Assignment for EE5101
Linear Systems

A0260167H

Li Weichen

e0974009@u.nus.edu

Nov 14th, 2022

Abstract

This assignment designed serval control systems for a modeled two-wheel motionless vehicle with evaluations for different parameters. In Task 1 I designed a controlling model with poles placement with different poles. In Task 2 I focused on designing LQR control with different Q/R. In Task 3 I designed a state observer. In Task 4 I decoupled the system into two SISO system. In Task 5 and 6 I focused on designing integral control for servo control and discussed with the reachability of the system.

Keywords: Linear System, Poles Placement, LQR, Observer, Decoupling, Integral Control

Contents table

1. Section 1 Introduction

1.1 Background

Self-sustaining two-wheelers not only demonstrate the development of control theory in the past few decades, but also have huge market potential. Due to this, many researchers from universities and companies are working on the subject.

Figure 1 Two-wheeled self-balancing electric car/motor

1.2 Modeling

The experimental system of the two-wheeled prototype is shown in Figure 2. A two-wheeler consists of three parts. There is a cart system corresponding to the rider's center of gravity movement, a steering system for steering (front), and a body (rear). The front part and the rear part are structures that are movable through a steering axis. A cart system and a steering system are driven by DC servo motor, and DC motors are controlled by servo amplifier which contains the velocity control system. Handle angle and cart position are measured by encoders. Attitude angles of the two-wheeled vehicle (roll angle and yaw angle) are measured by gyroscopes.

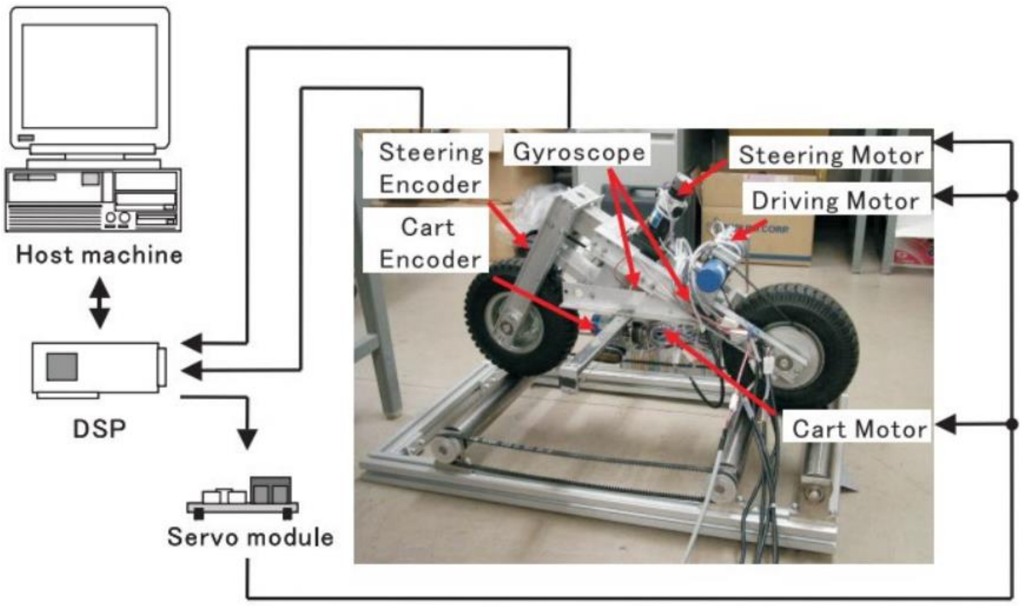

Figure 2 Composition of experimental system

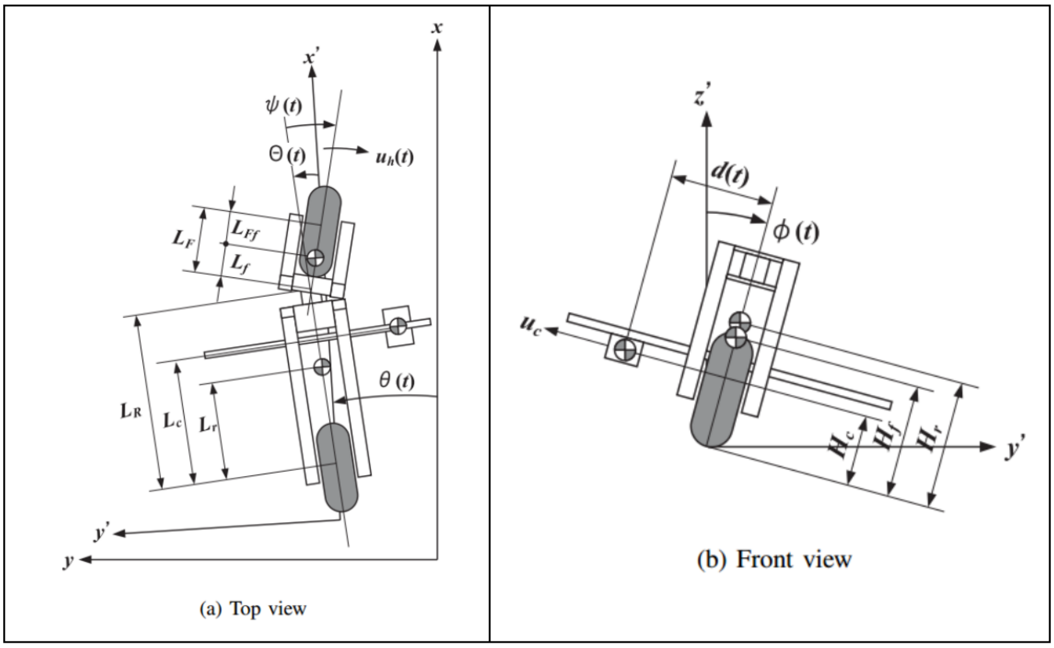

Figure 3 Two-wheeled vehicle structure model

The mechanical structure for the two-wheeled vehicle is given in Figure 3. The two-wheeled vehicle is stabilized by moving the cart position d(t) and adjusting the handle angle ψ(t) . The control inputs are the voltages u_c(t) and u_h(t) h u t to two DC servo motors, which drives the cart system and the steering system correspondingly. For the dynamic model, the relevant symbols are defined in Table 1.

Table 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

Thee state space linear model for the two-wheeled vehicle is derived to be:

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad x(0) = x_0 \\ y &= Cx \end{aligned} \tag{1}$$

where the state variable is:

$$x_0 = [d(t) \quad \phi(t) \quad \psi(t) \quad \dot{d}(t) \quad \dot{\phi}(t) \quad \dot{\psi}(t)]^T \tag{2}$$

and the matrices and the input vector are calculated by last 4 number of my matriculate number a = 0, b = 1, c = 6, d = 7:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 6.5 & -10 & -16 & 0 & 0 \\ -23.16 & 20.56 & 3.65 & -3.71 & -4.46 & 0.33 \\ 5 & -3.6 & 0 & 0 & 0 & -10.95 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 24 & 11.2 \\ 5.56 & -1.77 \\ 40 & 59.4 \end{bmatrix} \tag{3}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad D = \left[0\right], \quad u = [u_c(t), \quad u_h(t)]^T \tag{4}$$

Where g is the gravitational acceleration, $g \approx 9.8 m/s^2$.

1.3 Control System Design

After all, we get a linear state space model for the stationary two-wheeled vehicle. In the following, different control strategies will be explored to stabilize this vehicle to achieve its self-balance. We will target both the regulation and set point tracking problems. The initial condition for the two-wheeled vehicle system is assumed to be $x_0 = \left[0.2 \quad -0.1 \quad 0.15 \quad -1 \quad 0.8 \quad 0\right]^T$.

