A fuzzy-logic controller for an autonomous vehicle operation in an unknown environment

William E McCarthy
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds
Part of the Higher Education Commons

Repository Citation
https://digitalscholarship.unlv.edu/rtds/358

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A fuzzy-logic controller for an autonomous vehicle operation in an unknown environment

McCarthy, William E., M.S.

University of Nevada, Las Vegas, 1994
A FUZZY-LOGIC CONTROLLER FOR AN AUTONOMOUS VEHICLE OPERATING IN AN UNKNOWN ENVIRONMENT

by

William E. Mc Carthy

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Department of Mechanical Engineering
University of Nevada, Las Vegas
May 1994
The thesis of William E. Mc Carthy for the degree of Master of Science in Mechanical Engineering is approved.

Chairperson, Mohamed B. E. Trabia, Ph.D

Examiner Committee Member, Brendan J. O'Toole, Ph.D

Examiner Committee Member, Woosoon Yim, Ph.D

Graduate Faculty Representative, Shashi K. Sathisan, Ph.D

Graduate Dean, Ronald W. Smith, Ph.D

University of Nevada, Las Vegas
May 1994
ABSTRACT

A controller is developed to guide a four-wheeled vehicle through an unknown environment. The vehicle is equipped with an ultrasonic sensor that can rotate to survey the neighboring environment. A new path planning algorithm is developed that reduces the computational time while avoiding obstacles. The vehicle uses a fuzzy-logic controller to determine the corresponding change in steering. While fuzzy-logic controllers exhibit robustness under varying operating conditions, it is difficult to design a good controller when observations about the system are scarce or when the system has a large number of inputs and outputs. Due to this fact, the performance of the fuzzy-logic controller is improved using nonlinear programming techniques. The algorithm automatically generates the fuzzy rules and redefines the shape of the membership sets of input and output variables for an optimal performance of the controller. The effects of changing: the velocity of the vehicle, the range of the ultrasonic sensor, and the time step of the controller of the autonomous vehicle are discussed.
# TABLE OF CONTENTS

ABSTRACT ................................................................................................................................iii

LIST OF FIGURES .............................................................................................................. vi

LIST OF TABLES ................................................................................................................vii

ACKNOWLEDGEMENTS ............................................................................................. viii

CHAPTER 1  INTRODUCTION

1.1 General Background ..........................................................................................1

1.2 Literature Survey ...............................................................................................2

1.3 Project Objectives ...............................................................................................5

CHAPTER 2 MODELING OF THE VEHICLE

2.1 Assumptions .......................................................................................................8

2.2 Vehicle and System Variables .....................................................................8

2.3 Calculation of the New Position of the Vehicle ...................................10

CHAPTER 3  PATH PLANNING AND OBSTACLE AVOIDANCE

3.1 Path Planning Assumptions .........................................................................14

3.2 Ultrasonic Sensor ............................................................................................14

3.3 Path Planning Algorithm ..............................................................................16

3.3.1 Solutions to Procedure One .........................................................19

3.3.2 Solutions to Procedure Two ......................................................22

3.3.2 Solutions to Procedure Three ...................................................26

CHAPTER 4 FUZZY-LOGIC CONTROL

4.1 Fuzzy-Logic .......................................................................................................33

4.2 Optimizing of Fuzzy-Logic ........................................................................36

4.3 Automatic Generation of Fuzzy Inference Rules .......................................37

4.4 Selection of Membership Sets ..................................................................41

4.5 Application of the Optimization Algorithm .........................................44

4.6 Optimization Results ......................................................................................47

CHAPTER 5  PATH PLANNING RESULTS

5.1 Application of the Control System ..........................................................50

5.1.1 AGV in a Cluttered Environment ...........................................51

5.1.2 AGV in a Dead-End Alley .........................................................55
5.1.3 AGV in an Environment with Multiple Obstacles . . . 58
5.2 AGV Trials with Different Environments ....................... 61

CHAPTER 6 CONCLUSION ........................................... 66

REFERENCES .......................................................... 69
# LIST OF FIGURES

1. Variables of a Four Wheeled Vehicle ......................................................9
2. Calculation of the Instantaneous Center of Rotation ..............................11
3. New Position of the Vehicle .................................................................12
4. A Vehicle with an Ultrasonic Sensor Mounted on its Front ....................15
5. Sensed Region Cone .............................................................................16
6. Block Diagram of the Path Planning Algorithm ......................................18
7. Possible Solution to Procedure 1, case i ...............................................20
8. Possible Solution to Procedure 1, case ii ...............................................21
9. Possible Solution to Procedure 1, case iii .............................................22
10. Possible Solution to Procedure 2, case i .............................................23
11. Procedure 2 Sensing a Cluttered Environment .......................................25
12. All Possible Solutions to Procedure 3 ..................................................27
13. Possible Solution to Procedure 3, case i .............................................28
14. Possible Solution to Procedure 3, case ii .............................................29
15. Possible Solution to Procedure 3, case iii .............................................31
16. Possible Solution to Procedure 3, case iv ............................................32
17. Construction of the Imaginary Wall .....................................................32
18. A System using a Fuzzy Logic Controller ............................................33
19. Membership Sets for a Variable .............................................................34
20. Defuzzification Area .............................................................................36
21. A Modified Hooke-Jeeves Nonlinear Programming Algorithm ............43
22. Diagram of the Variables for an Autonomous Vehicle in an Unknown Environment without Obstacles ..................................................44
23. Controller for an Autonomous Vehicle in an Unknown Environment without Obstacles ..................................................45
24. Path of the Autonomous Vehicle Using Three, Five, and Seven Membership Sets ..................................................47
25. Path of the Autonomous Vehicle Using Seven Membership Sets ..........48
26. Select Vehicle Paths in a Cluttered Environment ....................................54
27. Select Vehicle Paths in an Environment with a Dead-End Alley ............57
28. Select Vehicle Paths in an Environment with Multiple Obstacles ..........60
29. Vehicle Path with Variable Vehicle Starting Orientation with an Obstacle ..................................................62
30. Vehicle Path with Variable Vehicle Starting Orientation with Multiple Obstacles ..................................................63
31. Vehicle Path with Variable Vehicle Starting Orientation with a Series of Long Slender Obstacles ..................................................64
32. Vehicle Path with Variable Vehicle Starting Orientation with a Complex Dead-End Alley ..................................................65
LIST OF TABLES

I. Possible Numerical Weights for Some Membership Sets ............ 37
II. Evaluation of the Extreme Response for a Single Input,
    Two Output System Described using Seven Membership Sets ....... 39
III. Preliminary Rules for a Single Input, Two Output System
     Described using Seven Membership Sets ................................. 40
IV. Rules for a Single Input, Two Output System
    Described using Seven Membership Sets ................................. 41
V. Membership Sets for the Correction of the Steering Angle ($\Delta \alpha$) ...... 48
VI. Membership Sets for the Steering Angle and
    the Orientation Error ($\alpha$, $\Delta \phi$) ........................................ 49
VII. Traversal Time in a Cluttered Environment, case (a) .............. 52
VIII. Traversal Time in a Cluttered Environment, case (b) .............. 53
IX. Traversal Time in a Cluttered Environment, case (c) ............... 53
X. Traversal Time in a Cluttered Environment, case (d) ............... 54
XI. Traversal Time in an Environment with a
    Dead-End Alley, case (a) .......................................................... 55
XII. Traversal Time in an Environment with a
     Dead-End Alley, case (b) .......................................................... 56
XIII. Traversal Time in an Environment with a
      Dead-End Alley, case (c) .......................................................... 56
XIV. Traversal Time in an Environment with a
     Dead-End Alley, case (d) .......................................................... 57
XV. Traversal Time in an Environment with
     Multiple Obstacles, case (a) ..................................................... 58
XVI. Traversal Time in an Environment with
     Multiple Obstacles, case (b) ..................................................... 59
XVII. Traversal Time in an Environment with
      Multiple Obstacles, case (c) .................................................... 59
XVIII. Traversal Time in an Environment with
       Multiple Obstacles, case (d) ................................................. 60
ACKNOWLEDGEMENT

