Fuzzy Logic for Mobile Robot Navigation Applications

Ben Church – 10006197

**Abstract:**

Contents

[Introduction 1](#_Toc478558323)

[Motivation 1](#_Toc478558324)

[Scope 1](#_Toc478558325)

[Background 2](#_Toc478558326)

[Perception 2](#_Toc478558327)

# Introduction

## Motivation

Mobile robot navigation is a broad field with around a century of invested research dating back to early autopilot systems, depending on what exactly one considers as the first mobile robot with navigation capabilities. We now profit from many applications using this technology such as scientific exploration with rovers, autonomous warehouses, and modern autopilot systems. The self-driving car is an example of how mobile robot navigation technology could impact many private lives, if present trends in the industry continue. While some may find the scope of its applications inspiring, the number of problems in mobile robot navigation may be daunting to potential innovators. This paper is written to the end of bridging the gap between user and designer, exploring mobile robot navigation functionality based on common-sense human experience, rather than various abstract mathematical treatments of the navigation problem removed from human experience.

Fuzzy logic applications are chosen as the focus of this paper because the author believes that mobile robot technologies stand to benefit from integration of a fuzzy rule-based programming interface, suitable for programming by average consumers. Simplified mobile robot customization through a fuzzy programming language consisting of perhaps several verbs, measurements, logical operators, and qualitative magnitudes, could offer typical consumers more sophisticated or better tuned robot behavior than technically knowledgeable professionals can pre-program. The author believes further that a guide matching mobile robot navigation problems with fuzzy logic solutions could be informative in designing such a fuzzy programming interface.

## Scope

While greater user freedom through simple programming is desirable, the designer may not wish to make all functionality open for customization. For example, drone hover stability controls are unlikely to improve with user tuning, whereas a user may wish to change the relationship between their robot’s velocity and the distance it follows the user from. The particular robot functionality made customizable through a fuzzy programming interface will always depend on the total functionality available for customization, determined by the application, and individual designer choices. This work is not meant to indicate when functionality ought to be implemented with fuzzy logic or be made user-programmable, much less to explore fuzzy navigation solutions in an exhaustive manner. It is instead intended as a reference, a toolbox, for designers who wish to implement navigation functionality with fuzzy logic, with consideration given to how such functionality could be made user-programmable.

To this end, the Background section provides an overview of mobile robot navigation, while it is assumed that the reader is familiar with fuzzy logic. The overview explores navigation models, behaviors, and problems from literature. It serves to introduce concepts explored in the context of fuzzy logic by works considered throughout the rest of the paper, and to inspire the organization of our fuzzy programming framework. The programming framework is then presented in the Proposed Framework section, where its organization is described in reference to the conceptions of mobile robot navigation explored in the Background section. Subsequent sections explore works which deal with various aspects of navigation using fuzzy logic, and their approaches are conveyed in our proposed framework.

## Background

We can consider mobile robots as engineering systems like any other. A set of measurements taken from the environment are the inputs which determine an output, a chosen action. To achieve useful, sustained navigation behavior, this systemic conception of robot behavior may be incorporated into a closed-loop control system. The closed-loop aspect of the system means that the robot determines subsequent action based on how the relationship between itself and the environment has changed as a result of previous behavior. Arkin represents behavior using a cognitive-inspired schema flowchart [Arkin1987], which amounts to a closed-loop control system using processes and information familiar to human experience. Figure 1 shows a general action-perception cycle, extended to show perception as a process.

