Fuzzy Logic for Mobile Robot Navigation Applications

Ben Church – 10006197

**Abstract:**

Contents

[1. Introduction 1](#_Toc479602896)

[i. Motivation 1](#_Toc479602897)

[ii. Scope 1](#_Toc479602898)

[iii. Background 2](#_Toc479602899)

[iv. Proposed Framework 3](#_Toc479602900)

[2. Perception 4](#_Toc479602901)

[v. Measurement type 4](#_Toc479602902)

[3. References 5](#_Toc479602903)

# 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 introduces navigation models, behaviors, and problems from literature. It serves to define 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 process. 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 include perception as a process.

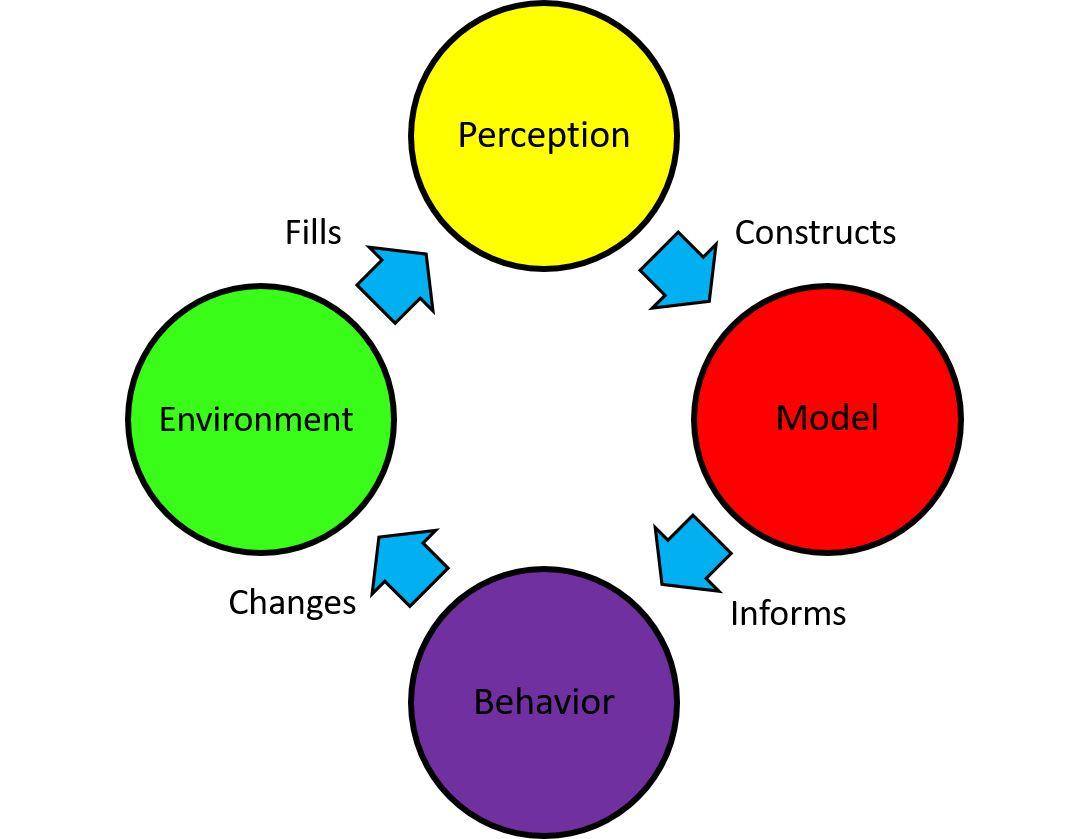


Figure 1: Action-perception cycle for environmental interaction

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 do so 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 immediately 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 not to be interpreted in the context of other readings. The environmental model begins when perception data is integrated and taken in context. A robot with a single sensor constitutes a special case; where no measurement context is possible, the robot’s perception is equivalent to its model of the environment. Solutions are more likely to fall under modelling than perception because of how they are classified here, however application for 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.

The modelling component of navigation may be further divided into producing features from environmental measurements the robot perceives, and combining features into the robot’s complete environmental model. The distinction here between features and models is similar to the distinction between the perception and modelling processes; a feature is homogeneous in its dimensions, which may be quite abstract, while the model is the result of the synthesis of all features under consideration. For example, multiple adjacent range sensor readings can be perceived and grouped into an obstacle feature. The robot may construct multiple, separate obstacle features from additional sensor readings. A model would be the combination of these obstacle features. It is from this model that behavior is determined.

