Xinran Li
260774237

COMP 417 A4 Report

- **Key Aspects of Solution**

In our assignment 4 second problem, we choose to use LQR to control the cart pole robot. In the assignment we are given the state $\vec{x} = [x, \dot{x}, \theta, \dot{\theta}]^T$ where vector_x is the position of the cart along the x axis, θ is the angle of the pole (from the zero point: downwards) in radians, and both cart velocity ($\dot{x}$) and angular velocity of pole ($\dot{\theta}$) are in unit per seconds.

By using LQR problem to show an optimal control, we are given the dynamic model of robot:

$$\ddot{x} = \frac{2ml\dot{\theta}^2 \sin\theta + 3mg\sin\theta\cos\theta + 4(u - b\dot{x})}{4(M+m) - 3m\cos^2\theta}$$

$$\ddot{\theta} = \frac{-3(ml\dot{\theta}^2 \sin\theta\cos\theta + 2((M+m)g\sin\theta + (u - b\dot{x})\cos\theta))}{l(4(M+m) - 3m\cos^2\theta)}$$

The dynamic model expression can be:

$$\frac{dx}{dt} = \dot{\vec{x}} = [\dot{x}, \ddot{x}, \dot{\theta}, \ddot{\theta}]^T$$

In order to apply LQR method, the dynamic system can be expressed as a linear approximation:

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}) = A\vec{x} + B\vec{u}$$
$$around\ the\ goal\ state\ \bar{x} = [0, 0, 0, \pi]^T$$

By given appendix, linear approximation is archived by taking a Taylor expansion to first order, the resulting A and B are Jacobian matrices. Since this dynamic gives a 3D model, the Jacobian matrices A, B compute orientation of the tangent plane to 3D model at given point which is goal state $\overline{x} = [0, 0, 0, pi]^T$. By definition of Jacobian Matrix in Figure2.1, we can find A, B by taking partial derivative of $\dot{x} = f(x, u)$, with respect of each variable in state vector x.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

(Figure2.1 Standard computation of Jacobian matrix)

After solving A, B, we need to find Q, R matrices for quadratic cost functions Jn(x, u), that Q, R are obtained by hand tuning in a similar style to PID gains. The quadratic cost functions as follows:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$

Q is a 4x4 diagonal matrix where each non-zero value in diagonal is the penalty of each vector in $[x, \dot{x}, \theta, \dot{\theta}]^T$. R is a 1x1 matrix that controls the cost of cart expenditure; the smaller value of R, more aggressive motion of the cart will achieve.

Since we know the A, B, Q, R, we can find the value of K that minimize Jn(x, u). With known K-value, goal state (g) and current state x, we can compute the feedback control u = -K(x-g). Next we can apply force to the cart based on current state x and get next feedback control to calculate next state $\dot{x}$ until we reach goal state.

In the assignment4, we discovered how to linear approximate a 3D dynamic model by Taylor Expansion with a given point $\overline{x}$. We could find A, B by linear approximation such that A, B are found as Jacobian matrices. This observation is demonstrated in handwriting Figure3.1.

Despite using LQR method to solve Q2, we can also use PID controller to solve the problem. We calculate the error function every time we reach a new position, so we know the target position and ball position difference, then we calculate pid by formula:

$$u(t) = \mathrm{MV}(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\, d\tau + K_{\mathrm{d}} \frac{de(t)}{dt}$$

We can tune the gains which are in 7 dimensions each in this particular question to find a steady state. This method can be easily approached but since in this assignment we are out of time to try, we keep this scenario as one of the outstanding items that we will try in the future.

Figure 3.1 (Handy demonstration of Linear approximation by Taylor Expansion)

Given:

$$\dot{\vec{x}} = [\dot{x}\ \ddot{x}\ \ddot{\theta}\ \dot{\theta}]^T \quad \vec{x} = [x\ \dot{x}\ \dot{\theta}\ \theta]^T \quad \bar{x} = [0, 0, 0, \pi]^T$$

$$\frac{dx}{dt}(\vec{x}) = \dot{\vec{x}} = f(\vec{x}, u) = A\vec{x} + Bu$$

By Taylor expression:

$$\dot{\vec{x}} = f(\vec{x}, u)$$

$$\dot{\vec{x}} = f(\bar{x}, \bar{u}) + \frac{Df}{Dx}\Big|_{\bar{x}} \cdot (\vec{x} - \bar{x}) + \frac{D^2 f}{Dx^2}\Big|_{\bar{x}} (\vec{x} - \bar{x})^2 + \cdots + \frac{Df}{Du}\Big|_{\bar{u}} (\vec{u} - \bar{u}) + \frac{D^2 f}{Du^2}\Big|_{\bar{u}} (u - \bar{u})^2 + \cdots$$

$\Rightarrow \dot{\vec{x}} \approx \frac{Df}{Dx}\Big|_{\bar{x}} \vec{x} + \frac{Df}{Du}\Big|_{\bar{u}} u$ since other than first order items are nearly 0.

$$\dot{\vec{x}} \approx \nabla_{\vec{x}} f(\vec{x}, u)[(\vec{x}, u)] \cdot \vec{x} + \nabla_u f(\vec{x}, u)[(\vec{x}, u)] \cdot u \equiv A\vec{x} + Bu.$$

where $\nabla_{\vec{x}}$ ∇_u are jacobian matrix as $\frac{Df}{Dx}\Big|_{\bar{x}}$, $\frac{Df}{Du}\Big|_{\bar{u}}$.

which are evaluated at goal state $\bar{x}$.

A, B matrices are evaluated as Jacobian matrices.

- **Practical Outcome Description**

In this section we describe how we compute A, B by using Taylor expansion.

