Optimal Control and Reinforcement Learning

Homework 2

March 26, 2019

Mingyu Zhao
New York University
Tandon School of Engineering
m.zhao@nyu.edu

Exercise 1: Dynamic Programming

(a) Use the dynamic programming algorithm to solve the finite horizon optimal control problem that minimizes J, for $N = 3$ and $x_0 = 5$.

According to the cost function, we can define the final cost for this problem:

$$J_N(x_N) = x_N^2$$

And define the cost-to-go starting at state x_n for $n \in [0, N-1]$:

$$J_n(x_n) = \min_{u_n}((x_n^2 + u_n^2) + J_{n+1}(x_n + u_n))$$

So the optimal cost $J^* = \min_{u_0,\ldots,u_{N-1}}(x_N^2 + \sum_{k=0}^{N-1}(x_k^2 + u_k^2))$ is equivalent to the cost-to-go starting at initial state x_0, which is $J_0(x_0)$.

Then, we can apply the dynamic programming algorithm to find the optimal cost, under the constraints $\forall n \in [0, N],\ x_n \in [0, 5]$.

1. Start from final state, when $n = 3$:

Compute the final cost for each final state:

$$J_3(0) = 0 \quad J_3(1) = 1 \quad J_3(2) = 4 \quad J_3(3) = 9 \quad J_3(4) = 16 \quad J_3(5) = 25$$

2. When $n = 2$:

Compute cost-to-go at time 2 for each possible state:

$$J_2(x_2) = \min_{u_2}((x_2^2 + u_2^2) + J_3(x_2 + u_2))$$

$$J_2(0) = \min_{u_2}((0^2 + u_2^2) + J_3(0 + u_2)) = \min\{0, 2, 8, 18, 32, 50\} = 0 \quad (u_2 = 0)$$

$$J_2(1) = \min_{u_2}((1^2 + u_2^2) + J_3(1 + u_2)) = \min\{2, 2, 6, 14, 26, 42\} = 2 \quad (u_2 = -1, 0)$$

$$J_2(2) = \min_{u_2}((2^2 + u_2^2) + J_3(2 + u_2)) = \min\{8, 6, 8, 14, 24, 38\} = 6 \quad (u_2 = -1)$$

$$J_2(3) = \min_{u_2}((3^2 + u_2^2) + J_3(3 + u_2)) = \min\{18, 14, 14, 18, 26, 38\} = 14 \quad (u_2 = -2, -1)$$

$$J_2(4) = \min_{u_2}((4^2 + u_2^2) + J_3(4 + u_2)) = \min\{32, 26, 24, 26, 32, 42\} = 24 \quad (u_2 = -2)$$

$$J_2(5) = \min_{u_2}((5^2 + u_2^2) + J_3(5 + u_2)) = \min\{50, 42, 38, 38, 42, 50\} = 38 \quad (u_2 = -3, -2)$$

3. When $n = 1$:

Compute cost-to-go at time 1 for each possible state:

$$J_1(x_1) = \min_{u_1}((x_1^2 + u_1^2) + J_2(x_1 + u_1))$$

$$J_1(0) = \min_{u_1}((0^2 + u_1^2) + J_2(0 + u_1)) = \min\{0, 3, 10, 23, 40, 63\} = 0 \quad (u_1 = 0)$$

$$J_1(1) = \min_{u_1}((1^2 + u_1^2) + J_2(1 + u_1)) = \min\{2, 3, 8, 19, 34, 55\} = 2 \quad (u_1 = -1)$$

$$J_1(2) = \min_{u_1}((2^2 + u_1^2) + J_2(2 + u_1)) = \min\{8, 7, 10, 19, 32, 51\} = 7 \quad (u_1 = -1)$$

$$J_1(3) = \min_{u_1}((3^2 + u_1^2) + J_2(3 + u_1)) = \min\{18, 15, 16, 23, 34, 51\} = 15 \quad (u_1 = -2)$$

$$J_1(4) = \min_{u_1}((4^2 + u_1^2) + J_2(4 + u_1)) = \min\{32, 27, 26, 31, 40, 55\} = 26 \quad (u_1 = -2)$$

$$J_1(5) = \min_{u_1}((5^2 + u_1^2) + J_2(5 + u_1)) = \min\{50, 43, 40, 43, 50, 63\} = 40 \quad (u_1 = -3)$$

