QArm Lab Procedure
Lead Through

Setup

1. Launch Quanser Interactive Labs and load the QArm workspace
2. Launch MATLAB and browse to the working directory for Lab 3 – Lead Through

Lead Through (Learn)

1. Open LeadThrough_Learn.slx. This model uses a welding operation to represent a lead through a learning process. You will first complete the mapping of the keyboard I/O to control the movement of the robotic arm. Then you need to move the arm through a pre-set operation path and store all the joint angle data.

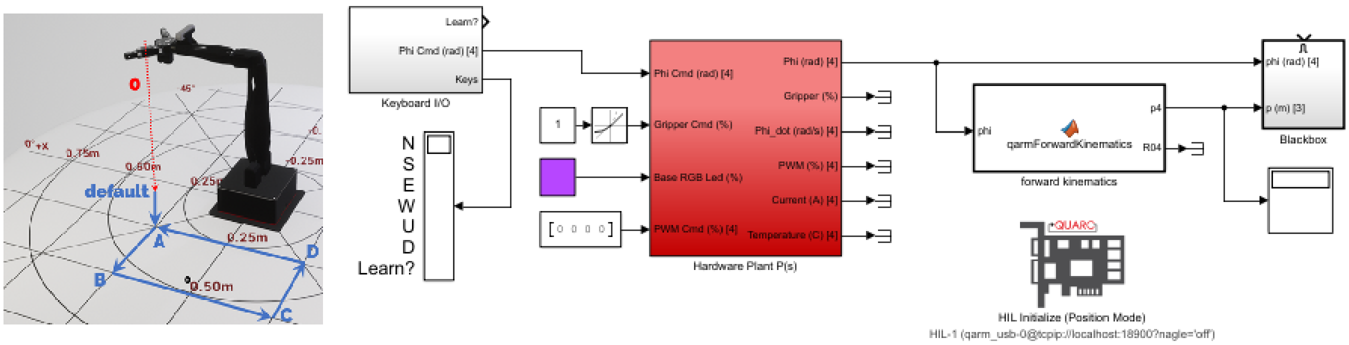

Figure 1: LeadThrough_Learn Simulink model.

2. Open the subsystem called *Keyboard I/O*. This subsystem, shown in Figure 2 contains an incomplete section called Map keys to end-effector speed. You will need to connect the keyboard inputs to the corresponding command inputs to the QArm.

