# Lab Procedure for Simulink

# Forward Kinematics

## Setup

1. It is recommended that you review Lab 1 - Application Guide before starting this lab.
2. It is recommended that 2 students run this lab together.
3. Hardware Preparation:
   a. Ensure that the QArm Mini is securely attached to the base.
   b. Verify that the manipulator is in the rest position.
   c. Confirm that the QArm Mini is connected to the PC and turn it ON (the light in the switch should be red).
   d. Check and update the latency setting as shown in Figure 1:
      i. Navigate to Device Manager > Ports
      ii. Select the appropriate device - USB Serial Port (COMx) Make a note of the COM port Number.
      iii. Go to Port Settings > Advanced > Latency
      iv. Set the latency to 2 ms

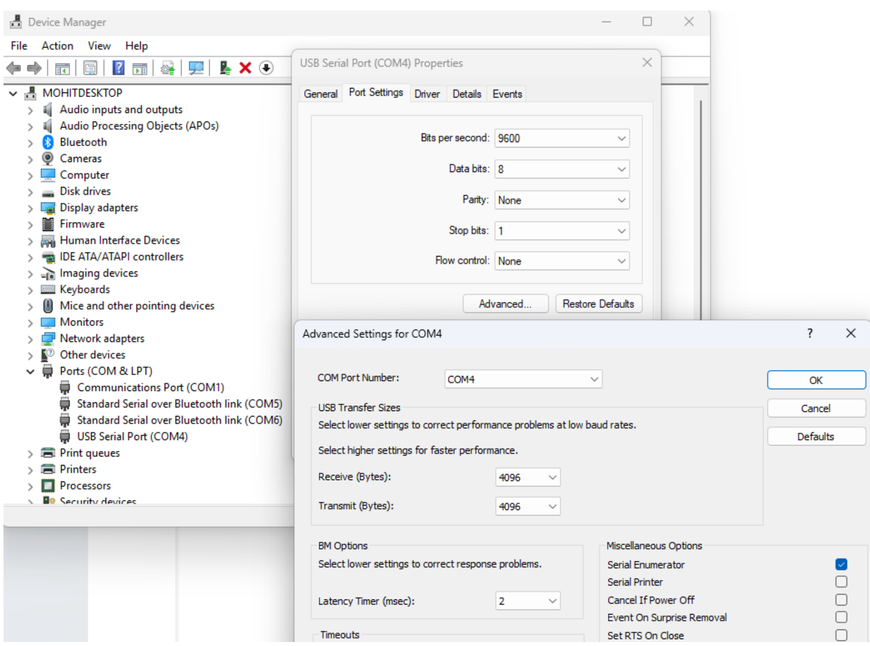


Figure 1. Latency Settings

## Workspace Identification

1. Open workspace_identification.slx

   a. In the Hardware tab, open **Hardware Settings -> Solver** and verify:

      i. Solver Type: Fixed

      ii. Solver: ODE2

      iii. Fixed Step Size: 1/30

   b. From the root level's **Application Layer**, double-click QArm Mini subsystem to navigate to the **Interface Layer**, and then the **Hardware Layer**, and double-click on the **HIL Initialize** block.

   c. Update the **Board Identifier** value to match the COM port you noted during setup.

2. Ensure the space around the QArm Mini is clear of any objects. The manipulator will lift itself to home positions as soon as the model starts.

3. Go back to the root level of your model (**Application Layer**). Build and deploy the model using the **Monitor and Tune** button on the Hardware or QUARC tab.

4. The manipulator starts in the home position. You can control each joint individually by adjusting the Joint Selector constant and using the up and down arrow keys on your keyboard.

   1 -> Base Joint

   2 -> Shoulder Joint

   3 -> Elbow Joint

   4 -> Wrist Joint

   **Note:** You can press **spacebar** at anytime to return the arm to the home position

5. Explore the manipular's limits by moving it to various positions using the keyboard. Observe its range of motion.

6. Adjust the Slider Gain block to control the gripper. You can change the values between 0-to-1, controlling when its fully open, closed or partially open/closed depending on the application.

7. Open the QArm Mini subsystem and to view the **Interface Layer.** Notice that the measured joint angles (theta values) are fed into the qArmMiniForwardKinematics function. The computed end-effector positions (**ee_pos**) are saved to the MATLAB workspace.

8. Try to capture all the points in the reachable workspace of the manipulator. Once you have recorded enough positions going through all the joint limits, stop the model.

9. Upon stopping, a graph will be generated, mapping all the end-effector positions reached during the motion of the arm. Does this graph represent the overall workspace of the manipulator well?

   Note: You can use the built-in plot tools in the graph to rotate the 3D render as shown in Figure 2.

10. Repeat steps 3 through 8 if you are not satisfied with the workspace graph generated. When complete, take numerous screenshots of the workspace in the xy, xz and yz planes, as well as an isometric view.

