UNSW
SYDNEY

MTRN4230 T2 2022 ROBOT-2: Robot Programming

Learning Outcomes

- Implement good safety practices in the use of robots.
- Employ a robot and robot programming environment effectively and efficiently to achieve a given task

Aims

- To gain experience in programming the UR5e robot

Due Date

- Week 4: By the end of your scheduled lab session

Prior to Lab Demonstration

Safety

1. The robot will be placed in manual mode during the lab so you can edit the TCP and add features.

Online Training

1. Complete Modules 9 – 11 of the e-series Pro Track.

Matlab or Polyscope?

You are welcome to use the RTDE scripting library provided, or you can use polyscope provided on the URsim Teach pendant to create the program for this assessment.

If you are using matlab you must submit only one matlab file with the following naming format:

- **yourZID_ROBOT-2.m**

If you are using URsim you must submit a file with the following naming format:

- **yourZID_ROBOT-2.urp**

Students are allowed to use any UR5e's not being used for marking to practice their solution. When they are ready to get marked, they will need to either fill out an online form or write their name on the whiteboard according to the demonstrator's preference. As soon as the one-hour mark of the allotted 3-hour lab time hit, the lab demonstrator will start marking students who have yet to be marked to put their name down to be marked.

In Person Assessment:

Part A: Trace digits

Students are required to demonstrate either (1) “Trace digits in ascending order” **OR** (2) “Trace digits in random order”. The description of the tasks will be expanded under their relative headings. Their marks are provided in the marking criteria.

Lucida Console of size 100 will be used. This is a monospace font. If you don’t have this font, don’t worry, try to find any monospaced font and use that at size 100. The font must not be a 7-segment style font, it must have curves in the relevant numbers.

1. **Basic task: Trace digits in ascending order [0 - 4%]**

This is the basic task. Students are required to program or teach the robot to follow a path which traces out the digits from 0 to 9, on a horizontal plane exactly 60 mm above the table with the final joint facing directly down.

- Write out digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 in order
- Start and end the motion at the robot home position.
- At the start of your program, you must display the sequence of digits to the user:
    - If using **MATLAB**, you are required to print the digits 0 to 9 to the console.
    - If using **POLYSCOPE (URsim)** you are required to use a **popup** to display the sequence of digits that are to be written.
- At the end of the program, you must indicate to the user that the program is complete:
    - If using **MATLAB**, you are required to print “Program Complete” to the console
    - If using **POLYSCOPE (URsim)** you are required to use a **popup** to display “Program Complete” to the user
- The bottom left hand corner of the A4 sheet of paper will be placed at the following x,y,z position [-588.53, -133.30, 0]. This position is vertically below the robot’s “HOME” position. **Please look at the Appendix for a diagram.**

2. **Advanced Task: Trace digits in random order [0 – 6%]**

This is the advanced task and is worth more than the basic task. Students are required to program or teach the robot to follow a path which traces out 10 random digits on a horizontal plane exactly 60 mm above the table with the final joint facing directly down. The demonstrator will provide the random 10 digits at the start of your demonstration, you will not know these beforehand and must be able to handle it on the spot.

- Write out a random series of 10 digits.
- Start and end the motion at the robot home position.
- At the start of your program, you must display the sequence of digits to the user:
    - If using **MATLAB**, you are required to print the digits 0 to 9 to the console.
    - If using **POLYSCOPE (URsim)** you are required to use a **popup** to display the sequence of digits that are to be written.
- At the end of the program, you must indicate to the user that the program is complete:
    - If using **MATLAB**, you are required to print “Program Complete” to the console
    - If using **POLYSCOPE (URsim)** you are required to use a **popup** to display “Program Complete” to the user
- The bottom left hand corner of the A4 sheet of paper will be placed at the following x,y,z position [-588.53, -133.30, 0]. This position is vertically below the robot’s “HOME” position. **Please look at the Appendix for a diagram.**

Part B: Translation and Rotation [0 - 2%]

Extend the basic or advanced program to trace the digits when the piece of paper is in an arbitrary pose on the table.

