Lab 4: Localization

Submission instructions: Students are expected to work in their assigned lab groups. The lab consists of three components, demonstration, lab report, and code submission. Instructions on the demonstration can be found in this handout. Lab reports and code submission must follow the guidelines established in this handout and for the course. For more information, see the ECSE211SubmissionInstructions.pdf on MyCourses.

Design objectives

1. Design a system to localize the robot using ultrasonic and light sensors, where the robot should move to a known starting position.
2. Evaluate the design and determine how well the system localizes the robot.

Design requirements

The following design requirements must be met:

- The system must localize the robot to the origin of the tiles grid system.
- Localization must use the ultrasonic and light sensors.
- The robot must localize approximately to the 0° direction using the ultrasonic sensor.
- The robot must provide input to select Rising Edge or Falling Edge. Hence, two versions of the localization routine must be available:
  - Rising Edge
  - Falling Edge
- The robot must localize to the (1, 1) gridpoint and 0° using the light sensor.
- The robot must wait for input once completing ultrasonic localization and orienting to 0°.

Demonstration (30 points)

The design must satisfy the requirements by completing the demonstration outlined below.

Design presentation (10 points)

Before demoing the design, your group will be asked some questions for less than 5 minutes. You will present your design and answer questions designed to test your individual understanding of the lab concepts. Each person will be graded individually.

You must present your workflow, an overview of the hardware design, and an overview of the software functionality. Visualizing software with graphics such as flow charts is valuable.

Demonstration procedure

As shown in **Figure 1**, the robot is placed along the 45° line (thick dotted) inside the bottom-left tile of the 4x4 field grid. The **brown-colored** walls represent walls used for the **ultrasonic localization**. Please note that the robot could be placed in **ANY ORIENTATION** and **POSITION** along this 45° line. For example, **Figure 2** shows another orientation and position of the robot's starting position compared to that in **Figure 1**.
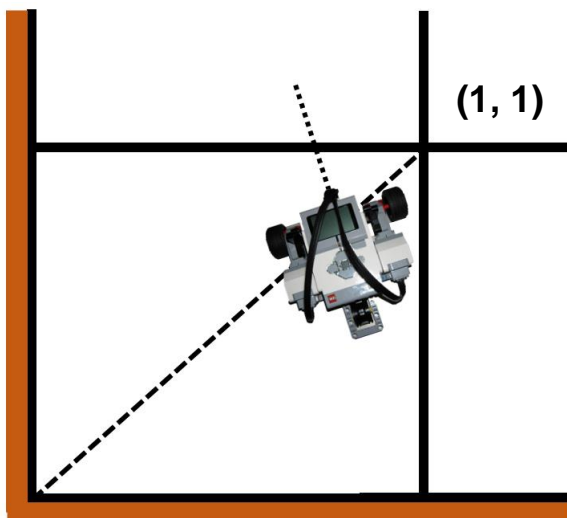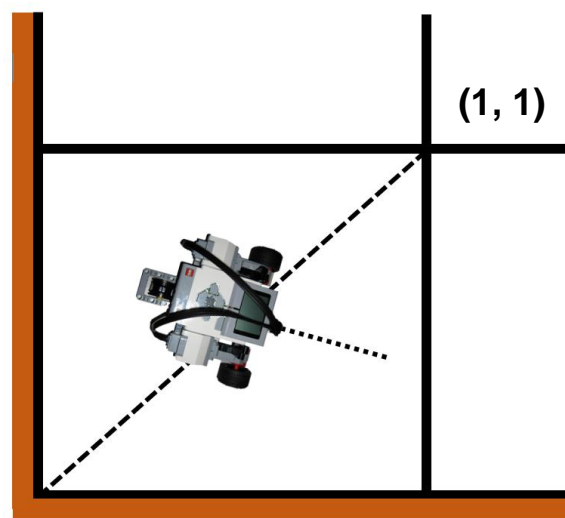

Figure 1. Robot's Starting Orientation A.

Figure 2. Robot's Starting Orientation B.

Indicate to the TA what version of the ultrasonic localization you would like to use. Upon completing the **ultrasonic localization**, the robot should orient to its estimated 0° axis and the TA will measure the *orientation error* with respect to its true 0° axis. Note that you should have a stopping criterion to ensure that the robot stays still while the *error angle* is being measured (e.g. `Button.waitForAnyPress()` can be used to separate the two procedures).

After the measurement is complete, the robot should then move to the (1, 1) point as shown in **Figure 1** and **Figure 2**. Once **light localization** is performed at this point, two error quantities are measured by the TA: the *Euclidean distance error* **ϵ** between the robot's actual position (X_F, Y_F) and the (1, 1) point, and the *orientation error* with respect to the 0° axis.

$$\epsilon = \sqrt{(X_F)^2 + (Y_F)^2}$$

Ultrasonic Localization (5 points)

- **5 points** are given for *orienting* the robot on its 0° axis within an error tolerance of **±10°**. A penalty of **2.5 points per ±10°** is used for angles beyond 10°. Hence, the following points are awarded for the actual robot's orientation:
  - ± [0, 10] °   →   **5 points**
  - ± (10, 20] °   →   **2.5 points**
  - ± (20, ∞) °   →   **0 points**

Light Localization (15 points)