Robot control schemes are designed based on desired functionality, and functionality can be described in relatable terms such as “exploration” or “destination seeking”. Multiple behaviors may be made available to the robot to implement a given functionality. Behaviors can likewise be described in recognizable language such as “wall-following” or “obstacle avoidance”. This use of behaviors breaks the higher level functionality down into components which can be implemented more easily since they are smaller, less abstract tasks. Working in the other direction, behaviors can be combined into new ways to produce custom functionality, providing a level of abstraction above hardware action for programming. This treatment of navigation as behaviors 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: Which behaviors when made available to the robot are sufficient, in some combination, to implement the desired functionality, and determining how to obtain a single, actionable output from these multiple behaviors.

## Proposed Framework

To the end of producing a useful guide to implementing fuzzy mobile robot navigation, the solutions and approaches surveyed in this paper are categorized under perception, modelling, or behavior processes. Solutions are further categorized into components of these processes because of the natural modularity of the processes’ building blocks. This provides a convenient organization scheme which presents solutions as they relate to the particular aspect of navigation which the designer wishes to make available to the user for fuzzy programming. This organization scheme is shown in Figure 2.

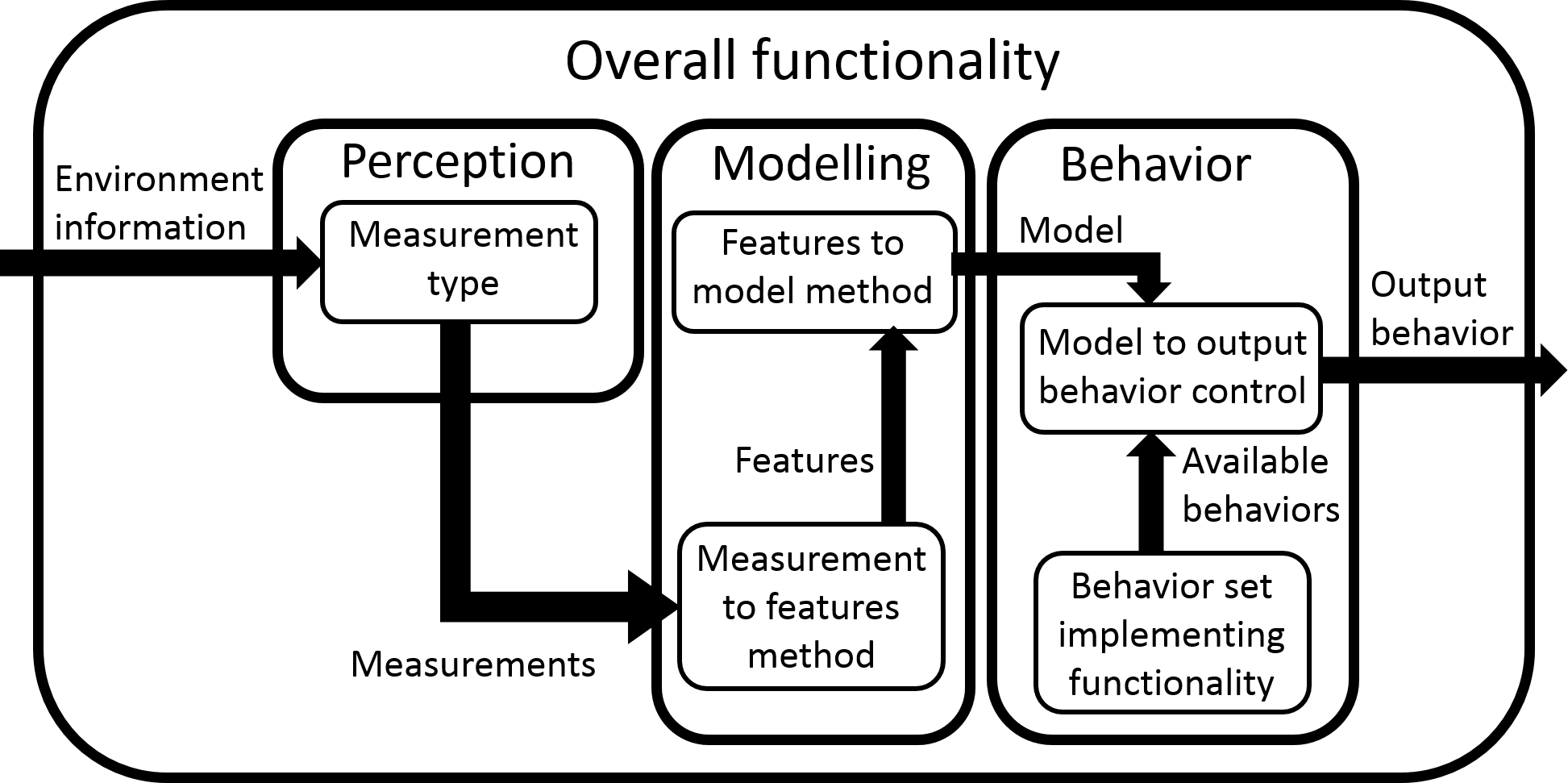


Figure 2: Topic categorization scheme, organized as a robot navigation process model

This is essentially a more detailed navigation model than shown in Figure 1. The process in Figure 2 is still closed loop, when the omitted environment is considered. The environment relates output behavior to environmental change relative to the robot, which is then perceived. The rounded rectangles in Figure 2 represent navigation process components which may be implemented in a fuzzy manner, and therefore surveyed here. The arrows represent data flow from between modules. Individual works often present solutions or approaches to multiple components of this framework. This is because a functional navigation implementation requires multiple components, conceptualized in this framework or another, and authors are typically providing insufficient information if they address only one such topic.