The transient response performance specifications for all the outputs y in state space model are as follows:

1) The overshoot := M_p is less than 10%.
2) The 2% settling time := t_s is less than 5 seconds.

2. Task 1 Poles Placement

Assume that all the six state variables can be measured in system of formula (1), I designed a state feedback controller using the pole place method, simulated the designed system and show all the six state responses to non-zero initial state with zero external inputs.

2.1 Reference model building

The reference model of a 2nd order is defined as:

$$H(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}. \tag{5}$$

First derive ζ and ω_n from design specifications:

$$10\%<M_p=e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}\Rightarrow\zeta>0.5912\Rightarrow\zeta=0.6 \tag{6}$$

$$t_s\cong\frac{4}{\zeta\omega_n}\Rightarrow\omega_n=\frac{4}{t_s\zeta}\Rightarrow\omega_n=\frac{4}{0.6*5}=1.33 \tag{7}$$

Reference model $\Rightarrow H_d(s)=\dfrac{1.778}{s^2+1.6s+1.778}=\dfrac{1.778}{(s-\lambda_1)(s-\lambda_2)}$, (8)

where λ_1,λ_2 represents the 2 roots of characteristic polynomial of reference model.

It can be obtained easily that $\mathrm{Re}(\lambda_1)=\mathrm{Re}(\lambda_2)$.

2.2 Design high order system

The characteristic polynomial for the system $\Phi_d(s)$ is defined as:

$$\phi_d(s):=\prod_{i=1}^{6}(s-\lambda_i) \tag{9}$$

As we should locate extra poles other than the dominant ones to be 2-5 times faster than the dominant ones, in my experiment, I design the other four λ_i as 3,3,3,3 times, 2,3,4,5 times and 5,5,5,5 times of $\mathrm{Re}(\lambda_1)$ respectively.

2.3 Controllability check for q

To observe the result more clearly, as B is matrix with size 6*2, I defined q as:

$$q=[0\quad 1] \tag{10}$$

Then I calculate the original controllability matrix of [B, AB, A^2B, A^3B, A^4B, A^5B] and feedback controllability matrix ω_c:= [Bq, ABq, A^2Bq, A^3Bq, A^4Bq, A^5Bq].

They are both full rank matrix, which shows the system is controllable.

The above checks remain the same for following part.

2.4 Solving K for poles replacement

The direct method would be computationally expensive and impossible to conduct. The Ackermann's formula can be shown by Caley-Hamilton Theorem that:

$$k^T = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]\omega_c^{-1}\phi_d(A)$$
$$\phi_d(A) = \phi_d(s)|_{s=A} \tag{11}$$

With k^T, we can solve the feedback transfer matrix A-BK by letting:

$$K = qk^T \tag{12}$$

The whole system is devised as:

$$\dot{x} = (A - BK)x + Bu$$
$$y = Cx \tag{13}$$

2.5 Simulation result and conclusion

The 6 variables are checked by adding dimensions to output matrix as C = eye(6). The disturbance and set point are assumed to be zero. The control signal and output in 10 seconds are monitored and plot as follows:

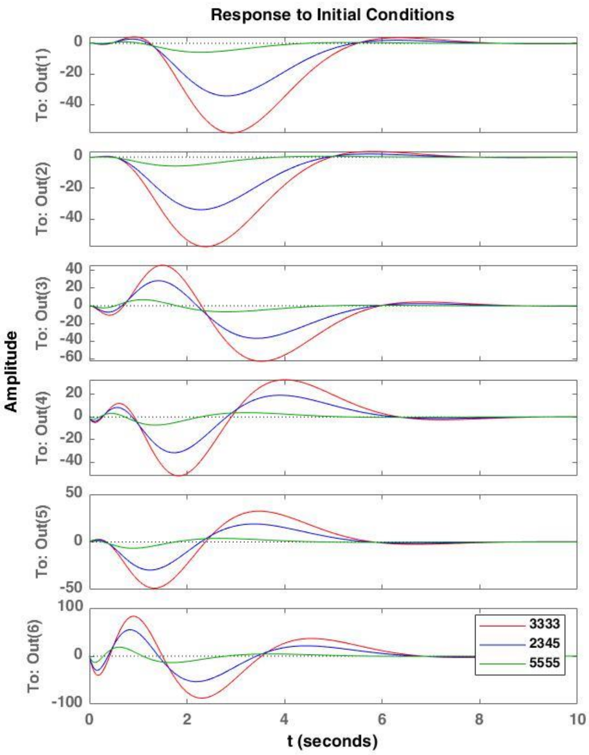

Figure 4. State monitor on poles placement

Other 4 poles from left to right are set as: 3-3-3-3, 2-3-4-5, 5-5-5-5 times of Re(λ). From up to down are:

$$x = [d(t) \quad \phi(t) \quad \psi(t) \quad \dot{d}(t) \quad \dot{\phi}(t) \quad \dot{\psi}(t)]^T$$

From the figure, it can be seen that the larger times other poles are of Re(λ), the better accuracy the system will have on initial conditions.

When other poles are set larger, it means how faster it decays than the dominant poles.

Meanwhile, all system have almost the same settling time.

This shows that it is the largest pole of that determines the quality of the system. This also deals with system in higher order.

2.6 Feedback result

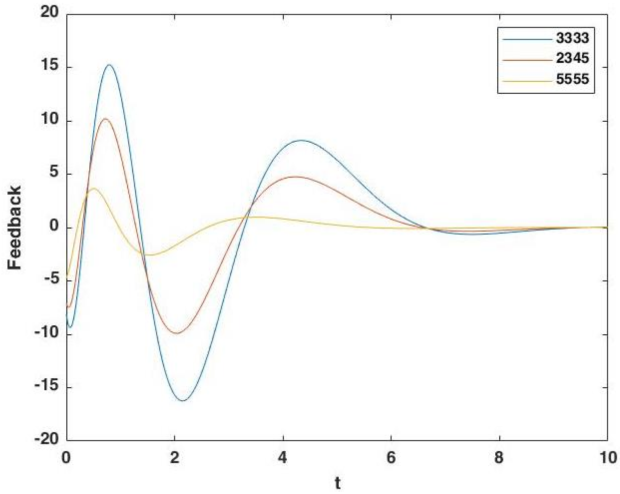

Figure5. Feedback monitor on poles placement

Feedback control is necessary to change the dynamic property of the system. In other words, good control (good closed-loop poles) is demanded.

From the figure we can see that as other poles getting larger, the cost is getting smaller, indicating a better performance on reducing variation.

This shows that the Ackermann's formula is effective for designing system's feedback.

3. Task 2 Quadratic Optimal Control

Assume that all the six state variables can be measured in system of formula (1), I designed a state feedback controller using the LQR method and simulated the designed system. In this step, both the disturbance and set point can be assumed to be zero. The cost function is defined as:

$$J = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)dt \tag{14}$$

Where u is defined as: $u = -Kx$. (15)

3.1 Solving ARE

To design feedback in formula (13), the most time-consuming part of LQR is solving the ARE as it involves a set of nonlinear equations. In my experiment, I solve ARE in steps as follows:

To find the positive definite solution of the Riccati equation:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{16}$$

I first form the 12*12 matrix:

$$\Gamma = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}, \tag{17}$$

and find its n stable eigenvalues.

Then, I found the eigenvector of Γ corresponding to stable λ_i, i = 1,2,..., 12, which means finding all λ_i whose real part are less than 0, and then form a matrix with dimension of 12*6 being:

$$\begin{bmatrix} v_i \\ \mu_i \end{bmatrix}, \quad i = 1, 2, ..., 12. \tag{18}$$

where v_i, μ_i are separated as two 6-dimensional vectors.

