Lab Procedure for Python

Forward Kinematics

Setup

1. It is recommended that you review Lab 1 – Application Guide before starting this lab.

2. Launch Quanser Interactive Labs, scroll to the "QBot Platform" menu item and then select the "Warehouse" world.

3. Run observer.py first to initiate receiving data feeds from the QBot Platform.

4. On a separate terminal, run play.py on the QBot Platform. When the script is run successfully, the QBot and the Environment should be spawn in QLab and User LEDs on the virtual QBot will turn blue, as shown in Figure 1.

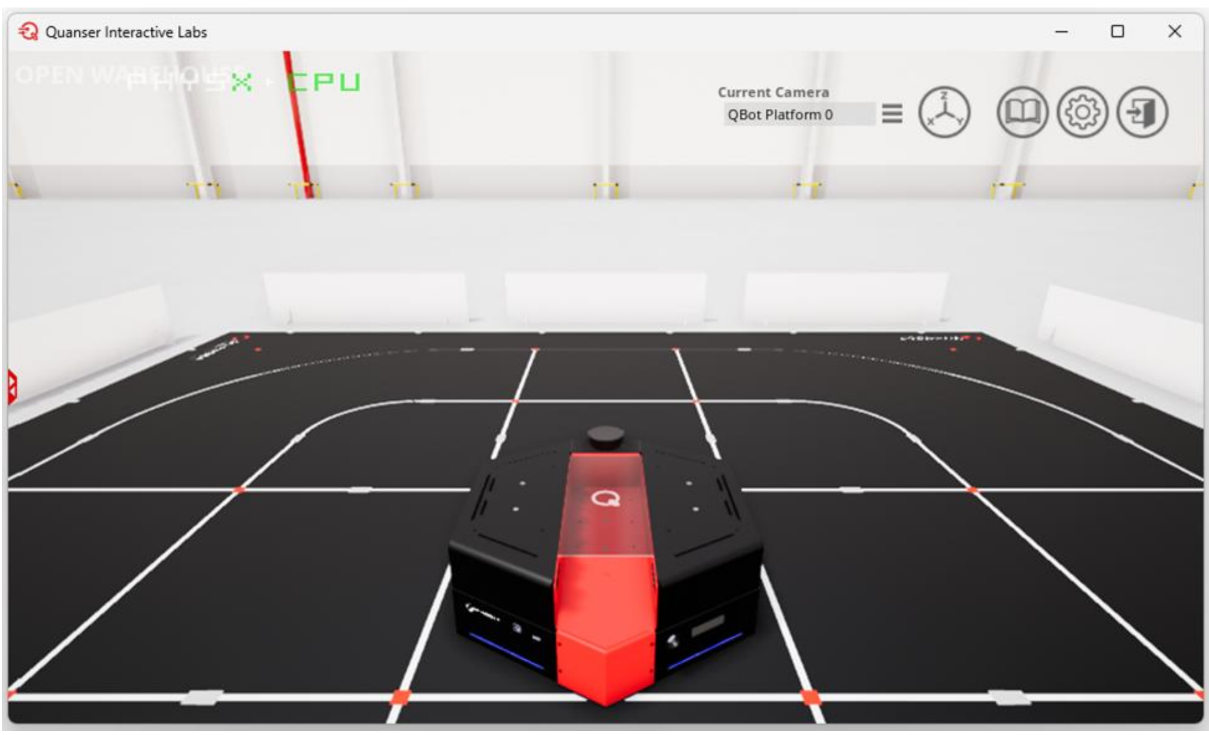

Figure 1. Successful set up of the Quanser Interactive Labs Workspace

Drive with Individual Wheel Speeds Command

1. Press and hold the Space Bar to arm the robot. Notice that the LEDs turn green. Keep this key pressed as you teleoperate the QBot for future labs as well.

   Note: If your robot is ever in a position where it may collide with obstacles, disarm the robot by releasing the Space Bar. The LEDs will turn blue again, indicating that the QBot Platform is disarmed.

2. While armed, use the following keys to move wheels while monitoring the **Motor Speed Plot** and **Body Speed Plot** window as shown in Figure 2. Take note of the wheel speeds, as well as what motion it produces for the QBot Platform body. **Body Speed Plot** displays turn speed from the gyroscope and body speeds from the forward kinematics model which you will compete later.

    a. press the "W" and "S" keys and determine the positive convention of the left wheel speed.

    b. press the "I" and "K" keys and determine the positive convention of the right wheel speed.

    c. move both sticks to drive the QBot around. Take note of the stick combination that produce the following results,

        i. forward or backward motion

        ii. turn in place clockwise or counterclockwise.