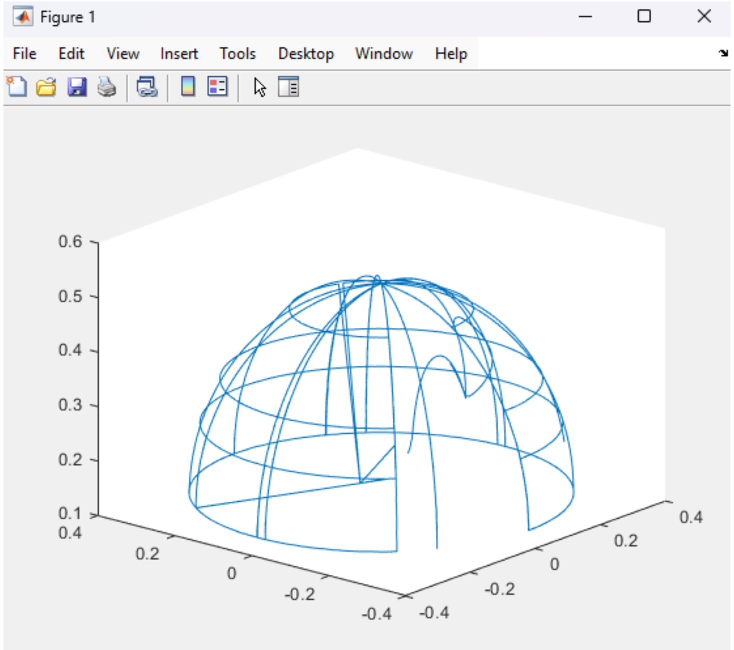


Figure 2. End Effector Positions Mapped

11. Close the graph and the Simulink model.

12. Proceed to the Lead Through section or turn off the arm and gently move it to its resting position (refer User Manual) if not proceeding immediately.

## Lead Through (Learn)

1. Open lead_learn.slx. This model simulates a welding application on a horizontal plane to represent the lead-through learning process.

   a. In the Hardware tab, open **Hardware Settings -> Solver** and verify:

      i. Solver Type: Fixed

      ii. Solver: ODE2

      iii. Fixed Step Size: 1/30

b. Navigate to the **HIL Initialize** block like step 1b in Workspace Identification.

c. Update the **Board identifier** to match the correct COM port you noted during setup.

2. Ensure the space around the QArm Mini is clear of any objects and verify that the manipulator is in the rest position (refer User Manual).

3. Grab a piece of paper and draw a simple closed-loop trajectory. Consider the QArm Mini's reach and joint limits while drawing. Mark a point on your trajectory as the start/end.

4. Place the paper at the base of the QArm Mini as shown below in Figure 3.

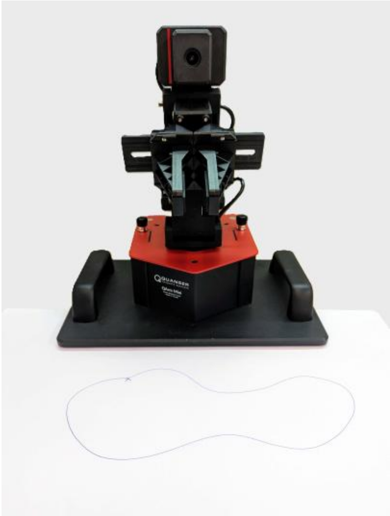

Figure 3. Closed Loop Trajectory

5. Go back to the root level of your model (**Application Layer**). Build and deploy the model using the **Monitor and Tune** button.

   **Note:** The arm operates in **PWM mode**, and as such, you should feel resistance when moving it.

6. **Recording the Motion:**

   a. Move the manipulator gently by hand such that its end-effector is approximately 1 cm above the start/end position of your trajectory.

   b. When ready, toggle the **Record (ON/OFF)** switch in your application to 1, to start recording the joint locations.

   c. Move your arm's end-effector along the trajectory towards the end. **Note**: Make sure that the end-effector does not touch the paper. Simply use the trajectory in the paper as a guide.

   d. When you reach the start/end position again, stop recording by toggling the **Record (ON/OFF)** switch to 0.

    e. This mimics one full cycle of a welding process along your custom trajectory.

7. Perform the experiment again, guiding the arm along a closed-loop trajectory near the base. Then, repeat the process with a closed-loop trajectory further from the base, near the joint limits. Compare the "EE position tracking Scope" data from both runs and take screenshots of the scope.

8. Move the arm back to its resting position.

9. Stop the model to complete the learning phase.

10. Once the model stops, a graph will be generated showing all the end-effector positions recorded during the arm's movement. Does the resulting graph accurately depict the simple closed-loop trajectory that was intended?

## Lead Through (Follow)

1. Open lead_follow.slx. This model reads the joint trajectories recorded in the Lead Through (Learn) phase and uses it to recreate the welding process automatically.

    a. Open **Model Settings / Configuration Parameters** and verify:

        i. Solver Type: Fixed

        ii. Solver: ODE2

        iii. Fixed Step Size: 1/30

    b. Find the **HIL Initialize** block in the Hardware Layer similar to step 1b in the first section.

    c. Update the **Board Identifier** to match the correct COM port you noted during setup.

2. Ensure the workspace is clear of any objects and verify that the manipulator is in the rest position.

3. Go back to the root level of your model (**Application Layer**). Build and deploy the model using the **Monitor and Tune**  button.

4. The arm will move to the start point of the recorded trajectory. The arm will follow the trajectory to the endpoint and continue repeating it.

5. The **End-Effector (EE) Position Tracking Scope** and **Joint Position Tracking Scope** have two inputs each:

    a. **Raw command trajectories:** These are the trajectories recorded at the joint level and end-effector task level during the learn phase.

    b. **Measured trajectories:** These are the trajectories measured from the QArm Mini at the joint and end-effector levels during the follow phase.

This allows for an apple-to-apple comparison between the recorded movement and the actual execution.

6. Open the **Joint Position** and **End Effector Position Tracking Scopes**. Observe how the arm follows the trajectory. How well does the arm track the desired position, both at the joint and end-effector level? Are there variances between when the end-effector is closer or further away from the base? Take screenshots of the arm's performance in the scopes.
7. Stop the model and turn off the arm.
8. Gently move the arm back to its resting position.