Therefore, shown in lecture note, we can have:

$$P[v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6] = [\mu_1 \quad \mu_2 \quad \mu_3 \quad \mu_4 \quad \mu_5 \quad \mu_6] \tag{19}$$

P is given by: $P = [\mu_1 \quad \mu_2 \quad \mu_3 \quad \mu_4 \quad \mu_5 \quad \mu_6][v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6]^{-1}$ (20)

With P, K can be obtained by:

$$K = -R^{-1}B^T P, \tag{21}$$

Then I simulated the system in formula (13).

3.2 Simulation Result with different Q

As is mentioned in lecture, Q and R are selected to be diagonal so that specific state and control variables are penalized individually with higher weightings if their response is undesirable.

Also, it is the relative ratio of Q to R matters. In my experiment, I set R as eye(2) and Q to different diagonal matrix eye(6), eye(6)*10, diag([1 1 4 5 1 4]) and diag([1 1 9 1 1 1]) respectively for simulation, the results on state and feedback are as follows:

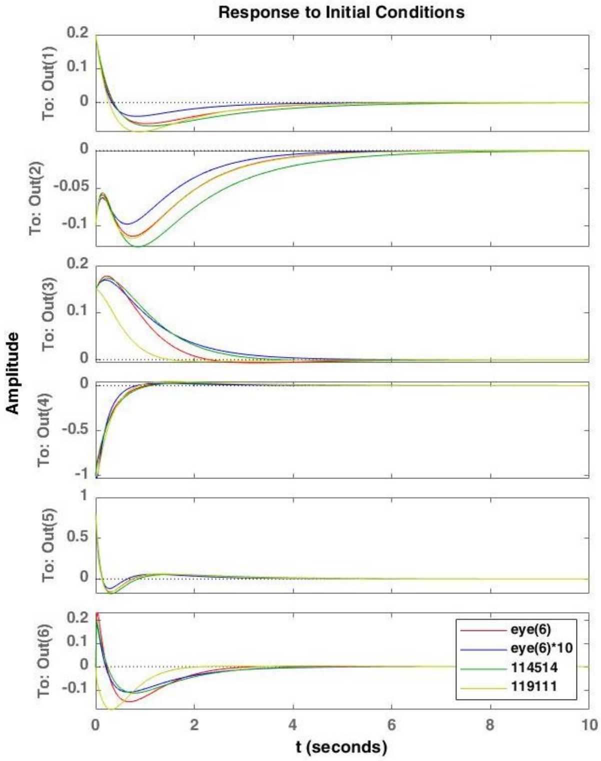

Figure 6. State monitor on LQR

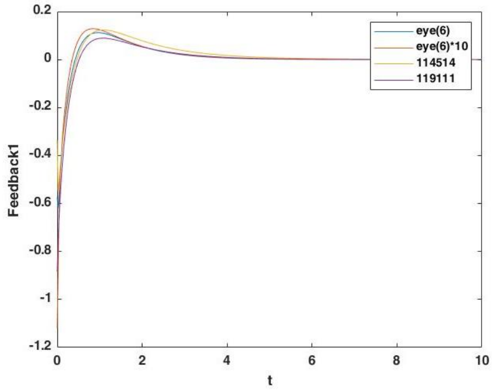

Figure 7. LQR feedback monitor on (u_1)

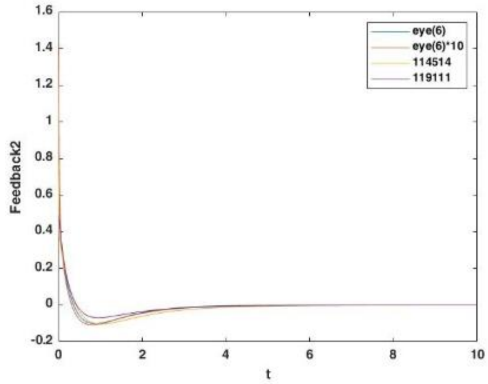

Figure 8. LQR feedback monitor on (u_2)

3.3 Conclusion

The larger the elements of Q are, the larger are the elements of the gain matrix K, and the faster the state variables approach zero. On the other hand, the larger the elements of R, the smaller the elements of K which leads to slower response but smaller energy cost.

In my simulation, the Q set as diag([1 1 9 1 1 1]) has the best performance, in that it has the fastest speed and the smallest power cost of control signal and best accuracy for most state. This shows that although Q are larger for better, the ratio of Q and R should be devised carefully.

4. Task 3 State Estimation

Assume only three outputs can be measured. I designed a state observer, simulate the resultant observer-based LQR control system, monitor the state estimation error, investigate effects of observer poles on state estimation error and closed-loop control performance. In this step, both the disturbance and set point can be assumed to be zero. The C is defined as:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

4.1 Full-order Observer

In the system of formula (1), we define the estimator:

$$\begin{aligned} &\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y - \hat{y}] \\ &\hat{y} = C\hat{x} \end{aligned} \tag{23}$$

Let the estimation error in the state be:

$$\tilde{x} = x - \hat{x} \tag{24}$$

Then, it follows that

$$\begin{aligned} \dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} = Ax + Bu - \{A\hat{x} + Bu + L(y - C\hat{x})\} \\ &= A(x - \hat{x}) - L(Cx - C\hat{x}) \\ &= A(x - \hat{x}) - LC(x - \hat{x}) \end{aligned}$$

$$\dot{\tilde{x}} = (A - LC)\tilde{x}, \quad \tilde{x}(0) = x(0) - \hat{x}(0) \tag{25}$$

For the observer problem, write its error characteristic polynomial:

$$\det[sI - (A - LC)] = \det[sI - (\tilde{A} - \tilde{B}\tilde{K})], \tag{26}$$

where $\tilde{A} = A^T$, $\tilde{B} = C^T$, $\tilde{K} = L^T$.

4.2 Poles placement

In this, I set other four poles different times of Re(λ_1). Then conduct poles placement solving $\tilde{K}$ for the system:

$$\dot{\tilde{x}} = (\tilde{A} - \tilde{B}\tilde{K})x + \tilde{B}u \tag{27}$$

Because $\tilde{B}$ is a matrix with size of 6*3, for clear observation the $\tilde{q}$ is defined as:

$$\tilde{q} = [0 \quad 0 \quad 1], \tag{28}$$

in replacement of formular (10) in Task 1.

At last, we can obtain L by letting $L := \tilde{K}^T$ in formula (26).

4.3 LQR control system

In 3.3 Conclusion I obtained the Q set as diag([1 1 9 1 1 1]) has the best performance for calculating matrix K for feedback. Here I directly apply the K calculated in the former part.

4.4 Simulation model

The system model is built in Simulink with screenshot as follows:

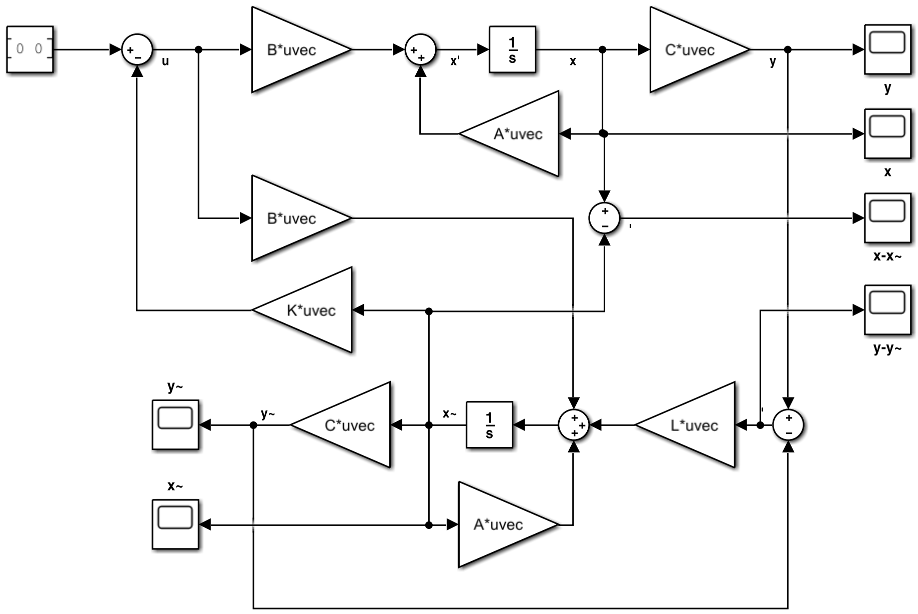

Figure 9. Simulink screenshot on observer

4.5 State estimation error

Though Y selects merely the former 3 variables of X by Y=CX, to plot Y more clearly, the result on each observer poles are plot as follows: (left: $x - \hat{x}$, right: $y - \hat{y}$).

