AI & Robotics: Lab Course
Weekly Exercise 1

Joaquim Ortiz de Haro & Jung-Su Ha
Learning & Intelligent Systems Lab, TU Berlin
Marchstr. 23, 10587 Berlin, Germany

Summer 2022

1 Motion Generation with Pseudo-inverse

Start from the skeleton code **course3-Simulation/02-kinematics/e01-circling.ipynb**.

a) Generate a motion so that the gripper moves in a circular path around the object on the x-z plane. Hint: You can use the coordinate transformation, i.e., from x-z to r-theta. Can you change the radius of the circle and the velocity?

b) Let the object move around (just uncomment the lines with **pos_obj[0] = ...**) and see how the robot moves.

2 Basic IK

The last example in **course3-Simulation/02-kinematics/kinematics.ipynb** demonstrates a minimalistic setup to use optimization for IK. The initial example creates a KOMO instance setup to solve a 1-time-step optimization problem (i.e., and IK problem). Start from this example to generate more interesting motion. The specific tasks are:

a) Vary between the left gripper and right gripper reaching for the object. Is there a difference to "the object reaching for the right gripper" vs. the other way around? And test the left gripper reaching for the right gripper.

b) Also constrain the gripper orientation when reaching for the object. For a start, try to add a **ry.FS.quaternionDiff** constraint. Why does this not work immediately? Try to change the object pose so that the constraint can be fulfilled. Be able to explain the result.

c) There are (in my view better) alternatives to apriori fixing the gripper orientation (the full quaternion). Instead add a **ry.FS.scalarProductXZ** constraint and try to understand. (Zoom into the little coordinate frame in the gripper center to understand conventions.) Play around with all combinations of scalarProduct?? and understand the effect. Further, add one more argument target=.1 to the addObjective method, and understand the result. Using multiple scalar product features, how could you also impose a full orientation constraint, and how would this differ to constraining the quaternion directly?

d) Think more holistically about grasping: How could it be realized properly? For example, think about optimizing a series of two or three poses, where the first might be a so-called pre-grasp, and the others model approach and final grap (from which the gripper-close command could be triggered). Create such a sequence. Note: Do all of this without yet explicitly using collision (pairCollision) features.

For your information, **tutorials/2-features.ipynb** gives an impression about what alternative features one can use to design motion.