I would first like to thank my parents, Walter and Mary Ann, for giving me the support that I needed to finish my graduate studies. I would like to express my deepest and most sincere appreciation to Mohamed Trabia, whom it is an honor to have as friend, for having the patience and dedication to guide me through my thesis work. And not to be forgotten is the faculty and staff of the Department of Mechanical Engineering for doing an excellent job of providing me with a great education.
Chapter 1
Introduction

1.1 GENERAL BACKGROUND

A robot can be defined as an automatic apparatus or a device that preforms tasks ordinarily ascribed to human beings, and exhibits a certain degree of autonomy. Research in the area of robotics has been going on since the invention of the computer, due to the increasing demand for automation. As major innovations continue in many areas such as computer systems, robotic programming languages, trajectory planning, compliant devices, and vision systems, the number of robotic systems and the intelligence level of the robots will increase.

Robots can be divided into: stationary and mobile robots. The stationary robots have a fixed base, whereas mobile robots have the capability to move there base location. Vehicles also can be categorized into those that are manually controlled and those that are autonomous. A manually controlled mobile robot uses a direct interface with a user in which all decisions are made by the user. In the field of autonomous control, the autonomous guided vehicle (AGV) is an area of particular interest. An AGV has clear potential advantages over manned vehicles in many different areas where human life may be endangered such as exploration and rescue missions in a hazardous material spill. This particular area
may arguably be described as one that contains some of the most challenging problems in the robotics science.

The environment of an autonomous robot may be fully known or unknown. The type of environment has a large influence on the complexity of the controller. The more that is known about the environment, the simpler the controller will be. An example of a known environment is that of a industrial factory floor, where robots move about from machine to machine. A planetary explorer is an application that would involve an unknown environment. Research in the area of an AGV comprise: vehicle modelling, path planning, obstacle detection and design of controllers.

1.2 LITERATURE SURVEY

Shiller and Gwo (1991) describe, among many other topics, a method of modelling the motion of a traditional four-wheeled vehicle. The motion of the vehicle is described using an instantaneous center of rotation and geometric identities to solve for the new position of the vehicle. A three-wheeled vehicle (i.e. tricycle) is used by Zhoa and BeMent (1992). The dynamic model they developed uses one wheel to steer the vehicle and the other two wheels are used to the drive the vehicle. The model was used to prove that simple controllers (i.e. systems that neglect the suspension and wheel effects) can be used with highly accurate results.

A third type of autonomous vehicle is the omnidirectional vehicle. Campion, Bastin and Novel (1993) described the modelling of a three wheel robot that has
the ability to change the orientation of each wheel independently. An omnidirectional robot is able to make abrupt direction changes when moving slowly due to the configuration of the wheels and motors that drive the wheels.

The area of object recognition is of great importance to any robot. Sound-wave (RADAR, SONAR, Ultrasonic sensors, etc...) and vision (cameras, fiber optics, etc...) systems are the two major object recognition devices. Vision systems, that have a fixed camera position, are used by many authors. Ishiguro, Kato and Tsuji (1993) used a system of multiple rotating cameras to detect stationary as well as moving objects. The system of multiple cameras is used to track and plot the existence of walls and moving objects. The vision system that was used by Pin and Wantabe (1992) is a stationary low resolution camera that has the ability to detect all obstacles. Daily et. al. (1988) used a vision system to provide topography as well as obstacle recognition. The problem with most vision systems is the large amount of computations that have to be done to interpret the picture.

The other major area of object recognition is through the use of sound-waves. An advantage of sonar systems is that they are computationally efficient. Manyika and Whyte (1993) used sonar for the tracking of a mobile robot. They proved that it is possible to use a monopulse system to provide highly accurate location information. Barshan and Kuc (1992) proved that is possible to provide highly accurate results using a wide-beam sonar system.
A major task of a controller for an AGV is the path planning in an unknown environment. Maze path planning has been the topic of several papers authored or co-authored by Lumelsky (1987, 1988, 1991). In the solution of the maze path planning, Lumelsky determines a desired direction to take when an obstacle is found. The algorithm then proceeds to follow the outline of the obstacle until the obstacle is no longer in the desired straight line path to the target. This method is repeated at every obstacle that is encountered. Trabia (1993) use a different approach to finding the target that still relies on a systematic procedure coupled with a more direct approach to reaching the target. Balch and Arkin (1993) used a gradient method to always move toward the target. After leaving a dead end alley, the entrance and direction of the dead end alley were stored in memory to ensure that the path is not explored again.

A mobile robot path planning algorithm will usually include an algorithm for obstacle avoidance. Takeuchi and Nagai (1988) devised a navigation algorithm that worked well but it relies on highly accurate stationary sonar sensors in an uncluttered area. Taylor and Kriegman (1993) used a system of stationary landmarks that a sonar sensor could detect to determine the AGV position and it used proximity sensors to evaluate where the obstacles are positioned. The previous two algorithms used stationary sonar beacons to locate the vehicle as well as the obstacles. NASA (1991) used a potential field that surrounds any obstacle to try to repeal the AGV from coming close to the obstacle using the theory of
magnetics. All of these systems depend on having prior knowledge of the environment.

A more difficult problem occurs when the AGV has no prior knowledge of the environment. Majumder, Naganathan and Kim (1993) described a navigation system that uses an ultrasonic sensor that sweeps a large arc motion to detect the location of all neighboring obstacles. The sweeping method is an accurate way of determining obstacle position. Pin and Watanabe (1993) used a low resolution optical camera to detect all of the obstacles. These methods along similar ones are used to determine obstacles in an unknown environment so that accurate navigational systems can be developed.

Fuzzy-logic is a new technique for problem solving based on human experiences, Zadeh (1973). The easiest way to explain the ideas behind fuzzy-logic is to demonstrate how fuzzy-logic works, so the example of driving a car will be used to demonstrate the theory behind fuzzy logic. If the car starts to deviate a small amount to the left of the desired target, the normal response is to turn the steering wheel a small amount to the right. If the car starts to deviate a large amount to the right of the desired target, the normal response would be a large turn of the steering wheel to the left. Fuzzy logic controllers, from other researchers experience, are very robust controllers. If there is a large amount of inputs and outputs that are coupled or not fully defined, the process of creating a successful controller becomes a difficult task. Fuzzy logic is used to provide the system with stability in areas that have high uncertainty values.
After a fuzzy logic controller has been designed, the controller must be tested to evaluate its reliability. After the controller proves to be reliable, the next question that needs to be addressed is there a better configuration of the fuzzy logic controller that will make the system work more efficiently. Several researchers have studied the above problem. Sugeno (1985) used structure and parameter identification process, similar to that used in traditional control, to select the rules of the fuzzy controller based on experimental data. Procyk and Mamdani (1977) developed a method for self-organized controllers. In this method, the rules are continually adjusted to improve the performance of the controller. Langari and Berenji (1992) presented a method for self-organizing fuzzy control system that has a "critic" program that evaluates the performance of the system and a "performance optimizer" program that rewards or penalizes the rules according to the "critic" output. Berenji (1992a and 1992b) suggested using neural networks to train fuzzy-logic controllers. Rashid and Heger (1993) developed a feedback system to tune a fuzzy-logic controller. Athalye et. al.(1993) used a simplex method to optimize an integral square error performance index of a fuzzy-logic controller. The optimized controller is then demonstrated to have a better performance than an ordinary PI controller.

