EXPERIMENT 2: MANIPULATOR INVERSE KINEMATICS

The purpose of this experiment is to investigate the Inverse Position Kinematics (IPK) of 4-DOF serial link robots. The following topics will be studied in this experiment.

Topics Covered

- The concept of inverse position kinematics (IPK)
- Geometric solution to IPK
- Existence of solution and multiple solutions to IPK

Prerequisites

- The robot has been setup and tested. See the Quick Start Guide for details.
- You have access to the User Manual.
- You are familiar with the basics of MATLAB® and SIMULINK®.

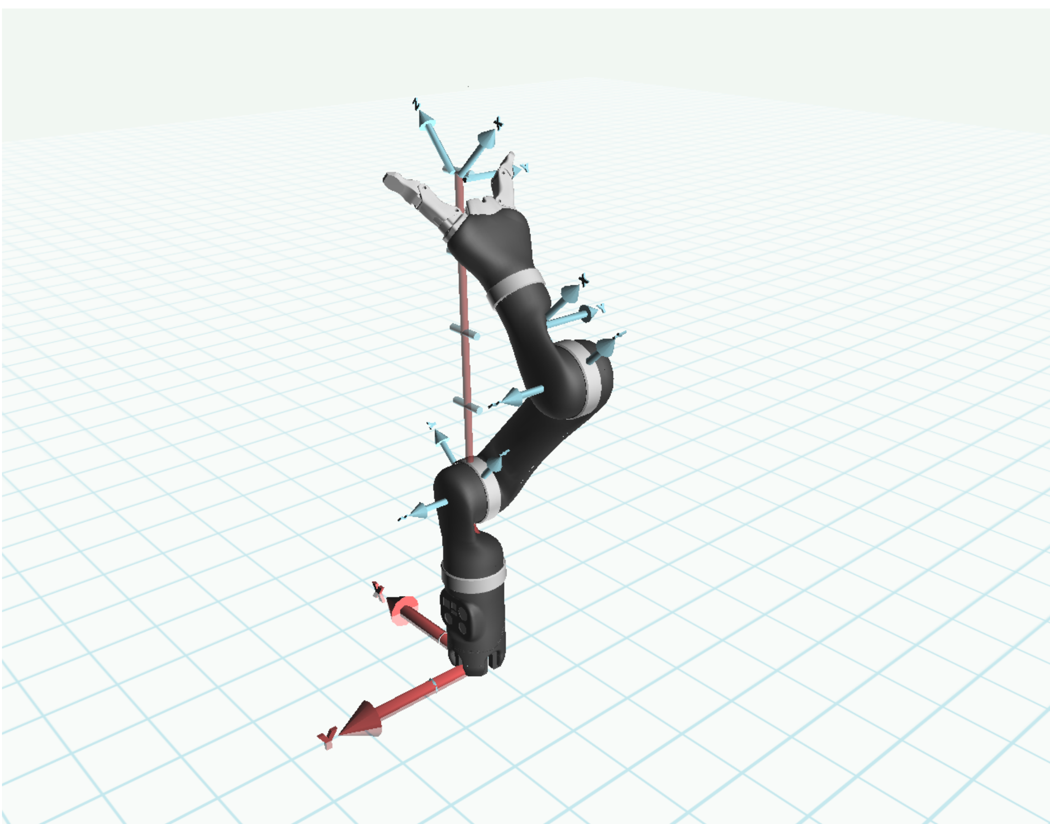

Figure 0.1: Inverse Kinematics is used to determine the joint angles of the robot given end-effector position.

1 Pre-Lab Preparations

1.1 Background

In most robotic applications, the robot end-effector needs to be able to reach a specific known position and/or orientation in space to achieve the desired task. Inverse kinematics solves the problem of finding the required joint angles to place the end-effector frame of the robot in a specific pose with respect to the base frame.