- **5 points** are given for orienting the robot along its 0° axis at point (1, 1) within an error tolerance of **±5°**. A penalty of **2.5 points per 5°** is used for angles beyond 5°. Hence, the following points are awarded for the actual robot's orientation:
  - ± [0, 5] °   →   **5 points**
  - ± (5, 10] °   →   **2.5 points**
  - ± (10, ∞) °   →   **0 points**

- **10 points** are given for reaching point (1, 1) within an error tolerance of **2 cm** using a *Euclidean distance*. A penalty of **2.5 points per 2 cm** is used for distances beyond 2 cm. Hence, the following points are awarded for the actual robot's position relative to the point (1, 1):
  - [0, 2] cm   →   **10 points**
  - (2, 4] cm   →   **7.5 points**
  - (4, 6] cm   →   **5 points**
  - (6, 8] cm   →   **2.5 points**
  - (8, ∞) cm   →   **0 points**

Provided materials

Sample code

No sample code is provided for this lab. Instead follow the guidelines given below:

- Create a `UltrasonicLocalizer` class that performs the two ultrasonic localization routines: `FallingEdge` and `RisingEdge`
    - Create a method called `fallingEdge` that performs the falling edge localization
    - Create a method called `risingEdge` that performs the rising edge localization

- Create a `LightLocalizer` class that performs the light localization routine

Physical material

In the lab, tiles with grids will be provided which will have walls for localization.

Implementation instructions

1. Using the class definitions above, implement the ultrasonic localization versions.
    a. For more information on rising and falling edge, see the tutorial provided.
2. Using the class definition above, implement the light localization routine.
3. Create a main class that:
    a. First, lets the user select the falling or rising edge routine,
    b. Second, pauses after ultrasonic localization and waits for user input before continuing to light localization.

Report Requirements

The following sections must be included in your report. Answer all questions in the lab report and copy them into your report. For more information, refer to ECSE211SubmissionInstructions.pdf. **Always provide justifications and explanations for all your answers**.

Section 1: Design Evaluation

You should concisely explain the overall design of your software and hardware. You must present your workflow, an overview of the hardware design, and an overview of the software functionality. You must briefly talk about your design choices before arriving at your final design. Visualizing hardware and software with graphics (i.e. flowcharts, class diagrams) must be shown.

Section 2: Test Data

This section describes what data must be collected to evaluate your design requirements. Collect the data using the methodology described below and present it in your report.

Test localization using rising edge (*10 independent trials*)
1. Place the robot in a tile corner along the 45° line (**Figure 1**), where two walls are present.
2. Choose a random orientation for the robot.
3. Run the ultrasonic localization routine using **rising edge**.
4. Note the ultrasonic angle of the robot.
5. Compute the ultrasonic angle error using the ultrasonic angle and the expected value.
6. Continue the localization using the light sensor to center at (1, 1) and 0°.
7. Note the final position and final angle of the robot.
8. Compute the Euclidean distance error and final angle error.
9. Report in a table: ultrasonic angle error, Euclidean distance error, and final angle error.

Test localization using falling edge (*10 independent trials*)
1. Place the robot in a tile corner along the 45° line (**Figure 1**), where two walls are present.
2. Choose a random orientation for the robot.
3. Run the ultrasonic localization routine using **falling edge**.
4. Note the ultrasonic angle of the robot.
5. Compute the ultrasonic angle error using the angle and the expected value.
6. Continue the localization using the light sensor to center at (1, 1) and 0°.
7. Note the final position and final angle of the robot.
8. Compute the Euclidean distance error and final angle error.
9. Report in a table: ultrasonic angle error, Euclidean distance error, and final angle error.

Section 3: Test Analysis

For each test, compute the mean and standard deviation of the ultrasonic angle error, Euclidean distance error, and final angle error. Be sure to show general formulas and sample calculations.

Section 4: Observations and Conclusions

1. Which of the two localization routines performed the best?
2. Was the final angle impacted by the initial ultrasonic angle?
3. What factors do you think contributed to the performance of each method?
4. How do changing light conditions impact the light localization?

Section 5: Further Improvements

1. Propose a software or hardware way to minimize errors in the ultrasonic sensor.
2. Propose another form of localization other than rising-edge or falling-edge.
3. Discuss how and when light localization could be used outside of the corner to correct Odometry errors, e.g. having navigated to the middle of a larger floor.

Frequently asked questions (FAQ)

1. **Do I need to implement my robot's localization routines using the same 0° axis shown in Figure 1 and Figure 2 (that is, along +y axis)?**
   Yes.

2. **Do I need to move to another position within the ultrasonic localization procedure?**
   No, you should only rotate about your starting position. Once the *orientation error* in part 1 is measured, the robot should then move to the (1, 1) point for **light localization**.

3. **What is meant by "design presentation"?**
   Before a lab demo, you and your partner will briefly present your design. This can include a basic visualization of how your code functions, such as a flow chart. You will then be asked a series of questions. These could be related to the lab tutorial and the initial lab code. For this part, a grade of 10 signifies full understanding, 5 signifies satisfactory understanding, while 0 shows no understanding at all. Note that memorized answers are discouraged and both partners should be responsible for understanding the design and their associated code, even if one of them did not write all of it.