1.3 PROJECT OBJECTIVES

A controller has been developed for an AGV in an unknown environment. A model for a four wheeled vehicle that has passive rear wheels and active front
steering is developed. The controller has a path planning algorithm that is capable of solving many different situations that can arise in traveling to a specified target. The AGV will use a rotating ultrasonic range sensor that is mounted on the front of the vehicle. The sensor will use a passive sweeping system to detect the presence of any obstacle. The ultrasonic sensor will provide accurate and quick calculations of the range, along any angle within the mechanical limits of the sensor, to an obstacle. The AGV will use a fuzzy-logic controller for the steering gain. An algorithm has been developed that will self generate a fuzzy logic controller. Through the use of nonlinear programming, the fuzzy-logic controller that was generated, will be optimized to create the most effective controller possible.
CHAPTER 2

MODELING OF THE VEHICLE

2.1 ASSUMPTIONS

The vehicle is assumed to have rigid wheels and a rigid body. The variations in terrain level and quality are assumed to be negligible. The wheels of the vehicle will not have any slippage during motion. The velocity of the vehicle is constrained to only be in the forward direction.

2.2 VEHICLE AND SYSTEM VARIABLES

The vehicle has several outputs that can be seen in Figure 1. The vehicle orientation, \( \theta \), is measured by counter-clockwise rotation about the z axis with respect to the x axis. The steering angle, \( \alpha \), is measured by counter-clockwise rotation about the z axis with respect to the vehicle's axis. A third vehicle variable is the velocity of the vehicle, \( V \). The angle to the target, \( \phi \), is defined by the vector that connects the vehicle mass center with the target and can be calculated as follows,

\[
\phi_i = \arctan \left( \frac{y_{\text{target}} - y_{\text{vehicle} \ i}}{x_{\text{target}} - x_{\text{vehicle} \ i}} \right)
\]

The distance to the target is \( D \), which is calculated as follows,
The last output variable is the distance to an obstacle, if any. This variable is determined through the use of an ultrasonic sensor which is discussed in section 3.2.

The vehicle has two inputs. The first input is the steering change, $\Delta \alpha$, and it is constrained not to allow the value of steering angle, $\alpha$, to be greater that $\pm$ the mechanical limit of $\alpha_{\text{limit}}$ degrees. The second input is the change in the acceleration pedal angle, $\Delta \beta$, which controls the velocity of the vehicle, according to the following relationship,

$$D_i = \sqrt{(x_{\text{target}} - x_{\text{vehicle}})^2 + (y_{\text{target}} - y_{\text{vehicle}})^2}$$  \hspace{1cm} (2)
The specific formula that is used to calculate the velocity is,

$$\Delta V = f (\Delta \beta, V)$$  \hspace{1cm} (3)

The value of $c_i$ is a proportionality constant that relates radians to meters per second. The change in the acceleration pedal angle, $\Delta \beta$, is constrained not to allow the vehicle's velocity be greater than $V_{\text{max}}$.

2.3 CALCULATION OF THE NEW POSITION OF THE VEHICLE

The controller determines the change in the steering angle and the change in the acceleration pedal change every time step, $\delta t$ to help the vehicle reach the specified target. The controller will calculate the new steering angle, $\alpha_i$, at the $i$-th time step as follows,

$$\alpha_i = \alpha_{i-1} + \Delta \alpha_i$$  \hspace{1cm} (5)

The vehicle is assumed to travel along a circular arc during each time step. The instantaneous center of rotation of the vehicle, Figure 2, has to be determined by calculating the radius of curvature, $\rho_i$, which is measured from the mass center of the vehicle as follows,

$$\rho_i = \sqrt{\left(\frac{L}{2}\right)^2 + \left(\frac{L}{\tan(\alpha_i)}\right)^2}$$  \hspace{1cm} (6)
where, \( l \) is the distance between the front axle and the rear axle of the vehicle. The mass center is assumed to be along the center line of the vehicle and at the midpoint between the two axles. The angle, \( \Delta \), that is formed by connecting the rear wheels and the mass center is calculated as follows,

\[
\Delta_i = \arctan \left( \frac{\tan \left( \alpha_i \right)}{2} \right) \tag{7}
\]

By adding the vehicle orientation and \( \Delta \), the initial angle, \( \eta_i \), that is used in formulation of the arc that is traveled can be calculated,

\[
\eta_i = \theta_i + \Delta_i \tag{8}
\]
Using equation 4, the length of the arc, $\gamma_i$, that is travelled can be evaluated as follows,

$$\gamma_i = \frac{c_1 \Delta \beta_i \delta t + v_{i-1} \delta t}{\rho_i}$$  \hspace{1cm} (9)$$

The change in the acceleration pedal angle, $\delta \beta_i$, is measured in radians. The new orientation, $\theta_{i+1}$, of the vehicle is calculated,
When the steering angle, $\alpha_i$, is positive, the vehicle will rotate CCW and when the steering angle, $\alpha_i$, is negative the vehicle will rotate CW.

The new position of the mass center of the vehicle is also dependent on the sign of the steering angle and can be calculated as follows,

$$
\theta_{i+1} = \theta_i + \gamma_i \quad \text{CCW (10)}
$$

$$
\theta_{i+1} = \theta_i - \gamma_i \quad \text{CW (11)}
$$

When the steering angle, $\alpha_i$, is equal to zero, the following set of equations is used to calculate the new position of the mass center because the value of $\rho_i$ will be infinity,

$$
x_{i+1} = x_i - \rho_i \sin(\eta_i) + \rho_i \sin(\gamma_i + \eta_i) \quad \text{CCW (12)}
$$

$$
y_{i+1} = y_i + \rho_i \sin(\eta_i) - \rho_i \sin(\gamma_i + \eta_i)
$$

$$
x_{i+1} = x_i + \rho_i \sin(\eta_i) + \rho_i \sin(\gamma_i - \eta_i) \quad \text{CW (13)}
$$

$$
y_{i+1} = y_i - \rho_i \sin(\eta_i) + \rho_i \sin(\gamma_i - \eta_i)
$$

When the steering angle, $\alpha_i$, is equal to zero, the following set of equations is used to calculate the new position of the mass center because the value of $\rho_i$ will be infinity,

$$
x_{i+1} = x_i + \left(c_i \Delta \beta_i \delta t + v_{i-1} \delta t\right) \cos(\theta_i) \quad (14)
$$

$$
y_{i+1} = y_i + \left(c_i \Delta \beta_i \delta t + v_{i-1} \delta t\right) \sin(\theta_i) \quad (15)$$
CHAPTER 3

PATH PLANNING AND OBSTACLE AVOIDANCE

3.1 PATH PLANNING ASSUMPTIONS

A path planning algorithm needs to have the ability to avoid the obstacles such as: buildings, trees, cliffs, walls, etc... All of these obstacles have one thing in common, which is, they are all inaccessible areas that the vehicle would not want to travel through. Therefore, the map of the environment is developed in 2-½ dimensions. The obstacles will have x and y coordinates and the z coordinate will either be a unit if the area is inaccessible or zero if the area is accessible. The environment is not known before the start of motion, so the vehicle will generate a localized map of the environment at each time step.

3.2 ULTRASONIC SENSOR

An ultrasonic sensor will be placed in the front of the vehicle. The controller can rotate the ultrasonic sensor, as seen in Figure 4, to any desired angle within certain mechanical limits. The range is governed by the farthest accurate distance possible for the sensor to measure. It is also assumed that the ultrasonic sensor is not capable of sensing through an obstacle. The details of how an
Ultrasonic sensor calculates the distance to an obstacle are discussed in detail in Leonard and Durrant-Whyte (1992).

