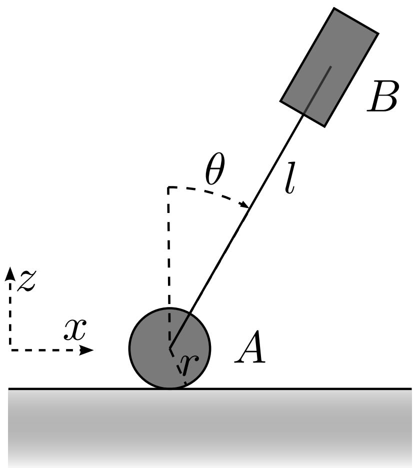
Robotics
Exercise 6

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

November 14, 2014

1 Fun with Euler-Lagrange



Consider an inverted pendulum mounted on a wheel in the 2D x-z-plane; similar to a Segway. The exercise is to derive the Euler-Lagrange equation for this system.
Tips:

- Strictly follow the scheme we discussed on slide 03:23.
- Use the generalized coordinates

$$q = (x, \theta) \tag{1}$$

  with x is the position of the wheel and θ the angle of the pendulum relative to the world-vertical.
- The system can be parameterized by
  - m_A, I_A, m_B, I_B: masses and inertias of bodies A (=wheel) and B (=pendulum)
  - r: radius of the wheel
  - l: length of the pendulum (height of its COM)
- Describe the **pose** of every body (depending on q) by the (x, z, ϕ) coordinate: its position in the x-z-plane, and its rotation relative to the world-vertical. Accordingly represent the (linear and angular) velocities as a 3-vector.
- In this 3-dim space, the mass matrix of every body is

$$M_i = \begin{pmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & I_i \end{pmatrix} \tag{2}$$

2 Optional: PD control to hold an arm steady

In our code, in `03-dynamics` you find an example (rename `main.problem.cpp` to `main.cpp`). Please change `../02-pegInAHole/pegInAHole.ors` to `pegArm.ors`. You will find an arm with three joints that is swinging freely under gravity (ignoring collisions).

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, i.e., $q^* = 0$ and $\dot{q}^* = 0$. Bonus: If you are successful, try the same for the arm in `pegArm2.ors`.

Play with `setDynamicsSimulationNoise` and check stability.

b) As above, try to hold the arm steady at $q^* = 0$ and $\dot{q}^* = 0$. But now use the knowledge of M and F in each time step. For this, decide on a desired wavelength λ and damping behavior ξ and compute the respective K_p and K_d (assuming $m = 1$), the same for each joint. Use the PD equation to determine desired accelerations $\ddot{q}^*$ (slide 03:31) and use inverse dynamics to determine the necessary u.