Robotics
Exercise 5

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

November 13, 2013

In the lecture we discussed PD force control on a 1D point mass, which leads to oscillatory behavior for high K_p and damped behavior for high K_d (slide 05:13). Slide 05:14 replaces the parameters K_p, K_d by two other, more intuitive parameters, λ and ξ: λ roughly denotes the time (or time steps) until the goal is reached, and ξ whether it is reached agressively ($\xi > 1$, which overshoots a bit) or by exponential decay ($\xi \leq 1$). Use this to solve the following exercise.

1 PD force control on a 1D mass point

a) Implement the system equation for a 1D point mass with mass $m = 3.456$. That is, implement the Euler intergration of the system dynamics that computes x_{t+1} given x_t and u_t in each iteration. (No need for the robot simulator—implement it directly.) Assume a step time of $\tau = 0.01$sec. Generate a trajectory from the start position $q_0 = 0$ that approaches the goal position $q^* = 1$ with high precision within about 1 second using PD force control. Find 3 different parameter sets for K_p and K_d to get oscillatory, overdamped and critical damped behaviors. Plot the point trajectory (e.g. using the routine `gnuplot(arr& q); MT::wait();`.)

b) Repeat for time horizon $t = 2$sec and $t = 5$sec. How should the values of K_p and K_d change when we have more time?

c) Implement a PID controller (including the integral (stationary error) term). How does the solution behave with only K_i turned on ($K_p = K_d = 0$); how with K_i and K_d non-zero?

2 A distance measure in phase space for kinodynamic RRTs

Consider the 1D point mass with mass $m = 1$ state $x = (q, \dot{q})$. The 2D space of $(q, \dot{q})$ combining position and velocity is also called phase space.

Draft an RRT algorithm for rapidly exploring the phase space of the point mass. Provide explicit descriptions of the subroutines needed in lines 4-6 of the algorithm on slide 03:58. (No need to implement it.)

Consider a current state $x_0 = (0, 1)$ (at position 0 with velocity 1). Pick *any* random phase state $x_{\text{target}} \in \mathbb{R}^2$. How would you connect x_0 with x_{target} in a way that fulfils the differential constraints of the point mass dynamics? Given this trajectory connecting x_0 with x_{target}, how would you quantify/meassure the distance? (If you defined the connecting trajectory appropriately, you should be able to give an analytic expression for this distance.) Given a set (tree) of states $x_{1:n}$ and you pick the closest to x_{target}, how would you "grow" the tree from this closest point towards x_{target}?