ENMT482 - Manipulators Assignment (25%) 2025v1

The overall goal of this assignment is to use the UR5 to perform some basic coffee making tasks using the new interchangeable *barist-effector* tools. You will also need to perform a time-and-motion study to quantify the potential benefits of automating this task.

Learning outcomes: The objective of this assignment is to get some experience applying the transforms you have learned about in class to the real world, and operate an industrial robot (without damaging it!). You will also learn to objectively analyse the potential benefits/downsides of automating a process.

Information: You will be provided with:

1. Coordinates for 2 or 3 identified points on each fixture (in the local frame and robot base frame).
2. Coordinates of the buttons, levers etc that you must actuate in local frames attached to each fixture.
3. Coordinates of relevant tool points in their local frame (not necessarily the same as robot TCP frame).
4. A model of the workspace in RoboDK, including the tools and UR5 robot to enable simulation and programming.

Note: There will be some things that you may need to measure, determine, or refine yourselves, e.g. how far to pull the lever on the Mazzer, orientation of tools required to complete a task, the exact location on a button where you need to press, etc.

You will need to complete to following:

1. Determine the transformation matrices between the coffee making equipment (Rancilio (espresso machine), Mazzer (grinder), etc.) and the robot base frame (= world frame).
2. Determine the transformation matrices between the tools and the robot's tool centre point (TCP) - named *Master Tool* in the RoboDK workspace.
3. Program the robot to complete the Demonstration tasks without colliding with itself or any other equipment within its workspace.

   Obviously, there will be intermediate steps involved, such as changing tools, robot configurations etc.

   Notes:

   I. During the entire sequence of tasks, the robot and it's tool must remain within the bounding box defined by the extents of the table.

II. In programming the robot in RoboDK, you must have the tool frame set to Master Tool, and the reference frame set to UR5 Base.

III. All tool changes must be done using the supplied routines, which must be called using precisely the same syntax as used in the Python example code.

4. Time-and-motion study

   This involves making a comparison between the time it takes a human operator (Rodney) and the UR5 running your finalised code to perform each task, and the complete coffee making process. This information can then be used to evaluate the financial viability of the robotic system - does it make sense to automate this task? If so, over what period does the robotic system pay itself off?

   Some quantitative information that you may want to use:

   - Cost to purchase UR5: $40k
   - Cost of tooling: $5k
   - Annual maintenance of UR5: $400 + shutdown for 3 hours
   - Hourly wage of experienced barista: $30 /hour
   - Human working conditions: 2 × 15 min breaks + 30 min lunch per 8 hour shift
   - Average cost of cup of coffee: $5

   You should discuss other factors that may influence the decision to automate this task. Some examples include:

   - Improvements that may increase the speed of the robotic system
   - What times of the day people typically buy coffee
   - Other costs not included above

   Finally, make a recommendation on whether automating this task would be worthwhile.

   You can find a video of Rodney making a coffee on the course Learn page.

5. Demonstrate your program working on the physical robot.

   Notes on the demonstration:

   - Failure to successfully run your code will result in 0 marks for this element of the assignment and no further opportunity to demonstrate - make sure your code works before you come to the demo!
   - Penalties will be applied for any collisions, or uncompleted tasks - see below

Format: These tasks and report must be completed in groups of three. Reports are to be submitted via Learn by 5pm Friday 17 October.

Resources: In addition to the simulation environment, 20-minute slots are available on the UR5s for testing your code on the real platform.

- Each group is guaranteed two sessions per week. If these sessions are not used, they expire at the end of each week.
- The sessions will be available during afternoons (14:00-17:00) of 8-Sept - 15-Oct. Groups will largely be able to choose the specific time of each session, using a sign-up sheet outside the lab. If you have constraints for which times you can attend, make sure you sign up early, as we will not put on additional slots.
- Each group will be assigned to one specific robot. You cannot change robots as the exact positions of the pieces of equipment on the carts are slightly different.
- If you are unable to make a slot for which you have signed up, please remove your name from the signup sheet so that the slot becomes available for others to use.