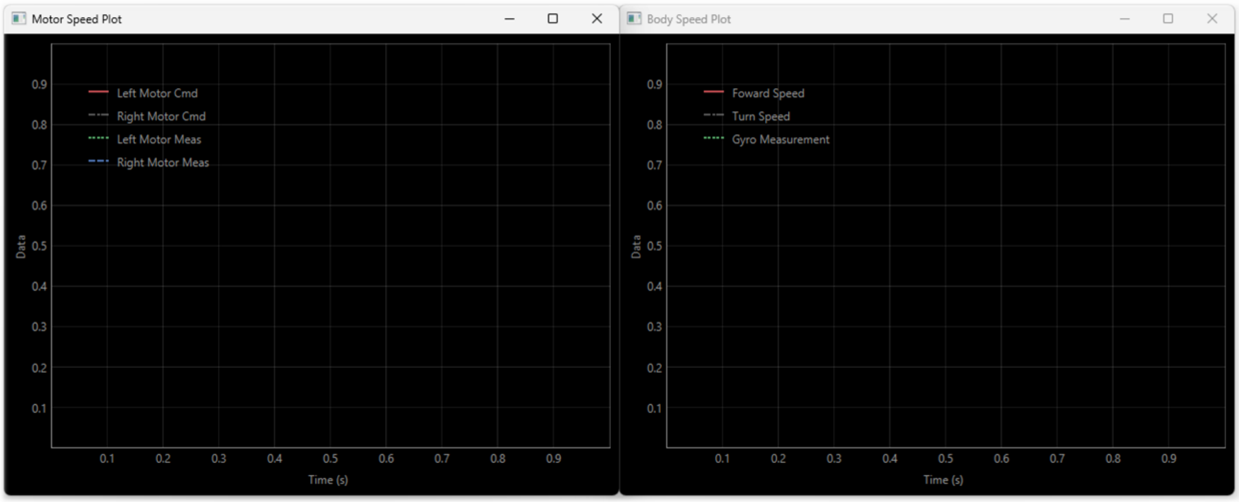

Figure 2. QBot Speeds Scope

3. What relationship do you observe between the wheel speeds and turn speed of the robot?

4. Look at the QLab window and use both wheels to control the QBot to follow the lines on the mat (straight or curved, or both). Take notes on the challenged experienced during this activity.

5. Using observer.py, open the downward facing camera feed as shown in Figure 3. Now, instead of looking at the QLab window, try making the robot follow the lines on the mat using only the downward facing camera feed. Document the strategies you utilized to follow the line and additional challenges faced when using only the camera feed.

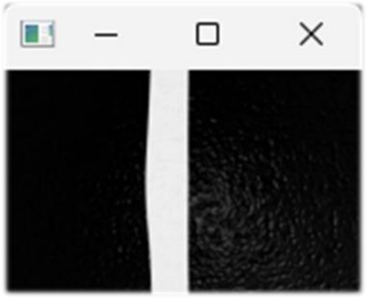

Figure 3. Downward Facing Camera Feed

6. Stop the code by pressing the “U” key.

Forward Kinematics Formulation

1. Open for_kin.py. In Section B, right click, find and right click on the **QBMovement()** class and select “Go to Definition”. This class handles the computation of forward kinematics as well as things needed in future labs, including inverse kinematics and line following, but these methods are incomplete This class also contains physical QBot parameters.

2. Using the predefined global variables with the QBot’s physical dimension, complete **diff_drive_forward_velocity_kinematics()** function under **QBMovement()** to formulate the QBot forward kinematics equation via geometric derivation.

3. Save your changes to **QBPMovement()**. Using the function you completed on the previous step, go back to the for_kin.py file and complete the missing code in Section F.

4. Run observer.py first, then run for_kin.py, and drive the QBot.

5. Using the scope display window, verify that the turn speeds output from your forward kinematics function closely matches that from the gyroscope. Revise your function if needed. When you’re satisfied with the output of your function, take a screenshot of the scope window before shutting down the script. Also monitor the forward speed of the robot.

6. What is the maximum forward or turn speed of the robot?

7. Stop your script using the “U” key. Ensure that you save a copy of your completed files for review later.