![Diagram of a vehicle with an ultrasonic sensor mounted on its front.](image)

**Figure 4** A Vehicle with an Ultrasonic Sensor Mounted on its Front

The ultrasonic sensor that is mounted on the vehicle has a total sweep range of ±ε degrees with respect to the vehicle orientation, θ, where ε is the mechanical limit of the sensor. The ultrasonic sensor forms a sensed cone that is illustrated in Figure 6. The sensor can only sense the part of the object that is within the cone, so if the vehicle is stationary, it will not be able to tell how long or how wide an obstacle as in the situation shown in Figure 5.

The path planning algorithm will send a signal to the ultrasonic sensor to scan a certain direction. The value of this angle will be continually changed depending on the solution technique that the path planning algorithm is using. The ultrasonic sensor will return, to the path planning algorithm, the distance to
an obstacle, if there is any, or it will return a null character. The null character will be interpreted by the path planning algorithm as there is no obstacle along the direction of this angle.

![Diagram of Obstacle, Range, and Sensed Region](image)

**Figure 5** Sensed Region Cone

### 3.4 PATH PLANNING ALGORITHM

The objective of the path planning algorithm is to determine the desired correction angle the vehicle orientation that is necessary to ensure travel along a collision free path. A controller will be used to determine the actual amount of change in the steering that is necessary to follow a collision free path. The
controller is used because the amount of steering change depends on the necessary
correction of the vehicle orientation as well as the current steering angle.

The path planning algorithm is separated into three distinct procedures to
minimize computations. The procedures are:

Procedure 1, will be used to find the desired path to travel along when
there is no obstacle in the direction of the vehicle orientation, \( \theta_v \).

Procedure 2, will be used to find the desired path to travel along when
there is at least an obstacle in the direction of the vehicle orientation, \( \theta_v \).

Procedure 3, will be used to find the desired path to travel along when
problems such as a dead-end alley or a highly cluttered environment occur.

A block diagram of the path planning algorithm is shown in Figure 6. At
the start of motion, the path planning flag (PPF) is set to zero, after motion starts
the PPF is free to assume the value of zero or one depending upon the current
situation of the vehicle.

The input to the path planning algorithm are the coordinates of the target
and the current position and orientation of the vehicle. The output of this
algorithm will be the vehicle orientation correction, \( \Delta \phi_v \). The vehicle orientation
correction, \( \Delta \phi_v \), is measured with respect to the vehicle orientation angle, \( \theta_v \).
Procedure one and procedure two are configured in such a way that they will only
return a value of the vehicle orientation correction, \( \Delta \phi_v \), if there is no obstacle in
this direction. The feasibility of the vehicle orientation correction, \( \Delta \phi_v \), is checked
as seen in Figure 6 as the decision blocks that ask if the path is feasible.
Figure 6  Block Diagram of the Path Planning Algorithm
3.4.1 SOLUTIONS TO PROCEDURE ONE

Procedure one is activated when no obstacle is detected in front of the vehicle and the PPF has a value of zero at the start of the time step. The procedure starts with the ultrasonic sensor pointed in the same direction as the vehicle orientation, $\theta_i$. The ultrasonic sensor is then rotated (swept) at incremented steps toward the vector that connects the vehicle with the target. One of the following scenarios may occur:

i. the sensor has rotated a safe angle, $\xi$, past the target, Figure 7.

ii. an obstacle is detected before the sensor has swept a safe angle past the target, Figure 8.

iii. the sensor has reached the mechanical limit of the sensor, Figure 9.

In case i, we sweep up to the safe angle, $\xi$, instead of sweeping up to the mechanical limit, $\varepsilon$, to reduce the search time. The safe angle, $\xi$, is a function of the range of the ultrasonic sensor and can be calculated as follows,

$$\xi_i = \arcsin \left( \frac{SD}{R} \right)$$  \hspace{1cm} (16)

where, $SD$ is the minimum acceptable distance that the vehicle should be from an obstacle. The range, $R$, is the maximum accurate distance that the ultrasonic sensor can sense. If the safe angle, $\xi$, is reached and no obstacle is sensed, then the vehicle orientation correction, $\Delta\phi$, is,
\[ \Delta \phi_i = \phi_i - \theta_i \]  

(17)

A possible solution to procedure one is case ii, which occurs when an obstacle is detected during the sweep. The angle of the ultrasonic sensor when it detects the obstacle is defined as \( \lambda_i \), which is measured with respect to the vehicle orientation \( \theta_i \). If the sweep direction is in the positive direction, then the vehicle orientation error, \( \Delta \phi_i \), can be calculated as follows,

\[ \Delta \phi_i = \lambda_i - \xi_i \]  

(18)
If the sweep direction is in the negative direction then, the vehicle orientation error, $\Delta \phi_i$, can be calculated as follows,

$$\Delta \phi_i = \lambda_i + \xi_i$$  \hspace{1cm} (19)

The safe angle, $\xi_i$, is calculated as follows,

$$\xi_i = \arctan \left( \frac{SD}{RO_i} \right)$$  \hspace{1cm} (20)

where $RO_i$ is the range to an obstacle.

---

Figure 8 Possible Solution to Procedure 1, case ii
The last possible solution of the first procedure is when the ultrasonic sensor sweep angle, $\lambda$, has reached the mechanical limit as seen in Figure 9. In this case, the value for the vehicle orientation correction, $\Delta \phi_i$, is,

$$\Delta \phi_i = \epsilon$$

(21)

![Target Diagram](#)

**Figure 9** Possible Solution to Procedure 1, case iii

### 3.4.2 SOLUTIONS TO PROCEDURE TWO

The second procedure is activated directly when the ultrasonic sensor algorithm detects an obstacle in front of the vehicle and the PPF is equal to zero
at the start of the time step. It may also be activated if there is no feasible response to the first procedure. The second procedure will start with the ultrasonic sensor pointed in the direction of the vehicle. The sensor is then rotated in incremental steps away from the vector that connects the vehicle to the target. The ultrasonic sensor sweep will continue until either:

i. the sensor has swept through a safe angle, $\kappa$, past an obstacle.

Figure 10.

ii. the mechanical limit of the ultrasonic sensor has been reached.

The safe angle, $\kappa$, is calculated using the following equation,

$$\kappa_i = \arctan \left( \frac{2SD}{RO} \right) \quad (22)$$

![Figure 10 Possible Solution to Procedure 2, case i](image)

Figure 10 Possible Solution to Procedure 2, case i
The obstacle range, RO, is the last calculated distance to an obstacle and the minimum alley width is set at twice the safe distance used in procedure one. If the ultrasonic sensor reaches the safe angle, \( \kappa_i \), the vehicle orientation correction, \( \Delta \phi_i \), can be evaluated as, if the sweep angle is positive,

\[
\Delta \phi_i = \lambda_i - \frac{\kappa_i}{2}
\]  

(23)

If the sweep angle is negative, the vehicle orientation correction can be calculated as, A CCW sweeping motion will result in a positive value of \( \lambda_i \).

\[
\Delta \phi_i = \lambda_i + \frac{\kappa_i}{2}
\]  

(24)

When an opening between two obstacles is detected such that the angle between the obstacles, \( \tau \), is less than \( \kappa_i \), the two obstacles are treated as if they were a solid obstacle and the sweep will continue. This may happen when the vehicle is in a cluttered environment as seen in Figure 11.

If the ultrasonic sensor sweep has reached the mechanical limit of the sensor, one of two things are possible. The first possibility is there is no obstacle along the mechanical limit when it was reached, then \( \tau \) is calculated as follows,

\[
\tau = \epsilon - \lambda_i
\]  

(25)

If the angle, \( \tau \), is less than \( \kappa_i \), it is determined that there is no feasible solution on this side of the obstacle and procedure two is restarted with the ultrasonic
sensor being swept in the opposite direction (i.e. toward \( \phi_i \)). If the angle, \( \tau \), is greater than \( \kappa_n \), the vehicle orientation correction \( \Delta \phi_n \) can be evaluated as if the sweep angle is positive,

\[
\Delta \phi_i = \lambda_i - \frac{\kappa_i}{2}
\]  \hspace{1cm} (26)

If the sweep angle is negative, the vehicle orientation correction can be calculated as,
\[ \Delta \phi_i = \lambda_i + \frac{\kappa_i}{2} \]  

(27)

The second possibility is that the ultrasonic sensor reaches the mechanical limit while being swept in the opposite direction without finding a feasible solution. The path planning algorithm will declare the area as highly cluttered and it activates the third procedure.

