ENMT482 – Robotics – Assignment (25%) 2020v1

The overall goal of this part of the 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 point of this part of the 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 two identified points on each piece of equipment (in the robot base frame).
2. Coordinates of the buttons, levers etc that you must actuate in local frames attached to each piece of equipment.
3. Coordinates of relevant tool points in their local frame (note 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. This model will also contain several programs to complete tool pick-up/drop-off from set standoff points. Please use these, but don't modify them as you may damage the expensive tools.

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

Tasks: You need to complete to following

1. Determine the transformation matrices between the coffee making equipment (coffee machine, grinder, tool stand, cup, tamper-brush) and the robot base frame (=world frame).
2. Determine the tool-transformation matrices between the tools and the robot's *tool centre point* (TCP).
3. Program the robot to complete the following series of tasks without colliding with itself or any other equipment within its workspace:
    a. Place *portafilter tool* under the grinder dosing head.
    b. Use *grinder tool* to turn the grinder on, wait 3s, turn the grinder off.
    c. Use *grinder tool* to pull the grinder dosing lever to deposit ground coffee in the *portafilter tool*.
    d. Scrape coffee from the rim of *portafilter tool*
    e. Tamp coffee.
    f. Move *portafilter tool* to the coffee machine (*Silvia*) standoff location (TA will manually insert into coffee machine).
    g. Pick up a coffee cup with *cup tool*.

h. Place a coffee cup on the drip-tray of the coffee machine under the *portafilter tool*.
i. Use *grinder tool* to turn coffee machine on, wait 3s, and turn off.
j. Hand cup of steaming coffee to Rodney

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

4. Time-and-motion study

This involves you making a comparison between the time it takes a human operator (Rodney) and the UR5 <u>running your finalised code</u> 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: $27 /hour
- Human working conditions: 2x 15 min breaks + 30 min lunch per 8 hour shift
- Average cost of cup of coffee: $4.50

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 Learn.

5. Demonstrate your program working on the physical robot.

Notes on the demonstration:

- Failure to successfully compile 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!
- The task must be finished within 10ish mins
- Penalties will be applied for any collisions, or uncompleted tasks – see below.

Demo days: October 14-16

Format: This task and report <u>must be completed in groups of two</u>. Reports are to be submitted via Learn by <u>5pm Thursday 15 October</u>.

Resources: In addition to the simulation environment, 10 & 20-minute slots are available on the UR5s for testing your code on the real platform.

- Each group will receive an allocated number of sessions per week. If these sessions are not used, they expire at the end of each week.
- These sessions will be available during afternoons (13:40-17:00) of 14-Sept – 14-Oct. Groups will largely be able to choose the specific time of each session, using a sign-up sheet in 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.

Report/project guidelines

For the UR5 report, the following are some guidelines on what to include, how it will be marked etc. While I won't impose a strict page limit, the report should probably be less than 10 pages, not including code in Appendix (This isn't a design report!)

Your report should include

- Introduction
- Frame assignment for the equipment and tools, along with some explanation about why you assigned frames the way you did.
- Determination of the transformation matrices between the equipment frames, tool frames, and the robot frame.
- Path planning and obstacle avoidance
- Time-and-motion results and discussion
- What you learned and what you suggest
- Code in an Appendix

Within this framework, you need to describe your methodology, why you chose it, and maybe reflect on the outcome/performance. You should also <u>briefly</u> explain things like the frame assignment and why it is important and how it relates to this assignment and the software running the robot, etc.

As you are all just about ready to go off into the world as professional engineers who will need to write reports for clients, colleagues etc, quality of writing is also very important. Spend some time making sure your report is concise and well written as this will also impact your mark.

Marking:

The report will be marked out of 10

The demonstration will be marked out of 5

Total marks for the lab: 15, contributing 25% to your final grade

An approximate marking guide for the report (out of 10) is given below:

10: Report is exceptionally well written and presented, and demonstrates excellent insight and effort on aspects such as optimising paths or use of certain commands to achieve rapid/optimal/fancy behaviour. Code is clever/cunning/elegant and demonstrates excellent knowledge of the robot and its programming environment. Code is well commented and easy to understand. Basically, going above and beyond…

7: Report is well written. Covers all the basic elements and shows a good degree of understanding of how to use the techniques taught in class. Good use of figures to illustrate aspects of the report. Code is commented and its flow/function can be easily understood.

5: Poorly written report, coding is very basic.

Marking for the demonstration:

5: All tasks completed satisfactorily

4: for only completing 7 of the 10 tasks listed (a-j)

3: for only completing 5 of the 10 tasks listed (a-j)

0: no tasks completed

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

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

Currently correct as at: 18/09/2020

** Notes on programming:

Remember that there are two stations (2019 and 2020 – named for the robot serial number), which are slightly different, so each has its own rdk file. If you start using one station, stick with it, as changing may cause collisions/inaccuracies.

- Station 2019: *UR5 Apparatus 2019.rdk*
- Station 2020: *UR5 Apparatus 2020.rdk*

Use of included programs:

In the RDK file there are several programs to perform final approach and tool attach/detach. You just need to get the robot to/from the appropriate point named *<Tool> Tool Approach* (e.g. *Cup Tool Approach*) and then call the required program (e.g. *Cup Tool Attach (Stand), Cup Tool Detach (Stand)*). Details of these approach points can be found in the RDK tree: *UR5 Apparatus 20xx > UR5 Base > Table Top > Tool Stand Plate Base > Tool Stand Base > [Tool] Top > [Tool] Approach*

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

Regarding the two programs to detach and attach the robot from the portafilter tool at the grinder; *Portafilter Tool Detach (Grinder), Portafilter Tool Attach (Grinder)*. To use the programs:

Detach - get the portafilter tool positioned on its stand, then call the program. The robot will operate the IOs and move back from the tool to the position so you can go and get the pointy one and continue the coffee making

Attach - To reattach, move the TCP to target, then call the program. The robot will move in and reattach to the tool. You then need to remove the tool from the grinder.

Any questions, please ask Rodney or myself.