4. When $n = 0$:

Compute cost-to-go at time 0 for state $x_0 = 5$:

$$J_0(x_0) = \min_{u_0}((x_0^2 + u_0^2) + J_1(x_0 + u_0))$$

$$J_0(5) = \min_{u_0}((5^2 + u_0^2) + J_1(5 + u_0)) = \min\{50, 43, 41, 44, 52, 65\} = 41 \quad (u_0 = -3)$$

So the optimal cost will be $J^* = J_0(x_0 = 5) = 41$, and the optimal sequences of x_k and u_k are $(x_0 = 5, x_1 = 2, x_2 = 1, x_3 = 0)$ and $(u_0 = -3, u_1 = -1, u_2 = -1)$ or $(x_0 = 5, x_1 = 2, x_2 = 1, x_3 = 1)$ and $(u_0 = -3, u_1 = -1, u_2 = 0)$. The different steps are also shown above.

(b) Consider the same problem as (a) with the additional constraint $x_3 = 5$ on the final state.

In this case, the final state space becomes $\{5\}$. Recompute backwards for each state.

1. Start from final state, when $n = 3$:

Compute the final cost for each final state:

$$J_3(5) = 25$$

2. When $n = 2$:

$$J_2(x_2) = \min_{u_2}((x_2^2 + u_2^2) + J_3(x_2 + u_2))$$

$$J_2(0) = \min_{u_2}((0^2 + u_2^2) + J_3(0 + u_2)) = ((0^2 + 5^2) + J_3(0 + 5)) = 50 \quad (u_2 = 5)$$

$$J_2(1) = \min_{u_2}((1^2 + u_2^2) + J_3(1 + u_2)) = ((1^2 + 4^2) + J_3(1 + 4)) = 42 \quad (u_2 = 4)$$

$$J_2(2) = \min_{u_2}((2^2 + u_2^2) + J_3(2 + u_2)) = ((2^2 + 3^2) + J_3(2 + 3)) = 38 \quad (u_2 = 3)$$

$$J_2(3) = \min_{u_2}((3^2 + u_2^2) + J_3(3 + u_2)) = ((3^2 + 2^2) + J_3(3 + 2)) = 38 \quad (u_2 = 2)$$

$$J_2(4) = \min_{u_2}((4^2 + u_2^2) + J_3(4 + u_2)) = ((4^2 + 1^2) + J_3(4 + 1)) = 42 \quad (u_2 = 1)$$

$$J_2(5) = \min_{u_2}((5^2 + u_2^2) + J_3(5 + u_2)) = ((5^2 + 0^2) + J_3(5 + 0)) = 50 \quad (u_2 = 0)$$

3. When $n = 1$:

Compute cost-to-go at time 1 for each possible state:

$$J_1(x_1) = \min_{u_1}((x_1^2 + u_1^2) + J_2(x_1 + u_1))$$

$$J_1(0) = \min_{u_1}((0^2 + u_1^2) + J_2(0 + u_1)) = \min\{50, 43, 42, 47, 58, 75\} = 42 \quad (u_1 = 2)$$

$$J_1(1) = \min_{u_1}((1^2 + u_1^2) + J_2(1 + u_1)) = \min\{52, 43, 40, 43, 52, 67\} = 40 \quad (u_1 = 1)$$

$$J_1(2) = \min_{u_1}((2^2 + u_1^2) + J_2(2 + u_1)) = \min\{58, 47, 42, 43, 50, 63\} = 42 \quad (u_1 = 0)$$

$$J_1(3) = \min_{u_1}((3^2 + u_1^2) + J_2(3 + u_1)) = \min\{68, 55, 48, 47, 52, 63\} = 47 \quad (u_1 = 0)$$

$$J_1(4) = \min_{u_1}((4^2 + u_1^2) + J_2(4 + u_1)) = \min\{82, 67, 58, 55, 58, 67\} = 55 \quad (u_1 = -1)$$

$$J_1(5) = \min_{u_1}((5^2 + u_1^2) + J_2(5 + u_1)) = \min\{100, 83, 72, 67, 68, 75\} = 67 \quad (u_1 = -2)$$

