CMPE 565 AUTONOMOUS ROBOTS
Assignment 5: Kinematics

Deadline: 15.03.2018

In this lab, you will implement to calculate the forward and inverse kinematics with the position error for the UR10 in "Virtual Robot Experimentation Report (VREP)".

- The environment is given in scene "assignment5.ttt" in scenes folder.
- Three nodes in rosws/src/ur10_kinematics folder are given to calculate forward kinematics and inverse kinematics (and bonus).

1 Forward Kinematics

- The node is implemented in UR10ForwardKinematics.cpp.
- You are expected to implement all `TODO` sections.
  - You should find the **DH** parameters and write correct **Chain Segments** according to the UR10 in V-REP.
  - You should read joint angles from a file.
  - You should calculate the **End Effector Position**.
  - You should publish the joint angles to UR10.
  - You should fill the **calculatedPosition** variable according to found End Effector Position.
  - You should show the error for five different set of joint angles.

2 Inverse Kinematics

- The node is implemented in UR10InverseKinematics.cpp
- You are expected to implement all `TODO` sections.
  - You should write **Chain Segments** from ForwardKinematic.
  - You should calculate joint angles for each **Target Point (/task/positions/targets)**.

  - You should give the joint angles to robot.
  - You should fill the **calculatedPosition** varable according to Target Point.
  - You should show the error between **Target Point** and **End Effector Position**

3 Bonus Kinematics

- The node is implemented in UR10BonusKinematics.cpp
- You are expected to implement all `TODO` sections.
  - You should write **Chain Segments** from ForwardKinematic.
  - UR10 should take the Cup and carry it to Bill.
  - You should calculate and show the error which is the distance between the **CupTarget position** and **End Effector Position**.

4 Deliverables

Your reports should have the following sections:

- Introduction (Brief description of the experiment, assumptions with reasons)
- The frame construction for UR10.
- DH parameters of that arm
- The code segment for forward kinematics
- The results for the forward kinematics experiments
- Explanation for the way of inverse kinematics calculation
- The code segment for inverse kinematics
- The results for the inverse kinematics experiments
- The errors between the actual position and the positions found by forward and inverse kinematics
- A video for each section
- Conclusion (Comments for the experiment results)

5 Tips

- TargerPoints, Cup and CupTarget Positions are according to the base of the arm of UR10. You can directly use them.
- You can use this website for learning Orocos KDL: `http://www.orocos.org/kdl`
- You can watch this video for finding DH parameters. However, in this video, CW degrees are positives degrees and 3 Finger Rule is reversed. Therefore, at the end, you will get the same DH Parameters. `https://www.youtube.com/watch?v=rA9tm0gTln8`
- In the KDL, KDL::Chain contains segments and KDL::Segment contains KDL::Joint and KDL::Frame. KDL::Joint is used for rotation axis and the KDL::Frame is used for the location of the next joint (or end effector).
- KDL::Frame has two parameters KDL::Rotation and KDL::Vector. KDL::Vector is the relative position of the next joint (or end effector) according to current joint position. KDL::Rotation is also used when the rotation angle of the next joint is not defined in the KDL. For example in UR10, some of the joints are rotates around the -X axis. Therefore you should rotate the joint by 180 degrees.
- You can also use DH parameters directly in KDL. Also, you can use any library and any method for this lab.
- You can use */UR10/gripper/closingAngle* for gripper, it took 0 - 10 values for closing the gripper.
- Every joint position is also published according to base position. You can add joints into your system iteratively.