Robotics
Exercise 6

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

November 21, 2013

1 Direct 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.

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 $\dot q^* = 0$. If you are successful, try the same for the arm in `pegArm2.ors`.

b) (Bonus) Try to use a PID controller that also includes the integral error

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

2 PD acceleration control to hold an arm steady

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 05:31) and use inverse dynamics to determine the necessary u.

Try this for both, `pegArm.ors` and `pegArm2.ors`.

3 The dynamic peg-in-a-hole problem

In the exercise 3 you generated nice collision-free trajectories for peg-in-a-hole using inverse kinematics.

a) Follow these reference trajectories using PD acceleration control (slide 05:31) and thereby solve the peg-in-a-hole problem with a noisy dynamic system.

b) Increase noise into the dynamic system (change to `setDynamicSimulationNoise(2.);`). Record the trajectory of the 3rd joint (`q(2)`) and plot it. Tune the PD parameters to get an oscillatory behavior.