MTRN4230 T2 2025 ROBOT-2: Robot Programming

Changelog

- 2025-06-12: All commands need to be entered through terminal. Altered mark allocation.

Learning Outcomes

LO1: Learn a robot environment and put it to use effectively and efficiently on a given task
LO3: Implement good safety practices in the use of robots.

Aims

To gain experience in programming the UR5e robot

Due Date

- Demonstration during your scheduled **lab slot in Week 4**

Prior to Assignment

Setup

1. You must have completed the ROBOT-1 assessment prior to using the robots for this assessment.
2. Ensure you have followed the Virtual Machine Installation Instructions (refer to the lab0 document on Moodle).

Safety

1. The TCP position of the robot has been edited for you such that it is located at the tip of the attached pen.

Submission and In-Lab Demonstration

To complete this task, you will need to use the RTDE scripting library provided within MATLAB to communicate the robot and create the program for this assessment.

You will be required to submit only one Matlab file with the following naming format:

- yourZID_ROBOT_2.m (e.g., z5555555_ROBOT_2.m)

Students are allowed to use any UR5e's not being used for marking to practice their solution.

Assessment

Part A: Trace digits

Students are required to demonstrate either (1) "basic task" OR (2) "advanced task". The description of the tasks will be expanded under their relative headings. Their marks are provided in the marking criteria. The general requirements that must be met are:

- You should be writing using the Hershey font (http://paulbourke.net/dataformats/hershey/) – This is provided within the RVC toolbox, refer to the example code within the RTDE toolbox (on Github) or section 7.5.1 of the textbook for usage information.
- The font should be used with the scale set to 0.04 (this is the scale as in the RTDE example file)
- The specific font spacing is up to you, but must meet the following requirements:
    - Characters must not overlap. o Spacing should look natural (not be larger than the width of a character).
    - The robot should be able to write 10 characters within 40 cm.
- Start and end the motion at the robot home pose ([-588.53, -133.30, 227.00, 2.221, 2.221, 0]).
- The position of the first stroke should be located at the following x, y, z position: [-588.53, -350, 0] (see appendix). The orientation should be such that the TCP is facing directly downwards. This position is vertically below the robot's "HOME" position.
- Your direction of writing must be perpendicular to the table (see figure in appendix)
- You must print the digits being written on the console before starting motion.
- At the end of the program, you must indicate to the user that the program is complete (print "Program Complete" to the console).
- Your program must have a globally defined variable that defines the height of the TCP (zcoordinate) when writing letters.

For this task, an attachment will be placed on the robot that holds a marker. It will be used to write on an acrylic sheet that is placed on the bench. The TCP of the robot will be updated such that it is at the end of the pen minus the thickness of the acrylic sheet. This means that when the z-coordinate of the pose is set to 0, the pen will be touching the surface.

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

This is the basic task. Students are required to program the robot to follow a path that traces out the digits 0-9 in order (i.e., 0123456789), on a horizontal plane. This should be done such that the TCP location (the end of the pen) is touching the table surface (set the z-coordinate of the pose to 0 when writing) with the final joint facing directly down.

Imagine you are tracing the digits with a pen, so the motion of the robot should be smooth. This includes raising and lowering the end effector by ~10mm between each separate stroke/letter.

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 the robot to follow a path which traces out 10 random alphanumeric digits (0-9, a-z) in any order. You should complete **everything from the basic task**, except the order is randomized and this includes letters as well. The demonstrator will provide the random 10 digits **through terminal input**, you will not know these beforehand and must be able to handle it on the spot (your demonstrator must be able to enter these into your code easily either as a prompt or by changing one line at the start).

Part B: Translation and Rotation [0 - 3%]

Extend the basic or advanced program to trace the digits when the piece of paper is in an arbitrary pose on the table. **The translation and rotation prompts will be entered by your demonstrator through the terminal.**

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 start position of writing the first stroke of your character and the angle at which you should write it.

Hint: Have a look at this video by Peter Corke which covers how to use the RVC toolbox to create transformation matrices.

- 1% will be awarded if you can program for an arbitrary translation.
- 1% will be awarded if you can program for an arbitrary rotation.
- 1% for both translation and rotation.

Part C: Mathematical Operations [0 - 3%]

In this task, you must extend the basic or advanced task such that you are able to solve simple mathematical operations[2]. This is defined as a single operation of addition, subtraction, or multiplication in only integer forms. Examples could be (1 + 3, 5 − 3, or 2 * 5).

Your program should be able to interpret when it is given an operation[1]. They will always be inputted in the format [*integer*] [*operation*] [*integer*] =, where [*integer*] will be replaced with an integer number and [*operation*] will be replaced by one of +, −, or ∗.

When writing the answers, your answers should be written down by the robot in the form of 'long-operations' (that is long-addition, long-subtraction, and long-multiplication). An example of this is given below.

The maximum number of digits that will be required for any of these numbers (either operand or the result) will be 5.

Program Input	1 + 2 =	5 − 2 =	10 ∗ 3 =
Program Output	1 2 + 3	5 2 − 3	10 3 × 30

[1] *Your program still needs to be able to achieve arbitrary translation and rotation while performing this task.*

[2] *Your program needs to take in commands through the terminal and determine if the entered command is in a mathematical format.*

Note: the sample letters code (RTDE example 8) has a minor bug when trying to draw the "−" symbol. Make sure you solve this to complete this task and display the subtraction operator.

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-1 assessment to show competency in safely using the robot
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 **through terminal input** 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. It should also consider the smoothness of the motion of the letters and between strokes.
Part B: Translation and Rotation	0 – 1 %	Demonstrate that you have programmed to consider an arbitrary **translation** in your program. Accepted through terminal.
	0 – 1 %	Demonstrate that you have programmed to consider an arbitrary **rotation** in your program entered through terminal.
	0 – 1 %	Both **translation** and **rotation**.
Part C: Mathematical Operations	0 – 3 %	The program accepts mathematical terminal input, solves basic math expressions, displays them in correct format and performs arbitrary translation and rotation.
Part D: Understanding of implementation	0 - 3%	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:

Insufficient (1%)	**Developing (2%)**	**Accomplished (3%)**
(i) Provide a basic description on the chosen approach.	(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.

Appendix

Top-down view of table