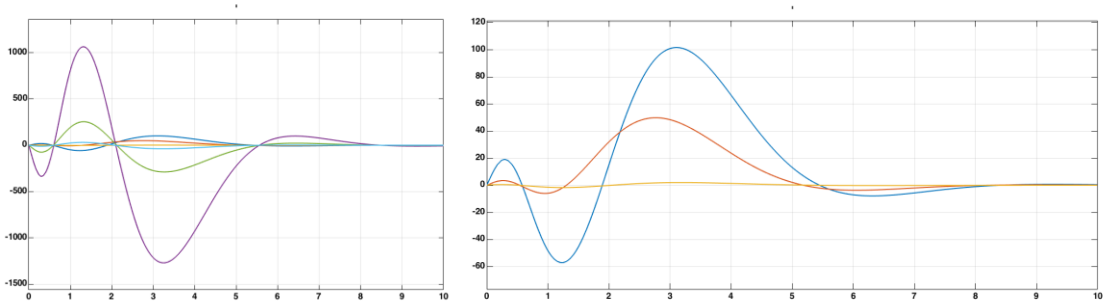

Figure 10. State estimation error on observer poles set 3,3,4,4 times of Re(λ_1).

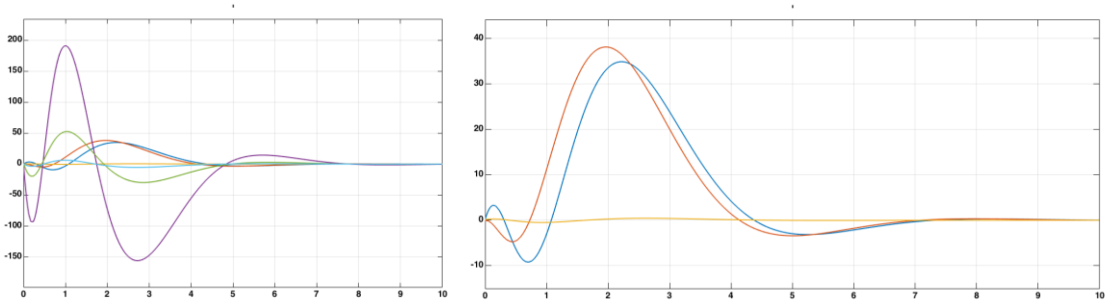

Figure 11. State estimation error on observer poles set 4,5,6,7 times of Re(λ_1).

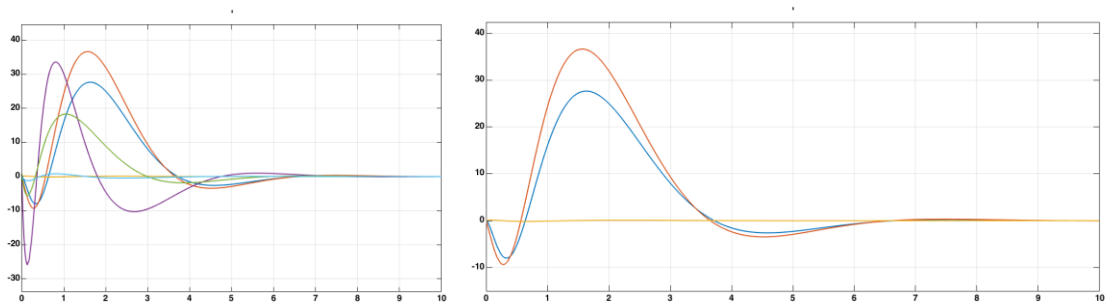

Figure 12. State estimation error on observer poles set 8,8,8,8 times of $\mathrm{Re}(\lambda_1)$.

The result in Figure 10-12 shows that larger times observer poles are of $\mathrm{Re}(\lambda_1)$, the faster speed would the state estimation error decay to balance, and also the better accuracy can be reached, which indicates a better performance.

4.6 Closed-loop control performance

As Y=CX, I plot X to show the full closed-loop control performance of each observer poles as follows:

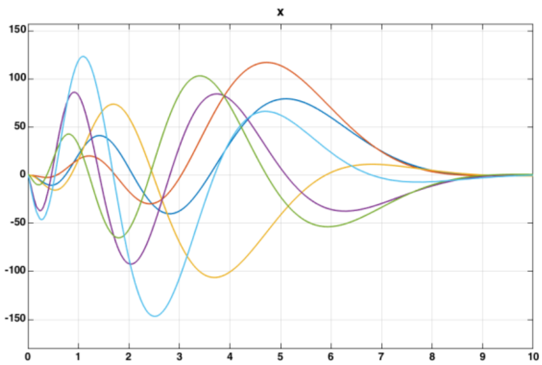

Figure 13. Closed-loop control performance on observer poles set 3,3,4,4 times of $\mathrm{Re}(\lambda_1)$.

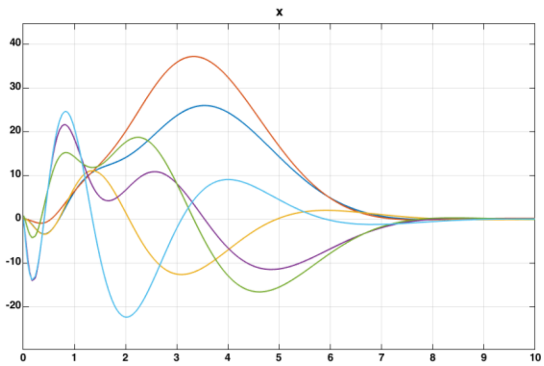

Figure 14. Closed-loop control performance on observer poles set 4,5,6,7 times of $\mathrm{Re}(\lambda_1)$.

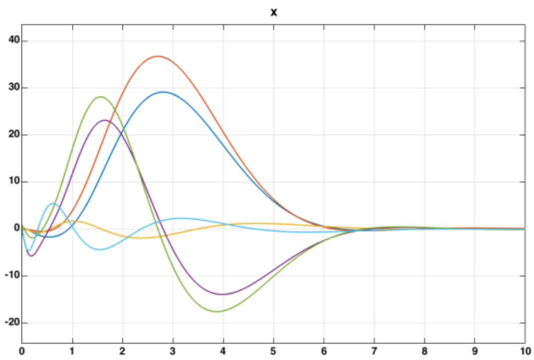

Figure 15. Closed-loop control performance on observer poles set 8,8,8,8 times of $\mathrm{Re}(\lambda_1)$.

The result in Figure 13-15 shows that larger times observer poles are of $\mathrm{Re}(\lambda_1)$, the faster speed would the state decay to balance, the better accuracy would the system achieve, which indicates a better closed-loop performance in engineering.

5. Task 4 Decoupling Control

Suppose we are only interested in the two outputs d(t) and ψ(t), a new output matrix is

$$C=\begin{bmatrix}1&0&0&0&0&0\\0&0&1&0&0&0\end{bmatrix} \tag{29}$$

then we get a 2-input-2-output system. I designed a decoupling controller with closed-loop stability and simulate the step set point response of the resultant control system to verify decoupling performance with stability. In this step, the disturbance is assumed to be zero.

