Robotics
exercise 2

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

October 23, 2013

1 Task spaces and Jacobians

a) Assume you want to control the position of the left hand (L) relative to the right hand (R), that is p^R, where p is a point on the left hand and p^R its coordinate w.r.t. the right hand's frame. What is the respective task map $\phi: \; q \mapsto p^R$ expressed on the basic transformations $T_{W\to L}$ and $T_{W\to R}$? What is its Jacobian?

b) Assume you would like to control the pointing direction of the robot's head (e.g., its eyes) to point to an external world point x^W. What task map can you define to achieve this? What is the Jacobian?

c) You would like the two hands or the robot to become parallel (e.g. for clapping). What task map can you define to achieve this? What is the Jacobian?

d) You would like to control a standard endeffector position p_{eff} to be at y^*, as usual. Can you define a 1-dimensional task map $\phi: \; \mathbb{R}^n \to \mathbb{R}$ to achieve this? What is its Jacobian?

2 IK in the simulator

Download the simulator code from http://userpage.fu-berlin.de/~mtoussai/source-code/libRoboticsCourse.13.tgz. (See last exercise for instructions.) The header `<src/Ors/roboticsCourse.h>` provides a very simple interface to the simulator—we will use only this header and some generic matrix functionalities.

Consider the example in `teaching/RoboticsCourse/01-kinematics` (rename `main.problem.cpp` to `main.cpp`). The goal is to reach the coordinates $y^* = (-0.2, -0.4, 1.1)$ with the right hand of the robot. Assume $W = \mathbf{I}$ and $\sigma = .01$.

a) The example solution generates a motion in one step using inverse kinematics $\delta q = J^\sharp \, \delta y$ with $J^\sharp = (J^\top J + \sigma^2 W)^{-1} J^\top$. Record the task error, that is, the deviation of hand position from y^* after the step. Why is it larger than expected?

b) Try to do 100 smaller steps $\delta q = \alpha J^\sharp \, \delta y$ with $\alpha = .1$ (each step starting with the outcome of the previous step). How does the task error evolve over time?

c) Generate a nice trajectory composed of $T = 100$ time steps. Interpolate the target linearly $\widehat{y} \leftarrow y_0 + (t/T)(y^* - y_0)$ in each time step.

d) Generate a trajectory that moves the right hand in a circle centered at $(-0.2, -0.4, 1.1)$, aligned with the xz-plane, with radius 0.2.