4. When $n = 0$:

Compute cost-to-go at time 0 for state $x_0 = 5$:

$$J_0(x_0) = \min_{u_0}((x_0^2 + u_0^2) + J_1(x_0 + u_0))$$

$$J_0(5) = \min_{u_0}((5^2 + u_0^2) + J_1(5 + u_0)) = \min\{92, 81, 76, 76, 81, 92\} = 76 \quad (u_0 = -3, -2)$$

So with the additional constraint, the optimal cost will be $J^* = J_0(x_0 = 5) = 76$, and the optimal sequences of x_k and u_k are $(x_0 = 5, x_1 = 2, x_2 = 2, x_3 = 5)$ and $(u_0 = -3, u_1 = 0, u_2 = 3)$ or $(x_0 = 5, x_1 = 3, x_2 = 3, x_3 = 5)$ and $(u_0 = -2, u_1 = 0, u_2 = 2)$. The different steps are also shown above.

(c) Consider the dynamic system of question (a) with a stochastic disturbance w_n added. The disturbance takes the values -1 and 1 with equal probabilities for all x_n and u_n, except if $x_n + u_n$ is equal to 0 or 5, in which case $w_n = 0$ with probability 1.

The system dynamic becomes:

$$x_{n+1} = x_n + u_n + w_n$$

where w_n is a random variable, so we need to minimize the expectation of cost at different time. So change our cost-to-go into:

$$J_n(x_n) = \min_{u_n} \mathop{\mathbb{E}}_{w_n} ((x_n^2 + u_n^2) + J_{n+1}(x_n + u_n + w_n))$$

And the optimal cost now is also equivalent with our new cost-to-go at time 0. Then, similarly, we can use dynamic programming algorithm to compute cost-to-go at different time backwards.

1. Start from final state, when $n = 3$:

Compute the final cost for each final state: (same as the previous problem)

$$J_3(0) = 0 \quad J_3(1) = 1 \quad J_3(2) = 4 \quad J_3(3) = 9 \quad J_3(4) = 16 \quad J_3(5) = 25$$

2. When $n = 2$:

Compute cost-to-go at time 2 for each possible state:

$$J_2(x_2) = \min_{u_2} \mathop{\mathbb{E}}_{w_2} ((x_2^2 + u_2^2) + J_3(x_2 + u_2 + w_2))$$

$$J_2(0) = \min_{u_2}((0^2 + u_2^2) + J_3(0 + u_2 + w_2)) = 0.5 \times \min\{0+0, 1+5, 5+13, 13+25, 25+41, 50+50\} = 0$$

$$J_2(1) = \min_{u_2}((1^2 + u_2^2) + J_3(1 + u_2 + w_2)) = 0.5 \times \min\{2+2, 1+5, 3+11, 9+21, 19+35, 42+42\} = 2$$

$$J_2(2) = \min_{u_2}((2^2 + u_2^2) + J_3(2 + u_2 + w_2)) = 0.5 \times \min\{8+8, 5+9, 5+13, 9+21, 17+33, 38+38\} = 7$$

$$J_2(3) = \min_{u_2}((3^2 + u_2^2) + J_3(3 + u_2 + w_2)) = 0.5 \times \min\{18+18, 13+17, 11+19, 13+25, 19+35, 38+38\} = 15$$

$$J_2(4) = \min_{u_2}((4^2 + u_2^2) + J_3(4 + u_2 + w_2)) = 0.5 \times \min\{32+32, 25+29, 21+29, 21+33, 25+41, 42+42\} = 25$$

$$J_2(5) = \min_{u_2}((5^2 + u_2^2) + J_3(5 + u_2 + w_2)) = 0.5 \times \min\{50+50, 41+45, 35+43, 33+45, 35+51, 50+50\} = 39$$

The optimal policy for stage 2 is:

$$\mu_2^*(0) = 0 \quad \mu_2^*(1) = -1 \quad \mu_2^*(2) = -1 \quad \mu_2^*(3) = \{-2, -1\} \quad \mu^*23(4) = -2 \quad \mu_2^*(5) = \{-3, -2\}$$

