Robotics
Exercise 6

Marc Toussaint
Lecturer: Duy Nguyen-Tuong
TAs: Philipp Kratzer, Janik Hager, Yoojin Oh
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

November 27, 2018

1 Riccati equation in discrete time (4 points)

Consider the time discrete linear quadratic system

$$f(x_t, u_t) = Ax_t + Bu_t$$

$$c(x_t, u_t) = x_t^\top Q x_t + u_t^\top R u_t$$

with the cost function

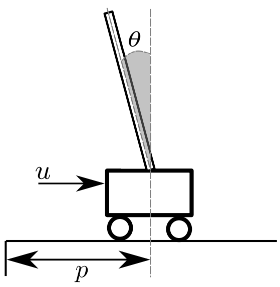
$$J^\pi = \sum_{t=0}^{\infty} c(x_t, u_t) \ .$$

The Bellman equation (slide 04.13) for this infinite-horizon discrete time system is

$$V(x) = \min_u \left[c(x,u) + V(f(x,u)) \right] \ .$$

Start with the Bellman equation and derive the Riccati equation for the system. Similar to the continuous case, you can assume a value function of the form $V(x) = x^\top P x$ with a symmetric matrix P.

2 Cart Pole Control (8 points)



In the last exercise we calculated the local linearization of the cart-pole around $x^* = (0,0,0,0)$. The solution is

$$\dot{x} = Ax + Bu \ , \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{c_2 g}{\frac{4}{3}l - c_2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{\frac{4}{3}l - c_2} & 0 \end{pmatrix} , \quad B = \begin{pmatrix} 0 \\ c_1 + \frac{c_1 c_2}{\frac{4}{3}l - c_2} \\ 0 \\ \frac{-c_1}{\frac{4}{3}l - c_2} \end{pmatrix}$$

with $g = 9.8ms^2$ the gravitational constant, $l = 1m$ the pendulum length and constants $c_1 = (M_p + M_c)^{-1}$ and $c_2 = lM_p(M_p + M_c)^{-1}$ where $M_p = M_c = 1kg$ are the pendulum and cart masses respectively.

a) Consider the local linearization of the cart-pole. Is the system controllable? (2 P)

b) We assume a stationary infinite-horizon cost function of the form

$$J^\pi = \int_0^\infty c(x(t), u(t))\ dt$$
$$c(x,u) = x^\top Q x + u^\top R u$$
$$Q = \text{diag}(c, 0, 1, 0)\ , \quad R = \mathbf{I}\ .$$

That is, we penalize position offset $c\|p\|^2$ and pole angle offset $\|\theta\|^2$. Choose $c = \varrho = 1$ to start with.
Solve the Algebraic Riccati equation

$$0 = A^\top P + P^\top A - PBR^{-1}B^\top P + Q$$

by initializing $P = Q$ and iterating using the following iteration:

$$P_{k+1} = P_k + \epsilon[A^\top P_k + P_k^\top A - P_k BR^{-1}B^\top P_k + Q]$$

Choose $\epsilon = 1/1000$ and iterate until convergence. Output the gains $K = -R^{-1}B^\top P$. (2 P)

c) Solve the same Algebraic Riccati equation by calling the `care` routine of the octave control package (or a similar method in another programming language).

For Python, install `scipy` (using *'python3 -m pip install scipy'*), use `from scipy import linalg` to import scipy in the python script and use `P=linalg.solve_continuous_are(A,B,Q,R)` to solve the ARE.

For Octave, install the Ubuntu packages `octave` and `octave-control`, perhaps use `pkg load control` and `help are` in octave to ensure everything is installed, use
`P=care(A,B, Q, R)` to solve the ARE. Output $K = -R^{-1}B^\top P$ and compare to b). (2 P)

(The solution is $K = (1.0000, 2.6088, 52.9484, 16.5952)$.)

d) Implement the optimal Linear Quadratic Regulator $u^* = Kx$ on the cart pole simulator in the function testMove().

1. For python please install pygame and pyopengl (using *' python3 -m pip install pygame'* and *' python3 -m pip install pyopengl'*), then you can run: *'jupyter-notebook py/04-riccati/04-riccati.ipynb'*
2. For C++ run: *'cd cpp/04-riccati'*, *'make'*, *'./x.exe'*

Simulate the optimal LQR and test it for various noise levels (by changing the variable *dynamicsNoise*). (2 P)