Lab: Introduction to RoboDK software

This lab will help you get familiar with the RoboDK software package so you can complete the lab assignment using it. RoboDK has been installed on the PCs in the Mechatronics Design Lab. For further information, please see the RoboDK online documentation: https://robodk.com/doc/en/Basic-Guide.html#Start

Before you start RoboDK (RDK), you should download the RDK station file from Learn: ENMT482S2 - Robotics > Manipulators > RoboDK Introduction > RoboDK Practice Station.rdk (https://learn.canterbury.ac.nz/pluginfile.php/1531492/mod_folder/content/0/RoboDK%20Practice%20Station.rdk?forcedownload=1)

Please do NOT use this station as the basis for your actual assignment. So use it for this lab to get familiar with the software, but please don't use it for the actual assignment. I will provide a more accurate version in the very near future.

1. Open RoboDK
2. Load *RoboDK Practice Station.rdk* using the folder icon on the toolbar or File > Open
3. Try navigating around the model: https://robodk.com/doc/en/Basic-Guide.html#Nav3D
4. Bring up the *Robot Panel* by double-clicking on the robot (UR5) in the item tree, or by right-clicking on the robot and selecting Options.
   https://robodk.com/doc/en/Interface.html#RobotPanel
   a. To control the robot, you can modify joint angles, or the pose in Cartesian space (*Tool Frame with respect to Reference Frame*). Alternatively, you can drag the joint-sliders. Even more alternatively, you can hover the cursor over the joint-slider (or coloured box in pose section) and roll the mouse scroll-wheel. You'll see the joint angles and pose updating. (You don't really want to go changing the top two sets of information – Tool Frame and Reference Frame)
   b. For a given pose, you can explore the other configuration options using the *Other Configurations* section. Try several other configurations.
   c. You can visualise the robot workspace (i.e. the extent of the robots reach) with the *WorkSpace* section.
   d. You can visualise the link frames (from DH parameters) with the *Frames* section. Note that pressing + and – makes the frames (axes larger or smaller). Also note that it is standard in robotics (and many other areas) to colour the axes so RGB correspond with XYZ, i.e. the x-axis is red and the z-axis is blue.
   e. You can Copy or Paste poses or joint angles using the icons/menus to the right of each section.

   → Note: For the UR5, the position part of pose is defined by Cartesian coordinates (XYZ), and the orientation part uses *angle-axis* (i.e. a unit vector describing an axis, scaled by the rotation about that axis – more on this from the lecture material)

5. In the item tree, you'll see that I have saved some waypoints (targets) clicking on these will send the robot to these points. Try it.
6. To set your own waypoint, move the robot to the desired position and click the *Teach Target* icon in the toolbar (looks like a target with an arrow in it). https://robodk.com/doc/en/Interface.html#RobotTarget
7. To change the target settings, right click the new target in the item tree and select *Options*.
    a. You can name your target something meaningful
    b. You can change the target type to a joint target (i.e. the robot goes to the joints angles, rather than the Cartesian pose) – this can be useful in forcing a specific configuration. You can also specify/adjust the configuration.
    c. More importantly, you can edit the location of the target (joint and or Cartesian pose).
    d. You can change the parent reference frame for the target (i.e. the frame in which it is expressed)
8. You can add reference frames to the model by clicking *Add a Reference Frame* on the toolbar. Targets and tools etc need to have a parent reference frame. To modify the name/location/parent of your new frame, right-click it in the item tree and select *Options* (you should be starting to see a trend…). https://robodk.com/doc/en/Interface.html#RobotReference
9. To program the robot, you can use the Python or Matlab APIs (https://robodk.com/doc/en/RoboDK-API.html#RoboDKAPI), or directly in RDK. https://robodk.com/doc/en/Robot-Programs.html#RobotProgram
    a. Click *Add New Program* in the toolbar
    b. Move the robot to the first waypoint (e.g. Home)
    c. Click *Add Joint Move*. (It is always good to make the first move to the starting position as a joint move, i.e. linear in joint-space)
    d. Move the robot to the next point (by clicking on it in the item tree), then click *Add Linear Move*.
    → You'll see the path shown in yellow. Use the path to see the difference *between Joint Move (MoveJ)* and *Linear Move (MoveL)*
    e. Many more instructions are available through the *Program* menu or by right-clicking the program in the item tree and hovering over *Add Instruction*.
    f. With the robot near one of the tools on the tool stand, try the *Simulation Event* instruction. You can attach and detach the closest tool to the robot.

    → Note: The *Set Rounding Instruction* does not render in RDK (i.e. in simulation, the robot stops at every waypoint), but does cause the real UR5 to move through the waypoints without stopping.

10. You can toggle collision checking via the icon in the toolbar.

Feel free to explore other aspects of the software, particularly the API's. With RDK open, you can control the robot from the Python console.