1.1.1 Solvability

In general, the problem of solving the kinematics equation of a manipulator arm is non-linear and non-trivial. The end-effector frame of a robot can be described by its transformation matrix, which is a 4$\times$ 4 matrix. Four of the elements are trivial (last row of the transformation matrix is [0 0 0 1]), and the remaining 12 elements of the transformation matrix are split into a 3$\times$1 vector position vector and a 3$\times$3 orientation matrix. However, the 9 parameters of the rotation matrix are not independent. The orientation of a frame with respect to another, can be described by 3 independent parameters called roll (rotation about x axis), pitch (rotation about y axis) and yaw (rotation about z axis).

1.1.2 Existence of Solutions

To achieve a desired pose in space, (x, y, z, roll, pitch and yaw parameters), six joint variables are necessary. Therefore, with a 4-DOF manipulator, the end-effector can not reach all positions in space with a desired orientation. As such, the 4DOF robot can be said to have no *dexterous workspace*, or volume of space where the robot end-effector can reach in any arbitrary orientation. The volume of space that the robot end-effector can reach with at least one orientation is called the *reachable workspace*.

For the 4-DOF MICO arm, since the first joint can freely move 360 $^\circ$, the reachable workspace is when $(x^2 + y^2 + (z - (L_0 + L_1))^2 \leq L2 + L3 + L4$. Note that $x^2 + y^2 + (z - (L_0 + L_1))^2$ represents the distance of the origin of the end-effector from frame J_2. If this distance is greater than the sum of links 2,3 and 4 lengths, the robot cannot reach that point. Therefore, for any point in space that satisfies the above inequality, we will have a solution, provided that the resulting joint angles fall within the physical limitations of all joints. For the 4-DOF MICO arm, joints 2 and 3 are limited to -220° $< \theta_2 <$ 40° and -230° $< \theta_3 <$ 50°.

In this experiment, we solve for the possible joint positions for various positions of the robot end-effector, and study the possible orientations. The desired end-effector position for the robot has three elements: x, y, and z. Looking at the schematic of the robot in Figure 1.1 (side view), it is obvious that variations in the last joint of the arm (J_4) won't affect the position of the end-effector; the last joint can only change the orientation of the end-effector. As a result, we solve for the first three joint angles (θ_1, θ_2 and θ_3) given x, y and z.

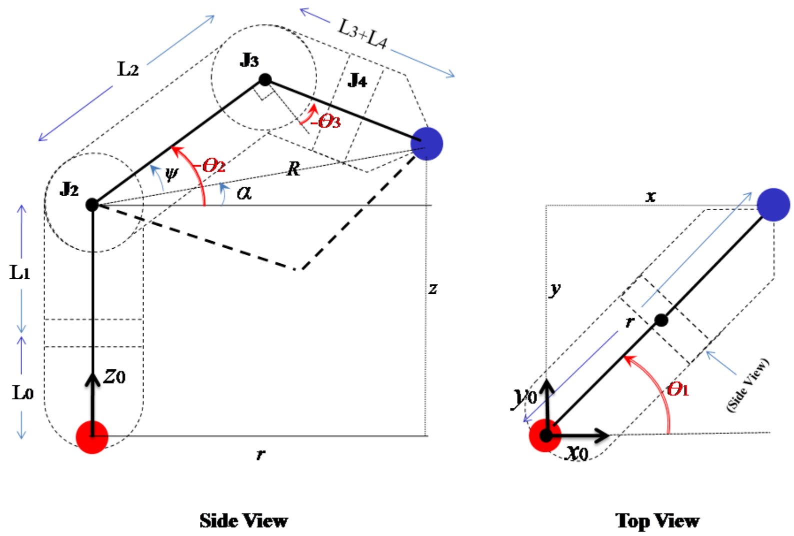

Figure 1.1: 2D top view, and side view schematics of the 4-DOF MICO robot arm.

The solutions to the inverse kinematics calculations can be either *numeric* or *closed-form*. Numeric solutions specify values, whereas closed-form solutions are expressions that are found using *algebraic* equations or *geometric* calculations. Here, we provide you with the closed-from solution using a geometric approach.

1.1.3 Geometric Solution

The geometric approach to inverse kinematics is an intuitive and usually simpler approach to solve for unknown joint angles given an end-effector position. In this method we try to decompose the spatial geometry of the arm into simple planar-geometry problems.

