Using Fuzzy Logic for Environmental Mapping in Robotic Applicaitons

Michael Folcik and Christopher J. Martinez
Department of Electrical and Computer Engineering
University of New Haven
West Haven, CT, USA
cmartinez@newhaven.edu

***Abstract*—Robotics has seen an exponential growth in research during the past decade. The main reason for the growth is due to advancement in hardware. With all the hardware advancements, autonomous mobile robots have become a very important topic. The number of applications for autonomous robots is continuing to grow. This paper examines how to map the surrounding environment in a building using fuzzy logic. A fuzzy inference system is developed which examines a small local area and continuously assemblies a global map of the environment. We have classified fifteen probable relative local locations and through a random exploration of the environment a complete global map is created. This paper will explain the fuzzy system and show how the space is mapped to a visual map along with a mathematical graph. Graph theory can later be applied to the environmental graph the system creates to determine shortest path and additional navigation in the environment.**

Keywords-fuzzy logic, robotics, maze-solving, environment mapping, graph theory

I. Introduction

Robotics has always been a popular research area. In recent years, the research area has seen exponential growth. The growth is due in part to the essential underlining hardware advancing, e.g. high-speed wireless communication, ad-hoc network algorithms, low power CPUs, GPS systems and etc. Another reason for growth in the area is due to the limitless potential applications. This research paper looks at how to improve intelligence in autonomous mobile robots.

An important aspect of autonomous mobile robots is the ability to have precise navigation in the robotic systems. Navigation in autonomous systems is a non-trivial task depending on the application. Some applications for autonomous robots include: land mine detection, exploration of landscapes such as Mars, urban vehicle navigation, maze solving contests in academia, and household tasks such as vacuum cleaning and lawn mowing. Each of these applications has their own difficulties in implementation. Some applications require extremely rudimentary intelligence while other will require advanced artificial intelligence. Autonomous lawn mowing, for example, only requires a set of boundaries as a group of GPS coordinates or underground land markers. Other applications require large amount of environmental data, as in the exploration of Mars, due to the number unknown conditions.

Our paper develops a fuzzy inference system (FIS) for autonomous applications. This paper examines the environmental mapping required in autonomous robots used in a search and rescue scenario. One possible example of a search and rescue application includes firefighters needing data about a building that is on fire. Since a burning building is a dangerous situation, autonomous robots can provide an additional layer of safety to the rescue workers. The robots could be placed inside a building through any windows or doorways. The robots will have to quickly create a floor plan of the building to wirelessly transmit to rescue workers outside the building. Given this scenario, the robot may not be able to rely on computer vision based on visual obstacles, so ultra-sonic sensors to measure wall distance are used in this paper.

II. Previous Research

The problem of environmental mapping is not new. Many researchers have looked into determining how to accurately map the environment. Two of the main difficulties of environmental mapping are how to efficiently scan the entire surface area of the environment and how to account for the physical limitations of the physical systems. The physical system, the robot, is made up of many mechanical parts and electrical components. The mechanical components result in large accumulating errors. The robot is designed to move forward, backwards, turn clockwise and turn counterclockwise. Ideally these movements will be precise and have no variations over time. Unfortunately, the mechanical components are not ideal. The motors that drive the robot wheels will have many variations over time. The motors may slip or each motor have slightly different characteristics resulting in the wheels spinning differently relative to the other wheels. The surface the wheels lay on will also lead to robots movements being uneven. These physical limitations result in motion drift. Figure 1 shows a visual example of motion drift. The electrical sensors also have similar variation in their operations. All of these errors lead to difficulties in accurately mapping the environment.

Most of the mapping research has fallen into two different techniques: topological and geometrical [1,2,3]. Topological

Non precise, Rough Estimates

Topological Mapping - Quick
- Lower resolution of Data
- Ideal for Navigation
- Ideal for Waypoint Nav

mapping results in a qualitative representation of the environment [1,3]. The physical system is not highly modeled resulting in non-precise mapping, but rough estimates are available. Topological mapping will allow the robot to have a generalized representation of its location in the environment. For a quick breakdown of the environment, only a topological mapping technique is important. Topological data allows for enough information for navigation and waypoint marking. Topological mapping is commonly represented as a set of graphs. Geometrical techniques, on the other hand, are concerned with producing a precise representation of the environment [2]. Geometrical techniques will take into account the physical system in more detail than the topological techniques.