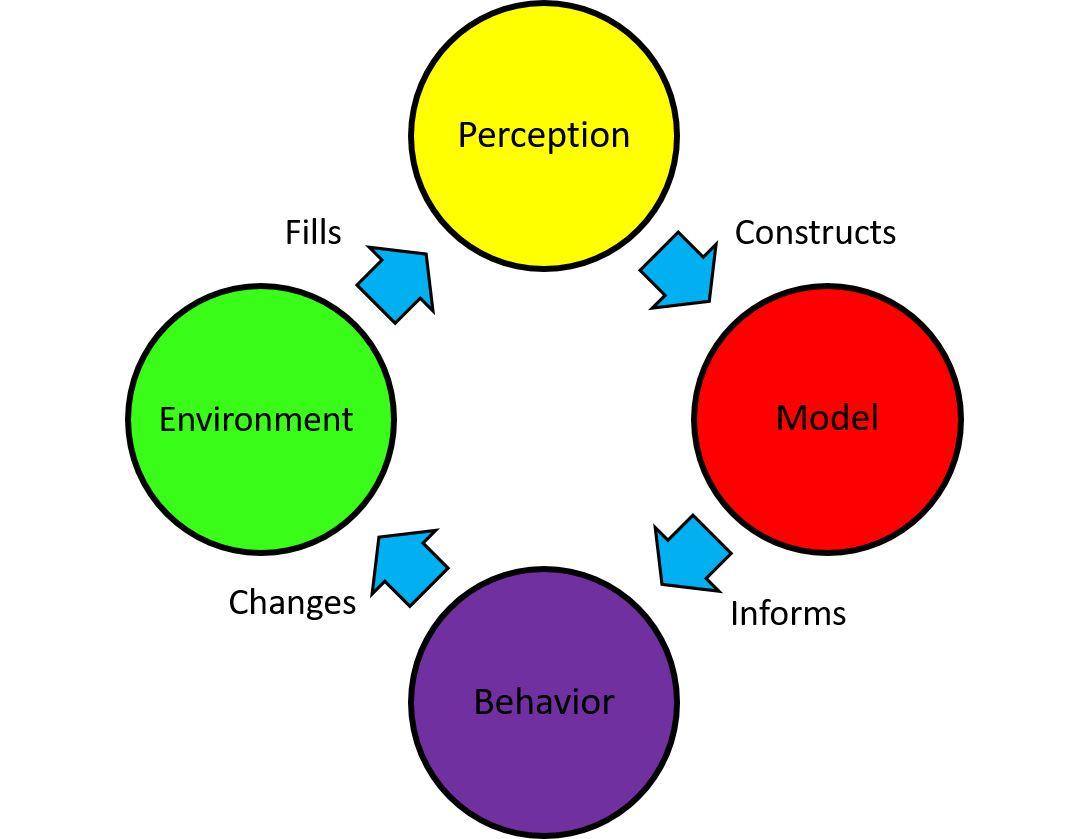


Figure : Action-perception cycle for environmental interaction

Navigation may be seen as a special case of an action-perception cycle where the changes caused by behavior are limited to changes in the robot’s spatial relationship with the environment. Many fuzzy logic solutions compartmentalize the navigation problem into such processes, more or less explicitly, therefore the proposed programming framework should as well.

The environment itself exists independently of the robot, and therefore cannot be incorporated into a navigation scheme, fuzzy or otherwise. This leaves three processes in the cycle for exploration. However, the distinction between perception and the internal model of the environment is not always clear; at which point is information taken from the environment considered a model thereof? For the purpose of organization, we consider perception as sensor readings which have let to be considered in context with other readings. The environmental model begins when perception data is integrated and taken in context. Solutions are more likely to fall on under modelling than perception because of how they are here classified, however both may be considered. This leaves the behavior component, which has received much attention in research due to its breadth, difficulty, and the natural suitability of fuzzy logic to its problems. We attempt to organize fuzzy solutions into one of these three components of navigation.

Since robots are designed based on desired functionality, and robot functionality can be described in relatable terms such as “exploration” or “destination seeking”, various behaviors are defined to implement the functionality. Behaviors, in this sense, are the building blocks of navigation as functionality. They are likewise described in recognizable language such as “wall-following” or “obstacle avoidance”, allowing the designer to abstract away the physical details of their implementation, such as how the motors should turn. This treatment of navigation as behavior was used by authors such as Brooks [Brooks1986], and Rosenblatt and Payton [Rosenblatt1989], to address problems such as behavior selection or arbitration. The programming framework should be able to treat each of these levels of behavior programming: Selecting behaviors sufficient to implement the desired functionality, determining how to obtain a single, actionable output from these multiple behaviors, and how to implement the different behaviors based on the environmental model.

## Proposed Framework

The conception of navigation functionality being the product of three programmable processes, perception, modelling, and navigation, form the highest layer of organization for the proposed programming framework.

Fuzzy rule-based navigation schemes can be represented as a set of sets of fuzzy implication relations or rules, {{R11, R12, …, R1n}, … {Rm1, Rm2, …, Rmp}} of the form:

# Perception

# References

[Arkin1989] - R. Arkin, “Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior”, IEEE International Conference on Robotics and Automation, 1989.

[Brooks1986] - R. Brooks, “A Robust Layered Control System for a Mobile Robot”, IEEE Journal of Robotics and Automation, vol. RA-2. pp. 14-23.

[Rosenblatt1989] - J. Rosenblatt, and D. Payton, “A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control”, IEEE International Joint Conference on Neural Networks, 1989.