Lab 4 – Server Client Programming

Overview:

We have been programming the robots with what can be described as offline control. Offline control involves developing the movement and actions of a robot in a simulation without the real robot. The program is then compiled, downloaded to the real robot, and run without further instructions from us. Even though we have not used a simulator, the compile, download, and run still makes what we have been doing offline control.

The benefits of offline control are that the process is practically and conceptually simple, and it lines up with our experience of compiling and running a program that does its own thing. The downside is that the process can take longer to debug since every time you want to investigate how an actuator moves or what a sensor reports, you have to edit the code, compile, download, and then run. Additionally, you cannot send live commands to the robot as part of the control.

An alternative to offline control is a type of server-client control. In a server-client control, the robot acts like a server (or client) and listens for messages that specify how to make an action or request sensor information. You still have to write, compile, and download the server or client code to the robot, but once it is running on the robot you can interactively command it or query its sensors wirelessly. Whatever offline code was previously put on the physical robot can be run on a separate pc and pipe the action and sensor commands through the wireless connection.

The advantages of this server-client control are the ease in debugging, ability to add operator commands during running, and removes the bulk of the process from the robot's onboard computer. The last point can be very powerful because it allows the robot's onboard computer to be very simple and the ability to use powerful computers and libraries on the client computer. The disadvantage of a server-client control is that setting up the wireless communication and action and sensor protocols can be complicated. Also, the system is susceptible to lag in the wireless communication.

Objectives:

In this Lab, you will write your own server-client program based off sample code I give you.

- Develop your own message protocol for sending robot movements and querying sensors.
- With your server-client protocol rewrite the line follow program from Lab 3.
    - Note, you will be using your server-client control setup for the rest of the course. It does not, and should not, be perfect right now, but you should keep in mind that you will be reusing and building up this code for the rest of the class.

Procedure

1) Setup
    a) Assemble the Base Robot.
    b) Get the server-client demo working
2) Sending motor commands
    a) Create a server-client protocol that allows you to send motor commands to the robot.

        i) With this protocol and the movement models from Lab 1 (forward distance and turning angle).
            (1) Program the robot to drive forward 25 cm. Do this 5 times and record the results.
            (2) Program the robot to turn 90 degrees. Do this 5 times and record the results.
    b) Compare to Lab 1
        i) Because we are going to be controlling the robot differently from the prior Labs, we want to make sure we are getting the same type of actions and behaviors.
        ii) Compare the distance and turning measurements from the server-client control to the distance and turning measurements from offline control in Lab 1 with a **<u>two-sided t-test.</u>**
        iii) For the offline control data, you can reuse your data from Lab 1 or rerun your code from Lab 1 and take new measurements. Note, the speed and duration should be the same used in the prior measurements.

3) Querying sensors
    a) Attach one touch sensor, the IR sensor, the gyro, and the line sensor to the robot. Connect each to a separate port.
    b) Create a server-client protocol that allows you to query a sensor port on the robot and get back a data reading. The following are 3 phases that should be present in your code. The following should occur on the robot's (server) side.
        i) Phase 1: Receives a sensor request for port n.
        ii) Phase 2: Look up what sensor type is attached to port n. You have to manually set or record which sensors are attached to which ports in the server/robot code.
        iii) Phase 3: Based on the sensor type, query the sensor, format the data (if necessary), and send it back to the client.
        iv) Take screenshots for the 3 phases and put them in the Results. This is the data/results for this section. Make sure you talk about or explain what is going on in the screenshots (aside from the caption) when you put them in the report.
    c) Create a script on the client side that continuously queries sensor readings from all four ports and prints them out.
        i) Take a screenshot of it working for your results.

4) WASD control
    a) Create a program that causes the robot to drive forward, back, turn left, or turn right if you press the WASD keys. The space bar should cause the robot to stop.
    b) Take a video. Put a screenshot of the video in the Results as the 'data' for this task. Make sure to talk about what happened and explain the screenshot.

5) Line follow
    a) Convert your line follow code from Lab 3 so that it works on the client side with the motor commands and sensor queries.
        i) Your line following code should be running on the external PC and controlling the robot via the motor commands and sensor queries protocols you just developed.
    b) Take a video. Put a screenshot of the video in the Results as the 'data' for this task. Make sure to talk about what happened and explain the screenshot.

Questions

1. Explain the difference between a 1-sided and 2-sided t-test? What are we using a 2-sided test here instead of 1-sided?
2. Was there a statistical difference between telling the robot to move or turn with the offline control from Lab 1 and the server-client control you created? If there was no difference, what factors could lead to a difference? If there was a difference, where do you think it came from? Hint, think about the cons for server-client control.
3. What were some of the designed considerations you thought about when coming up with the message protocol for motor commands and sensor query?
4. What are some of the applications of controlling a robot live like you did with the WASD program?
5. What was the most difficult part of this Lab?

Write up reminders

0. Note, these are just general reminders of what should be in each section. Anything that is mentioned in the procedure, class discussions about Labs, or the Lab expectations should still be included in the report even if it is not directly mentioned here.
1. Methods
    a. Describe and explain offline and server-client control. Use figures and diagrams. Make sure to explain and walk through any figure or diagram you present.
    b. Describe your message protocol for sending motor commands and query sensors. That includes the different tokens you use and what they mean.
    c. Briefly describe the Lab procedure and tasks.
2. Results
    a. <u>Any piece of data you put in the Results should be accompanied by some text (aside from the caption) that gives some background on the Task and explains the data we are looking at.</u>
    b. Task 2 – Table and/or plots for your data and the statistical analysis (mean, std, t-test).
    c. Task 3 – Screenshots of the phases and screenshot of the live sensor query.
    d. Task 4 – Screenshot(s) for the video you took. Video to be submitted via Canvas.
    e. Task 5 – Screenshot(s) for the video you took. Video to be submitted via Canvas.
3. Supplementary Material and Submission
    a. Submission – One zipped folder named after the team members usernames. In the folder should be the report, a folder for your code, and a folder for all the videos.
    b. Supplementary Material – Should contain the following sub-sections
        i. Section 1
            1. List of videos you submitted and a 1 line description of each
            2. List of python files you submitted and a 1-2 line description of each
        ii. Section 2
            1. Screenshots/pdfs of your code

Rubric – Lab 4

General formatting (10)

- Title, date, name, section headings
- Miscellaneous style and formatting

Abstract (15)

- Gives a brief intro or overview of the Lab and/or topics covered.
- Gives a brief summary of the Results and main points from the Discussion.
- Well written and concise without typos or grammar issues.

Methods (20)

- Possess all relevant equations and algorithm explanations
- Tasks and procedures are described.
- Looks nice and is easy to follow.

Results/Supplementary (40)

- Figures and text for all experiments.
- Data and analysis for all experiments. Includes any required statistical summaries or tests.
- Figures are properly formatted (labels, legends, captions, etc.).
- Required videos and code.

Discussion (15)

- All questions are answered.
- Well written and concise without typos or grammar issues.