Precise Representation

Geometrical Mapping: Higher Resolution of data Captured

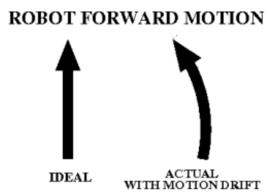

Figure 1. Visual example of the effects of motion drift in robotic movement.

The papers by Kuipers [1] and Thrun [3] provide some insight into current topological mapping techniques. The work in [1] applies a model of the human cognitive map to determine topological information. The focus of the paper is to develop a complete methodology based on sensory, control, causal, topological, and metrical criteria. The work done by [3] develops a map based on particle-filtering techniques. Their research uses Bayesian and Bayes filters to provide a highly detailed motion model of the robot. The downfall of [3] is that a large amount of computational power is needed for the robot to develop the map. Our research is a topological technique that uses a FIS to develop a qualitative representation of the environment.

III. Fuzzy Inference System Model

Many of the existing mapping techniques require many different types of sensors (laser distance sensors, GPS systems, tilt sensing, etc) and sensors with high sensitivity. The goal of this research is to develop a robust mapping technique with a minimal number of sensors that can have any degree of sensitivity and still develop a reasonable topological map of the environment. By using fuzzy logic we are able to determine position with great flexibility and compensate for the error due to the physical components of the robot. We design the fuzzy logic mapping to be used on a robot similar to Figure 2. The robot in Figure 2 has the following components: 3 ultra-sonic distance sensors (one on right side, left side, and front of the robot), a magnetic compass, and a gyroscope that can be used to realign the robot due to the motion drift. The magnetic compass and gyroscope are not used in the FIS but are instead used to find direction and error in the motor movement. The ultra-sonic sensors are sampled regularly and the data is passed into the FIS.

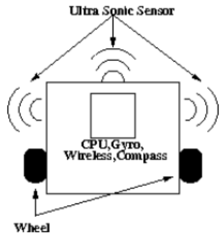

Figure 2. Robot with components.

Fuzzy logic is not new in robotic systems. Fuzzy logic has previously been used in different control systems. The work done in [4] used fuzzy logic to control the speed of the motors on the robot. Based on the distance from the wall the fuzzy logic would determine how fast the motors should turn. Research done in [5] used a combination of neural network and fuzzy logic to create an Adaptive Neuro-Fuzzy Inference System (ANFIS). The ANFIS system that was created was able to learn the maze the robot was in and determine how the robot should navigate. The outputs of the ANFIS system told the robot to follow the wall, turn around, turn the corner, etc.

As we have previously discussed, all robotic systems have noise that is present in the sensors and in the robotic motion. The options to dealing with these problems are to either develop a complex physical model for the robotic motion or spend a large amount of money on the most precise sensors available. The purpose of using FIS is to mask the errors and noise in the system. The fuzzy logic can use cheap sensors and generalize the distance into quantitative measurements such as very close, close, medium, far and very far. The motion drift can also be accounted for because the sensor data is quantitative so the movements do not need to be highly predictable. We have decided to create a FIS to make the robot insensitive to the errors that would occur in operation.

The goal of the FIS in environmental mapping is to determine the robot's relative location. The interior of a building is very similar to a maze. The relative location of the robot can be one of the following situations: the robot is in free open space; the robot is in a dead end; or the robot has a wall or walls next to it. The interior of building is broken down into a set of fifteen potential absolute locations inside a building. Figure 3 shows a breakdown of the absolute locations. Each of the absolute locations will be given an index value the robot can use in its processing; we will refer to the absolute locations as the absolute location index where the index number refers to the numbers found in Figure 3. We have required that the robot have some sense of magnetic direction to determine north, south, west and east used to distinguish absolute position index 1 from absolute position index 2. FIS is perfect for this application because we are only concerned with a rough floor of the system. Figure 4 shows how the FIS output will be an approximation of actual locations inside a building. The FIS will find the best fit from the actual environment and our fifteen absolute locations.