5.1 Decoupling by state feedback

Consider the system in formula (1), the open loop transfer function matrix:

$$G(s)=C(sI-A)^{-1}B \tag{30}$$

Let the state feedback controller $u=-Kx+Fr$. (31)

The closed loop system is

$$\begin{aligned}\dot{x}&=(A-BK)x+BFr\\y&=Cx\end{aligned}, \tag{32}$$

and the transfer function matrix of the feedback system is

$$H(s)=C(sI-A+BK)^{-1}BF. \tag{33}$$

The relationship between open loop G(s) and the closed loop H(s) is derived as

$$\begin{aligned}H(s)&=G(s)[I-K(sI-A+BK)^{-1}B]F\\&=G(s)[I+K(sI-A)^{-1}B]^{-1}F.\end{aligned} \tag{34}$$

5.2 Calculation of σ_i, B^*, C^*

As C is a 2*6 matrix, partition C into 2 row vectors.

$$C=\begin{bmatrix}c_1^T\\c_2^T\end{bmatrix} \tag{35}$$

The σ_i, i=1, 2, …, m, is defined as an integer by

$$\sigma_i=\begin{cases}\min(j\,|\,c_i^T A^{j-1}B\neq 0^T,\ j=1,2,...,n);\\ n,\quad if\quad c_i^T A^{j-1}B=0^T,\ j=1,2,...,n.\end{cases} \tag{36}$$

G(s) can be written as

$$G(s)=\begin{bmatrix}s^{-\sigma_1}&0\\0&s^{-\sigma_2}\end{bmatrix}[B^*+C^*(sI-A)^{-1}B], \tag{37}$$

where B^* and C^* are

$$B^* = \begin{bmatrix} c_1^T A^{\sigma_1 - 1} B \\ c_2^T A^{\sigma_2 - 1} B \end{bmatrix}, \quad C^* = \begin{bmatrix} c_1^T \phi_{f1}(A) \\ c_2^T \phi_{f2}(A) \end{bmatrix}, \tag{38}$$

where $\phi_{fi}(A) = A^{\sigma_i} + \gamma_{i1} A^{\sigma_{i-1}} + \cdots + \gamma_{i\sigma_i} I$, (39)

in which $\phi_{fi}(s) = s^{\sigma_i} + \gamma_{i1} s^{\sigma_{i-1}} + \cdots + \gamma_{i\sigma_i} = \prod_{j=1}^{\sigma_i} (s - \lambda_j)$, (40)

where λ_i are sorted in ascending sequence of their real part.

5.3 System model building

The matrix F and K can be obtained with B^* and C^* as

$$F = (B^*)^{-1}, \quad K = (B^*)^{-1} C^*. \tag{41}$$

Then we can build the closed loop system in formula (32).

5.4 Initial and step response

The initial and step response of the system are shown as follows.

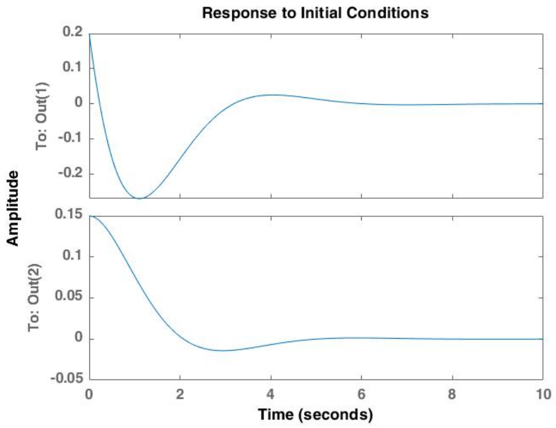

Figure 16. Initial response on system decoupled by state feedback

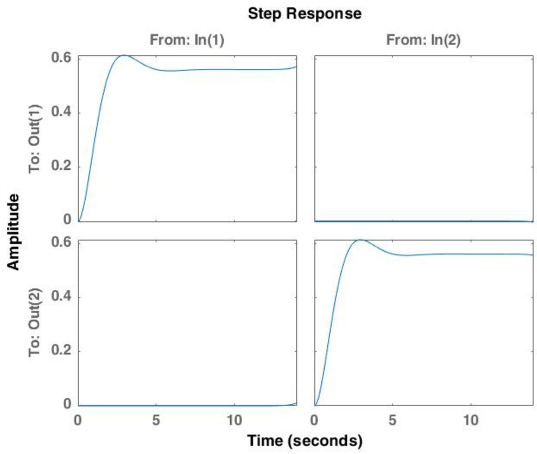

Figure 17. Step response on system decoupled by state feedback

5.5 Conclusion

The result from Figure 16-17 shows that the decoupled system internally stable. Figure 17 shows the success in decoupling the system into two SISO system, with 2% setting time and overshoot amplitude both meeting the requirements in Section 1 Introduction.

6. Task 5 Servo Control

The operating set point for the three outputs calculated by my matricular number, where C set as formula (22).

$$y_{sp} = -\frac{1}{10} C A^{-1} B \begin{bmatrix} \text{-0.5+}\frac{\text{a-b}}{20} & \frac{\text{0.1+b-c}}{\text{a+d+10}} \end{bmatrix}^T = \begin{bmatrix} 2.551 & 2.611 & 1.544 \end{bmatrix}^T \tag{42}$$

Therefore, the objective of the controller is to maintain the output vector around this operating set point as close as possible.

Assume there are only three cheap sensors to measure the output. I designed a controller such that the plant (vehicle) can operate around the set point as close as possible at steady state even when step disturbances are present at the plant input. In my simulation, I assume the step disturbance for the two inputs, $\omega = [-1, 1]^T$ takes effect at time $t_d = 10s$.

6.1 LQR for augmented system

The augmented servo control system can be expressed by

$$\begin{gathered} \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A & O \\ C & O \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} B \\ O \end{bmatrix} u \\ \dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u \\ y = [C \quad 0] \begin{bmatrix} x \\ v \end{bmatrix} = \bar{C}\bar{x} \end{gathered} \tag{43}$$

Then, we solve K with LQR method in Task 2, with $\bar{A}, \bar{B}$ and $\bar{C}$.

6.2 Servo control system

Assume all states can be measured, with noise set as ω start at $t_d = 10s$, I firstly build the servo control system as follows.

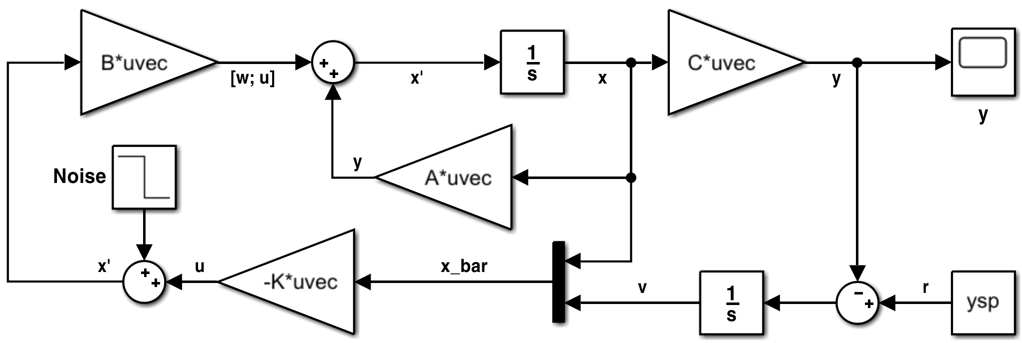

Figure 18. Integral control system by state error feedback

The performance of this servo control is as follows.

The result shows that with all state can be measured, the designed system can adjust to the impulse noise and return to the balance within the requirements. With this successful servo controller, I then further build an observer to replace the states with their estimates in the controller.

