Exercise: Pole Balancing

Advanced Control lecture
at Ecole Centrale Paris

Anne Auger and Dimo Brockhoff

`firstname.lastname@inria.fr`

Jan 11, 2013

Abstract

Balancing a pole on a moving cart is a standard benchmark problem of control engineering. A related control problem has to be solved within the Segway personal transporter. In this exercise, we implement the most basic pole balancing problem (one single pole mounted on a cart that is only able to move in one dimension where we abstract from friction).

Please keep your code for the next exercises!

1 Simulating the Pole Balancing Problem

Choose your favorite language (recommended: MATLAB/SciLab[1]) and implement the pole balancing problem from the lecture. To this end, use the simple Euler method to approximate the ODEs for angle and position accel-

[1] Useful commands to look at: `sign`, `fprintf`, `disp`

erations, given by

$$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t \left[\frac{-F_t - m_p l \dot{\theta}_t^2 \sin\theta_t}{m_c + m_p}\right]}{l\left[\frac{4}{3} - \frac{m_p \cos^2\theta_t}{m_c + m_p}\right]}$$

$$\ddot{x}_t = \frac{F_t + m_p l\left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right]}{m_c + m_p},$$

the linear controller mentioned in the lecture, and the variables and parameters as described in Table 1.

Recommended Procedure:

a) Start with functions/methods for computing $\ddot{\theta}$ and $\ddot{x}$.

b) Continue with a function for the linear controller.
The response F_t of the controller is based on the four constants k_1, k_2, k_3, and k_4 (choose them from $[0, 1]$) as well as on $F_m = 100N$:

$$F_t = F_m \text{sgn}(k_1 x_t + k_2 \dot{x}_t + k_3 \theta_t + k_4 \dot{\theta}_t).$$

The function $\text{sgn}(x)$ is the signum function, giving 1 if $x > 1$, 0 if $x = 0$, and -1 otherwise.

c) Combine all parts to a single script/function that simulates the system for 120s and a fixed parameter setting using the Euler method:

$$\begin{aligned} x_{t+1} &= x_t + \tau \dot{x}_t \\ \dot{x}_{t+1} &= x_t + \tau \ddot{x}_t \\ \theta_{t+1} &= \theta_t + \tau \dot{\theta}_t \\ \dot{\theta}_{t+1} &= \dot{\theta}_t + \tau \ddot{\theta}_t \end{aligned}$$

d) Output the number of iterations until your simulation results in an unstable pole (θ not in $[-12°, +12°]$) to see whether a given parameter setting is producing a good controller.

e) Test your controller by choosing 1000 different (randomly chosen[2]) settings for k_1, k_2, k_3, and k_4.

[2] Look out for the command `rand(n,1)`.

Table 1: System parameters and variable names for the pole balancing problem.

Symbol	Name	Description
θ	Pole Angle	measured (in radians) relatively to the upright position, initial value in $[-0.1, +0.1]$
$\dot{\theta}$	Pole Velocity	angular velocity of the pole in rad/s
$\ddot{\theta}$	Pole Accelaration	acceleration of the pole in rad/s^2
x	Card Position	measured relatively to the middle of the track (in m), initial value in $[-1, +1]$
$\dot{x}$	Card Velocity	velocity of the cart (in m/s)
$\ddot{x}$	Card Acceleration	acceleration of the cart (in m/s)
g	Gravitational Acceleration	acceleration due to gravity (g =9.81 m/s^2)
m_c	Cart Mass	1.0 kg
m_p	Pole Mass	0.1 kg
l	Pole Length	distance from pivot to the pole's center of mass (l=0.5m)
t	Time	measured in s
F_t	Force	force applied to the cart at time t (in N, always $F_t \neq 0$ for a bang-bang controller)
h	Track Limit	$\pm$2.4m from track center
r	Pole Failure Angle	$\pm 12^\circ$ from vertical ($12^\circ \approx 0.209$rad)
τ	Time Step	discrete integration time step for the simulation ($\tau = 0.02$s)
F_m	Controller Constant	constant of linear controller (set to $F_m =$ 100N)
k_1, k_2, k_3, k_4	Controller Constants	further constants of controller (in $[0, 1]$, to be optimized)

2 Questions

a) Is it easy to find parameter values that produce a stable controller?

b) Are different starting conditions x and θ of the system simulation equally difficult for the linear controller?

c) What is the influence of the simulation accuracy τ?

3 Non-Mandatory Questions

If you have more time, are you able to answer the following two questions?

a) Are the found good controllers also robust to changes in the cart and the pole mass?

b) How would you find robust parameter values of a controller for (more or less) arbitrary starting conditions and masses?

c) Is is easier to build a bang-bang controller or one that allows arbitrary (continuous) forces to be applied in each step?

d) Is this also true if the starting position of the pole is exactly in the middle?