Micro controller -> ARM MBED MCU Prototyping Kit v10.2 Ref: 2456112

mBed NXP LPC11U24

A. *Fuzzy Logic Rules Set and Defuzzification*

The FIS we created is based on a mamdani fuzzy model. The system is created with three inputs and one output. The three inputs correspond to the three ultra-sonic sensors: LEFT SIDE, RIGHT SIDE, and FRONT SIDE. The output that is

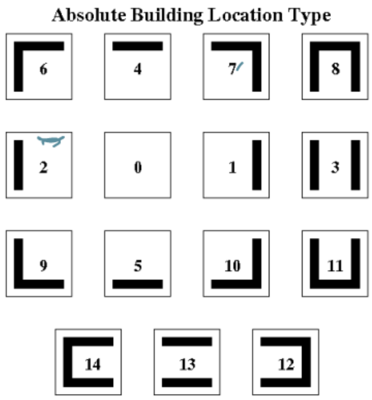

Figure 3. Types of building locations environments. The indexes are used in the robot software as absolute environmental locations.

created is the RELATIVE LOCATION in the maze. The following settings were used: And method = min function, Or method = max function, Implication = min function, Aggregation = max function, and Defuzzification = centroid function. The output of the FIS is the relative location. The location is not the same as the absolute location presented in Figure 3. The relative location does not have any direction associated with it so the location is relative to what is in front and to the side of the robot. The relative location will later be translated to an absolute location.

Fuzzy logic works by assigning each input with a membership function. The membership function is then used to classify the properties of the input in to a set of quantitative measurements. Figure 5 shows the membership function for one ultra-sonic sensor, and all three ultra-sonic sensors have the same membership functions. The membership function will output a probability on how likely it is the robot is CLOSE to a wall and the probability of the robot being FAR from the wall.

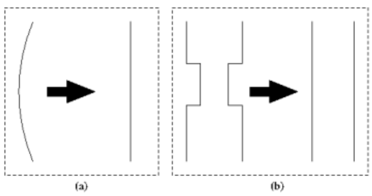

Figure 4. Fuzzy approximations in a building. (a) a curve wall is seen as a straight wall (b) a hallway with inlays is seen as a hallway.

The values of determining far and close can be manipulated to match any scenario.

The output of FIS is also mapped into a set of membership functions (Figure 6). A set of fuzzy rules and defuzzification will determine which membership function the output is mapped into. The relative location output can be one of eight choices: 1) no walls, 2) a left wall, 3) a right wall, 4) a left and front wall, 5) a right and front wall, 6) front wall, 7) a left, right and front wall, and 8) a left and right wall.

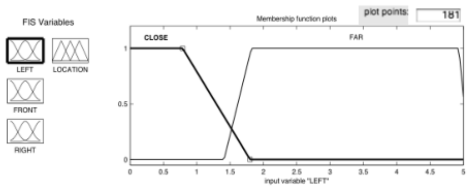

Figure 5. Membership function for ultra-sonic inputs.

The matching from the inputs to the output is done through a set of fuzzy rules. The rules used in this research are intuitive but can be expanded in the future. The rules set up which membership function from the inputs should give the correct values for the output membership functions. The rules implemented in the fuzzy system are as follows:

- **IF** (LEFT is FAR) and (FRONT is FAR) and (RIGHT is FAR) **THEN** (LOCATION is NOWALLS)
- **IF** (LEFT is CLOSE) and (FRONT is FAR) and (RIGHT is FAR) **THEN** (LOCATION is LEFTWALL)
- **IF** (LEFT is FAR) and (FRONT is FAR) and (RIGHT is CLOSE) **THEN** (LOCATION is LEFTWALL)
- **IF** (LEFT is CLOSE) and (FRONT is CLOSE) and (RIGHT is FAR) **THEN** (LOCATION is LEFTFRONTWALLS)
- **IF** (LEFT is FAR) and (FRONT is CLOSE) and (RIGHT is CLOSE) **THEN** (LOCATION is RIGHTFRONTWALLS)
- **IF** (LEFT is FAR) and (FRONT is CLOSE) and (RIGHT is FAR) **THEN** (LOCATION is FRONTWALLS)
- **IF** (LEFT is CLOSE) and (FRONT is CLOSE) and (RIGHT is CLOSE) **THEN** (LOCATION is LEFTFRONTRIGHTWALLS)
- **IF** (LEFT is CLOSE) and (FRONT is FAR) and (RIGHT is CLOSE) **THEN** (LOCATION is LEFTRIGHTWALLS)

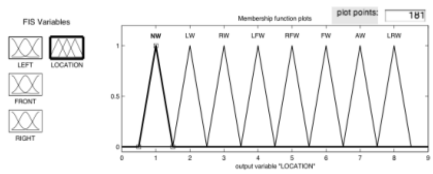

Figure 6. FIS output membership function

An visual example showing how the inputs enter the system and the corresponding generated output is given in Figure 7.

IV. USING THE FUZZY LOGIC OUTPUT TO FIND LOCATION IN THE ENVIRONMENT

The FIS output only gives a relative approximate environment location based on the current position of the robot. To find the absolute environment location the direction the robot is facing needs to be taken into account. The robot is equipped with a magnetic compass to determine North, South, East, and West direction inside the building. Given the relative output along with the robot direction heading we can determine the absolute position index shown in Figure 3. For example, the FIS output determines the robot has a wall in front of it. In order to get a detailed picture of the environment we need to be able to determine if the wall in front is North facing wall of the building (absolute position index 1), South facing wall (absolute position index 2), East facing wall (absolute position index 3), or West facing wall (absolute position index 4). In order to determine which absolute position index our robot is currently situated in, the FIS and direction are processed together. A map can easily be created to determine the location. Another benefit of the FIS system and the absolute location look up is a second level of protection against errors. It is possible that a sensor may give erroneous data or fail to produce any data. The wrong information passed into the FIS will produce incorrect interpretations. The robot will determine the appropriate location it thinks it has found and if the location is revisited it can recalculate its absolute position index by using superposition of the two erroneous locations. For example, consider a situation where the left ultra-sonic sensor breaks during operation. The robot is in absolute position index 10 facing south. Given that only the right sensor

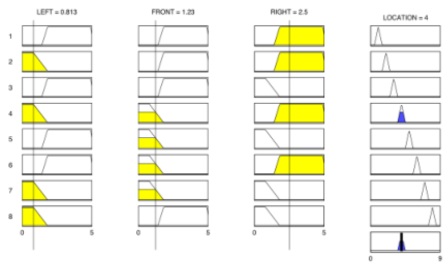

Figure 7. FIS example.

and front sensor are functioning the fuzzy system tells the robot that it only has a front wall so the robot infers it is in position index 5. The robot then makes a 180º turn and now the fuzzy system determines a right wall is present. Given that the location has been visited before we can sum up the given position indices and make a new determination that the absolute location is not absolute position index 5 but index 10. By having the direction element in our system, the robot has a built in redundancy element.

A. Simulation

A simulation was created to test the FIS for accuracy on a number of different building environments. The simulation looked at creating a maze to simulate the inside of a building. The maze was designed to model as a grid. The robot will move one grid cell at a time in north, south, east, or west direction. Once the robot is in a cell the robot will sample the ultra-sonic sensors. The ultra-sonic sensors were modeled in the simulation to resemble non-ideal operation. At each cell the simulation calculated the distance to a wall for each ultra-sonic sensor and adjusted the distance measurement by adding noise through a Gaussian distribution. The simulation showed that the FIS was always able to accurately generate an environment map of the maze. One item that was not examined in this research was how to efficiently explore the open maze surface. The simulation explored the maze by randomly picking a direction to move when the robot entered a new cell. The simulation resulted in a large number of the cells being revisited.

