MTRN4230 Lab 05

1. Aim

The aim of this lab session is to understand how to use the RVC toolbox to construct a model of the UR5e robotic arm, calculate the forward kinematic solution as well as the Jacobian. The necessary content required to complete this task was covered in Lecture 4 and Lecture 5. You will need to watch those to understand the theory before attempting this task. **This task is directly relevant to Project 1 which will be released after Quiz 1.**

2. Pre-lab

Prior to the lab, you must be prepared to demonstrate your solution to ROBOT-2.

Pre-lab checklist:

- Ensure that you have watched Lecture 4 and 5
- Ensure that you understand how to create robotic arms using the RVC Toolbox.

3. Lab Activities

The demonstrator will be marking student's solutions for ROBOT-2, so this lab task will be self-directed. It will be directly contributing to progress for Project-1.

Lab Work Tasks:

Manual Jacobian Calculation

In this section, you will go through and use matlab to manually calculate the forward kinematics and the Jacobian. You can use the same method to solve all questions presented in the lectures.

Use the following joint angle configuration for all following questions: Make sure you use radians for all angle related calculations not degrees.

q = [0, -75.00, 90.00, -105.00, -90.00, 0.00] (degrees)

The DH table for the UR5e robotic arm is presented below:

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	π/2
Joint 5	0	0	0.0997	-π/2
Joint 6	0	0	0.0996	0

Forward Kinematics

Instead of utilising the inbuild forward kinematic function 'fkine to determine the forward kinematics, create your own function.

1. Create a function which calculates the frame transformation matrix. It should take in variables theta, offset, length and twist as input and output the transformation matrix.

$$^{i-1}T_i = \begin{pmatrix} cos\theta_i & -sin\theta_i \, cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

θ – angle; d_i – offset; a_i –length; α_i – twist

Figure 1 Frame Transformation Matrix

2. Determine results for $^0T_1$, $^1T_2$, $^2T_3$, $^3T_4$, $^4T_5$, $^5T_6$, using the function, you have created.
3. Determine results for the interim transformations: $^0T_2$, $^0T_3$, $^0T_4$, $^0T_5$
4. Determine the forward kinematic solution for:

$$^0T_6$$

5. Verify your result with the inbuild **'fkine'** function.
6. Knowledge check: Do you understand what $^0T_2$, $^0T_3$, $^0T_4$, $^0T_5$represent? Try to explain it to the person next to you! Otherwise ask the demonstrator to help you understand.

Jacobian

The Jacobian allows us to map joint velocities to the end effector velocities (x,y,x,r,p,y velocities). This is very important for robotic arm control as if we want to move the end effector at a certain velocity in the "x" direction we need to know the joint velocities required for this movement.

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{\theta}\\ \dot{\phi}\\ \dot{\varphi}\end{pmatrix} = \begin{pmatrix} \frac{\delta f_1}{\delta q_1} & \frac{\delta f_1}{\delta q_2} & \cdots & \frac{\delta f_1}{\delta q_n}\\ \frac{\delta f_2}{\delta q_1} & \frac{\delta f_2}{\delta q_2} & \cdots & \frac{\delta f_2}{\delta q_n}\\ \frac{\delta f_3}{\delta q_1} & \frac{\delta f_3}{\delta q_2} & \cdots & \frac{\delta f_3}{\delta q_n}\\ \frac{\delta f_4}{\delta q_1} & \frac{\delta f_4}{\delta q_2} & \cdots & \frac{\delta f_4}{\delta q_n}\\ \frac{\delta f_5}{\delta q_1} & \frac{\delta f_5}{\delta q_2} & & \frac{\delta f_5}{\delta q_n}\\ \frac{\delta f_6}{\delta q_1} & \frac{\delta f_6}{\delta q_2} & & \frac{\delta f_6}{\delta q_n} \end{pmatrix} \begin{pmatrix}\dot{q}_1\\ \dot{q}_2\\ \cdot\\ \cdot\\ \cdot\\ \dot{q}_n\end{pmatrix}$$

- $J(\boldsymbol{q})$ is the **Jacobian** where, $J_{ij} = \frac{\delta f_i}{\delta q_j}$
- The Jacobian is a $6 \times n$ matrix

The Jacobian will have "N" columns for an "N" link robot. For the case of the 6 axis ur5e robot, the Jacobian will have 6 columns. Each row in the Jacobian matrix represents the effect of each joint on a spatial velocity. That means the first row of the Jacobian represents the effect in the "x" direction.

$$ {}^{0}T_i = \begin{pmatrix} {}^{0}R_i & {}^{0}\mathbf{o}_i \\ 0 & 1 \end{pmatrix} = \begin{array}{c} \begin{array}{cccc} \phantom{r_{1,1}} & \phantom{r_{1,2}} & \mathbf{z}_i & \mathbf{o}_i \end{array} \\ \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & o_{1,4} \\ r_{2,1} & r_{2,2} & r_{2,3} & o_{2,4} \\ r_{3,1} & r_{3,2} & r_{3,3} & o_{3,4} \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} $$

1. Knowing that for the calculation of the Jacobian you can determine vectors for Z_i and O_i from the frame transformation matrix (Shown in the image above). Using the ${}^{0}T_1$, ${}^{0}T_2$, ${}^{0}T_3$, ${}^{0}T_4$, ${}^{0}T_5$, ${}^{0}T_6$ you calculated in the forward kinematics section determine $Z_0, O_0, Z_1, O_1, Z_2, O_2, Z_3, O_3, Z_4, O_4, Z_5, O_5, Z_6, O_6$. (Hint: You can determine Z_0, O_0 from ${}^{0}T_0$).
2. Utilising the following formula, calculate $J_1, J_2, J_3, J_4, J_5, J_6$. (Hint: In our question, all joints are revolute)

$$J_i = \begin{pmatrix} \mathbf{z}_{i-1} \times (\mathbf{o}_n - \mathbf{o}_{i-1}) \\ \mathbf{z}_{i-1} \end{pmatrix} \text{ if joint } i \text{ is revolute;}$$

Or

$$J_i = \begin{pmatrix} \mathbf{z}_{i-1} \\ 0 \end{pmatrix} \text{ if joint } i \text{ is prismatic.}$$

3. Calculate the overall Jacobian knowing that, $J = [J_1 \quad J_2 \quad J_3 \quad J_4 \quad J_5 \quad J_6 \;]$
4. Verify your result with the 'jacob0' function
5. What does it mean when the determinant of the Jacobian is equal to 0?

Now you have the knowledge to use the RVC Toolbox to calculate the forward/inverse Kinematics and Jacobian. Attempt some of the other questions presented in the lectures.

4. Post Lab

- Make sure you revise content from lectures 1-4 for the quiz this Friday.
- For further resources:
  - Velocity Kinematics in 2D: https://robotacademy.net.au/masterclass/velocity-kinematics-in-2d/
  - Velocity Kinematics in 3D: https://robotacademy.net.au/masterclass/velocity-kinematics-in-3d/
  - RVC Textbook refer to relevant sections in Chapter 8