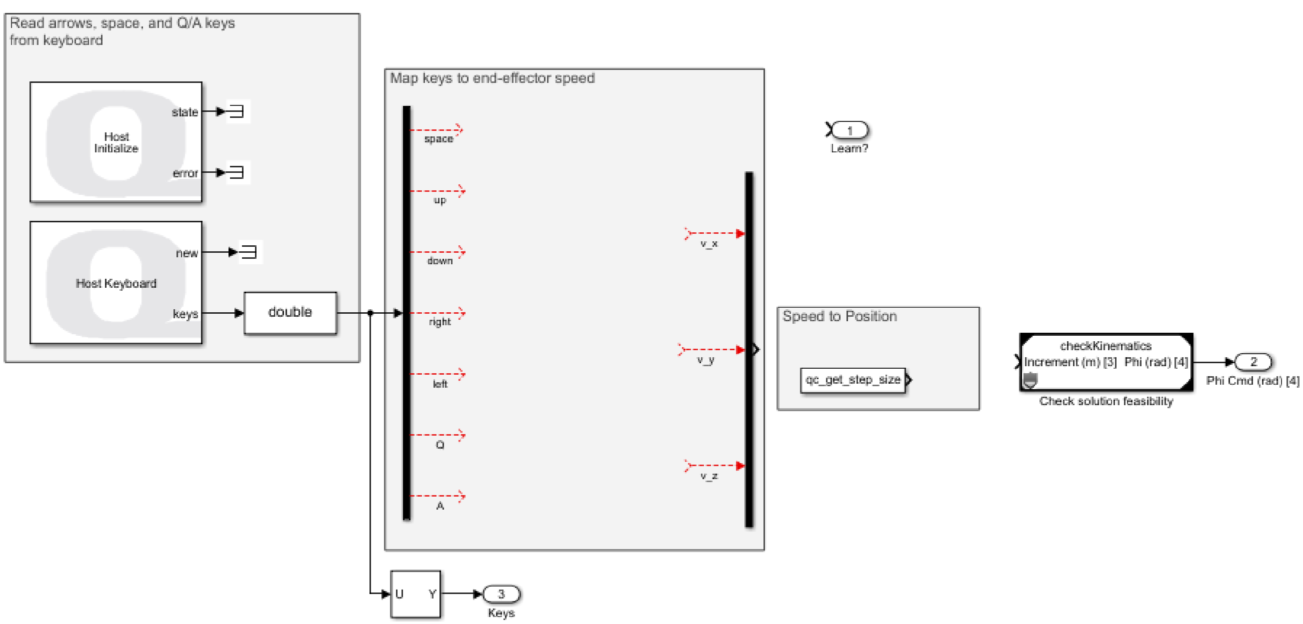

Figure 2: Incomplete Forward Kinematics subsystem.

3. Double click on the *Host Keyboard* block and take note of the keys that the block outputs, and in what specific order. Close the block.

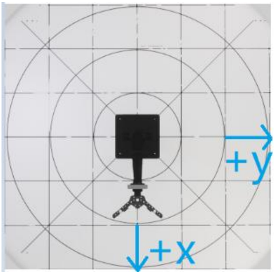

Figure 3: Orientation of the QArm Workspace

4. Note that the **demux** breaks the $7\text{x}1$ vector output from the *Host Keyboard* block into 7 individual signals. Connect the first output of the demux (labelled **space** for you) to the first output of this subsystem (labelled **Learn?**). This will ensure that while you hold the space key on your keyboard, the **Learn?** signal will remain high, indicating that the manipulator record the trajectories.

5. View the manipulator from the top in a way that the manipulator's end-effector points south in the home position, as shown in figure 3. The **up** and **down** arrow keys should drive the end-effector correspondingly in the North and South directions, which in turn correspond to a positive or negative x-direction. Add a gain block to the two channels labelled up and down. Set the value of these gains to $+1$ or -1 depending on the correct directions.

   Hint: The **down** arrow key would move the manipulator in the South direction, which is +X.

6. Repeat the process above for the **demux** outputs labelled right and left to drive the end-effector along the y axis.

7. Repeat the process above for the **demux** outputs labelled Q and A to drive the end-effector along the z axis ($+z$ is out of the view towards you and $-z$ is into the view).

8. Add a **sum** or **add** block to add the outputs of the **gain** pairs from steps 5 and wire that to the input on the **mux** labelled v_x. Do the same for the **gain** pairs from step 6 and 7 to the **mux** inputs labelled v_y and v_z, respectively.

9. Place a **gain** block between the output of the **sum** or **add** blocks and the **mux** you wired it to in step 8. This will help you set the correct magnitude and handle units for command conversion. We want the maximum speed of the end-effector to be 5 cm/s or 0.05 m/s. Set the **gains** to 0.05. Thus, as you tap the **up** or **down** arrow keys, the output of the corresponding **add** block equals -1, 0 or 1, and the v_x input sees -0.05, 0 or 0.05 m/s. Do the same for the v_y and v_z channels.

10. After the three speed signals are grouped through the **mux** block, you need to convert the speed to a position increment. The constant qc_get_step_size is the simulation time step. You need to multiply the speed output of the **mux** with time step **constant** block to get the position increment. Use a **product** block. Wire the output of the **product** block to the **checkKinematics** block.

11. Return to the root level and double click on the subsystem called *Blackbox*. This subsystem records the joint angles and position data to the workspace. Add a **To Workspace** block and wire the **phi (rad) [4]** input to it. Double click on the block and change the variable name to **phi_trajectory**. Under the **Save format** dropdown menu, select **Structure**. Under *Sample time* section, replace -1 with to qc_get_step_size.

12. Repeat step 11 with another **To Workspace** block connected to the **p (m) [3]** input. Use the variable name **p_trajectory** and set the other settings as step 11.

