Robotics
Exercise 5

Marc Toussaint
Lecturer: Peter Englert
TAs: Matt Bernstein, Danny Driess, Hung Ngo
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

November 22, 2016

1 PD control to hold an arm steady

Download the code framework here: https://ipvs.informatik.uni-stuttgart.de/mlr/16-Robotics/e05a-code.tbz2
Unpack the code and copy the folder 'dynamics' into 'robotics15/share/teaching/RoboticsCourse/'

In the main.cpp we provide a dynamics simulation of a robot arm that simulates the system for 1000 timesteps. The task is to write a controller that holds the robot arm steady, i.e., $q^* = 0$ and $\dot{q}^* = 0$.

a) Apply direct PD control (*without* using M and F) to each joint separately and try to find parameters K_p and K_d (potentially different for each joint) to hold the arm steady.

b) Try to do the same with a PID controller that also includes the integral error

$$u = K_p(q^* - q) + K_d(\dot{q}^* - \dot{q}) + K_i \int_{s=0}^{t} (q^* - q(s))\ ds\ .$$

c) Now use the knowledge of M and F (slide 03:29) in each time step (both matrices are already provided in the for loop of the code). Use the PD equation to determine desired accelerations $\ddot{q}^*$ and use inverse dynamics to determine the necessary u (slide 03:30 - 03:31).

Optional:

Try different starting positions.

Try the same controllers for the arm in `pegArm2.ors`.

Play with `setDynamicsSimulationNoise` and check stability.

2 Local linearization and Algebraic Riccati equation

Slide 04:02 describes the cart pole dynamics, which is similar to the Segway-type system of Exercise 4, but a little simpler. We'll solve the cart pole in this exercise. The state of the cart-pole is given by $x = (p, \dot{p}, \theta, \dot{\theta})$, with $p \in \mathbb{R}$ the position of the cart, $\theta \in \mathbb{R}$ the pendulums angular deviation from the upright position and $\dot{p}, \dot{\theta}$ their respective temporal derivatives. The only control signal $u \in \mathbb{R}$ is the force applied on the cart. The analytic model of the cart pole is

$$\ddot{\theta} = \frac{g \sin(\theta) + \cos(\theta) \left[-c_1 u - c_2 \dot{\theta}^2 \sin(\theta) \right]}{\frac{4}{3} l - c_2 \cos^2(\theta)} \tag{1}$$

$$\ddot{p} = c_1 u + c_2 \left[\dot{\theta}^2 \sin(\theta) - \ddot{\theta} \cos(\theta) \right] \tag{2}$$

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) Derive the local linearization of these dynamics around $x^* = (0, 0, 0, 0)$. The eventual dynamics should be in the form

$$\dot{x} = Ax + Bu$$

Note that

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial \ddot p}{\partial p} & \frac{\partial \ddot p}{\partial \dot p} & \frac{\partial \ddot p}{\partial \theta} & \frac{\partial \ddot p}{\partial \dot\theta} \\ 0 & 0 & 0 & 1 \\ \frac{\partial \ddot \theta}{\partial p} & \frac{\partial \ddot \theta}{\partial \dot p} & \frac{\partial \ddot \theta}{\partial \theta} & \frac{\partial \ddot \theta}{\partial \dot\theta} \end{pmatrix} , \quad B = \begin{pmatrix} 0 \\ \frac{\partial \ddot p}{\partial u} \\ 0 \\ \frac{\partial \ddot \theta}{\partial u} \end{pmatrix}$$

where all partial derivatives are taken at the point $p = \dot p = \theta = \dot\theta = 0$.

The solution (to continue with the other parts) is

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

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

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

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:

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

Choose $\epsilon = 1/1000$ and iterate until convergence. Output the gains $K = -R^{-1}B^\top P$. (Why should this iteration converge to the solution of the ARE?)

c) Solve the same Algebraic Riccati equation by calling the `are` routine of the octave control package (or a similar method in Matlab). For Octave, install the Ubuntu packages `octave3.2`, `octave-control`, and `qtoctave`, perhaps use `pkg load control` and `help are` in octave to ensure everything is installed, use `P=are(A,B*inverse(R)*B',Q)` to solve the ARE. Output $K = -R^{-1}B^\top P$ and compare to b).

(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(). Download the code framework here: `https://ipvs.informatik.uni-stuttgart.de/mlr/16-Robotics/e05b-code.tbz2`

Unpack the code such that the folder structure is 'robotics15/share/teaching/RoboticsCourse/riccati/main.cpp'

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