Lab 4 – SLAM

Austin Holmes – U#31858364

Control of Mobile Robots – Section 001F23

INTRODUCTION

Lab four of the Control of Mobile Robots focuses on localization within a map of known geometry or containing a set of features with known global position. The localization capabilities developed in this lab are then used to support global navigation functionality.

TASK 1 – TRILATERATION

The first task described in this lab requires that the robot determine its position within the below environment by way of trilateration, or assessing its own position based on its distance from 3 known points. Once position is ascertained, the robot navigates to each cell in the grid at least once before stopping.

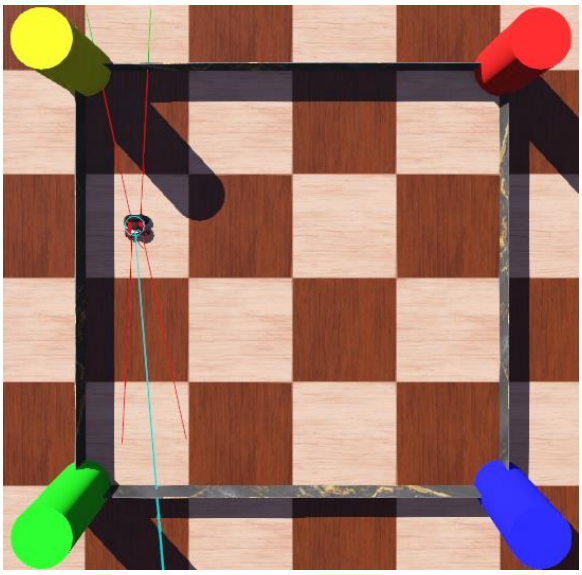

Figure 1 – Trilateration Test Environment

TASK 2 – WALL LOCALIZATION

Task 2 involves probabilistic localization based on a predetermined motion model and analysis of the cardinal wall characteristics of the present cell. This information is compared against a known map in order to determine relative position, which is again used to support navigation of the full grid.

Figure 2 – Wall Localization Test Environment 1

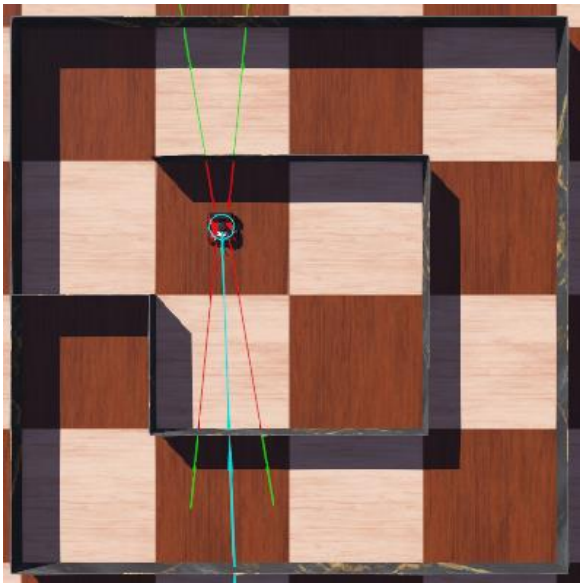

Figure 3 – Wall Localization Test Environment 2

Figure 4 – Wall Localization Test Environment 3

```
+  -  -  -  -  +
|  .  ▪  .  .  |
|  .  .  .  .  |
|  .  .  .  .  |
|  .  .  .  .  |
+  -  -  -  -  +
Cell: X: -1, Y: -2, Confidence: 0.25378815917253783
Exact Coordinates: X: -1.00, Y: 2.00, θ: 90
[[0.02 0.25 0.02 0.02]
 [0.02 0.02 0.02 0.02]
 [0.02 0.02 0.02 0.02]
 [0.02 0.25 0.25 0.02]]
```

Figure 5 – Example Presence and Probability Grid / Stats

KEY VALUES

A. Grid Dimensions

The dimensions of the global grid (in cells) are provided to the program as a parameter in order to set bounds for navigation. For the purposes of this lab, these dimensions are determined by the size of the overall bounding box formed by the outer walls of each environment.