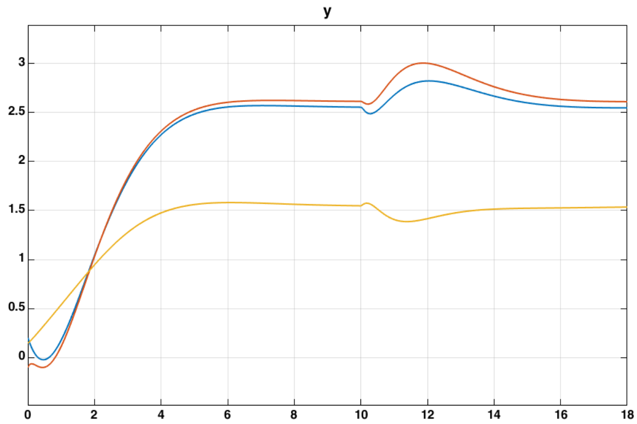

Figure 19. Step response of integral control system

6.3 Result with an observer

Assume there are only three cheap sensors to measure the output. The C is defined as formula (22). An observer is established with steps in Task 3 to calculate L. The whole system is shown as follows.

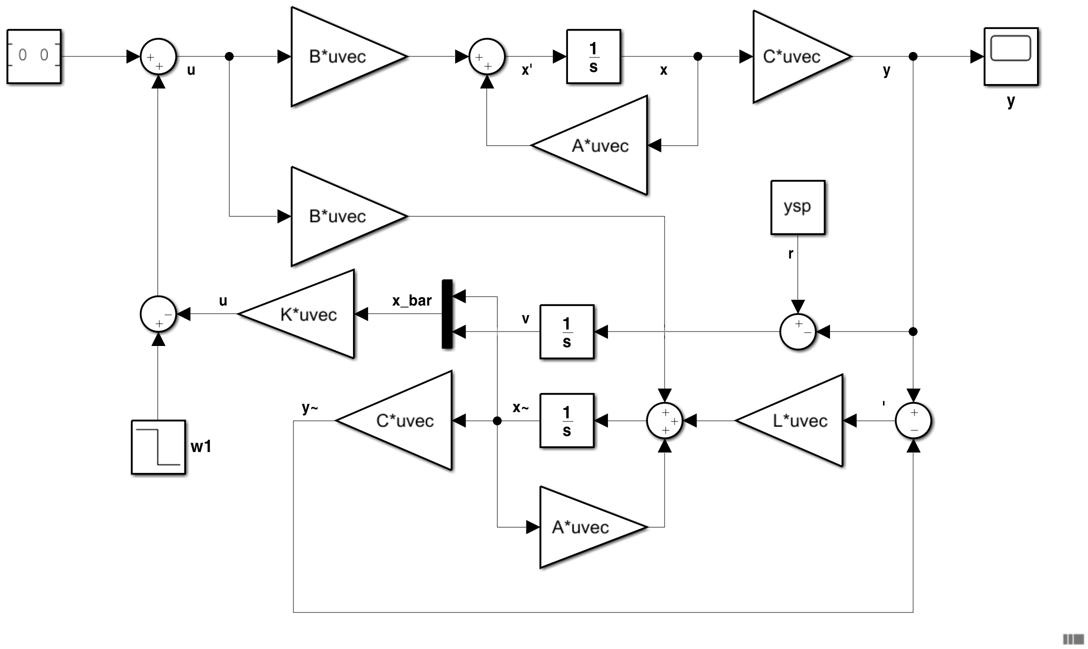

Figure 20. Integral control system by estimated state error feedback

The performance of this servo control by estimation is as follows.

The results shows that although with worse accuracy and setting speed compared with original integral control system with all six states, the system can still reach the balance within the requirements, which indicates with the cheap sensors we can also achieve the ideal result in engineering.

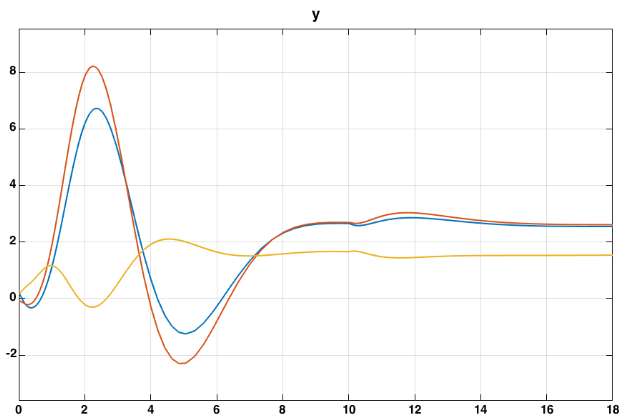

Figure 21. Step response of system controlled by estimation

7. Task 6 Arbitrary Servo Control

With the knowledge of the multivariable integral control using state space model in Chapter 9, it is a classical way to solve the set point tracking problem even when a constant disturbance is involved. However, we cannot maintain the three outputs at an arbitrary constant set point with zero steady-state error.

7.1 Mathematical proof

In the integral control system build in Task 5 by LQR, the augmented servo control system was expressed by formula (43). However, if we check the rank for the controllability matrix of this augmented system, we can find that

$$rank\begin{pmatrix} A & B \\ C & 0 \end{pmatrix} = 8 < n+m = 6+3 = 9, \tag{44}$$

which shows that this system is not fully controllable.

The augmented system can be divided into a fully controllable sub-system with rank = 8 and a uncontrollable system with rank = 9-8 = 1.

From the former part, we already knew that the original open-loop system in formula (1), is a fully controllable subsystem with rank = 6 of this augmented system. Therefore, it can be determined easily that the uncontrollable subsystem is in the other 3 dimension in the feedback matrix K. This indicates that the 3 dimensional y_{sp} can only be controlled in 2 dimensions, that is, the reachable y_{sp} locates in a plane in its 3D space. In formula (30), the open-loop system decay to stable when G(s) = 0, which can obtain the parametric 2D plane equation as

$$y_{sp} = CA^{-1}B[u_1 \quad u_2]. \tag{45}$$

in the 3D space of y_{sp} with arbitrary real value parameters $[u_1, u_2]$.

7.2 Simulation for verification

The integral control system is built in Figure 18. Different value of y_{sp} within or outside the plane defined in Equation (45) are tried to verify the conclusion. The results are as follows.

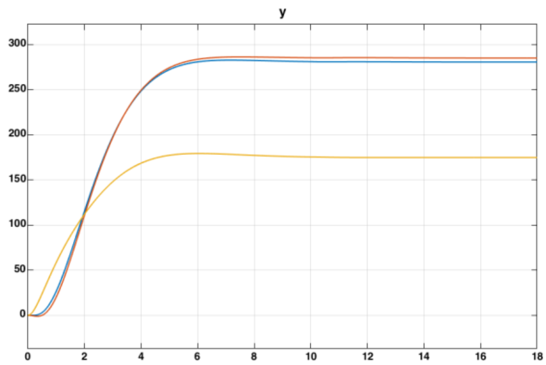

Figure 22. Step response with y_{sp} on the plane y_{sp} = C/A*B*[2; 5], $y_{sp} \approx$ [281 285 174].

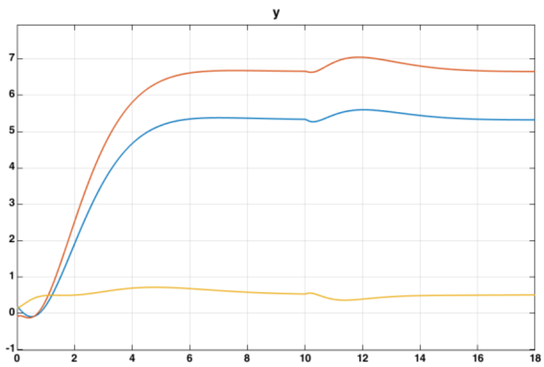

Figure 23. Step response with y_{sp} not on the plane, y_{sp} = [1 10 2].