3. When $n = 1$:

$$J_1(x_1) = \min_{u_1} \mathop{\mathbb{E}}_{w_1} ((x_1^2 + u_1^2) + J_2(x_1 + u_1 + w_1))$$

$$J_1(0) = \min_{u_1}((0^2 + u_1^2) + J_2(0 + u_1 + w_1)) = 0.5 \times \min\{0 + 0, 1 + 8, 6 + 19, 16 + 34, 31 + 55, 64 + 64\} = 0$$

$$J_1(1) = \min_{u_1}((1^2 + u_1^2) + J_2(1 + u_1 + w_1)) = 0.5 \times \min\{2 + 2, 1 + 8, 4 + 17, 12 + 30, 25 + 49, 56 + 56\} = 2$$

$$J_1(2) = \min_{u_1}((2^2 + u_1^2) + J_2(2 + u_1 + w_1)) = 0.5 \times \min\{8 + 8, 5 + 12, 6 + 19, 12 + 30, 23 + 47, 52 + 52\} = 8$$

$$J_1(3) = \min_{u_1}((3^2 + u_1^2) + J_2(3 + u_1 + w_1)) = 0.5 \times \min\{18 + 18, 13 + 20, 12 + 25, 16 + 34, 25 + 49, 52 + 52\} = 16.5$$

$$J_1(4) = \min_{u_1}((4^2 + u_1^2) + J_2(4 + u_1 + w_1)) = 0.5 \times \min\{32 + 32, 25 + 32, 22 + 35, 24 + 42, 31 + 55, 56 + 56\} = 28.5$$

$$J_1(5) = \min_{u_1}((5^2 + u_1^2) + J_2(5 + u_1 + w_1)) = 0.5 \times \min\{50 + 50, 41 + 48, 36 + 49, 36 + 60, 41 + 65, 64 + 64\} = 42.5$$

The optimal policy for stage 1 is:

$$\mu_1^*(0) = 0 \quad \mu_1^*(1) = -1 \quad \mu_1^*(2) = -2 \quad \mu_1^*(3) = -2 \quad \mu_1^*(4) = \{-3, -2\} \quad \mu_1^*(5) = -3$$

4. When $n = 0$:

Compute cost-to-go at time 0 for state $x_0 = 5$:

$$J_0(x_0) = \min_{u_0} \mathop{\mathbb{E}}_{w_0} ((x_0^2 + u_0^2) + J_1(x_0 + u_0 + w_0))$$

$$J_0(5) = 0.5 \times \min\{50 + 50, 41 + 49, 36 + 50.5, 37 + 57.5, 42.5 + 68.5, 67.5 + 67.5\} = 43.25$$

The optimal policy for stage 0 is:

$$\mu_0^*(5) = -3$$

So the optimal policy is $\pi^* = \{\mu_0^*, \mu_1^*, \mu_2^*\}$. And the optimal expected cost is $J^* = J_0(x_0) = 43.25$

Exercise 2: Controllability

2.1 Which of these systems are stable when $u_n = 0$?

The discrete linear dynamical system $x_{n+1} = Ax_n$ is stable if and only if the eigenvalues of the matrix A satisfy that $|\lambda| < 1$. So we can apply eigendiagonalization on these dynamical systems:

(a)

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1.5 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.526 & 0 & 0.851 \\ -0.000 & -1 & 0.000 \\ -0.851 & 0 & 0.526 \end{bmatrix} \begin{bmatrix} -0.618 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.618 \end{bmatrix} \begin{bmatrix} 0.526 & 0 & -0.851 \\ -0.000 & -1 & 0.00000 \\ 0.851 & 0 & 0.526 \end{bmatrix}$$

(b) (same as the previous one)

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1.5 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.526 & 0 & 0.851 \\ -0.000 & -1 & 0.000 \\ -0.851 & 0 & 0.526 \end{bmatrix} \begin{bmatrix} -0.618 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.618 \end{bmatrix} \begin{bmatrix} 0.526 & 0 & -0.851 \\ -0.000 & -1 & 0.00000 \\ 0.851 & 0 & 0.526 \end{bmatrix}$$

(c)

$$AV = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & -0.5 & -1 \\ 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & -0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

(d)

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & -0.5 & -1 \\ -0.1 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} -0.843 & -0.840 & -0.468 \\ -0.344 & 0.437 & 0.883 \\ 0.414 & -0.323 & -0.050 \end{bmatrix} \begin{bmatrix} 0.7 & 0 & 0 \\ 0 & 0.24 & 0 \\ 0 & 0 & -0.44 \end{bmatrix} \begin{bmatrix} -0.547 & -0.227 & 1.114 \\ -0.722 & -0.488 & -1.877 \\ 0.144 & 1.286 & 1.362 \end{bmatrix}$$

So the systems of case (a) and (b) are unstable when $u_n = 0$, while the systems of (c) and (d) are stable.

