AI & Robotics: Lab Course
Weekly Exercise 4

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

Summer 2022

1 Grasp, lift & drop a hopping ball

In **05-grasp/main.ipynb**, you already find a minimal setting where a dropping and hopping ball is grasped. The hopping is realized in simulation by an *imp* – which is a little callback code that can perturb the state of the simulation after each simulation step. This helps simulating stochastic effects and failures.

a) Use your OpenCV code from Exercise 3 instead of cheat perception to continuously track the sphere. Keep your model object always tracking the real red sphere.

b) The current code uses plain IK with a pseudo inverse from Exercise 1 to generate motion (you can stack the Jacobians and residuals for multiple objectives, see the motion slides). Try to replace this by a KOMO_IK method that solves, in each iteration, for the optimal final grasp pose, and then generate continuous motion by commanding a small constant velocity towards that final grasp pose. Close the gripper when the final grasp pose is reached.

c) Implement a similar behavior using a KOMO_path method with a constant final acceleration (recall Exercise 2). When would you need to re-optimize the path?

d) The current code sometimes makes the arm occlude the red ball. Can you choose a better alignment to avoid ball occlusion?

e) After the successful grasp, lift the gripper (e.g., using KOMO to compute a good final lift/drop pose for the ball).

f) Then call `openGripper("R_gripper")` to drop the ball again.

g) Ideally, repeat grasp-lift-drop robustly in an endless loop.

2 Throw the ball

Do everything as above, but instead of just dropping the ball, throw it (far) using the robot.

a) How can you design a motion that has a desired object velocity and the point of `openGripper`?

b) Change the object shape to `capsule` and try to throw it. Can you also control the angular velocity of the thrown object?

3 Grasping a box

Repeating the above exercise for a box is rather involved, esp. if you do not cheat perception and have to retrieve the box orientation (long axis) using opencv. Therefore, for this exercise it is fine for you to use cheat perception.

a) Replace the red sphere by a sphere-swept box of size [.06, .06, .15, .01] (in both, the **RealWorld** and your model world) and let the box lie on the table (i.e., the box's z-axis is orthogonal to the world's z-axis).

b) When localizing the box, retrieve both, the center and the rotation matrix. Note that the third column in the rotation matrix is the unit vector which points into the direction where the box is 15cm long. Let A be this direction.

c) Design a box gripping final pose, where the gripper x-axis is orthogonal to A, and orthogonal to world-z. Does this suffice to design a good box grasp? (Cf. Exercise 2)