Demonstration tasks

The following series of tasks must be performed by the robot to produce a cup of coffee:

a. Pick up the *Rancilio tool* and place it on the Mazzer Scale pan.

b. Use the *Mazzer tool* to unlock the Mazzer Scale.

c. Use the *Mazzer tool* to turn the Mazzer on, wait 15s, and turn the Mazzer off.

d. Use the *Mazzer tool* to pull the Mazzer dosing lever until the scale reports 20$\pm$0.1g of coffee grounds has been deposited in the *Rancilio tool*.

e. Use the *Mazzer tool* to lock the Mazzer Scale.

f. Remove the *Rancilio tool* from the Mazzer.

g. Open the WDT fixture, and place the *Rancilio tool* into the WDT fixture.

h. Release the *Rancilio tool* and close the WDT fixture.

i. Use the *Mazzer tool* to turn the WDT rotor five full revolutions.

j. Open the WDT fixture, remove the *Rancilio tool* and close the WDT fixture.

k. Place the *Rancilio tool* into the PUQ fixture, and wait 2 seconds while the machine tamps the coffee grounds.

l. Remove the *Rancilio tool* from the PUQ fixture, and insert it into the Rancilio group head.

m. Use the *Mazzer tool* to operate the cup dispenser.

n. Use the *cup tool* to pick up the dispensed cup, and place it on the Rancilio Scale pan.

o. Use the *Mazzer tool* to unlock the Rancilio Scale.

p. Use the *Mazzer tool* to operate the Rancilio hot water switch until the scale reports 32$\pm$0.1g of water has been dispensed in the cup.

q. Use the *Mazzer tool* to lock the Rancilio Scale.

r. Use the *cup tool* to carefully pick up the cup of coffee and place it in the customer zone.

s. Remove the *Rancilio tool* from the group head.

t. Position the *Rancilio tool* over the Rancilio Tool Cleaner fixture silicone brush, and actuate for 5s.

u. Position the *Rancilio tool* over the Rancilio Tool Cleaner fixture bristle brush, and actuate for 5s.

v. Return the *Rancilio tool* to the tool stand.

NB: Your robot should start and end in the (left or right) <u>Home</u> position.

Further details on these tasks can be found in the GitLab README:
https://gitlab.com/uc_mech_wing/robotics_control_lab/uc-02-2024.

Assessment

This assignment will constitute 25% of your course grade and will be distributed as follows:

- Milestone demonstration: 3%
- Final demonstration: 12%
- Report: 10%

Demonstration Marking Criteria:

For each of the Demonstration tasks above, your will receive 1 mark for each task correctly demonstrated.

Collisions with equipment will be penalised at -1 per collision, up to a maximum of -3.

Robot or tool moving outside the bounding box will receive a penalty of -1 for the demonstration.

Note: you cannot receive a negative mark for either demonstration.

Milestone demonstration: The milestone will require you to demonstrate tasks (b), (c) and (m) on the robot. You may demonstrate this milestone during any of your lab timeslots, however, it must be completed by 5pm Friday 26th September. Your mark will be out of 3, determined by your ability to meet the Demonstration Marking Criteria above.

Final demonstration: The final demonstration will require you to perform tasks (a-v) on the robot. You may demonstrate your finished program at any of your lab timeslots, however, you must have demonstrated by 5pm Friday 17th October. The final two days (Thurs 16th and Fri 17th Oct) will be reserved for demonstration purposes only. Your mark will be out of 22, determined by your ability to meet the Demonstration Marking Criteria above. There is no hard time limit for demonstrations, however, you should aim for around 15 minutes.

Notes on programming

Note: The following information may be updated throughout the assignment as we update files.

Correct as of: 01/09/2025

Remember that there are three coffee carts which are slightly different, so each has its own .rdk file. You MUST use the .rdk file corresponding to the robot you have selected to use, and you must use the same robot/file for the duration of the assignment. Failure to do so may result in collisions/inaccuracies, since each robot station is not exactly the same. The group names for this assignment on LEARN also reflect the station, so make sure you are consistent.

Use of included programs:

In the `tools.py` module within the `robodk_stations` directory on GitLab, there are numerous methods to perform relevant tasks within the station, such as changing/actuating the robot tools and actuating the cart fixtures. These methods must be called from the `Tools` class. Examples of how to use these programs can be found in the `robodk_basics.py` file in `robodk_stations`. Remember, you MUST use the tool methods when picking up/dropping off tools at the tool stand.

The methods can be called at any time. However, for changing tools, you may want to move the robot to an 'approach' position before calling the relevant tool change method to avoid collisions.

YOU MUST NOT MODIFY THE METHODS OR TARGETS ALREADY IN THE FILE!! Doing so may cause the robot to collide with the tools/equipment, resulting in very expensive damage.

In addition to the `Tools` class, there are also various visual programs which make no physical changes, but can be used to help with visualising the simulation process. These can be found in the *Visual Programs* directory with the robodk station tree.

NB: Despite the fact that the visual programs have no real-world effect, if they are to be called from code that is running online, they must have their "Run on Robot" property set to true, otherwise RoboDK will throw an error. To set the property, right-click on the visual program in the station tree and check the "Run on Robot" box.

Refer to the GitLab README for further information:
`https://gitlab.com/uc_mech_wing/robotics_control_lab/uc-02-2024`

Any questions, please ask Chris or Rodney.