3.4.3 SOLUTION TO PROCEDURE THREE

Procedure three can be activated directly when the PPF is equal to one or when procedure two has not found a feasible solution to the vehicle orientation correction, \( \Delta \phi \). The third procedure is used normally to solve the problems of a dead-end alley, but it is robust enough to solve many other problems such as highly cluttered areas. Procedure three uses a left hand steering convention to solve for the vehicle orientation correction when such problems occur.

The first step of procedure three is to set the PPF to one. Procedure three remains active as long as PPF is equal to one. This flag is assigned a value of zero at the start of motion. Only procedure three can alter the value of the flag after motion has started.

Procedure three uses modified versions of procedures one and two to solve for the vehicle orientation correction. Figure 12 illustrates the four possible scenarios that are possible in a highly cluttered environment. The different solution techniques occur when the following conditions are true:
Figure 12 All Possible Solutions to Procedure 3
i. the vehicle has an obstacle covering the entire sensed cone of the ultrasonic sensor, Figure 13.

ii. the vehicle has an obstacle in front of it, but there is an opening to the left of the vehicle, Figure 14.

iii. the vehicle has no obstacle in front of it but there is an obstacle to the right of the vehicle, Figure 15.

iv. procedure three has determined that the highly cluttered region is no longer a problem, Figure 16.

The ultrasonic sensor for all cases will start pointed in the same direction as the vehicle orientation, $\theta_v$. For case i, the ultrasonic sensor will be swept towards the CCW mechanical limit. When sensor sweep has reached the

![Figure 13](image-url) Possible Solution to Procedure 3, case i
mechanical limit while sensing an obstacle for the entire sweep, the vehicle orientation correction can be seen in Figure 13 and is calculated as follows,

\[ \Delta \phi_i = \epsilon \]  \hspace{1cm} (28)

Figure 14 Possible Solution to Procedure 3, case ii

The second possible solution to procedure three starts by sweeping the ultrasonic sensor towards the CCW mechanical limit as seen in Figure 14. The solution to the second occurs when the ultrasonic sensor has swept a safe angle past an obstacle. The safe angle, \( \kappa_i \), can be calculated as follows,
\[ \kappa_i = \arctan \left( \frac{2SD}{RO} \right) \] (29)

If the safe angle is reached the vehicle orientation error can be calculated as follows,

\[ \Delta \phi_i = \lambda_i - \frac{\kappa_i}{2} \] (30)

If the ultrasonic sensor is swept and no feasible opening is detected, then the vehicle orientation correction is calculated as follows,

\[ \Delta \phi_i = \epsilon \] (31)

The third solution of procedure three is activated when there is no obstacle in front of the vehicle as seen in Figure 15. Case iii is used to follow the wall of the obstacle that is on the right hand side of the vehicle. This is done by sweeping the ultrasonic sensor towards the CW mechanical limit until the obstacle is detected. The vehicle orientation correction then can be calculated as,

\[ \Delta \phi_i = \lambda_i + \xi_i \] (32)

where \( \lambda_i \) is the sweep angle and \( \xi_i \) is the safe angle where the safe angle can be calculated as follows,

\[ \xi_i = \arctan \left( \frac{SD}{RO_i} \right) \] (33)
The final solution occurs when the path planning algorithm has determined that the vehicle is no longer in the highly cluttered environment that initiated procedure three as seen in Figure 16. In this situation the controller will calculate the vehicle orientation error as follows,

$$\Delta \phi_i = -\epsilon$$

The next step the controller takes is to build an imaginary wall perpendicular to the vehicle as seen in Figure 17. The imaginary wall is constructed to deter the path planning algorithm from re-entering an area that has already been determined to be highly cluttered. The length of the wall can be determined by doubling the calculated distance from the vehicle position when procedure three was initiated to the vehicle position when procedure three, case iii, was first activated.
Procedure three will continue to set the vehicle orientation correction to the CW mechanical limit until the target orientation error is less than zero and greater than $-\pi$. Once this constraint is met, procedure three will set the PPF to zero and the full path planning algorithm will be used.

Figure 16 Possible Solution of Procedure 3, case iv

Figure 17 Construction of the Imaginary Wall
4.1 FUZZY-LOGIC

The design of a fuzzy-logic controller can be broken down into several basic steps. The initial step is the fuzzification of the system. This means that the system's inputs and outputs have to be determined. Usually, inputs are the control signals to the system such as current in the case of electric motor. Outputs are the measure of error with respect to certain parameters of performance such as displacement or velocity.

The fuzzification of a system starts by defining each of the inputs and outputs variables using linguistic terms, e.g., small, medium, large, etc.. For each
linguistic term, the smallest and the biggest variable values that are in the membership set need to be defined. The membership function, with a minimum value of 0 and a maximum value 1, will be chosen that describes the truthfulness, \( \mu \), of any value in this set. A zero value means totally false while one means totally true. Linguistic terms may overlap to describe the fact that a certain value of a variable can be described, with some truthfulness, as either medium or big.

![Membership Sets for a Variable](image)

**Figure 19** Membership Sets for a Variable
After the system has been fuzzified, the fuzzy inference rules need to be developed. The fuzzy inference rules are what the fuzzy-logic controller uses to describe the relations between the measured system outputs and inputs. These relations are in the form of,

\[
\text{If } \text{<condition A> and <condition B> then <consequence C>}. 
\]

The fuzzy-logic inference rules during the decision making process will assign a value of truthfulness, \( \mu \), to the input. The value of the truthfulness, \( \mu \), of the input is a function of the outputs that control the input decision, and can be evaluated using,

\[
\mu \text{ of } C = \min (\mu \text{ of } A, \mu \text{ of } B, ...) \tag{35}
\]

The input control signal should reflect the results of the rules activated in the preceding step.

The controller is now ready to send the input signal to the system, however, since the results are fuzzy, we need to defuzzify them and send a numerical value to the system. Several defuzzification algorithms have been proposed, Kosko (1992). The method used for the controller that is being developed is the moment of area algorithm. This algorithm starts by calculating the area of each activated input membership set, with the height of the membership equal to the value of truthfulness that had been assigned by equation 35. An illustration of the area that needs to be calculated is seen in Figure 20.
The centroid of the area is then evaluated for each activated input. The defuzzification is completed by calculating the true response as follows,

\[
\text{value of the input variable} = \frac{\sum \text{centroid}_i \times \text{area}_i}{\sum \text{area}_i}
\]  

(36)

4.2 OPTIMIZING OF FUZZY-LOGIC

The first questions that arise when designing a fuzzy-logic controller are what are the "correct" membership functions and what are the "correct" rules? To a certain extent, the answer to either question may not need to be an accurate one since the defuzzification method, equation 36, averages the results of activated rules. Also, experience shows that when the number of outputs is higher than two,
it becomes increasingly difficult to justify a set of rules. A generalized method to automatically generate fuzzy inference rules and to modify the number, the range, and the shape of the membership functions for these rules using nonlinear programming has been developed.

4.3 AUTOMATIC GENERATION OF FUZZY INFERENCE RULES

For a system with "n" inputs and "m" outputs, the rules can be represented as "n" hyper-surfaces of "m+1" dimensions. To quantitatively describe the relation between different membership sets, we assign integer numerical values to each membership set as in Table I. We use these numbers to identify a rule. For example, if \( I_1(-2,0) \) is equal to one, it means that the first input is of the positive small set, if the first output is of the negative medium set and the second output is of the zero set. Other sets may be added, or deleted, to better describe a particular system.