# Perception

Works which employ a fuzzy treatment of perception separately from modelling are somewhat rare because of how the two are distinguished in the organization scheme. Because of our definition of perception pertains to measurements of environmental information, it consists of only one component, the type of measurement performed. In fact fuzzy perception is little more than the determination of membership set values using a physical measurement as the crisp input. For the similarity of this topic to fundamental fuzzy logic theory, it is treated briefly. Using the resulting fuzzy set memberships, whatever measurement they fuzzyify, is treated in the Modelling Section.

## Measurement type

The measurement type is characterized by the physical quantity which is being measured. For mobile robot navigation, this is almost invariably a linear range, or angular heading. Exceptions exist, usually in a non-fuzzy context, such as Xu *et al.* [Xu2011] and Memon *et al.* [Memon2016] who used infrared thermal sensors to detect fires with mobile robots. Since navigation is often implemented with multiple range sensors and a single GPS for perception, fuzzy treatment of angular headings exclusively at the perception level are more common than treatment of range at this level.

### Range

Perhaps because it is one of the earliest papers on fuzzy mobile robot navigation, Sugeno and Nishida [Sugeno1985] provide some treatment of perception. They designed a control scheme to navigate a robotic model car through a crank-shaped course. They represent the relationship of the robot to the course with a set of measurements, which are input to fuzzy navigation rules before any context is inferred. This may have been sufficient for their application given the particular course layout, shown in Figure 3, and the feasibility of constructing navigation rules directly from physical measurements, also shown. The linear measurements determine the membership values of corresponding fuzzy size sets, Small, Medium or Large.

Figure 3: Model of the arena used by Sugeno and Nishida [Sugeno1985]

### Heading

Sugeno and Nishida’s [Sugeno1985] treatment of the x3 angular measurement was the same as the linear distances; its corresponding fuzzy sets were Out, Forward, and In. It is not discussed further because of its similarity to the range measurements.

# Modelling

## Measurement to Features

## Features to Model

# Behavior

# 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, 1986.

[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.

[Xu2011] - H. Xu, H. Chen, C. Cai, X. Guo, J. Fang, and Z. Sun, “Design and Implementation of Mobile Robot Remote Fire Alarm System”, International Conference on Intelligence Science and Information Engineering, 2011.

[Memon2016] - S.F. Memon, I.H. Kalwar, I. Grout, E. Lewis, and Y.N. Panhwar, “Prototype for Localization of Multiple Fire Detecting Robots in a Dynamic Environement”, International Conference on Computing for Sustainable Global Development, 2016.

[Sugeno1985] - M. Sugeno, and M. Nisida, “Fuzzy Control of Model Car”, Fuzzy Sets and Systems, vol. 16 pp. 103-113, 1985.