The motion of the 4-DOF MICO robotic arm can be decomposed in to a moving plane that rotates around the z axis when the first joint (J_1) changes. Figure 1.1 shows two different planar views of the robot arm. The left image shows the side view of the moving plane (perpendicular to the plane), and he right shows the top view (from the first joint axis looking down).

Looking at Figure 1.1 (side view), we can geometrically derive r and z as follows

$$r = (L_2\cos(\theta_2) - (L_3 + L_4)\sin(\theta_2 + \theta_3)) \tag{1.1}$$
$$z = L_0 + L_1 - L_2\sin(\theta_2) - (L_3 + L_4)\cos(\theta_2 + \theta_3) \tag{1.2}$$

The top view will result in x and y as follows

$$x = r\cos(\theta_1) \tag{1.3}$$
$$y = r\sin(\theta_1) \tag{1.4}$$

where r is shown in Equation 1.1. Therefore,

$$\theta_1 = \text{atan2}(y, x) \tag{1.5}$$

Also, from Figure 1.1 (side view) we can see that

$$R = \sqrt{r^2 + (z - (L_0 + L_1))^2}. \tag{1.6}$$

To calculate the joint angles θ_2 and θ_3, we need to first calculate α and ψ. The angle α can be calculated using the following equation:

$$\alpha = \text{arctan}(z - (L_0 + L_1), r) = \text{arctan}\frac{z - (L_0 + L_1)}{\sqrt{x^2 + y^2}} \tag{1.7}$$

where arctan is calculated using the atan2 command. To geometrically calculate ψ, given x, y and z, we will use the *law of cosines*. The law of cosines, also known as the cosine formula or cosine rule, describes the relationship between the lengths of the sides of a triangle and the cosine of one of the angles as shown in Figure 1.2 where $c^2 = a^2 + b^2 - 2ab\,\text{cos}(\gamma)$.

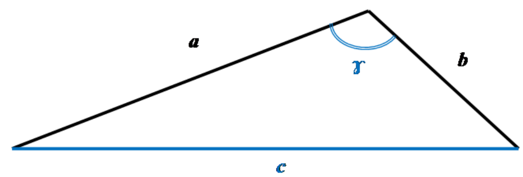

Figure 1.2: Law of cosines.

Applying the law of cosines to the triangle represented by R, L_2 and $L_3 + L_4$ and the angle ψ we have

$$(L_3 + L_4)^2 = R^2 + L_2^2 - 2L_2R\,\text{cos}\,\psi \Rightarrow \psi = \text{arccos}\left(\frac{R^2 + L_2^2 - (L_3 + L_4)^2}{2L_2R}\right) \tag{1.8}$$

where R is defined in Equation 1.6.

Also, applying the law of cosines to the same triangle with angle $\pi/2 - \theta_3$ we can derive θ_3 as follows

$$\theta_3 = \text{arcsin}\left(\frac{L_2^2 + (L_3 + L_4)^2 - (x^2 + y^2 + (z - (L_0 + L_1))^2)}{2L_2(L_3 + L_4)}\right) \tag{1.9}$$

Now, looking at Figure 1.1, we can achieve the following two sets of solutions for the joint angles

Solution set 1:

$$\begin{aligned} \theta_1 &= \text{arctan}({}^y/_x) \\ \theta_2 &= -(\alpha + \psi) \\ \theta_3 &= \text{arcsin}\left(\frac{L_2^2 + (L_3 + L_4)^2 - (x^2 + y^2 + (z - (L_0 + L_1))^2)}{2L_2(L_3 + L_4)}\right) \end{aligned} \tag{1.10}$$

Solution set 2:

$$\begin{aligned} \theta_1 &= \text{arctan}({}^y/_x) \\ \theta_2 &= -(\alpha - \psi) \\ \theta_3 &= -\pi - \text{arcsin}\left(\frac{L_2^2 + (L_3 + L_4)^2 - (x^2 + y^2 + (z - (L_0 + L_1))^2)}{2L_2(L_3 + L_4)}\right) \end{aligned} \tag{1.11}$$

1.2 Pre-Lab Exercise

1. Discuss the two solution sets in Equation 1.10 and Equation 1.11. How should one chose the right solution?
2. Solve for the inverse kinematics of the robot where $x = 0$, $y = 0$ and $z = 0.7$ m

   Note: $L_0 + L_1 = 0.2755$, $L_2 = 0.29$, $L3 = 0.1233$ and $L_4 = 0.16$
3. Draw the schematic of the robot in the configuration from Question 2, and show the configurations related to the solutions. Are there other solutions for this position?
4. Does the $x = 0$, $y = 0$ and z change if you rotate the first joint?