<table>
<thead>
<tr>
<th>Set</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical Weight</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

In what follows, a set of guidelines have been developed that can help in generating the fuzzy inference rules,
**Guideline 1:** If all outputs belong to the zero membership set \((Z)\), the input also belongs to zero membership set \((Z)\).

For most systems, this relation will always be valid since zero error requires no control correction.

**Guideline 2:** To ensure smooth response of the fuzzy-logic controller, all input hyper-surfaces have to be smooth.

Smoothness, is not meant in the regular algebraic term, continuous \(f\) and \(df/dx\), since the inputs and the outputs are integers. We use the following definition to ensure smoothness of the input hyper-surface,

*The absolute difference in weight of the input resulting from any two adjacent rules should be less than or equal to one or,*

\[
|I_i (O_1, O_2, \ldots, O_m) - |I_i (O_1 V O_1 \pm 1, O_2 V O_2 \pm 1, \ldots, O_m V O_m \pm 1)| | \leq 1 \tag{37}
\]

Satisfying the above inequality ensures that the results of any two adjacent inference rules cannot be, for example, NM and PB.

The initial step in the self generating fuzzy inference rules algorithm is, based on the behavior of the system, decide what should the appropriate control inputs (in linguistic terms) be when the all the outputs are at extremes. Table II shows an example of how to setup the extreme conditions of a two output, one input, fuzzy-logic controller.
Table II Evaluation of the extreme response for a single-input two-output system described using seven membership sets

<table>
<thead>
<tr>
<th>Output1 →</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output2 ↓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>NM</td>
<td>NS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PB</td>
</tr>
<tr>
<td>NS</td>
<td>NM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PB</td>
</tr>
<tr>
<td>Z</td>
<td>NB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PB</td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>NB</td>
<td></td>
<td></td>
<td></td>
<td>PB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td>NB</td>
<td></td>
<td></td>
<td>PB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
</tr>
</tbody>
</table>

After the extreme inputs have been chosen, connect vectors between the extremes outputs and the origin (where all outputs are zero). The vector connecting the origin with the extreme output state \( (O_1, k, \ldots, O_m)^T \) for the \( i \) th input passes by many different the states, and the states that are effected by this vector,

\[
(O_1, O_2 \ldots, O_m)^T_i = \text{integer} \left( \frac{i}{k} (O_1, k, \ldots, O_m)^T_i \right)
\]  

(38)

The inputs along this vector can be evaluated using the following equation,

\[
I_i \left( \text{integer} \left( \frac{i}{k} (O_1, k, \ldots, O_m)_i \right) \right) = \text{integer} \left( \frac{i}{k} I_i (O_1, k, \ldots, O_m)_i \right)
\]

(39)
where, \( i = 0, 1, \ldots, k \) if \( k > 0 \)
\[ \begin{align*}
i &= 0, -1, \ldots, k \quad \text{if} \quad k < 0
\end{align*} \]

If more than one vector passes by the same cell the weights of the inputs are averaged to nearest integer number and recorded accordingly.

In table III, the rules that are in bold are the original rules that were defined in table II. The rules that have been generated by equations 30 and 31 are in italics.

**Table III** Preliminary rules for a single-input two-output system described using seven membership sets

<table>
<thead>
<tr>
<th>Output1 ( \rightarrow )</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output2 ( \downarrow )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>NS</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>Z</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>PM</td>
<td>NB</td>
<td>NM</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
</tr>
</tbody>
</table>

Table III has several rules that were generated by equations 20 and 21 that violate guideline 2. As a result, we use the inequality defined in equation 18 to modify Table III starting from the origin toward the boundaries of the table. The
results are shown in Table IV and the elements that have been modified by
equation 20 are shaded.

Table IV  Rules for a single-input two-output system described using seven
membership sets

<table>
<thead>
<tr>
<th>Output1 →</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output2</td>
<td>NB</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td></td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>NM</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td>NM</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td></td>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
</tr>
</tbody>
</table>

4.4 SELECTION OF MEMBERSHIP SETS

The effectiveness of a controller can be evaluated by monitoring a
performance index. Performance index may be the time to reach target or the
deviation from a desired path. Our objective is to minimize this performance index
by varying the number, the range, and the shape of the membership functions for
the inputs and outputs of the system. We limit the following discussion to
membership functions of trapezoidal and triangular shapes. However, the ideas
presented here can be easily modified to deal with membership functions of other shapes. A trapezoidal membership function, Figure 13, is described using its four vertices. These four vertices are separated into the two limiting points of the set and the two intermediate values that define the set's full truthfulness range. Therefore, the variables used to describe any input or output are the number of the membership functions and the values of the four vertices describing each set. The variables used to describe the j the output are as follows,

\[ \begin{align*}
V_{O_j} (NB,1), & \quad V_{O_j} (NB,2), \quad V_{O_j} (NB,3), \quad V_{O_j} (NB,4), \\
V_{O_j} (PB,1), & \quad V_{O_j} (PB,2), \quad V_{O_j} (PB,3), \quad V_{O_j} (PB,4)
\end{align*} \]

Practical solutions require minimum and maximum limits on the number of linguistic sets. Two membership sets are the minimum needed to get realistic results. A membership function describing linguistic term "K" is subject to the following constraints,

\[ V_{O_j} (K,i+1) \geq V_{O_j} (K,i) \quad i = 1,2,3 \]

Additional constraints may be added to describe a particular system.

A modified form of a Hooke-Jeeves optimization method, Rekalitis et al. (1983), is used. The algorithm, which is graphically explained in Figure 14 for a function with two variables, but can be expanded to optimize any number of variables. The method starts by testing two points, along x_i direction, at ± Δx_i with respect to the initial guess. The initial guess point is replaced by the point
with the better performance index. The process is repeated for all the other variables as long as no constraint is violated and the algorithm continues to find points with a better performance index. If a constraint is violated, the algorithm will move that point, along the $x_i$ direction, within the feasible range. If the program cannot find a better neighboring point while searching in the $x_i$ direction, it reduces the step $\Delta x_i$ till a satisfactory answer is obtained or a maximum number of iterations has be reached.
4.5 APPLICATION OF THE OPTIMIZATION ALGORITHM

The optimization algorithm will be applied to the problem of guiding the autonomous vehicle that is developed in Chapter 2 to a specified target. Figure 23 shows a block diagram of the control system that is used when the vehicle is moving at a constant velocity of \( V \). It should be noted that the vehicle will be operating in an unknown environment without obstacles. The systems variables that are used to control the vehicle can be seen in Figure 22. The input to the controller is the correction of the steering angle, \( \Delta \alpha \). The measured outputs are the steering angle, \( \alpha \), which is measured with respect to the vehicle axis, the target orientation error, \( \Delta \phi \), and the distance to the target, \( D \). In this model, the distance

![Diagram of the Variables for an Autonomous Vehicle in an Unknown Environment without Obstacles](image)

**Figure 22** Diagram of the Variables for an Autonomous Vehicle in an Unknown Environment without Obstacles
to the target is not expressed in fuzzy terms and is only used to stop the vehicle once it reaches the neighborhood of the target, i.e., distance to the target becomes less than Dmin. Physically, the maximum steering angle is ±45 degrees with respect to the orientation of the vehicle. Therefore, the results of the fuzzy-logic controller will be adjusted so that this physical constraint is not violated.