2.2 Which of these systems are controllable?

The linear dynamical system $x_{n+1} = Ax_n + Bu_n$ is controllable if and only if the matrix:

$$\begin{bmatrix} B & AB & A^2B & \cdots & A^{k-1}B \end{bmatrix}$$

has full row rank (where k is the size of vector x_n). In these systems, $x_n \in \mathbb{R}^3$, so $k = 3$.

Thus, we can justify the controllability of each system by calculating the matrix above and testing if it has full row rank. I wrote a simple Matlab function to help me leverage the dense computation:

```
function res = is_controllable (A, B)
    T = [B, A * B, A^2 * B];
    res = rank(T) == 3;
endfunction
```

And test the controllability of each system with different input:

```
A1 = [ 1 0 1; 0 1.5 0; 1 0 0 ]
B1 = [ 0; 0; 1 ]
is_controllable(A1, B1) % ans = 0

A2 = [ 1 0 1; 0 1.5 0; 1 0 0 ]
B2 = [ 0; 1; 1 ]
is_controllable(A1, B1) % ans = 1

A3 = [ 0.5 0 0.5; 0 -0.5 -1; 0 0 0.5 ]
B3 = [ 1; 0; 1 ]
is_controllable(A1, B1) % ans = 1

A4 = [ 0.5 0.5 0; 0 -0.5 -1; -0.1 0 0.5 ]
B4 = [ 0; 1; 0 ]
is_controllable(A1, B1) % ans = 1
```

(a) $rank(T) = 2$, the system is not controllable.

(b) $rank(T) = 3$, the system is controllable.

(c) $rank(T) = 3$, the system is controllable.

(d) $rank(T) = 3$, the system is controllable.

2.3 For which of these systems can you find a control law u_n to stabilize the system?

For the systems (b), (c) and (d) we can find a control law u_n to stabilize it.

We can solve the Ricatti equation by iteration to get a list of optimal control input when we have chosen the cost matrix Q and R:

$$P = Q + A^{\top} P A + APB(B^{\top} PB + R)^{-1} B^{\top} PA$$

However, since we are only required to stabilize the system, and the systems (c) and (d) are already stable, we do can simply set $u_n = 0$ for them.

For system (a), I simply set $Q = I$ and $R = 1$, then solve the Ricatti equation starting from $P_{init} = Q$ (also use Matlab to leverage the computation):

```
function [P, K] = solve_ricatti (Q, R, A, B)
    P = Q; % start from Q
    counter = 0;
    u = [];
    while true
        P_new = Q + A' * P * A - A * P * B * pinv(B' * P * B + R) * B' * P * A;
        if norm(P_new - P) < 0.00001
            break;
        end
        P = P_new;
        K = pinv(R + B' * P * B) * B' * P * A;
        counter = counter + 1;
    end
endfunction
```

Finally the value of P converges to:

$$P = \begin{bmatrix} 2181.29 & -1209.03 & 1359.80 \\ -1209.03 & 673.64 & -753.58 \\ 1359.80 & -753.58 & 849.57 \end{bmatrix}$$

And the K can be computed with $K = (B^{\top} PB + R)^{-1} B^{\top} PA$:

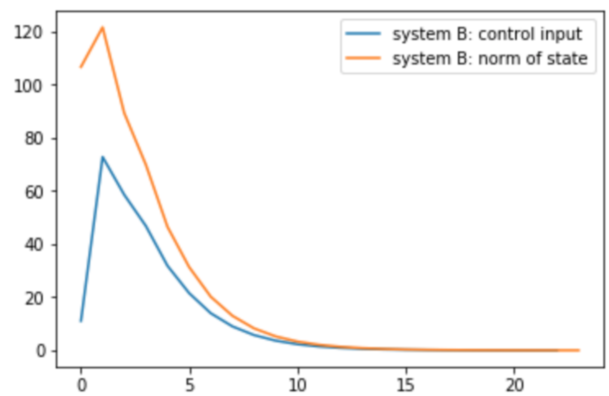
$$K = \begin{bmatrix} 14.4684 & -7.0308 & 8.8399 \end{bmatrix}$$

For system (b), test this controller with the initial state $x_{init} = \begin{bmatrix} 5 & 5 & 5 \end{bmatrix}$.