2 In-Lab Exercise

2.1 Simulation

The QUARC model for this exercise is "MICO_Inverse_Kinematics_Simulation.mdl" a snapshot of which shown in Figure 2.1.

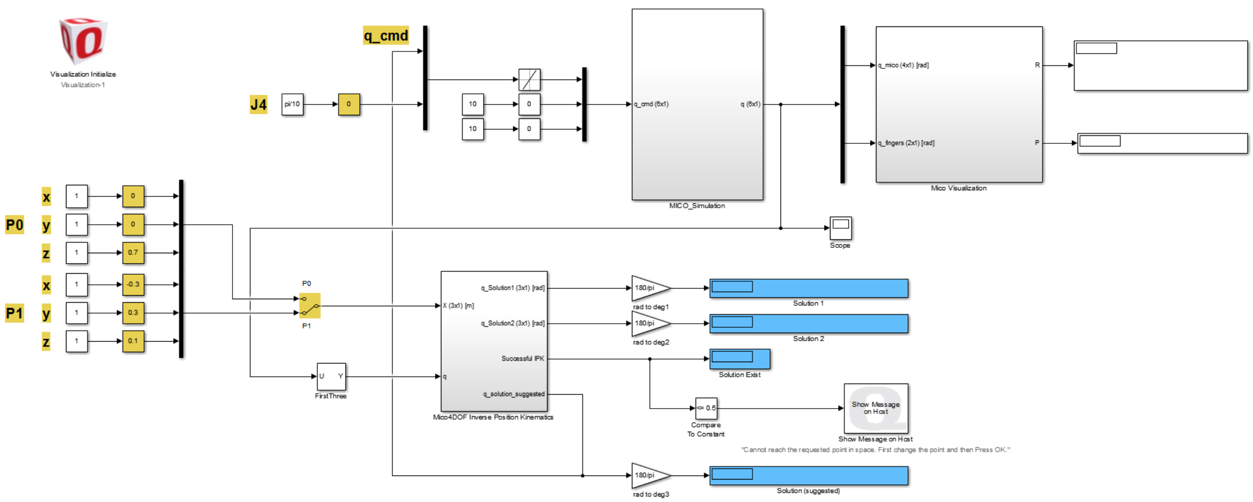

Figure 2.1: Snapshot of the controller model "MICO_Inverse_Kinematics_Simulation.mdl"

Compile and run the model. A the Quanser 3D Viewer window should open, showing a visualization of the robot. Go through the following steps and answer the corresponding questions.

1. Set the x, y, and z values of the **P0** input command to 0, 0, and 0.7 m respectively (Use the slider gains highlighted in yellow). Observe the outputs "Solution 1" and "Solution 2" and verify your Pre-lab calculations. Compare the pose of the robot with the schematic from Question 3 in the Pre-Lab exercise.
2. How is the y axis of the end-effector frame oriented with respect to the global (or base frame)? Hold down the middle mouse button, and move the mouse to rotate the camera view of the 3D Viewer. Describe the orientation of the robot in relation to the rotation matrix?
3. Change the last joint angle, **J4**, so that the x axis aligns with the global y axis. What is J4's joint angle?
4. Assume you want to program the robot to pick up an object, the centre of which located at $x = -0.3$ m, $y = 0.3$ m, and $z = 0.1$ m. Using the simulation model, derive the required joint angles. Switch the input position command to **P1**, and set the x, y and z values to verify that the robot end-effector safely goes from one point to the other (The end-effector of the virtual robot does not hit the ground).
5. What is the path of the robot end-effector when you switch between **P1** and **P0**? How can one control this path?

Once you feel comfortable with the above exercise on the virtual robot, you are ready to proceed to the next section.