13. At the root level, you want the *Blackbox* to record data when the **Learn?** signal is high. Which signal do you connect that accomplishes this?

14. Copy the *Forward Kinematics* MATLAB function to this model.

15. Prior to running the model, open the model's Configuration Parameters and verify that they are configured as follows:

    1. Solver type: Fixed-step

    2. Solver: ode4 (Runge-Kutta)

    3. Fixed-step size (fundamental sample time): 500 Hz

16. View the manipulator from the top as shown in Figure 4. You can hold the left-click mouse button and drag to change the camera view in Quanser Interactive Labs.

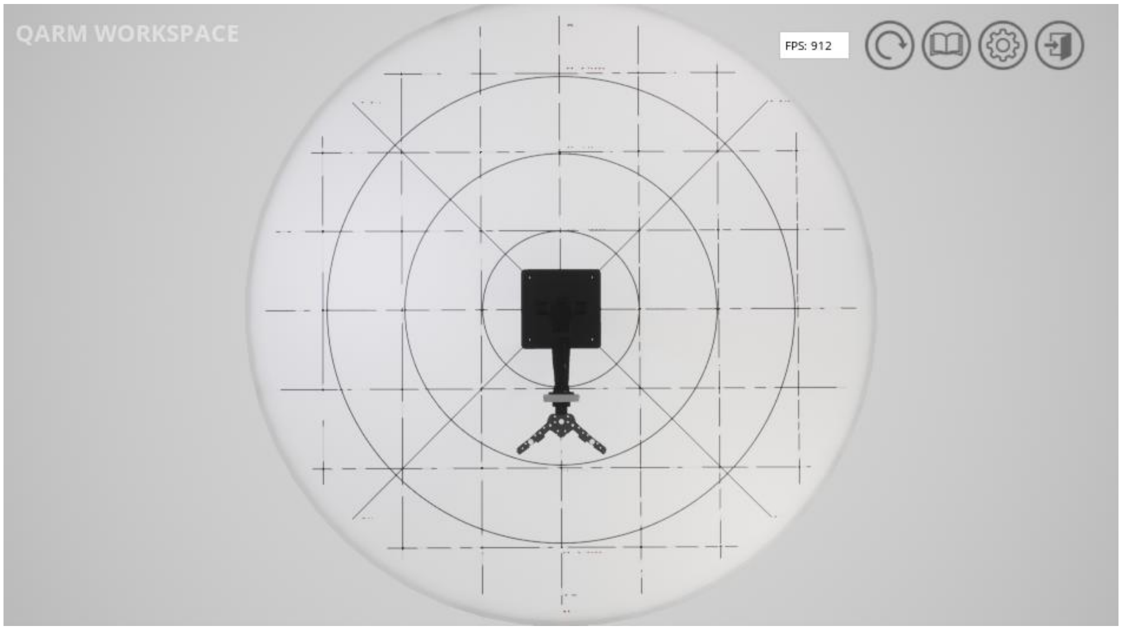

Figure 4 Top view of the manipulator

17. Run the model using the green Play button under the Simulation Tab of your model. Once started, the model will command 0 rad angles to all four of the manipulator's joints.

18. At the root level in your Simulink model, use Q and A to move the manipulator away or towards the table (along z axis). Use the four arrow keys to move the end-effector in the corresponding direction specified by the arrow keys. Do so with the **Space** key depressed (not recording).

19. Move the end-effector to $[0.25, 0.25, 0.10]$ (approximately). Monitor the end-effector's location through the display in the Simulink model.

20. The welding procedure is to start from the **default** $[0.25, 0.25. 0.10]$ to **A** $[0.25, 0.25, 0.01]$ to **B** $[0.25, 0.50. 0.01]$ to **C** $[-0.25, 0.50, 0.01]$ to **D** $[-0.25, 0.25, 0.01]$ then to **A**, and back to **default.**. This emulates holding the end-effector by hand and driving it along the desired path, while it learns the joint positions. Practice several times to get familiar with the keyboard input and its corresponding direction.

