Path Following and Obstacle Avoidance Fuzzy Controller for Mobile Indoor Robots

Mousa AL-Akhras, Maha Saadeh, Emad AL-Mashakbeh
Computer Information Systems Department
King Abdullah II School for Information Technology
The University of Jordan
Amman 11942, JORDAN
mousa.akhras@ju.edu.jo, saadeh.maha@yahoo.com, e.mashakbeh@gmail.com

Heba Saadeh
Department of Medical and Molecular Genetics
Medicine School
King’s College London, Guy’s Hospital
London, SE1 9RT, UK
heba.saadeh@kcl.ac.uk

Abstract: This paper provides the design and implementation details of a remotely controlled, battery-powered robot that is able to follow a pre-defined path and to avoid obstacles. The proposed system is divided into three modules for path following, path searching and decision making. Path following controller is responsible for following a pre-defined path, path searching is responsible for returning to that path when the robot leaves that path to avoid an obstacle. The third module is responsible for deciding which of the first two modules to follow. Fuzzy Logic was employed in the proposed system to take the robot’s light and ultrasonic sensory data as input and send commands to the robot motors to control the robot’s velocity and steering angel. Lego Mindstorms NXT robot was used to realise the proposed design.

Key–Words: Robot, Fuzzy Logic, Lego Mindstorms NXT, Obstacle Avoidance, Path Following.

1 Introduction

The aim of this paper is to build a controlling system for a mobile indoor robot using Fuzzy Logic (FL). The purpose of this controlling system is to enable a robot to follow a pre-defined path (like the one that leads visitors to different parts of a certain establishment like a hospital) and to avoid any obstacle the robot may face in its way. This scenario is adopted in this paper as depicted in Figure 1.

![Figure 1: General Fuzzy Controlled Robot System.](image)

The biggest challenge that has motivated us to use FL is to enable smooth movement for the robot which is a major factor that motivated researchers to use FL in controlling problems. The robot shown in Figure 1 is provided with 3 light sensors and one ultrasonic sensor that follows a certain path. This robot is faced with an obstacle and it avoids it and returns back to its path. The design details of this robot is provided in the rest of the paper. It was decided that a robot control scheme can be used to process sensory information and provide control action decisions. Data is collected via an ultrasonic sensor (also infra red sensor can be used) and light sensors and then transmitted to the brain where all of the calculations are taken place. These data have to be processed in the brain of the robot using the Fuzzy Inference System (FIS) and then out to the motors which in turn perform robot drive and steering commands. The FL controller consists of a rule-base that fuzzifies range values and produces defuzzified control actions as shown in 1. The control actions are crisp values that represent the robot’s velocity and steering directions. Any information about the robot’s environment is provided solely by the attached sensors. No feedback of kinematic or dynamic information (such as position and velocity) is employed. The use of such basic sensors allows us to examine the power of FIS when constrained by minimal observability. From a practical point of view, this constraint is realistic relative to most autonomous mobile systems with limited computational resources. It has been demonstrated that
such systems exhibit satisfactory performance when utilising reactive and/or behavioral control strategies. The rest of paper is organised as follows: section 2 reviews the related literature. Section 3 gives full details of the proposed system. Section 4 concludes the paper and presents for future work.

2 Literature Review

In this section the needed literature is reviewed. Section 2.1 reviews several attempts for path following and obstacle avoidance. Section 2.2 gives the reader background information on fuzzy logic. Section 2.3 introduces Lego Mindstorms NXT.

2.1 Path Following and Obstacle Avoidance

Pires and Nunes (1) presented an obstacle avoidance, behaviour-based architecture for a Reactive Shared-Control (RSC) system designed as a FL controller. The main aim was to provide easier and safer robotic wheelchair navigation. An improvement to the previous architecture was introduced by Bento et al. (2) through combining a FL implementing path tracking behaviour with the obstacle avoidance behaviour to enhance the navigation of the wheelchair.

