

MTRN4230 T2 2022 Project 1: Kinematics and Trajectory Planning

Learning Outcomes

- Analyse and critically evaluate robot performance using robot mechanics
- Formulate models of robot manipulators

Aims

- To understand the kinematics of the UR5e Robotic arm by manually calculating its forward kinematics and Jacobian.
- To understand the behaviour of different types of trajectories and being able to discuss their advantages and disadvantages
- To understand the implication of singularities and how they can be mitigated.

Due Date

- Report submission on Moodle: Week 8, Friday 17:00 pm (AEST)
- Submission format: pdf

A and B are two sets of joint angle positions (theta1 to theta6) of a UR5e robot:

- **Joint Variables/Position A:** [-90, -173, 132, 220, 0, 0] (degrees)
- **Joint Variables/Position B:** [-90, -60, 90, 0, 90, 0] (degrees)

Part A: Kinematic Modelling (2.5 %)

Calculation of Forward Kinematics

You will need to post relevant screenshots or insert matrices into your report submission to show completion of each task. Correct labelling and captioning must be used.

1. Construct a model of the UR5e robot using the RVC Toolbox and the DH Table provided. (This is not assessed; it is a pre-requisite for questions 2 onwards)

Table 1 DH Parameters of the UR5e

Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1625	$\pi/2$
Joint 2	0	-0.425	0	0
Joint 3	0	-0.3922	0	0
Joint 4	0	0	0.1333	$\pi/2$
Joint 5	0	0	0.0997	$-\pi/2$
Joint 6	0	0	0.0996	0

- Using the frame transformation matrix presented in lectures 3 and 4, calculate the following frame transformation matrices for joint variables A and joint variables B and subsequently calculate the following interim results:
 - ${}^0T_1, {}^0T_2, {}^0T_3, {}^0T_4, {}^0T_5$ of joint variable A
 - ${}^0T_1, {}^0T_2, {}^0T_3, {}^0T_4, {}^0T_5$ of joint variable B
- Calculate the forward kinematic solution 0T_6 for joint variables A & B.
- Verify your solution with the “**fkine**” function of the RVC Toolbox for joint variables A & B.
- Verify your solution with either URsim or the real robot for joint variables A & B.

Part B: Calculation of Jacobian (5.5%)

- Build up on the calculations of the forward kinematics to calculate the Jacobian for the end effector at joint variables A & B.
 - Provide a brief explanation about what the Jacobian and Inverse Jacobian are ,where and why it is used in the scope of robotic arms. (~ 100 words)
 - Manual calculation of the Jacobian at joint variables A & B. Include annotated screenshot of your code giving an overview of how you have formed the Jacobian from the matrices derived from the calculation of forward kinematics.
 - Verify your solution with the “**jacob0**” (robot_name.jacob0([joint variables])) function of the RVC Toolbox for joint variables A & B.
 - Using the calculated Jacobian calculate the instantaneous joint velocities for the end effector to achieve a desired linear movement of 250mm/s in the positive x direction while passing through **Joint Variables A** and maintaining the same orientation (equivalent to move!).
 - Hint: You may need to use pinv() (pseudoinverse) for finding the inverse of the Jacobian as just using inv() may not give you the correct matrix. Why do you think this is the case? Is it possible to inverse a square matrix? Is it possible to inverse a rectangular matrix?
 - Comment on whether these values for joint velocities seem reasonable and why (~100 words).
- Calculate the end effector velocities when driving the joints at the following speeds at **Joint Variables B**: <1, 0.1, 0.1, 0.1, 0.1, 0.1> (all rad/s)
 - Comment on whether these values seem reasonable and why (~100 words)



Part C: Trajectory Generation (8%)

Poses 1,2,3 and 4 of the end effector of the UR5e robot are given below:

- **Pose 1:** [-588.53, -133.30, 100, 2.2214, -2.2214, 0.00] (mm and radians)
- **Pose 2:** [-688.53, -133.30, 100, 2.2214, -2.2214, 0.00] (mm and radians)
- **Pose 3:** [-688.53, -233.30, 100, 2.2214, -2.2214, 0.00] (mm and radians)
- **Pose 4:** [-588.53, -233.30, 100, 2.2214, -2.2214, 0.00] (mm and radians)

For the following series of questions, you are required to use URsim for questions 1, 2 and 3; and use the real UR5e robot for question 4.

1. Write a program using **"movej"** draw a trajectory with the following poses: (It should be a square) (This is not assessed; it is a pre-requisite for questions 3 onwards)
 - a. **Pose 1** (start)
 - b. **Pose 2**
 - c. **Pose 3**
 - d. **Pose 4**
 - e. **Pose 1** (end)
2. Write a program using **"movej"** draw a trajectory with the four poses above. (This is not assessed; it is a pre-requisite for questions 3 onwards)
3. Compare and contrast both types of trajectories in **URsim** discussing the following: (Generate relevant plots for each of the following to place into your report, along with your discussions. You may wish to overlay both movement types on the relevant plots.) (~ 200 - 300 words for all parts)
 - a. What observations can be made about the path (the actual square) between both move types.
 - b. What observations can be made about each individual translational axis component of the paths between both move types.
 - i. X translational Component
 - ii. Y translational Component
 - iii. Z translational Component
 - c. What observations can be made about the joint positions of all 6 joints on the UR5e for both move types?
 - d. What observations can be made about the joint velocity profile of both move types?
 - e. What observations can be made about the joint acceleration profile of both move types?
4. Now collect the same data on the **real UR5e** in the lab. You might want to save the pose, joint position, joint velocity and joint acceleration data as a **".mat"** file so that you can generate plots from the data file at home. (Come to the lab prepared with the two programs (or one) so that you can collect data as quickly as possible)
 - a. What observations can you make about the data you collected in URsim compared to the real UR5e in the lab. Explain why the differences between URsim and the real robot are present. (~ 100 words)
5. Identify what type of trajectory generation method is employed by the UR5e. (Trapezoidal or Polynomial) (~ 150 - 200 words)



