Pathfinding with Obstacle Avoidance Using Fuzzy Logic

Jelle van Dijk
Bachelor student Computer Science
University of Amsterdam
Student ID: 10989048

Gawan Dekker
Bachelor student Computer Science
University of Amsterdam
Student ID: 11025654

Victor Gladys
Bachelor student Computer Science
University of Amsterdam
Student ID: 10523626

I. Introduction

Self driving cars are being developed at almost every major car and tech company in the world right now. Although they are still relatively early in development, they are considered by many to be the future of personal transportation. To try and gain a better understanding of autonomous moving vehicles, in this project we will attempt to recreate a simple one using a fuzzy logic system (FLS).
The goal of this project is to create and design a FLS that can guide a robot through different environments without colliding with any obstacles. Because of limited time and recourses we will not implement this system for an actual vehicle or robot, instead, we will work with a simulation. Both the FLS and the simulation will be written in python. The environment and obstacles will be represented in a number of grids. The simulated robot will get a starting position in this grid and a position as the target to be moving towards.

II. Literature Review

There is already a lot of literature about trying to make a robot traverse an environment using fuzzy logic. In this section, we will briefly discuss three papers that are relevant to our problem.

A. Artificial Neural Fuzzy Logic Algorithm for Robot Path Finding (X. Bajrami; A. Dërmaku; N. Demaku)

In "Artificial Neural Fuzzy Logic Algorithm for Robot Path Finding" [1], X. Bajrami, A. Dërmaku and N. Demaku create and compare two fuzzy logic implementations to guide a robot from point A to point B in a simulated environment with obstacles.
The first implementation uses a combination of FLSs with mamdani-type inferrence and handmade rulebases and membership functions to guide the robot. The input (distances to closest objects and goal, angle towards goal and preferred turn) is used to determine acceleration levels for the left and right wheels of the robot for different objectives (like obstacle avoidance) and to determine different weights for those objectives.
The second implementation used an FLS with a Sugeno-type inference system, where the memberships functions were trained using a neuro-adaptive learning method.
After running tests in a simulated environment, the two implementations were compared.
The second implementation in the paper is not too relevant to our project, since we are working with mamdani-type inference, however the first implementation comes close to what we are implementing. We are using similar input, although (for the time being) we use one FLS with an incomplete rulebase to guide the robot.

B. A Novel Hybrid Fuzzy A Robot Navigation System for Target Pursuit and Obstacle Avoidance (A. P. Gerdelan; Dr. N. H. Reyes, Ph.D.)*

In "A Novel Hybrid Fuzzy A* Robot Navigation System for Target Pursuit and Obstacle Avoidance" [**?**], the researcher created a hybrid system to control a robot. The first layer of the system consisted of an A* algorithm. The A* algorithm is a path finding algorithm, it is considered to be very fast. The A* path finding layer calculated the optimal route to from the position of the robot to the end point, this route consisted of way points. The second layer of the system is a fuzzy logic system. This system had as input information about the next way point as well as about the nearest obstacle. With these pieces of information a speed and turning speed for the robot were chosen.
This paper used some techniques that can be very useful for our project. Since they used a moving end point in their simulations, their test scenarios where harder, however the same principles they used to control there robot will apply to the robot in the simulations used in our project.
A problem that the researchers in [**?**] had, was that in some situation the fuzzy logic system would try to avoid an obstacle and by doing so, would not be able to move to the next way point. This is something to keep in mind for our project, but could be fixed by tweaking the membership functions.

C. Adaptive two layer fuzzy control of a mobile robot system (M. Mohammadian; R. J. Stonier)

In "Adaptive two layer fuzzy control of a mobile robot system" [3], a genetic algorithm has been used to adapt fuzzy rules in a two layer fuzzy logic system, which is then used by two robots to navigate towards a target without colliding. The genetic learning has been applied to generate a new layer of fuzzy rules that can be integrated into an already existing rulebase.

The first of the two fuzzy layers is used to determine the angle at which to continue moving, whilst the second layer determines the speed of the robot. By encoding the rulebase into a bitstring, Mohammadian *et al.* were able to modify the rulebases with a genetic algorithm using cross-over and single-point mutations over a number of generations. To do this they used a modified cross-over procedure that ensured that these bitstrings were cut only at points that defined boundaries between rules.
The paper concludes in noting that, altough most of their tests resulted in a positive outcome, some had trouble at the corners of the tested driving areas. This suggests that genetic algorithms find a maximum in optimizing fuzzy systems that is not necessarily easily applicable to a change in, or extension of, the initial learning environment.

III. Proposed Approach

A. Data

Most of the data will be ~~personal knowledge~~. The first versions of membership functions and rules will be created ~~by us~~. When possible the newer versions of the membership functions and rules will be extrapolated from the test result of the previous evolution of the system.

B. Design

The general design of our system is still unknown at this point. As inputs we will use the distances to the nearest objects at several degrees of rotation and the distance and angle to the target location. As outputs, we will use accelleration and degree of rotation of the robot. However, these might all still be subject to change.

C. Implementation

The two most important parts, the simulation and the fuzzy logic system, we be created by us in python (exact algorithms are not yet known).

References

[1] X. Bajrami, A. Dërmaku, and N. Demaku, "Artificial neural fuzzy logic algorithm for robot path finding," *IFAC-PapersOnLine*, vol. 48, no. 24, pp. 123–127, 2015.

[2] A. P. Gerdelan and N. H. Reyes, "A novel hybrid fuzzy a* robot navigation system for target pursuit and obstacle avoidance," in *Proceedings of the First Korean-New Zealand Joint Workshop on Advance of Computational Intelligence Methods and Applications*, vol. 1, 2006, pp. 75–79.

[3] M. Mohammadian and R. J. Stonier, "Adaptive two layer fuzzy control of a mobile robot system," in *Evolutionary Computation, 1995., IEEE International Conference on*, vol. 1. IEEE, 1995, p. 204.