EE 5885 Deep Reinforcement Learning and Control

Homework #3

Due April 25, 2025 at 11:59 pm

Problem 1 (100%)

Consider an inverted pendulum on a cart as shown in Figure 1.

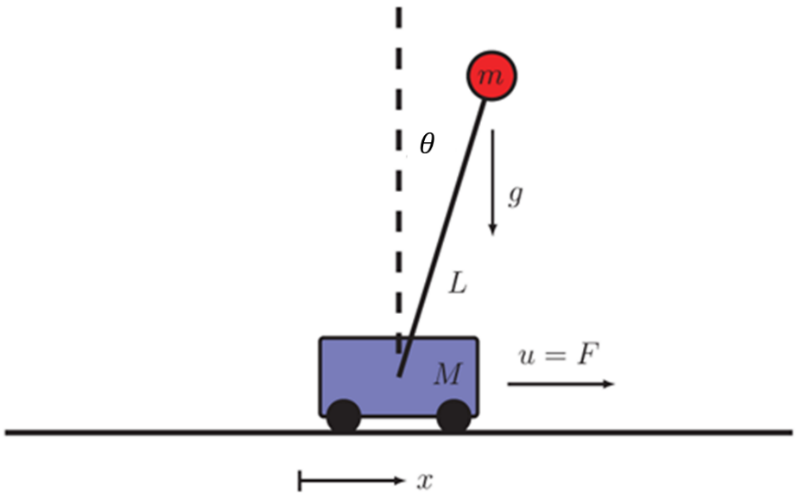

Fig. 1 Schematic of inverted pendulum on a cart

We will implement a stabilizing controller for the inverted pendulum. The full nonlinear dynamics are given by

$$\dot{x} = v$$

$$\dot{v} = \frac{-m^2L^2g\cos(\theta)\sin(\theta) + mL^2(mL\omega^2\sin(\theta) - \delta v) + mL^2u}{mL^2(M + m(1-\cos(\theta)^2))}$$

$$\dot{\theta} = \omega$$

$$\dot{\omega} = \frac{(m+M)mgL\sin(\theta) - mL\cos(\theta)(mL\omega^2\sin(\theta) - \delta v) + mL\cos(\theta)u}{mL^2(M + m(1-\cos(\theta)^2))}$$

where x is the cart position, v is the velocity, θ is the pendulum angle, ω is the angular velocity, m is the pendulum mass, M is the cart mass, L is the pendulum arm, g is the gravitational acceleration, δ is a friction damping on the dart, and u is a control force applied to the cart.

One stabilizing point corresponds to the pendulum up ($\theta = 0$) configuration. At the stabilizing point, $v = \omega = 0$. The cart position x is a free variable, as the equations do not depend explicitly on x. It is possible to linearize the equations around the pendulum up configuration, yielding the following linearized dynamics:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\delta}{M} & b\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -b\frac{\delta}{ML} & -b\frac{(m+M)g}{ML} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ b\frac{1}{ML} \end{bmatrix} u, \quad \text{for} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ v \\ \theta \\ \omega \end{bmatrix}$$

where $[x_1, x_2, x_3, x_4]^T = [x, v, \theta, \omega]^T = \mathbf{x}$ and $b = 1$.

Your task is to design and implement an LQR controller to stabilize the inverted pendulum configuration ($\theta = 0$) assuming full-state measurements (i.e., state variables, x, v, θ, ω, are all observable). The discrete-time cost function is specified as

$$J = \sum_{t=1}^{T} (\mathbf{x}_t - \mathbf{w}_r)^{\mathbf{T}} \mathbf{Q} (\mathbf{x}_t - \mathbf{w}_r) + \mathrm{u}_t^{\mathrm{T}} \mathrm{R} \mathrm{u}_t$$

where $\mathbf{w}_r$ is the desired state; $\mathbf{Q} = \boldsymbol{I}_{4\times4}$ with $\boldsymbol{I}_{4\times4}$ being an identity matrix, and $\mathrm{R} = 0.01$. We will initialize the simulation at $\mathbf{x}_0 = [2,\ 0,\ 2,\ 0]^T$ with an initial pendulum angle $\theta_0 = 2$. The desired state is $\mathbf{w}_r = [0,\ 0,\ 0,\ 0]^T$.

A starter code for the LQR implementation to solve the optimal control problem above is provided in the Matlab script, main_pendcar_LQR.m. Complete the backward-pass recursion and forward-pass recursion by adding the key equations of the LQR algorithm. Note that in the starter code, the continuous-time dynamics model is converted to a discrete-time dynamics model using c2d function. The resulting discrete-time dynamics model is given as

$$\mathbf{x}_{t+1} = A_d \mathbf{x}_t + B_d \mathrm{u}_t$$

with A_d and B_d being the state and control matrices of the discrete-time dynamics model.

Plot figures that show the evolution of state variables $[x, v, \theta, \omega]^T$ and the control signals u over time and discuss the results. Include the source code in your submission.