If using **MATLAB**, we will provide you with the origin of a plane parallel to the table and a rotation about the Z-axis (e.g. X = -300mm, Y = -300mm RZ = - 30 deg), all in base frame coordinates. This coordinate will indicate the bottom left corner of the A4 page. To consider this, we will adjust the paper when testing to match your starting position.

- **Hint:** Have a look at the following video by Peter Corke which covers how to use the RVC toolbox to create transformation matricies.

If using **URsim,** you are allowed to "Edit" a pre-set frame of reference or use the "Teach this plane" function. You are also allowed to teach the robot the position and orientation of the page during the assessment if you would like.

- Hint: Refer to Module 10 of the e-Series Pro Track

1% will be awarded if you can program for an arbitrary translation.

1% will be awarded if you can program for an arbitrary rotation.

Part C: Mimicking a human [0 - 2 %]

Extend the basic or advanced program to mimic human writing by considering the stroke order of the digits themselves and by always making smooth motion along the path.

Stroke Order: [0 – 1%]

The robot must take the same stokes and movements you would as a human when writing a specific digit. This includes the transition from one digit to the next as well.

Smooth Motion: [0 – 1%]

The robot must have a reasonably constant velocity during the entire program. It must not display any jerking motions while drawing a digit.

Part D: Understanding of Implementation [0 - 5%]

Following the demonstration of the task, you will be required to answer questions verbally from the demonstrator on how you have approached this task.

The demonstrator will ask you discuss the following:

1. What was your approach to solve this task? (How does your program work?)
2. What are some of the advantages of your chosen approach?
3. What are some limitations of your chosen approach?
4. What would you do to improve or develop your solution in the future?
5. What did you learn about the robot's capabilities, while completing this task? Did you encounter any problems, what were they and how did you solve them?

Example Demonstration Situations – In Person students:

- If you are attempting the basic task without any arbitrary translation or rotation you will only need to present the basic task. You may also get marks for Part C if you took stroke order and/or smooth motion into consideration.
- If you are attempting the basic task and have programmed for any arbitrary translation and/or rotation as mentioned in Part B, at the start of your demonstration you demonstrator will either provide you with arbitrary values for rotation and translation (if programming in matlab) or allow you to edit or teach a plane using URsim. You may also get marks for Part C if you took stroke order and/or smooth motion into consideration.
- Similarly for the advanced task, you may have implemented the arbitrary translation/rotation. You will be given marked accordingly.

Online Students only Assessment:

The assessment for online students will be identical to the tasks that in person students need to complete. You will also be required to send your program to the demonstrator to run during the lab. Your lab demonstrator will release a form at the start of the session. When you are ready to enter the queue during your lab time slot, please fill it out. The following two dot points are the extra considerations made for online students. It is recommended that you use MATLAB if possible, to visualise the output poses of your program.

- You will also have an opportunity to make a practice or an assessed submission, preference will be given to students who are ready to be assessed.
- The tasks and marking criteria will be identical to the in-person assessment, but you will also be given 10 minutes preparation time to setup your code and send it through to your demonstrator.

The Marking Procedure:

1. When the demonstrator contacts you:
    a. Let them know if you are attempting the basic or advanced task.
    b. Let them know if you are also attempting the arbitrary rotation or translation component.
    c. The demonstrator will the provide you with instructions accordingly. If attempting the advanced task, they will provide you with the sequence of random 10 digits. If attempting the translation and/or rotation component as well, they will also provide you with what the translation and rotation components are.
    d. You will then have 10 minutes to make the necessary changes to your program
2. Once you are happy with your program, you will be required to send it to your demonstrator via MS Teams.
3. Please note that we will not run any submissions that do not match the expected safety checksum or do not follow the specified submission format.
4. The demonstrator will then run your program and you will be able to see it via a Teams video call.
5. After your demonstration of the task is finished, you will then need to verbally answer questions from the demonstrator on how you have completed the task.