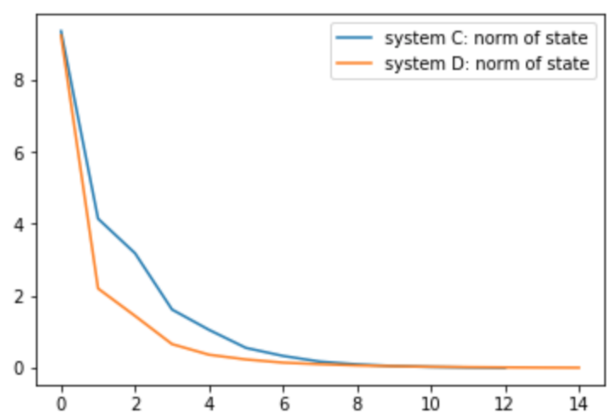
```
function [u_traj, x_traj, counter] = test_controller (A, B, K, x_init)
    u_traj = [];
    x_traj = [];
    counter = 0;
    x = x_init;
    while true
        u = -K * x;
        x = A * x + B * u
        u_traj = [u_traj, u];
```

```
            x_traj = [x_traj, norm(x)];
            counter += 1;
            if norm(x) < 0.01
                break
            end
        end
    endfunction
```

The test result is illustrated below:



For system (c) and (d), test their stability when setting $u_n = 0$:



Exercise 3: Linear Quadratic Regulators

Answered in the Jupyter notebook *Linear Quadratic Regulator.ipynb*.

Exercise 4: Cart-Pole Model

Answer for Question 1

The dynamic equations of the Cart-Pole model are written as

$$\dot{x} = v$$
$$\dot{v} = \frac{f + m_p \sin\theta(l\omega^2 + g\cos\theta)}{m_c + m_p \sin^2\theta}$$
$$\dot{\theta} = \omega$$
$$\dot{\omega} = \frac{-f\cos\theta - m_p l\omega^2 \cos\theta\sin\theta - (m_c + m_p)g\sin\theta}{l(m_c + m_p\sin^2\theta)}$$

Written in matrix form:

$$\dot{X} = F(X, U)$$

where X, U and F here are:

$$X = \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix} \quad U = \begin{bmatrix} f \end{bmatrix}$$

$$F(X,U) = \begin{bmatrix} v \\ \frac{f + m_p \sin\theta(l\omega^2 + g\cos\theta)}{m_c + m_p\sin^2\theta} \\ \omega \\ \frac{-f\cos\theta - m_p l\omega^2\cos\theta\sin\theta - (m_c+m_p)g\sin\theta}{l(m_c + m_p\sin^2\theta)} \end{bmatrix} \triangleq \begin{bmatrix} F_1(X,U) \\ F_2(X,U) \\ F_3(X,U) \\ F_4(X,U) \end{bmatrix}$$

Use Taylor expansion around $(\bar{X}, \bar{U})$, that is $\bar{x} = 0, \bar{v} = 0, \bar{\theta} = \pi, \bar{\omega} = 0, \bar{f} = 0$:

$$F(\bar{X} + \Delta X, \bar{U} + \Delta U) \approx F(\bar{X}, \bar{U}) + \left[\frac{\partial F}{\partial X}\right]_{\bar{X},\bar{U}} \Delta X + \left[\frac{\partial F}{\partial U}\right]_{\bar{X},\bar{U}} \Delta U$$

$$\dot{\bar{X}} + \Delta\dot{X} \approx F(\bar{X}, \bar{U}) + \left[\frac{\partial F}{\partial X}\right]_{\bar{X},\bar{U}} \Delta X + \left[\frac{\partial F}{\partial U}\right]_{\bar{X},\bar{U}} \Delta U$$

$$\Delta\dot{X} \approx \left[\frac{\partial F}{\partial X}\right]_{\bar{X},\bar{U}} \Delta X + \left[\frac{\partial F}{\partial U}\right]_{\bar{X},\bar{U}} \Delta U$$

