Robotics
Exercise 9

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

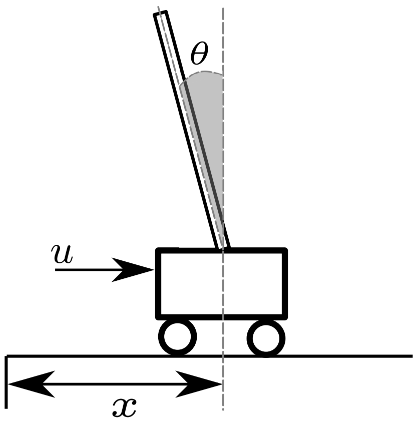
December 12, 2013

1 Aggressive quadcopter maneuvers

In the article D. Mellinger, N. Michael and V. Kumar (2010): *Trajectory generation and control for precise aggressive maneuvers with quadrotors* www.seas.upenn.edu/~dmel/mellingerISER2010.pdf the methods used for aggressive quadcopter maneuvers are described (see also the videos at http://www.youtube.com/watch?v=geqip_0Vjec).

Read the essential parts of the paper to be able to explain how the quadcopter is controlled. (Neglect the part on parameter adaptation.)

2 Cart pole swing-up



The cart pole (as described, e.g., in the Sutton-Barto book) is a standard benchmark to test stable control strategies. We will assume the model known.

The state of the cart-pole is given by $\boldsymbol{x} = (x, \dot{x}, \theta, \dot{\theta})$, with $x \in \mathbb{R}$ the position of the cart, $\theta \in \mathbb{R}$ the pendulums angular deviation from the upright position and $\dot{x}, \dot{\theta}$ 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

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

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

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) Implement the system dynamics using the Euler integration with a time step of $\Delta = 1/60s$. Test the implementation by initializing the pole almost upright ($\theta = .1$) and watching the dynamics. To display the system, start from the code in `course/07-cartPole`. The state of the cart pole can be displayed using OpenGL with the `state.gl.update()` function.

b) Design a controller that stabilizes the pole in upright position *and* the cart in the zero position – any heuristic is allowed (we will use Ricatti methods later). You may want to assume that the range of θ and $\dot{\theta}$ are limited to some small interval around zero (theoretically the implication is that the local linearization of the system is a good approximation). Your controller then needs to ensure that the system does not escape such an interval. (It would be rather hard to design a general controller that can handle any initial state and return the system stably to the target state.)

Test your controller on two problems:

– When the dynamics are deterministic (as above) but the initial position is perturbed by $\theta = .1$.

– When additionally the dynamics are stochastic (add Gaussian noise with standard deviation $\sigma = .01$ to the system state in each Euler integration step).