2.2 Experiment

The QUARC model for this exercise is "MICO_Inverse_Kinematics_Experiment.mdl" the snapshot of which shown in Figure 2.2.

Before running the model, manually move the robot arm with the power turned off into a comfortable pose where the robot end-effector is away from the table (preferably a elbow-up pose), and turn on the robot to hold its pose. Make

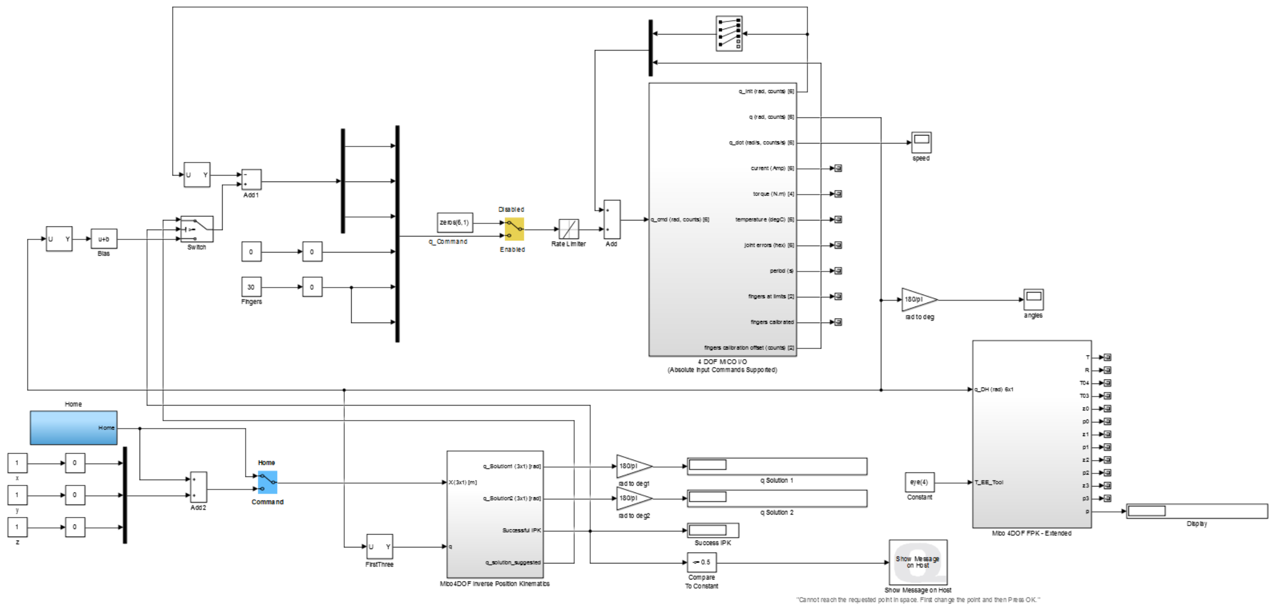

Figure 2.2: Snapshot of the controller model "MICO_Inverse_Kinematics_Experiment.mdl"

sure that the robot is disabled (the yellow manual switch is set to "Disabled"), and the blue manual switch is set to "Home".

Caution

Be sure to set the Ports in the 4-DOF MICO I/O block to the correct ports for your serial card. For more information, refer to the User Manual.

1. Set x, y, and z values of the **P0** input command to 0, 0, and 0.7 m respectively (Use the slider gains highlighted in yellow). Observe the outputs "Solution 1" and "Solution 2" and verify the values by comparing them to the ones you observed from the previous section. Compare the pose of the robot with the pose of the virtual robot you observed in the previous section. Enable the robot motion and observe the robot moving to **P0**.
2. What is the actual position of the robot end-effector (the output P from the forward Kinematics block highlighted in blue). What is the error between the commanded position and the output position? What do you think is the source of error?
3. Change the **P1** components to $x = -0.3$, $y = 0.3$ and $z = 0.1$ and switch the input position command to **P1**. Observe robot motion and compare it to the motion you observed on the virtual robot.
4. While the position command **P1** is selected, change **P0** to **P0** = [0.5 0 0.7] and then switch the input command to **P0**. Can the robot reach this point? Why?

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com