Tutorial: 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 Lab and Mech Eng PC Lab. For further information, please see the RoboDK online documentation.

Before you start RoboDK (RDK), download an RDK station file from the RoboDK Stations on GitLab. If you intend to use your own device for this assignment, follow the instructions on the GitLab README on installing the requirements.

For the purposes of this introduction, you may choose any of the three stations to practice on. However, when you go to start work on the assignment, make sure you are using the correct station corresponding to the robot you are working on. This is a very important detail, and failing to use the correct station file on the robot could cause damage to the expensive equipment.

1. Open RoboDK
2. Load *Robot_X_2025.rdk* with File → Open or using the folder icon on the toolbar (Fig. 1).

Figure 1

3. Try navigating around the model.
4. Bring up the *Robot Panel* by right-clicking on the robot and selecting Options or double-clicking on the robot (UR5) in the item tree (Fig. 2).

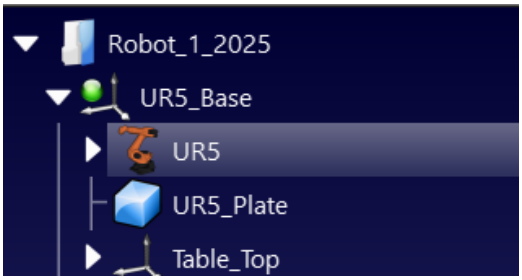

Figure 2

   a. To control the robot, you can modify joint angles, or the pose in Cartesian space (*Tool Frame with respect to Reference Frame*)(Fig. 3). Alternatively, you can drag

the joint-sliders (Fig. 4). 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).

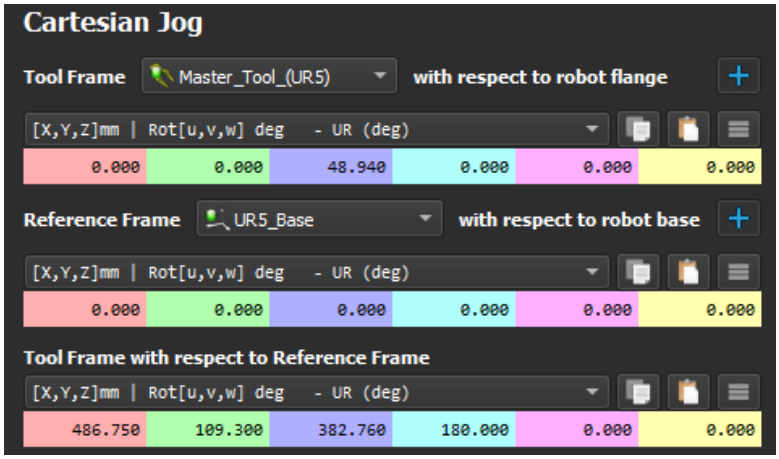

Figure 3

Figure 4

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 corresponds 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. Python scripts can be used to program the robot - this will be the basis for your assignment.

   a. With the RoboDK workspace open, bring up a terminal and navigate to the *Python-Embedded* directory in the RoboDK program files. On the lab machines, this can be done by opening *Command Prompt* and typing:

      ```
      cd /d C:\RoboDK\Python-Embedded
      ```

      (Another way to do this would be to navigate to the directory in file explorer and then enter *cmd* in the address bar.)

   b. Enter:

      ```
      python {filepath},
      ```

      replacing `{filepath}` with the script you want to run, e.g. `"D:\ENMT482-assignment\my_program.py"`. This script can be modified (in VSCode for example: see below) and run again in the same manner.

      Try running the `robodk_basics.py` example using this method.

   c. Run the RDK_code_example_adv_v2025.py file on LEARN under ENMT482-25S2 → Manipulators → 2025 UR5 Lab assignment, for an example on how to define and use transforms.

      Note: If you are getting an error because numpy (or some other package) isn't installed, you can use pip to install packages on the machines in the lab:

      ```
      python -m pip install {package}
      ```

To integrate VSCode (found in `C:_Shortcuts\_Programming\` on the Mechatronics Lab machines), the *python* extension must first be installed. Then, select the python.exe in the *Python-Embedded* directory as the python interpreter. You should then be able to run your python script directly from VSCode, provided RoboDK is open.

→ Note: You can toggle collision checking via the icon in the toolbar which may help to identify collisions in your program when moving around the cart. However, by default, this will identify any contact between the robot and environment, including tools. The *collision map* can be used to ignore collisions between specified objects. This can be found under the drop-down on the collision check button in the toolbar, or with Shift-X.