By given code, we need to calculate A, B and tunning Q, R by testing.

$$\dot{\vec{x}} = \frac{dx}{dt}(\vec{x}) = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \ddot{\theta} \\ \dot{\theta} \end{bmatrix}^T = \begin{bmatrix} \dot{x} \\ \dfrac{2ml\dot{\theta}^2 \sin\theta + 3mg\sin\theta\cos\theta + 4(u - b\dot{x})}{4(M+m) - 3m\cos^2\theta} \\ \dfrac{-3(ml\dot{\theta}^2 \sin\theta\cos\theta + 2((M+m)g\sin\theta + (u - b\dot{x})\cos\theta))}{l(4(M+m) - 3m\cos^2\theta)} \\ \dot{\theta} \end{bmatrix}$$

By Taylor Expansion we know:

$$\dot{\vec{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = A \begin{bmatrix} x \\ \dot{x} \\ \dot{\theta} \\ \theta \end{bmatrix} + B[u] \quad \text{where:}$$

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial \dot{x}} & \frac{\partial f_1}{\partial \dot{\theta}} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial \dot{x}} & \frac{\partial f_2}{\partial \dot{\theta}} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial \dot{x}} & \frac{\partial f_3}{\partial \dot{\theta}} & \frac{\partial f_3}{\partial \theta} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial \dot{x}} & \frac{\partial f_4}{\partial \dot{\theta}} & \frac{\partial f_4}{\partial \theta} \end{bmatrix} \qquad B = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{bmatrix}$$

where:

$$f_1 = \dot{x}$$

$$f_2 = \frac{2ml\dot{\theta}^2 \sin\theta + 3mg\sin\theta\cos\theta + 4(u - b\dot{x})}{4(M+m) - 3m\cos^2\theta}$$

$$f_3 = \frac{-3(ml\dot{\theta}^2 \sin\theta\cos\theta + 2((M+m)g\sin\theta + (u - b\dot{x})\cos\theta))}{l(4(M+m) - 3m\cos^2\theta)}$$

$$f_4 = \dot{\theta}$$

Since calculating partial derivative is too complicated, we use MATLAB to compute the values. According to handwriting above, A is jacobian matrix and denoted “jac_A” in MATLAB. B is denoted as “jac_B”.

After calculating partial derivatives, we substitute values into the A, B matrices, where m=0.5 M=0.5 l=0.5 g=9.82 b=1.0, and $\bar{x} = [x, \dot{x}, \theta, \dot{\theta}]^T = [0, 0, 0, pi]^T$. Here we substitute goal state values in A, B because we need to find tangent plane's orientation at goal state $\bar{x}$ (as we discussed in Key Aspects of Solution). Thus we compute final value of matrix A, B. The Matlab code is attached in the homework file.

Final result:

$$A = \begin{bmatrix} 0, & 1, & 0, & 0 \\ 0, & -8/5, & 0, & 1473/250 \\ 0, & -24/5, & 0, & 5892/125 \\ 0, & 0, & 1, & 0 \end{bmatrix}$$

$$B = [0, \quad 8/5, \quad 24/5, \quad 0]^T$$

A snap view of MATLAB code:

```
% declare symbols
syms theta xdot thetadot m l g u b M x

x_doubledot = (2*m*l*thetadot^2*sin(theta)+3*m*g*sin(theta)*cos(theta)+4*(u-b*xdot))/(4*(M+m)-3*m*cos(theta)^2);
theta_doubledot = -3*(m*l*thetadot^2*sin(theta)*cos(theta)+2*((M+m)*g*sin(theta)+(u-b*xdot)*cos(theta)))/(l*(4*(M+m)-3*m*cos(theta)^

% jac_A calculate the jacobian matrix of A
jac_A=jacobian([xdot,x_doubledot,theta_doubledot,thetadot],[x,xdot,thetadot,theta]);
% print result
disp(jac_A);

% subsitute variable with real values
% m=0.5 M=0.5 l=0.5 g=9.82 b=1.0
% x_goal_state=[x, xdot, thetadot, theta]=[0, 0, 0, pi]
jac_A=subs(jac_A,{g,m,M,l,b,theta,thetadot},{9.82,0.5,0.5,0.5,1.0,pi,0});
% print final result of A
disp(jac_A)

%jac_B calculate the jacobian matrix of B
jac_B=jacobian([xdot,(2*m*l*thetadot^2*sin(theta)+3*m*g*sin(theta)*cos(theta)+4*(u-b*xdot))/(4*(M+m)-3*m*cos(theta)^2),-3*(m*l*theta
%print result
disp(jac_B);

% subsitute variable with real values
% m=0.5 M=0.5 l=0.5 g=9.82 b=1.0
% x_goal_state=[x, xdot, thetadot, theta]=[0, 0, 0, pi]
```

Please refer to the attached file named "AB_jacobian.m" for more coding details.

Once A, B are calculated, we tune the Q, R value so pendulum can swing up and cart stop moving when θ = pi.

As for cost function Q, we want a competitively higher penalty of theta since angle is important to keep the cart pole upwards. For cost function R, we want lower penalty of R to have more aggressive motion of cart.

At last, we calculate feed back control u = K(x-x̄) to approach the control (force) we apply to next state based on current state x.

- **Time Break Down**

The second question took us 3 hours to figure out, as 2hrs are spent to figure out how to use Taylor Expansion to linearly approximate dynamic model. With Youtube videos' help and professor's answers on Discussion board, we understood the linear approximation and wrote our understanding in Figure3.1. Then we used MATLAB to calculate A, B value. Finally, based on our understanding of LQR method, we wrote the feedback control u = K(x-g) in computeControl function in cartpole_control.py. Q, R valued were obtained during testing.

Reference: https://www.youtube.com/watch?v=1YMTkELi3tE&t=909s