Compute the Jacobian matrices in this equation:

$$\frac{\partial F}{\partial X} = \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial v} & \frac{\partial F_1}{\partial \theta} & \frac{\partial F_1}{\partial \omega} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial v} & \frac{\partial F_2}{\partial \theta} & \frac{\partial F_2}{\partial \omega} \\ \frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial v} & \frac{\partial F_3}{\partial \theta} & \frac{\partial F_3}{\partial \omega} \\ \frac{\partial F_4}{\partial x} & \frac{\partial F_4}{\partial v} & \frac{\partial F_4}{\partial \theta} & \frac{\partial F_4}{\partial \omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m_p g}{m_c} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m_c g}{(m_c + m_p) l} & 0 \end{bmatrix}$$

$$\frac{\partial F}{\partial U} = \begin{bmatrix} \frac{\partial F_1}{\partial f} \\ \frac{\partial F_2}{\partial f} \\ \frac{\partial F_3}{\partial f} \\ \frac{\partial F_4}{\partial f} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{m_c} \\ 0 \\ \frac{1}{m_c l} \end{bmatrix}$$

Then discretize the system with one-step Euler integration:

$$\Delta X_{n+1} = \Delta X_n + \left[\frac{\partial F}{\partial X}\right]_{\bar{X},\bar{U}} \Delta t \Delta X_n + \left[\frac{\partial F}{\partial U}\right]_{\bar{X},\bar{U}} \Delta t \Delta U$$

$$\begin{bmatrix} \tilde{x}_{n+1} \\ \tilde{v}_{n+1} \\ \tilde{\theta}_{n+1} \\ \tilde{\omega}_{n+1} \end{bmatrix} = \left(I + \begin{bmatrix} 0 & \Delta t & 0 & 0 \\ 0 & 0 & \frac{m_p g}{m_c}\Delta t & 0 \\ 0 & 0 & 0 & \Delta t \\ 0 & 0 & \frac{(m_c+m_p)g}{m_c l}\Delta t & 0 \end{bmatrix} \right) \begin{bmatrix} \tilde{x}_n \\ \tilde{v}_n \\ \tilde{\theta}_n \\ \tilde{\omega}_n \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_c}\Delta t \\ 0 \\ \frac{1}{m_c l}\Delta t \end{bmatrix} \tilde{f}_n$$

which can be written as:

$$\begin{bmatrix} \tilde{x}_{n+1} \\ \tilde{v}_{n+1} \\ \tilde{\theta}_{n+1} \\ \tilde{\omega}_{n+1} \end{bmatrix} = A \begin{bmatrix} \tilde{x}_n \\ \tilde{v}_n \\ \tilde{\theta}_n \\ \tilde{\omega}_n \end{bmatrix} + B\tilde{f}_n$$

where

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & \frac{m_p g}{m_c}\Delta t & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & \frac{(m_c+m_p)g}{m_c l}\Delta t & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \frac{1}{m_c}\Delta t \\ 0 \\ \frac{1}{m_c l}\Delta t \end{bmatrix}$$

Answer for Question 2

Write a cost function that will help stabilize the resting position.

Now, we can write down the cost function:

$$L = \tilde{X}_N^\intercal Q \tilde{X}_N + \sum_{k=0}^{N-1} (\tilde{X}_n^\intercal Q \tilde{X}_n + \tilde{U}_n^\intercal R \tilde{U}_n)$$

where Q and R can be customized.

And the regulator for the original nonlinear dynamic becomes:

$$\min_{\tilde{U_0},\ldots,\tilde{U_{N-1}}} L(\tilde{X}, \tilde{U})$$

$$s.t. \quad \tilde{X}_{n+1} = A\tilde{X}_n + B\tilde{U}_n$$

How would you (approximately) solve this optimal control problem?

We can use iterative method to solve the Ricatti equation of the linearized and discretized problem, then use the solution to approximate the actual solution.

What will be the form of the optimal controller f_n?

The form of the optimal controller f_n will be:

$$f_n = \bar{f} + K_n \begin{bmatrix} x_n - \bar{x} \\ v_n - \bar{v} \\ \theta_n - \bar{\theta} \\ \omega_n - \bar{\omega} \end{bmatrix}$$

Answer for Question 3

Answered in the Jupyter notebook *Cart-Pole Model.ipynb*.

Exercise 5: Direct Transcription

Answered in the Jupyter notebook *Direct Transcription Method.ipynb*.

Answer for Question 1