Marking Criteria

The overall mark for this assessment is 15% of the final course mark. It has been distributed as below. Late submissions are not permitted without a special consideration application being approved.

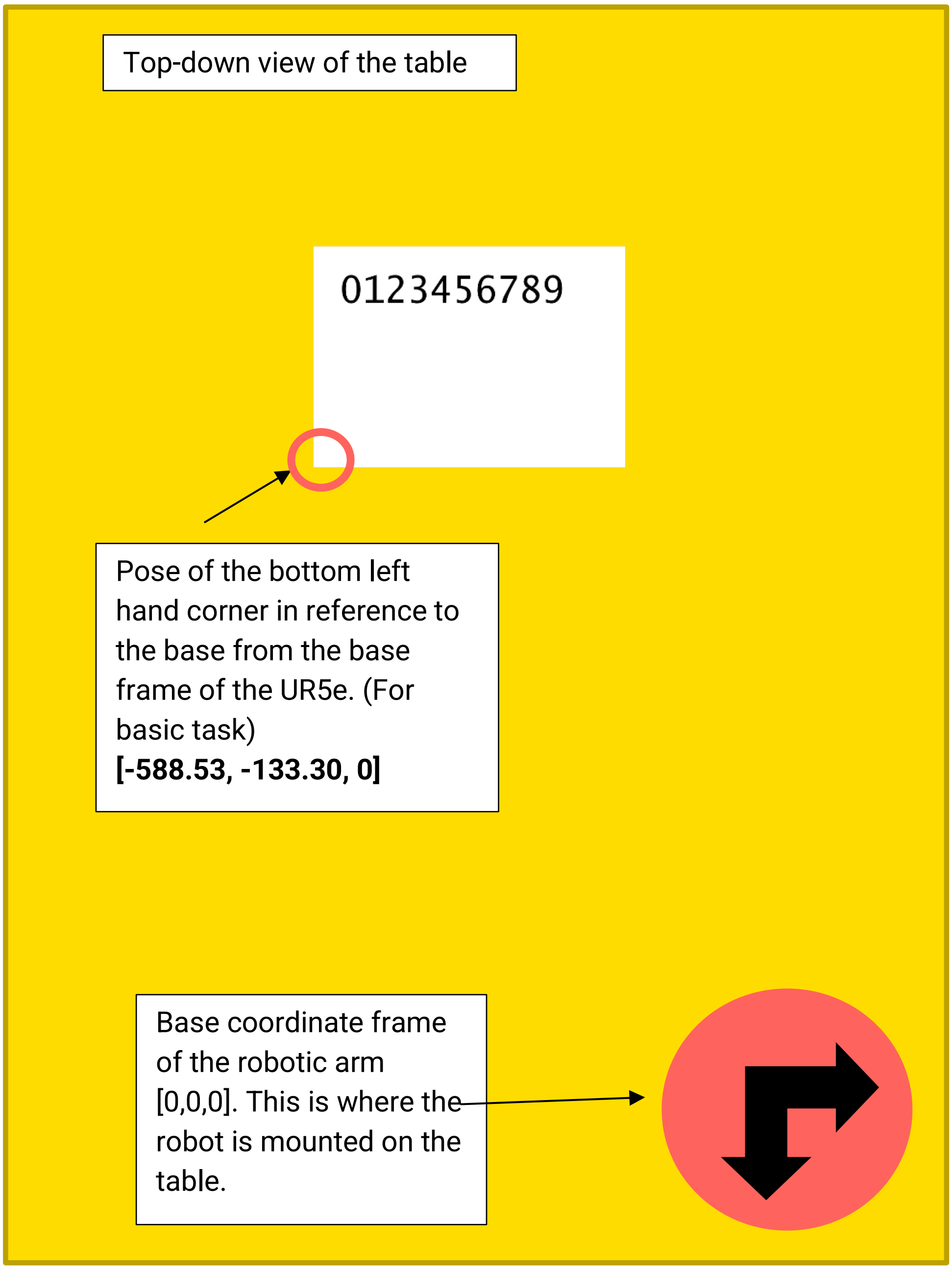
Item	Value	Description
Safety Requirements	0%* (hurdle)	Pass mark in ROBOT-2 assessment to show competency in safely using the robot
Online UR5e training	0%* (hurdle)	Show E-series Pro Track Certificate to demonstrator prior to attempting the UR5e Operation task
Part A: Trace digits (**Choose either the Basic or Advanced task**)	0 – 4 %	**Basic Task Completion**. You should be able to trace out the digits from 0 – 9 in order. Your program must start and end at the home position and it must display the required messages at the start and end of the program.
	0 – 6 %	**Advanced Task Completion**. You should be able to trace out the sequence of random 10 digits as provided to you by your demonstrator at the start of your demonstrator. Your program must start and end at the home position and it must display the required messages at the start and end of the program.
Part B: Translation and Rotation	1%	Demonstrate that you have programmed to consider an arbitrary **Translation** in your program.
	1%	Demonstrate that you have programmed to consider an arbitrary **Rotation** in your program
Part C: Mimicking a human	0 – 1%	Demonstrate that you have considered **the stroke order** of the digits in your program. This includes the lifting and dropping of the TCP while writing subsequent strokes when necessary as well as during the transition from one digit to the next.
	0 – 1%	Demonstrate that you have considered the **smoothness of motion** in your program. The robot must demonstrate a jerk free, reasonably constant velocity during the entire program.
Part D: Understanding of implementation	0 - 5%	Answer questions verbally from the demonstrator on how you have approached this task. **Please have a look at the rubric below for more detail on the marking of the responses.**