The fuzzy inference rules that will be used can be seen in Table V. The output variable, $\Delta \phi$, will replace Output 1 and the output variable, $\alpha$, will replace Output 2. In addition to the constraints of the preceding section, there will be several more constraints placed on the problem,

i. The same number of membership sets is used for all variables.
ii. The positive and the negative fuzzy sets are symmetrical for all the fuzzy variables since lack of this constraint will result in unsubstantiated bias to left or right steering. The zero set will also be symmetrical about zero.

iii. We use the same membership set functions for $\alpha$ and $\Delta \phi$, since they share the same units (degrees) and experience shows that they can be described using the same linguistic terms.

iv. We specify that the two extreme variable values in the biggest, or smallest linguistic sets are equal to ±180 degrees for $\alpha$ and $\Delta \phi$.

v. $\Delta \alpha$ has the same membership sets as $\alpha$ and $\Delta \phi$ except that the two extreme variable values in the biggest, or smallest linguistic sets are equal but they are free to assume any value.

The performance index that is used as an objective function is the path length between the start and the target points assuming constant velocity motion.

The autonomous vehicle model that was tested is the same model that is developed in Chapter 2. The vehicle starts at (0,0)m and the specified target is (100,100)m. The neighborhood of the target is described by a circle with the radius equal to $D_{\text{min}}$, and $D_{\text{min}}$ is set at 1.5m. The vehicle original orientation, $\theta$, is at -135 degrees with respect to the x axis. The original steering angle, $\alpha$, is zero degrees. The vehicle has a length, l, of 3m.
4.6. OPTIMIZATION RESULTS

Figure 24 represents the best results when the number of membership sets is fixed to three, five, and seven respectively. The starting values for the sets are the same (i.e. the range and shape of PS was the same for all three trials) where applicable. The sets were also not configured in what could be considered a "normal" arrangement. Seven membership sets resulted in the best performance with a path length of 152.2m compared to absolute minimum path length of 144.96m. Observation of Figure 24 shows also that the controller with the seven membership sets exhibits the least amount of oscillatory behavior. Table V and Table VI lists the optimized membership sets for all tested cases. Note that for the case of the seven membership sets, the small set is completely enclosed within

![Graph showing path of autonomous vehicle using three, five, and seven membership sets.](image-url)
Table V Membership Sets for the Correction of Steering Angle ($\Delta \alpha$)

<table>
<thead>
<tr>
<th>Mem. Set \n(K)</th>
<th>V(K,1)</th>
<th>V(K,2)</th>
<th>V(K,3)</th>
<th>V(K,4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three Sets</td>
<td>Z</td>
<td>-12.57</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td>12.57</td>
<td>12.86</td>
<td>98.65</td>
</tr>
<tr>
<td>Five Sets</td>
<td>Z</td>
<td>-14.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td>0.57</td>
<td>0.86</td>
<td>38.03</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>0.57</td>
<td>9.0</td>
<td>74.65</td>
</tr>
<tr>
<td>Seven Sets</td>
<td>Z</td>
<td>-13.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>PS</td>
<td>29.95</td>
<td>30.23</td>
<td>30.52</td>
</tr>
<tr>
<td></td>
<td>PM</td>
<td>0.57</td>
<td>5.51</td>
<td>79.23</td>
</tr>
<tr>
<td></td>
<td>PB</td>
<td>41.0</td>
<td>41.3</td>
<td>66.0</td>
</tr>
</tbody>
</table>

Figure 25 Path of the Autonomous Vehicle Using Seven Membership Sets
the medium set.

The optimization was repeated for the seven membership set case, but the starting configuration of the membership sets were what could be considered "normal". Figure 25 represents the results of the normal seven membership sets along with the best result of the seven membership set as seen in Figure 24. From observations of the figure, it is apparent that the seven membership sets from Figure 24 show better results in both the categories of shortest path and resistance to oscillatory motion.

Table VI Membership Sets for the Steering Angle and the Orientation Error ($\alpha$, $\Delta\phi$)

<table>
<thead>
<tr>
<th>Mem. Set (K)</th>
<th>V(K,1)</th>
<th>V(K,2)</th>
<th>V(K,3)</th>
<th>V(K,4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>-12.57</td>
<td>0</td>
<td>0</td>
<td>12.57</td>
</tr>
<tr>
<td>PS</td>
<td>12.57</td>
<td>12.86</td>
<td>180.0</td>
<td>180.0</td>
</tr>
<tr>
<td>Five Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>-14</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>PS</td>
<td>0.57</td>
<td>0.86</td>
<td>38.03</td>
<td>38.31</td>
</tr>
<tr>
<td>PM</td>
<td>0.57</td>
<td>9.0</td>
<td>180.0</td>
<td>180.0</td>
</tr>
<tr>
<td>Seven Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>-13.0</td>
<td>0</td>
<td>0</td>
<td>13.0</td>
</tr>
<tr>
<td>PS</td>
<td>29.95</td>
<td>30.23</td>
<td>30.52</td>
<td>30.81</td>
</tr>
<tr>
<td>PM</td>
<td>0.57</td>
<td>5.51</td>
<td>79.23</td>
<td>79.52</td>
</tr>
<tr>
<td>PB</td>
<td>41.0</td>
<td>41.3</td>
<td>180.0</td>
<td>180.0</td>
</tr>
</tbody>
</table>
CHAPTER 5
PATH PLANNING RESULTS

5.1 APPLICATION OF THE CONTROL SYSTEM

The control system that is developed for an autonomous guided vehicle (AGV) uses a fuzzy-logic controller to make the decisions on how much to change the steering angle to follow a collision free path that is prescribed by a path planning algorithm. The fuzzy-logic controller (FLC) that is used by the autonomous vehicle will have the same configuration of the membership sets and fuzzy inference rules as the fuzzy-logic controller that is developed in Chapter 4. The path planning algorithm that is used to evaluate the vehicle orientation correction to ensure a collision free path is developed in Chapter 3. The controller uses the same criteria as in Figure 23 to stop the vehicle, if the distance to the target is less than \( D_{\text{min}} \) then stop.

The control system in this section is tested in three different environments as seen in Figures 26, 27, and 28. For all the tests of the AGV path planning algorithms and fuzzy-logic controller, the following conditions are true:

i. The start of the vehicle is \((0,0)\).

ii. The target of the vehicle is \((100,100)\).

iii. The environment is \(100 \times 100\) units.

vi. The initial value of the steering angle, \(\alpha\), is 0 degrees.
v. The mechanical limit of the ultrasonic sensor, $e$, is $\pm 45$ degrees.

vi. The vehicle has constant acceleration for one time step and zero acceleration for all of the other time steps.

vii. The initial velocity of the vehicle is zero.

viii. The vehicle will stop when the distance to the target, $D$, is less than one unit.

For each trial of the controller the following conditions are variables:

i. The initial vehicle orientation, $\theta$, is 30, 45, or 60 degrees respectively.

ii. The initial value of the acceleration pedal angle change, $\Delta \beta$, is 0.1 radians or 0.4 radians.

iii. The value of the proportionality constant, $c_i$, is 20.

iv. The time step, $\delta t$, will be 0.05 seconds or 0.3 seconds.

v. The farthest accurate distance that the ultrasonic sensor can sense, $R$, is 5, 10, or 20 units.

The results are arranged in tables that describe trials with the same time step and vehicle velocity. Each table presents the combinations of three different sensor ranges with three different vehicle orientations.

5.1.1 **AGV IN A CLUTTERED ENVIRONMENT**

The following set of tables shows how the control system works for all the different configurations of the control system when the vehicle is travelling
through a cluttered environment. It can be seen from these tables that when the ultrasonic sensor range is set to five units, a majority of times the vehicle will crash. When the ultrasonic sensor range is ten units, more than 50 percent of the trials reach the target. The cause of all these crashes is the control systems inability to make correct decisions about the path to travel along since the controller possesses very limited information about the surrounding environment. When an ultrasonic sensor range of 20 units is used, a perfect success rate is obtained. A plot of the paths travel for several of these trials can be seen in Figure 26.