B. Cell Length

The square dimension of each cell is also passed as a parameter such that any rectangular environment can be analyzed as a grid of variable resolution.

C. Landmarks

Specifically used for Task 1, a list of landmarks which includes color and corresponding position is used to identify the position of visible landmarks.

D. Cell Walls

For Task 2, the wall characteristics in terms of whether there is a wall in a given cardinal direction are given to the program to support the probabilistic identification of cells based on walls.

Note that this list is not exhaustive and omits all entries mentioned in previous lab reports or which are not imperative to the purposes of this report.

KEY FUNCTIONS

A. Probability to Logs-Odd (and Inverse)

To ease probabilistic calculations, probability values are converted to Logs-Odd during any mutative procedures. The conversion is given as below, where p is the fractional probability, and l is the Logs-Odd representation:

$$l = \ln \frac{1}{1-p}$$

Along with its inverse:

$$p = 1 - \frac{1}{1+e^{l}}$$

B. Update Cell Probability

Cell probability calculations are handled principally by this function, which accepts a cell within the known grid as well as the most recent immediate probability of the robot being in that cell and updates the associated probability according to the following, where the p is the final probability, p_i is the input probability, and p_{i-1} is the existing probability (where all values are in Logs-Odd form):

$$p = p_{i-1} + p_i$$

This function also supports a “force” parameter which changes this calculation to:

$$p = p_i$$

With that, a given cell can be provided an asserted value which does not depend on prior values, in this lab enabling the assertion of a starting position.

C. Move Probably

Per the specifications of the motion model for this lab assignment, movements made during probabilistic navigation are assigned an 80% chance of succeeding. To streamline the process of applying these probabilities during motion, the move_probably function(s) perform the specified motion, updating the cell probabilities of the current and next cell at the same time according to the update_cell_probability function.

D. Detect Wall (Cardinal)

Wall detection for the purposes of probabilistic cell estimation is handled by this function, which applies probabilities (confidence levels) to positive and negative wall detection results. Per the lab specifications, a detected wall is assigned a confidence of 90%, and a lack of detected wall is assigned a confidence of 70%. The overall confidence value for being in a cell matching the detected wall characteristics is then provided as follows, where c is the overall confidence as a fractional probability, and c_i is the confidence for the wall detection currently being assessed:

$$c = \prod_{i=1}^{4} c_i$$

E. Wall Estimate Cell

Making heavy used of detect_wall_cardinal, this function is responsible for applying probabilities to the cell grid based on the walls immediately around the robot in the cardinal directions. The result of the wall detection function is checked against the known values in the Cell Walls parameter and the confidence provided alongside that result is distributed evenly among matching cells according to update_cell_probability.

F. Normalize Cell Probabilities

To avoid the sum of all fractional probabilities exceeding 1, a normalization function is called after every probability update which divides the probability value of each cell by the sum of all probabilities, bringing the sum to 1, as given by the below formula, where p_c is the probability for a given cell:

$$p_c = \frac{p_c}{\sum_{i=0}^{n} p_i}$$

G. Guess Cell

Guess_cell simply returns a reference to the cell with the greatest probability of presence, or the first of any tied cells.

H. Find Landmarks

This function performs a full rotation, storing all landmarks which were visible at any point during the rotation.

Note that this list is not exhaustive and omits all entries mentioned in previous lab reports or which are not imperative to the purposes of this report.