Part D Grading Rubric for responses:

Part D Grading Rubic	**Poor (1 – 2 %)**	**Insufficient (2 - 3%)**	**Developing (3– 4%)**	**Accomplished (4 – 5%)**
	(i) Provide a basic description on the chosen approach.	(i) Provide a good explanation of the approach with some details of specifics. (ii) List some of the advantages /limitations of method.	(i) Provide thorough explanation of the approach indicating that they have complete understanding of the solution. (ii) Explain and discuss some advantages and limitations of their approach. (iii) Provide some insight on how to improve the program in the future.	(i) Provide a thorough explanation of their overall approach demonstrating that they have a complete understanding of the solution. (ii) Discuss the advantages and disadvantages of their solution (iii) Based on this knowledge(ii), provide some insight on how to improve the program in the future. (iv) Demonstrate understanding of the robot's capabilities by discussing how it impacted the design of the solution or how it could pose challenges to other students' solutions.

1. Appendix

2. Top-down view of the table



Pose of the bottom left hand corner in reference to the base from the base frame of the UR5e. (For basic task)
[-588.53, -133.30, 0]

Base coordinate frame of the robotic arm [0,0,0]. This is where the robot is mounted on the table.

3. **TCP Path showing an example of TRACING the digit '1'**

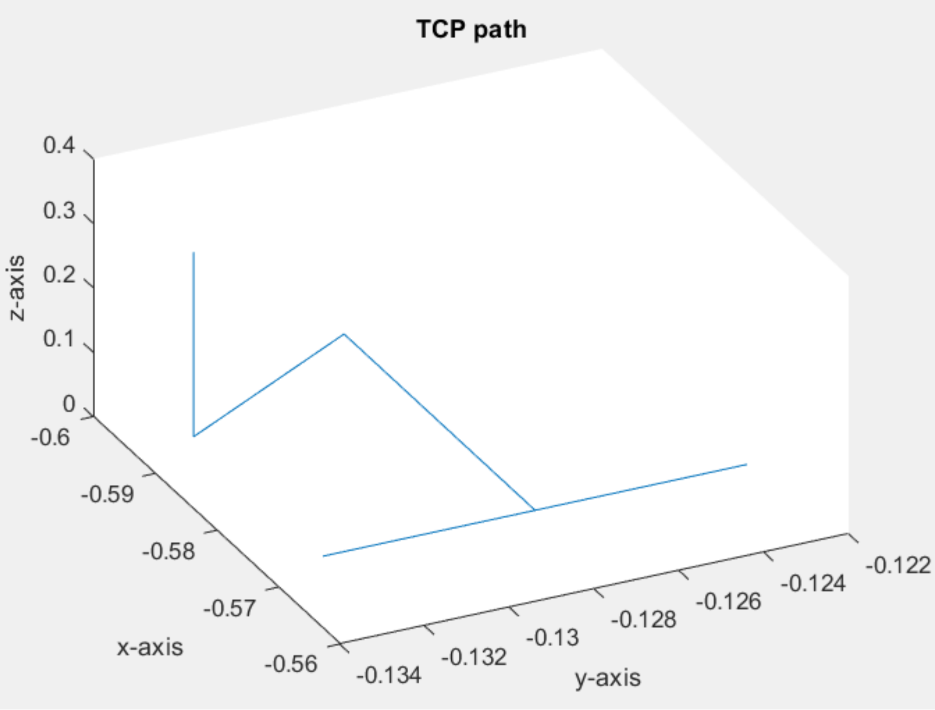


Figure 1 Angled View

4. **TCP Path showing an example of STROKES on the digit 1.**
   The left image displays the top view of the number. The right image displays a side view displaying the strokes that it took. Movec was used to generate these strokes.

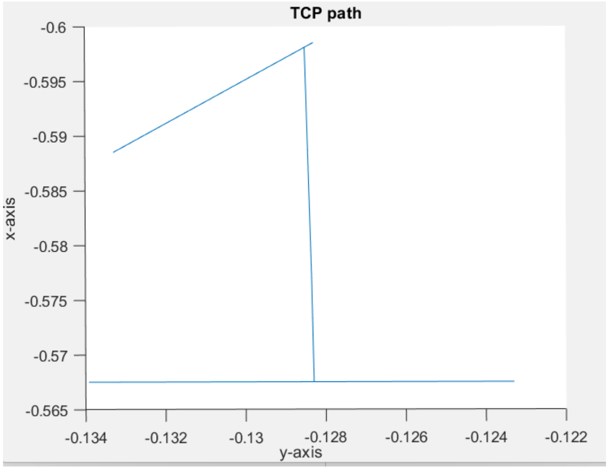


Figure 2 Top View

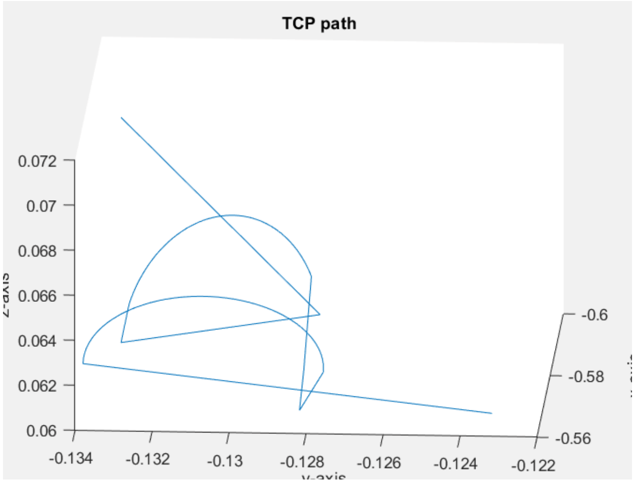


Figure 3 Side View