- a. Based on your understanding of the trapezoidal and polynomial trajectory profiles, discuss the advantages and disadvantages of each and justify why you think the UR5e uses the one that you've observed.
6. Discuss different industrial applications which would use "**movej**" and which would use "**move**". Justify your answer based on the observations that you have made. (~ 100 - 200 words) (Hint: Think about the different industrial uses for robotic arms)

Part D: Singularities (4%)

Note: The following section is only to be done on URsim, not the real robot. You will need to disable the safety planes in your URsim installation to be able to generate the path.

For the following, you will need to use **joint position A** [-90, -173, 132, 220, 0, 0] as given above and **pose C** is [100, -127.58, 571.29, -1.571, -0.017, 1.57] (mm and radians)

1. Write a program using "**movej**" to plan a path from **Joint Position A** to **Pose C** taking 5 seconds for the entire duration. (This is not assessed; it is a pre-requisite for questions 3 onwards)
2. Write a program using "**move**" to plan a path from **Joint Position A** to **Pose C** taking 5 seconds for the entire duration. (This is not assessed; it is a pre-requisite for questions 3 onwards)
3. Is the UR5e able to make the desired motions in both cases? If not, discuss what happened. (~200 words)
 - a. Insert a screenshot of the output (robot path or error) in your report. If the motion encountered a singularity, based on your understanding of singularities explain what behaviour you would have observed by the joint positions and joint velocities if it were possible. You may wish to include a fake plot (drawn by you) to aid in your explanations.
4. Discuss the different types of singularities (wrist, elbow, shoulder) and applications where they can be problematic and how they can be mitigated. (~200 words)

Marking Criteria

Item	Value	Description
Part A.2	0.5	Correct matrices for Joint Positions A & B. (0.05 mark per correct matrix, 10 all together)
Part A.3	1	Correct forward kinematic solution for Positions A and B using a manual method (you are recommended to create a function in MATLAB)
Part A.4	0.5	Correct forward kinematic solution with rvc fkine function for Positions A and B
Part A.5	0.5	Correct pose and joint position visible on screenshot inserted on the report.
Part B.1.a	0.5	Demonstrates understanding of what the Jacobian and Inverse Jacobian are. Provides further information about why it is useful to calculate and provides 1 or 2 examples of applications it is useful in.
Part B.1.b	2	Correct matrices for the Jacobian at positions A & B. Correct Matrices for z_0 to z_5 and O_0 to O_5 shown with an explanation of how these matrices were obtained.
Part B.1.c	0.5	Correct Jacobian matrices obtained at poses A & B using rvc jacob0 function.

Part B.1.d	0.5	Correct resultant joint velocities for the specified point
Part B.1.e	1	Demonstrates understanding of robot velocity limits to accurately predict and conclude whether the derived joint velocities are correct with justification
Part B.2	1	Correct resultant cartesian velocities calculated. Provides accurate conclusion of whether the derived cartesian velocities are possible with justification.
Part C.3.a	0.5	Inclusion of a correctly labelled and titled plot with a brief explanation of what visible differences are present in the movements of movel and movej
Part C.3.b	1.5	Inclusion of a (or multiple) correctly labelled and titled plot for X, Y and Z translation data obtained from the UR5e for both movel and movej. Provides an explanation of the observations made by analysing the data and the plots retrieved. Provides reasoning as to why the observed behaviour is present.
Part C.3.c	1	Inclusion of a (or multiple) correctly labelled and titled plot for all joints for the movel and movej command. Provides an explanation of the observations made by analysing the data and the plots retrieved. Provides reasoning as to why the observed behaviour is present.
Part C.3.d	0.5	Inclusion of a (or multiple) correctly labelled and titled plot showing the velocity profile of the UR5e. Correct identification of trajectory generation method is employed.
Part C.3.e	0.5	Inclusion of a (or multiple) correctly labelled and titled plot showing the acceleration profile of the UR5e. Correct identification of trajectory generation method is employed.
Part C.4	1	Inclusion of multiple correctly labelled and titled plots showing the differences between the pose, joint position, joint velocity, and joint acceleration data retrieved from the real UR5e for both movel and movej. Explanations about what differences are visible between the real data and the ones obtained from URsim expected. Justification of why this difference is evident is expected.
Part C.5	1.5	Identification of what trajectory generation method employed by the UR5e. A discussion of the pros and cons of the trapezoidal and polynomial trajectory generation algorithms expected with a conclusion justifying UR5e's chosen trajectory generation method.
Part C.6	1.5	Discussion of the pros and cons of movel and movej and how this behaviour can be beneficial or detrimental for specific applications. Identification of some industrial applications required.
Part D.3	2	Screenshots or correctly labelled plots of output to be included in the report. Discussion of what happened during both motions and why. Prediction of what theoretical behaviour would have been witnessed required and what effects this may have had on the robot.
Part D.4	2	Discussion of what types of singularities for robotic arms and provides multiple solutions on how they can be mitigated. Provides example industrial applications where singularities can be very problematic.