7.3 Controllability decomposition

In the former part I have discussed that the 3D output can only be controlled in a 2D plane. It is naturally to ask whether the system can be designed controlling 2 or less output to arbitrary value. To construct this system, we select to control two outputs by defining C as formula (29). Then we construct the integral control system with steps in Task 5. We define the new y'_{sp} as

$$y'_{sp} = \begin{bmatrix} y_{sp1} & y_{sp3} \end{bmatrix}. \tag{46}$$

for the state to be reached. All the other parameters remain the same for this augmented system.

7.4 Simulation result on arbitrary integral control

The integral system is built as follows, with y_sp representing the new y'_{sp}. It is observed that all the new y'_{sp} are reachable with whatever real values.

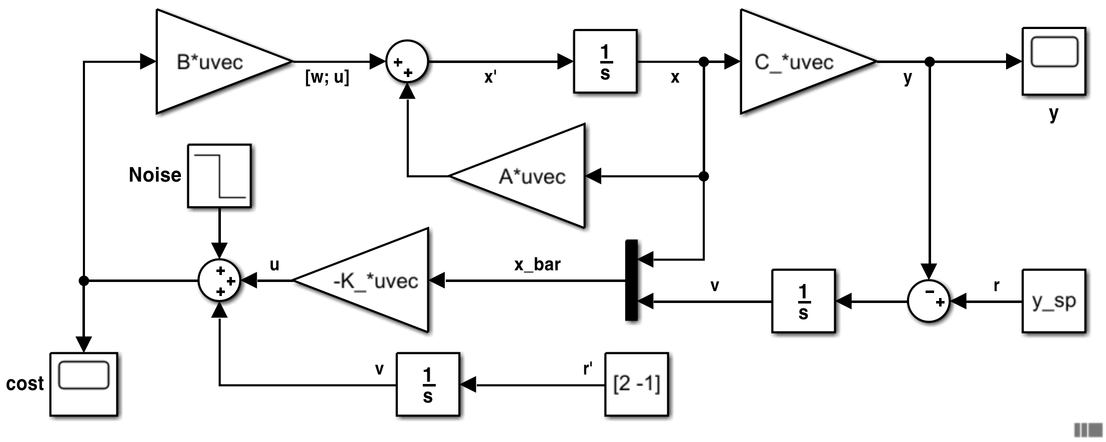

Figure 24. Arbitrary integral control system by state error feedback

It is also worth mentioning that, adding a 2D constant integral control after feedback obtains the resemble result on servo controlling. Some results are plot as follows

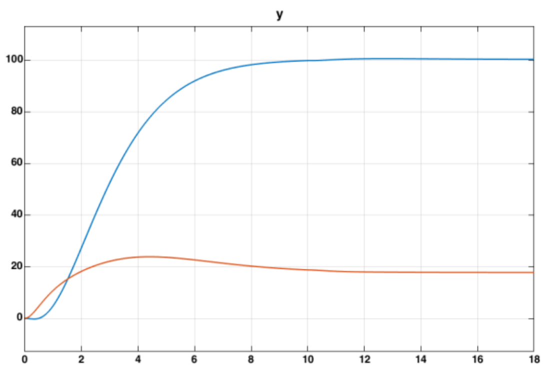

Figure 25. Step response on system with integral feedback [-20 100]

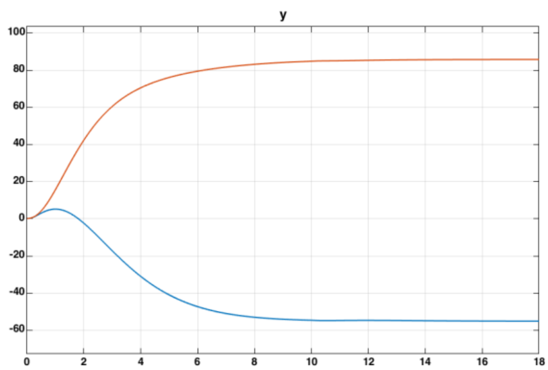

Figure 26. Step response on system with integral feedback [100 -20]

This illustrates that y_{sp} with whatever dimension, controls the output by mapping itself into the 2D feedback space, and further illustrates that a system cannot control the output with dimension larger than the lines of its B matrix. However, the relationship between y_{sp} and this equivalent signal is still unknown and deserves further study.

Appendix

1. ABCD.m

```
clc
clear
a = 0; b = 1; c = 6; d = 7;
Mf = 2.14 + c/20; Hf = 0.18;
Mr = 5.91 - b/10; Hr = 0.161;
Mc = 1.74; Hc = 0.098;
LFf = 0.05; LF = 0.133;
Lr = 0.128; LR = 0.308 +(a-d)/100;
Lc = 0.259; g = 9.8;
Jx = 0.5+(c-d)/100; mux = 3.333 - b/20 + a*c/60;
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;
den = Mf*Hf*Hf+Mr*Hr*Hr+Mc*Hc*Hc+Jx;
a51 = -Mc*g/den; a52 = (Mf*Hf+Mr*Hr+Mc*Hc)*g/den;
a53 = (Mr*Lr*LF+Mc*Lc*LF+Mf*LFf*LR)*g/(LR+LF)*den;
a54 = -Mc*Hc*alpha/den; a55 = -mux/den; a56 =
Mf*Hf*LFf*gama/den;
b51 = Mc*Hc*beta/den; b52 = -Mf*Hf*LFf*delta/den;
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
    a51 a52 a53 a54 a55 a56
    5 -3.6 0 0 0 -gama];
B = [0 0
    0 0
    0 0
    beta 11.2
    b51 b52
    40 delta];
C = eye(6);
D = [0];
t = 0:0.02:10;
x0 = [0.2 -0.1 0.15 -1 0.8 0];
para = [-0.5+(a-b)/20; 0.1+(b-c)/(a+d+10)];
ysp = -0.1*C(1:3,:)/A*B*para;
save('ABCD.mat','A','B','C','D','t','x0','ysp')
```

2. **pole.m**

```
%% Initial parameters
clc
clear
load('../ABCD.mat');
%% quality limitations
% Because Mp <= 0.1
kesi = sqrt(log(0.1)^2/(pi^2+log(0.1)^2));
kesi = ceil(kesi*10)/10; % bigger than max
% Because ts = 0.02
omega = 4/(5*kesi);
syms s;
eqn = s^2+2*kesi*omega*s+omega^2 == 0;
sols = vpasolve(eqn, s);
save('poles.mat','sols','kesi','omega')
%% plot Cart position, handle angle and bike angle
K1 = poleplace(3,3,3,3); K2= poleplace(2,3,4,5);
K3=poleplace(5,5,5,5);
A_BK1 = A-B*K1; A_BK2 = A-B*K2; A_BK3 = A-B*K3;
sys1 = ss(A_BK1, B, C, D); sys2 = ss(A_BK2, B, C, D);
sys3 = ss(A_BK3, B, C, D);
initial(sys1,'r',sys2,'b',sys3, 'g', x0, t);
xlabel('t');ylabel('Amplitude');legend('3333','2345',
'5555');
%% plot feedback
[y1, t, x1]=initial(sys1, x0, t); u1 = -K1*x1';
[y2, t, x2]=initial(sys2, x0, t); u2 = -K2*x2';
[y3, t, x3]=initial(sys3, x0, t); u3 = -K3*x3';
plot(t,u1(2,:),t,u2(2,:),t,u3(2,:));
legend('3333','2345','5555');
xlabel('t'),ylabel('Feedback'),
%% pole placement
function K = poleplace(pole3, pole4, pole5, pole6)
    load('../ABCD.mat');
    load('poles.mat');
    % check AB
    if rank([B A*B A^2*B A^3*B A^4*B A^5*B]) < 6
    error('AB not controlable');
    end
    % check ABq
    q = [0;1]; Bq = B*q;
    Wc = [Bq A*Bq A^2*Bq A^3*Bq A^4*Bq A^5*Bq];
    if rank(Wc)<6
```

```
        error('ABq not controlable');
    end
    % find K
    poles = [sols(1) sols(2) pole3*real(sols(1))
pole4*real(sols(1)) pole5*real(sols(1))
pole6*real(sols(1))];
    phids = double((A-poles(1)*eye(6))*(A-
poles(2)*eye(6))*(A-poles(3)*eye(6))*(A-
poles(4)*eye(6))*(A-poles(5)*eye(6))*(A-
poles(6)*eye(6)));
    K = q*([0 0 0 0 0 1]*(Wc^-1)* phids);
end
```

