on
```