To overcome the uncertainly of the unknown and dynamic robot behaviour, an adaptive robust fuzzy controller was proposed by Giap et al. (3), the Lyapunov function theory was used to guarantee the stability of the tracking errors. A more sophisticated and intelligent path following method based on human driving behaviour for mobile robots with nonholonomic constraints was proposed by Liu (4). The method suggests the use of two FL controllers: Robot Position Controller (RPC) and Robot Orientating Controller (ROC). The values of the position error and orientation error determine which fuzzy controller is used to control the movement of the robot.

2.2 Fuzzy Logic

Fuzzy Logic (FL) was introduced by Lotfi Zadeh in his 1965’s seminal paper (5). FL provides a method of reducing as well as explaining the system’s complexity and it is used for controlling purposes. Fuzzy sets are functions that map an input value, which might be a member of a set, to a number between zero and one (not only zero or one like in the crisp logic), indicating its degree of membership. A degree of zero means that the value is not in the set, and a degree of one means that the value is completely represented in the set. Intermediate values indicate partial membership in the set.

Considering the example of determining the car’s speed based on its distance from someone’s home. The basic three processing steps of fuzzy logic are explained as follows:

- **Fuzzification**: The translation of real-world concepts (values) to fuzzy world values using user-defined membership functions. For example the distance of over 100 km is translated into the value of 1 (Far). Distance less than 1 km is translated into the value of 0 (NOT Far). A distance between 1 km and 100 km is translated into a value between 0 and 1. The same is applied to the current speed which may be slow, medium, or fast.

- **Rule Evaluation**: Apply a set of rules like: If SPEED=SLOW and HOME=FAR then GAS=INCREASE.

- **Defuzzification**: After computing the fuzzy rules and evaluate the fuzzy variables, fuzzy concepts need to be translated back to the real-world. A membership function is needed for each output variable. For example Gas=Increased corresponds to applying 10% increase to the current gas level.

Several researchers have elaborated on the work of Zadeh, notably the fuzzy logic controller introduced by Mamdani (6). Since its introduction, fuzzy logic control method has been successfully applied in enormous number of control problems, e.g., (7) (8).

2.3 Lego Mindstorms NXT

Lego Mindstorms NXT is a programmable robotics kit released by Lego® in 2006 (9). It replaced the first-generation Lego Mindstorms, which was called the Robotics Invention System. It comes with a special programming environment called the NXT-G programming software, although other unofficial languages are also exist, such as Not eXactly C (NXC) which is a simplified version of C.

3 The Proposed System

This sections gives details about the proposed system. Section 3.1 explains the general framework of the sys-
tem. Section 3.2 describes the used hardware design. Section 3.3 describes the system modules.

3.1 Fuzzy Inference Controller

In this paper the design for a path following and obstacle avoidance controller system using FL is proposed. The general flow of the system is shown in Figure 2. The robot reads the sensory data provided through the light and the ultrasonic sensors and sends commands to the motors to follow the path and to avoid the obstacles.

Figure 2: System Flowchart.

The proposed design consists of three modules as illustrated in Figure 3, these modules are: Path Following Controller (PFC), Path Searching (PS) and Decision. The PFC module uses two-stages FIS to control the robot’s movements. Consequently, the robot can move freely and smoothly on a pre-defined path. During its movements the robot can also detect and avoid any obstacle on the way. When the robot moves far away from the path, to avoid a certain obstacle, the PS module is then used (called). It allows the robot to return back to the correct path. The final decision which controls the direction and velocity of the robot motors will be taken by the decision module depending on whether there is an obstacle on the robot way or not. The relation among these three modules is illustrated in Figure 3. Further detailed explanation for each module will be presented shortly.

Figure 3: System Structure.

3.2 Robot Hardware Design

The hardware design for the proposed system is illustrated in Figure 4. The design includes three light sensors used to track the path, one ultrasonic sensor to detect any obstacle on the path, and three motors. One motor is used to control the steering angle and two back motors are used to control the robot’s velocity.

Figure 4: Hardware Design.

3.3 System Modules

In this section, the design details of the three system modules are given.