21. When ready, repeat the procedure while holding down the **space** key as all the joint positions and the end-effector locations will be stored in a Blackbox. If you make a mistake, stop the model and restart to erase stored data. Do not release the **space** key during the welding procedure.

22. After one complete cycle, stop the Simulink model and close it.

Lead Through (Follow)

1. Open LeadThrough_Follow.slx. This model will take the trajectory data as inputs to recreate the welding process.

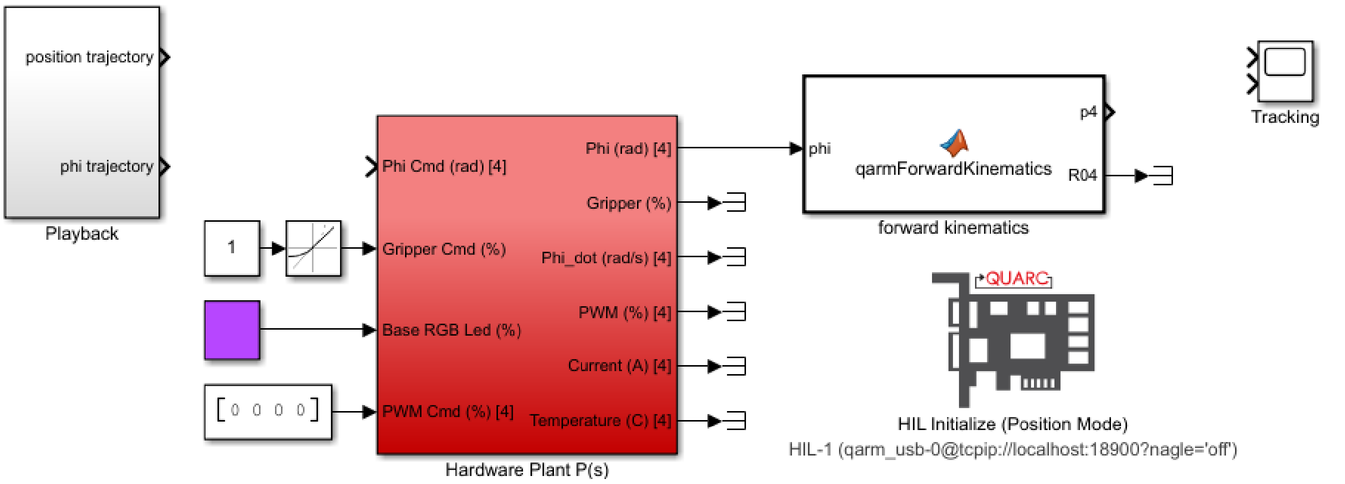

Figure 5 LeadThrough_Follow Simulink model

2. Open the *Playback* subsystem, where you will read the position and angles recorded by the *Blackbox* of the LeadThrough_Learn.slx. Add a **From Workspace** block, and double click to open it. Set the *Data:* field to **`out.p_trajectory`** and uncheck the **`Interpolate data`** option. Replace 0 in the *Sample time* field with qc_get_step_size. Set the *Form output after final data value by* field to **Cyclic Repetition** to repeat trajectories. Connect it to the output port labelled **position trajectory**.

3. Repeat step 2 to read the workspace data **`out.phi_trajectory`** with another **From Workspace** block.

4. At the root level, connect the output **position trajectory** to the first input of the **Tracking** scope and connect the output **phi trajectory** to the first input of the plant **Phi Cmd (rad) [4].**

5. Copy the *qarmForwardKinematics* MATLAB function you completed in the previous lab to this model.

6. Connect the output *p4* of the *qarmForwardKinematics* function to the second input of the **Tracking** scope**.**

7. Prior to running the model, open the model's Configuration Parameters and verify that they are configured as follows:

    1. Solver type: Fixed-step

    2. Solver: ode4 (Runge-Kutta)

    3. Fixed-step size (fundamental sample time): 500 Hz

8. Run the model using the green Play button under the Simulation Tab of your model. Use the **tracking** scope to comment on how well the manipulator tracks the recorded trajectory and the repeatability performance metric. Take a screenshot of a full welding cycle in the scope.

9. Stop the model, close it, then close MATLAB and Quanser Interactive Labs.