Lab Procedure for Python
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

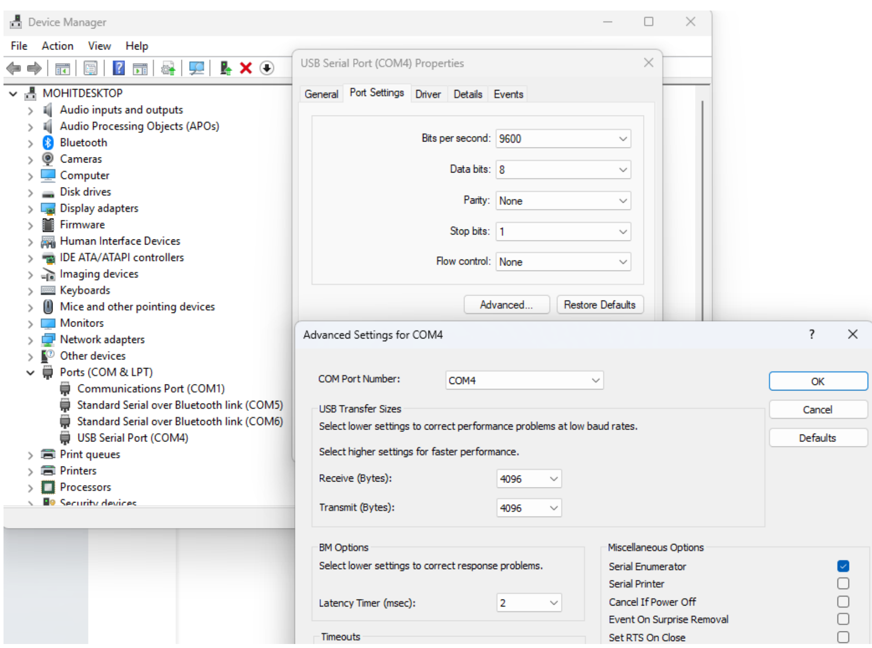

Figure 1. Latency Settings

Workspace Identification

1. Launch **Visual Studio Code** and browse to the lab directory via the **File > Open Folder** menu. Once in the correct directory, open workspace_identification.py.

    a. Update the **id** parameter in line 18 to match the COM port you noted during setup.

2. Ensure the space around the QArm Mini is clear of any objects.

3. Run the script using the ▷ button in the top-right corner.

4. The manipulator starts in the home position. Note the following actions you can take.

    a. Tap any of the 1 through 4 NUMBER keys on your keyboard once (not the NUMPAD keys). This selects the corresponding joint on the QArm Mini.

    b. Press and hold the UP or DOWN ARROW keys to increase or decrease the selected joint's position respectively.

    c. At any point, holding the SPACE BAR will record the end-effector's position for plotting and data logging.

    Note: As the script only records points when the SPACE BAR is held, it is recommended that you try to maintain continuity in action. As such, always ensure the SPACE BAR is held before using any of the ARROW keys, and you can let the SPACE BAR go once the manipulator stops moving.

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

    Note: To reach the limits of some joints, you may have to maneuver other joints out of the way first.

6. Starting at any position you find comfortable, hold the SPACE BAR, and move the arm joint after joint to cover the outer-most reaches of the arm's workspace, for example, as seen in the image below.

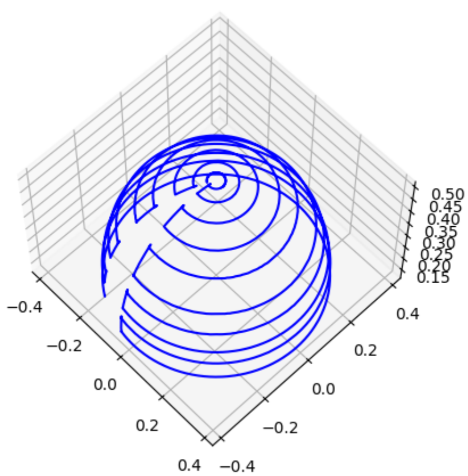

Figure 2. End Effector Positions Mapped

7. When you are satisfied with the points collected, let the SPACE BAR go, and tap CTRL-C until the script outputs **Received user terminate command**.

   The collected data is plotted for you in a 3D plot. Click and drag your mouse cursor to rotate the plot. Use the save button to save various views as image files. Make sure you record images representing the isometric view, XY planar view (from the top) as well as XZ planar view (from the left of the arm).

   Close the 3D plot to complete the script.

8. Repeat steps 6 and 7, but this time, limiting the motion of the arm in the vertical sagittal plane (joint 0 does not need to be moved at all). Ensure to collect all required views.

9. 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. Launch **Visual Studio Code,** navigate the correct directory like step 1 in Workspace Identification. Once in the correct directory, open lead_through_learn.py. This script simulates a welding application on a horizontal plane to represent the lead-through learning process.

   a. Update the **id** parameter in line 23 to match the correct COM port you noted during setup.

2. Ensure the workspace is clear of any objects. 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. Run the script using the button in the top-right corner.

   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, hold SPACE BAR to record the end effector positions and joint angles for plotting and data logging.

   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 letting go of the SPACE BAR key.

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

   Note: As the script only records points when the SPACE BAR is held, it is recommended that you try to maintain continuity in action. As such, always ensure the SPACE BAR is held before moving the arm along the custom trajectory, and you can let the SPACE BAR go once the manipulator stops moving. The data points are saved in file named 'end_effector_data.csv' in the opened directory.

7. When you are satisfied with the points collected, tap CTRL-C until the script outputs **Received user terminate command**. The collected data is plotted for you in a 3D plot. Click and drag your mouse cursor to rotate the plot. Does the resulting graph accurately depict the simple closed-loop trajectory that was intended?

8. Close the 3D plot to complete the script.

9. 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 3D plot data from both runs and take screenshots of the scope.

10. Move the arm back to its resting position.

Lead Through (Follow)

1. Launch **Visual Studio Code** navigate the correct directory like step 1 in Workspace Identification. Once in the correct directory, open lead_through_follow.py. This script reads the joint trajectories recorded in the Lead Through (Learn) phase from the 'end_effector-data.csv' and uses it to recreate the welding process automatically.

    a. Update the **id** parameter in line 24 to match the correct COM port you noted during setup.

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

3. Run the script using the button in the top-right corner.

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. What do you observe? Does the manipulator follow the custom trajectory?

6. Stop the script, tap CTRL-C until the script outputs **Received user terminate command**. The 3D plot produced allows for an apple-to-apple comparison between the recorded movement and the actual execution. Click and drag your mouse cursor to rotate the plot.

    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?

7. Take screenshots of both the plots that showcases the manipulators performance.

8. Close the 3D plot to complete the script.

9. Turn off the arm, gently move it back to its resting position.