3.3.1 Module One: Path Following Controller

Path Following Controller (PFC) module controls the movement of the robot over a pre-defined path. As shown in Figure 3, PFC module consists of a two-stage fuzzy system in which the inputs of the second
stage depend on the output of the first one. The detailed structure of this module is depicted in Figure 5, including the input and the output parameters of each stage.

![Figure 5: Path Following Controller Module.](image)

The first stage controls the steering angle that should be taken to keep the robot on the correct path. This inference system has three inputs and one output. The input parameters are the values read by the Left light Sensor (LS), Centre light Sensor (CS) and Right light Sensor (RS). Each input has three membership functions; white, gray and black that are defined in Figure 6 (a). Since we use Lego Mindstorms NXT light sensor each membership function lies in the range 0 to 1000.

Mamdani inference system (6) is used in this stage. First, each input value is converted into a fuzzy value that determines the degree to which it belongs to each fuzzy set according to the membership function representing that set (i.e. white, gray and black). This process is called fuzzification. The three inputs’ fuzzy values are then evaluated according to the fuzzy rules listed in Table 1. In this step the values are applied to the rules antecedents. The minimum fuzzy value will be chosen since AND operator is used. The results from antecedent evaluation are applied to the consequent part of the rules. Then the membership functions of all rules consequent are combined into single fuzzy set in rules aggregation step. This set is used in the defuzzification step to obtain the final output. The output from this stage is the Steering Angle (SA). This angle can have the same value passed from the first stage or an adjusted value according to obstacles’ presence. The state linguistic variable is used to trigger the PS module. It has two values: Clear and Not Clear. Clear means that the robot is moving on the correct path and there is no obstacle to avoid. Not Clear means that the robot moves away from the correct path to avoid an obstacle. The input and output variables membership functions for this stage are shown in Figure 7. The fuzzy rules are listed in Table 2.

### 3.3.2 Module Two: Path Searching

When PFC module output state is Not Clear, PS module will be activated; which means that the robot moved away from the correct path to avoid a detected obstacle. The goal of this module is to put the robot back on the correct path. The implementation of PS module depends on the testing environment including the robot’s size and the obstacle’s size. Figure 8 illustrates the flowchart of this module.

### 3.3.3 Module three: Decision Module

The decision module, shown in Figure 9, is used to decide which output should be sent to the robot motor according to PFC module state.

### 4 Conclusions and Future Work

In this paper we propose the design and implementation for a path following and obstacle avoidance con-
controller robotic system. The controller uses fuzzy logic to control the robot’s movement on a pre-defined path. The design consists of three modules: Path Following Controller (PFC), Path Searching (PS) and decision module. As a future work we will implement this design on other scenarios and evaluate its effectiveness in following complicated paths and multiple obstacles using Lego Mindstorms NXT robot kit.

References:


Table 2: Second Stage Fuzzy Rules.

1- If SA is Leftmost and OD is Near then LM is Medium and RM is Medium and SA is Right and State is Not-clear.
2- If SA is Left and OD is Near then LM is Slow and RM is Slow and SA is Leftmost and State is Not-clear.
3- If SA is Forward and OD is Near then LM is Medium and RM is Medium and SA is Left and State is Not-clear.
4- If SA is Right and OD is Near then LM is Slow and RM is Slow and SA is Rightmost and State is Not-clear.
5- If SA is Rightmost and OD is Near then LM is Medium and RM is Medium and SA is Left and State is Not-clear.
6- If SA is Leftmost and OD is Far then LM is Slow and RM is Slow and SA is Leftmost and State is clear.
7- If SA is Left and OD is Far then LM is Medium and RM is Medium and SA is Left and State is clear.
8- If SA is Forward and OD is Far then LM is Fast and RM is Fast and SA is Forward and State is clear.
9- If SA is Right and OD is Far then LM is Medium and RM is Medium and SA is Right and State is clear.
10- If SA is Rightmost and OD is Far then LM is Slow and RM is Slow and SA is Rightmost and State is clear.