I. Coordinate to Cell

A general utility, this function converts a global x and y coordinate to a cell reference according to the cell in which the coordinates lie per the following equation, where l is the cell length parameter, c_x is the x coordinate of the cell, c_y is the y coordinate of the cell, x is the input x coordinate, and y is the input y coordinate:

$$x = \left\lfloor \frac{x}{l} \right\rfloor$$

$$y = \left\lfloor -\frac{y}{l} \right\rfloor$$

LANDMARK LOCALIZATION

Following the collection of all visible landmarks by find_landmarks, the first three are used in the following calculation, which returns the coordinates *(x, y)* of the observer (robot):

c_{ix}: x-coordinate of i^{th} landmark

c_{iy}: y-coordinate of i^{th} landmark

c_{ir}: radius of i^{th} landmark

$$A = 2c_{2x} - 2c_{1x}$$

$$B = 2c_{2y} - 2c_{1y}$$

$$C = c_{1r}^2 - c_{2r}^2 - c_{1x}^2 + c_{2x}^2 - c_{1y}^2 + c_{2y}^2$$

$$D = 2c_{3x} - 2c_{2x}$$

$$E = 2c_{3y} - 2c_{2y}$$

$$F = c_{2r}^2 - c_{3r}^2 - c_{2x}^2 + c_{3x}^2 - c_{2y}^2 + c_{3y}^2$$

$$x = \frac{CE - FB}{EA - BD}$$

$$y = \frac{CD - AF}{BD - AE}$$

The positional value obtained from this calculation is then converted to a cell reference via guess_cell and passed to the grid navigation algorithm described in FULL GRID NAVIGATION.

STATISTICAL WALL LOCALIZATION

By invoking the procedure provided by detect_wall_cardinal and wall_estimate_cell at every stationary timestep wherein the robot should be at the approximate center of some cell, probabilities of increasing confidence are applied over each cell in the map grid. As this is effectively done upon initialization and following every motion action, the robot develops an increased confidence for the cell in which it lies as opposed to only using the motion model.

Due to the confidence values assigned to wall detection and motion, however, the application of wall probabilities only decreases the confidence level of the cell in which the robot actually is, leading to reduced predictability. This is true when the robot is provided its starting position, though, and remains a useful tool for determining the position of the robot when no starting position is provided, or in contexts where the robot may have low motion confidence and the map has a relatively unique cell distribution.

FULL GRID NAVIGATION

Upon determining the present cell, the robot proceeds to attempt to navigate every unobstructed cell within the grid, continuing until the sum of the sets of “visited” cells and “occupied” cells is equivalent to the product of the grid dimensions. The algorithm used to do so is as follows:

Let an “available” cell be one which has no wall obstructing motion to it which is within the bounds of the map grid and has not been visited before and contains no obstacle.

Let “attempt” mean to navigate to the referenced cell if it is “available,” adding that movement to the list “previous moves.”

1. Add current cell to set “visited.”
2. Attempt to move left.
3. If not, attempt to move up.
4. If not, attempt to move down.
5. If not, attempt to move right.
6. If not, perform the opposite of the previous move and removing it from the list.
7. Return to step 1.

While this algorithm does not seek to optimize movement, it guarantees that every reachable cell will be reached given sufficient time, and uses minimal memory and computational resources.

CHALLENGES

Aside from simple conversion errors between array values and cell indices (which occupied a

disappointing amount of my time on this lab due to my own incohesion), the principle difficulties in this lab seem to arise from the confidence values and the manner in which I calculate probabilities based on wall detection. As a motion model of 80% success does not at all reflect the decision model of the navigation algorithm nor the success rate of motion, the probability distribution is made immediately chaotic, and the lack of confidence in wall detection has a similar effect. This results in conditions where the robot becomes entrapped in motion loops due to shifting and often oscillating position estimates.

CONCLUSION

Probabilistic robotics is an essential tool for real-world localization and mapping as it allows a more heuristic approach which allows for existing conclusions to be updated with more accurate ones as data is collected. That said, assigning probabilities which align with the actual functionality of the robot is essential as it may define the difference between excellent navigation and a nervous automaton which cannot confidently ascertain its position.

NOTES

Confidence values should likely be increased dramatically for future usage.

VIDEOS

Included in submission archive.