Figure 8. Sample building example used in simulation

Figure 8 shows one example of a maze generated for testing the FIS system. The robot generated two sets of data from the FIS output. The first set of data is an array that can be used to generate a visual map of the environment. Figure 9 displays the corresponding environment map array. Each number in the map array corresponds to one of the fifteen absolute position indices in Figure 3. It is possible for the robot to use the map array to aid in navigation. In addition to the map array, the FIS output was used to generate a graph of the maze. In the graph, every vertex is a grid cell and the edges show the adjacent grid cells that are unobstructed. Figure 10 show the corresponding graph for the example maze in Figure 8. With the robot generating a map from the FIS, graph theory can be applied to aid in navigation and determine characteristic of the building. The graph can be used to generate shortest path distance between any two spots in the building. The graph can also be used to determine critical path analysis. The critical path will determine the traveling bottlenecks in the building. The robot will also be able to determine which areas are rooms and hallways by applying walk, path and cycle analysis.

V. CONCLUSION

Autonomous robots have limitless potential applications and many commercial applications for them have been developed or currently are in development. In order for many autonomous robots to perform their goals effectively they will need to be able to create maps of their environment. There are many previously well established techniques, but this paper lays out a very flexible and robust technique. By using fuzzy

6	7	8	6	7
2	1	3	2	1
9	5	0	5	10
14	13	0	13	12
14	13	5	13	12

Figure 9. Environment map created from FIS. Numbers correspond to absolute location index.

logic, our robot is able to map out the environment with low precision sensors. Another advantage of the system is that the mapping can still be achieved even if a sensor becomes disabled during the robot's mission. Two important environment maps are created. The first map is a graphical representation that can easily be used by a human observer. The other map is a graph-based map that is well suited for the robot to use in navigation. By using basic graph theory the robot can determine the shortest path between two positions and determine other analyses of the building.

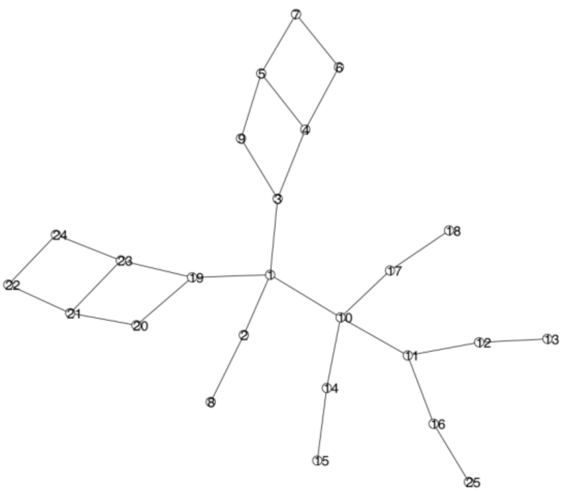

Figure 10. Environment map as a graph from example maze.

REFERENCES

[1] B. Kuipers, "The spatial semantic hierarch". *Artifical Intelligence*. Vol 119, 191-233, 2000.

[2] S. Thrun, D. Fox, and W. Burgard. "A probabilistic approach to concurrent mapping and localization for mobile robots". *Machine Learning*, Vol 31, 29-53, 1998

[3] S. Thrun, M. Montemerlo, D. Koller, B. Wegebreit, J. Nieto, and E. Nebot. "FastSLAM: An efficient solution to the simltaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, 2004.

[4] James Wolfer and Chad George, "Fuzzy logic control for robot maze tranversal: an undergraduate case study," *World Congress on Computer Science, Engineering, and Technology Education*, March 2006.

[5] Petru Rusu, Emil M. petriu, Thom E. Whalen, Aurel Cornell, and Hans J. W. Spoelder, "Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation," IEEE Transactions on Instrumentation and Measurements, Vol. 52, No. 4, 1335-1340 August 2003.