3. LQR.m

```
%% Initial parameters
clc; clear
load('../ABCD.mat');
%% plot eye
Q1 = eye(6);
Q2 = Q1*10;
Q3 = diag([1 1 4 5 1 4]);
Q4 = diag([1 1 9 1 1 1]);
R = eye(2);
K1 = getK(Q1,R); K2 = getK(Q2,R); K3 = getK(Q3,R); K4
= getK(Q4,R);
A_BK1 = A-B*K1; A_BK2 = A-B*K2; A_BK3 = A-B*K3; A_BK4
= A-B*K4;
sys1 = ss(A_BK1, B, C, D); sys2 = ss(A_BK2, B, C, D);
sys3 = ss(A_BK3, B, C, D); sys4 = ss(A_BK4, B, C, D);
initial(sys1,'r',sys2,'b',sys3, 'g', sys4, 'y',x0,
t);
xlabel('t'),ylabel('Amplitude'),legend('eye(6)','eye(
6)*10','114514', '119111')
%% plot feedback
[y1, t, x1]=initial(sys1, x0, t); u1 = -K1*x1';
[y2, t, x2]=initial(sys2, x0, t); u2 = -K2*x2';
[y3, t, x3]=initial(sys3, x0, t); u3 = -K3*x3';
[y4, t, x4]=initial(sys4, x0, t); u4 = -K4*x4';
figure(1)
plot(t,u1(1,:),t,u2(1,:),t,u3(1,:),t,u4(1,:));
legend('eye(6)','eye(6)*10','114514', '119111');
xlabel('t'),ylabel('Feedback1'),
figure(2)
plot(t,u1(2,:),t,u2(2,:),t,u3(2,:),t,u4(2,:));
legend('eye(6)','eye(6)*10','114514', '119111');
xlabel('t'),ylabel('Feedback2'),
%% LQR
function K = getK(Q, R)
load('../ABCD.mat');
gamma=[A -B/R*B'; -Q -A'];
[vector,value]=eig(gamma);
value=sum(value);
vec=vector(:,find(real(value)<0));
P=vec(7:12,:)/vec(1:6,:);
K=real(inv(R)*B'*P);
end
```

4. observer.m

```
%% Initial parameters
clc
clear
load('../ABCD.mat');
C = C(1:3,:);
%% pole placement
load('../pole/poles.mat')
%poles = [sols(1) sols(2) 3*real(sols(1))
3*real(sols(1)) 4*real(sols(1)) 4*real(sols(1))];
%poles = [sols(1) sols(2) 4*real(sols(1))
5*real(sols(1)) 6*real(sols(1)) 7*real(sols(1))];
poles = [sols(1) sols(2) 8*real(sols(1))
8*real(sols(1)) 8*real(sols(1)) 8*real(sols(1))];
%% full order pole placement
q = [0;0;1]; B_q=C'*q;
Wc = [B_q A'*B_q A'^2*B_q A'^3*B_q A'^4*B_q
A'^5*B_q];
assert(rank(Wc(:,1:6))==6);
phids = double((A'-poles(1)*eye(6))*(A'-
poles(2)*eye(6))*(A'-poles(3)*eye(6))*(A'-
poles(4)*eye(6))*(A'-poles(5)*eye(6))*(A'-
poles(6)*eye(6)));
K_ = q*([0 0 0 0 0 1]/Wc*phids);
L = K_';
%% LQR for K
R = eye(2);
K = getK(diag([1 1 9 1 1 1]),R);

%% LQR
function K = getK(Q, R)
load('../ABCD.mat');
gamma=[A -B/R*B'
    -Q -A'];
[vector,value]=eig(gamma);
value=sum(value);
vec=vector(:,find(real(value)<0));
P=vec(7:12,:)/vec(1:6,:);
K=real(inv(R)*B'*P);
end
```

5. servo.m

```
%% Initial parameters
clc
clear
load('../ABCD.mat');
C = C(1:3,:);
disturbance = [-1; 1];
ysp1 = C/A*B*[2;5];
ysp2 = [1 10 2];
%% Get new ABCD for LQR
assert(rank([A B;C zeros(3,2)]) == min(size([A B;C
zeros(3,2)])));
Abar=[A zeros(6,3);-C zeros(3,3)];
Bbar=[B;zeros(3,2)];
%% LQR for Kbar
Q = diag([1 1 1.5 1.5 1 1 1 1 1]); R = eye(2);
gamma=[Abar -Bbar/R*Bbar'
   -Q -Abar'];
[vector,value]=eig(gamma);
value=sum(value);
vec=vector(:,find(real(value)<0));
P=vec(10:18,:)/vec(1:9,:);
K=real(inv(R)*Bbar'*P);

%% observer pole placement
load('../pole/poles.mat')
%poles = [sols(1) sols(2) 3*real(sols(1))
3*real(sols(1)) 4*real(sols(1)) 4*real(sols(1))];
%poles = [sols(1) sols(2) 4*real(sols(1))
5*real(sols(1)) 6*real(sols(1)) 7*real(sols(1))];
poles = [sols(1) sols(2) 8*real(sols(1))
8*real(sols(1)) 8*real(sols(1)) 8*real(sols(1))];
%% full order pole placement
q = [0;0;1]; B_q=C'*q;
Wc = [B_q A'*B_q A'^2*B_q A'^3*B_q A'^4*B_q
A'^5*B_q];
assert(rank(Wc(:,1:6))==6);
phids = double((A'-poles(1)*eye(6))*(A'-
poles(2)*eye(6))*(A'-poles(3)*eye(6))*(A'-
poles(4)*eye(6))*(A'-poles(5)*eye(6))*(A'-
poles(6)*eye(6)));
K_ = q*([0 0 0 0 0 1]/Wc*phids);
L = K_';
```

6. servoctrl2.m

```
%% Initial parameters
clc
clear
load('../ABCD.mat');
C = C(1:3,:);
disturbance = [-1; 1];
%% reachability
ctrlable = rank([A B
    C zeros(3,2)]) - rank(A);
%% Get ABCD for LQR
C_ = [C(1,:); C(3,:)];
assert(rank([A B;C_ zeros(2,2)]) == min(size([A B;C_
zeros(2,2)])));
Abar=[A zeros(6,2);-C_ zeros(2,2)];
Bbar=[B;zeros(2,2)];
Cbar = [C_ zeros(2,2)];
%% LQR for new
Q = diag([1 1 1.5 1.5 1 1 1 1]); R = eye(2);
gamma=[Abar -Bbar/R*Bbar'
    -Q -Abar'];
[vector,value]=eig(gamma);
value=sum(value);
vec=vector(:,find(real(value)<0));
P=vec(9:16,:)/vec(1:8,:);
K_=real(inv(R)*Bbar'*P);
%% y_sp can be arbitary
y_sp = [3 -5];
```