**Table VII**  Traversal Time in a Cluttered Environment, case (a)

<table>
<thead>
<tr>
<th>orientation range ↓</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>79.0 sec.</td>
<td>78.95 sec.</td>
<td>79.1 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>72.6 sec. (B)</td>
<td>crash</td>
<td>71.9 sec.</td>
</tr>
<tr>
<td>5 units</td>
<td>crash</td>
<td>crash (E)</td>
<td>71.8 sec.</td>
</tr>
</tbody>
</table>

*note: Letters in the cell correspond to those on Figure 26*
Table VIII  Traversal Time in a Cluttered Environment, case (b)

<table>
<thead>
<tr>
<th>orientation → range</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>19.3 sec.</td>
<td>19.3 sec.</td>
<td>19.3 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>crash</td>
<td>18.3 sec.</td>
<td>crash</td>
</tr>
<tr>
<td>5 units</td>
<td>18.0 sec. (A)</td>
<td>crash</td>
<td>18.3 sec.</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 26

Table IX  Traversal Time in a Cluttered Environment, case (c)

<table>
<thead>
<tr>
<th>orientation → range</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>78.0 sec.</td>
<td>77.4 sec. (C)</td>
<td>77.7 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>73.8 sec.</td>
<td>74.6 sec.</td>
<td>crash</td>
</tr>
<tr>
<td>5 units</td>
<td>crash</td>
<td>crash</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 26
Table X  Traversal Time in a Cluttered Environment, case (d)

<table>
<thead>
<tr>
<th>orientation → range ↓</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>19.8 sec.</td>
<td>19.8 sec.</td>
<td>19.8 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>20.1 sec.</td>
<td>20.4 sec.</td>
<td>20.1 sec.</td>
</tr>
<tr>
<td>5 units</td>
<td>crash (F)</td>
<td>crash</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 26

Figure 26  Select Vehicle Paths in a Cluttered Environment
5.1.2 AGV IN A DEAD-END ALLEY

The following tables show the traversal time for the autonomous guided vehicle operating in an environment that has a dead-end alley. For all of the trials the control system, when the ultrasonic sensor range was 5 units the vehicle crashed. All but two trails of the control system crashed when the ultrasonic sensor range was 10 units. The crashes in these trials are caused by the controller not having enough space to make the u-turn necessary to exit from a dead-end alley. Path A, Figure 27, is a good example of the problem, although the vehicle made it safely to the target, the vehicle did come within 2 units of the wall. All of the trials when the ultrasonic sensor range was 20 units reached the target safely.

Table XI Traversal Time in an Environment with a Dead-End Alley, case (a)

<table>
<thead>
<tr>
<th>orientation →</th>
<th>range</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20 units</td>
<td>111.80 sec.</td>
<td>110.8 sec.</td>
<td>112.6 sec.</td>
</tr>
<tr>
<td></td>
<td>10 units</td>
<td>122.3 sec.</td>
<td>121.8 sec. (A)</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 27
Table XII  Traversal Time in an Environment with a Dead-End Alley, case (b)

<table>
<thead>
<tr>
<th>orientation → range \ degree</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>27.8 sec. (B)</td>
<td>28.15 sec.</td>
<td>28.15 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>crash</td>
<td>crash</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 27

Table XIII  Traversal Time in an Environment with a Dead-End Alley, case (c)

<table>
<thead>
<tr>
<th>orientation → range \ degree</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>113.7 sec.</td>
<td>116.7 sec.</td>
<td>112.5 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>crash (C)</td>
<td>crash</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 27
Table XIV Traversal Time in an Environment with a Dead-End Alley, case (d)

<table>
<thead>
<tr>
<th>orientation →</th>
<th>range ↓</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>32.4 sec.</td>
<td>32.1 sec.</td>
<td>29.1 sec. (D)</td>
<td></td>
</tr>
<tr>
<td>10 units</td>
<td>crash</td>
<td>crash</td>
<td>crash</td>
<td></td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 27

Figure 27 Selected Vehicle Paths in an Environment with a Dead-End Alley
The multiple obstacle environment of Figure 28 provides a difficult problem. When the time step, $\delta t$, is equal to 0.05 seconds the controller succeeds in reaching the target 17 out of 18 times. When the time step, $\delta t$, is equal to 0.3 seconds the controller succeeds in reaching the target only when the ultrasonic sensor range is 20 units. The results of all the tests are seen in Tables XV through XVIII. In Figure 28 some selected vehicle paths are shown. The placement and shapes of the obstacles in Figure 28 are selected to confuse the path planning algorithm. When procedure three is activated by the path planning algorithm, the vehicle will attempt to make a left hand turn. The left hand turn that the vehicle takes will move the vehicle closer to Obstacle 3, in Figure 28, making it harder for the vehicle to reach the target.

Table XV Traversal Time in a Multiple Obstacle Environment, case (a)

<table>
<thead>
<tr>
<th>orientation range</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>78.5 sec.(A)</td>
<td>78.65 sec.</td>
<td>84.5 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>87.45 sec.</td>
<td>87.5 sec.</td>
<td>121.2 sec.</td>
</tr>
<tr>
<td>5 units</td>
<td>88.3 sec.</td>
<td>88.85 sec.</td>
<td>84.3 sec.</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 28
Table XVI  Traversal Time in a Multiple Obstacle Environment, case (b)

<table>
<thead>
<tr>
<th>orientation → range ↓</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>25.6 sec.</td>
<td>46.75 sec.</td>
<td>25.9 sec.(E)</td>
</tr>
<tr>
<td>10 units</td>
<td>29.6 sec.</td>
<td>29.1 sec.</td>
<td>35.3 sec.</td>
</tr>
<tr>
<td>5 units</td>
<td>28.8 sec.(B)</td>
<td>crash</td>
<td>32.4 sec.</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 28

Table XVII  Traversal Time in a Multiple Obstacle Environment, case (c)

<table>
<thead>
<tr>
<th>orientation → range ↓</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>78.3 sec.(C)</td>
<td>129.0 sec.</td>
<td>78.3 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>crash</td>
<td>crash</td>
<td>crash</td>
</tr>
<tr>
<td>5 units</td>
<td>crash</td>
<td>crash</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 28
Table XVIII  Traversal Time in a Multiple Obstacle Environment, case (d)

<table>
<thead>
<tr>
<th>orientation → range ↓</th>
<th>30 degrees</th>
<th>45 degrees</th>
<th>60 degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 units</td>
<td>37.5 sec.</td>
<td>36.9 sec.</td>
<td>28.5 sec.</td>
</tr>
<tr>
<td>10 units</td>
<td>crash</td>
<td>crash(D)</td>
<td>crash</td>
</tr>
<tr>
<td>5 units</td>
<td>crash</td>
<td>crash</td>
<td>crash</td>
</tr>
</tbody>
</table>

* note: Letters in the cell correspond to those on Figure 28

Figure 28  Selected Vehicle Paths in an Environment with Multiple Obstacles
5.2 AGV TRIALS WITH DIFFERENT ENVIRONMENTS

The AGV controller is modified according to the findings of the previous section. It is again tested in several other environments as seen in Figures 29, 30, 31, and 32. For all of the following trials of the AGV controller the conditions stated in previous section are valid except for the following modifications:

i. The range of the ultrasonic sensor is 20 units due to the results of previous section. This distance proved to be the most reliable regardless of the values of all other parameters were varied.

ii. The acceleration pedal angle change, $\Delta \beta$, for each test is 0.3 radians for one time step. After the first time step, the acceleration pedal angle change, $\Delta \beta$, is set at zero so that the vehicle will have a constant velocity. The value of the acceleration pedal angle change was chosen to force the AGV to operate at a high velocity. If the AGV controller can demonstrate its robustness at this high velocity, it is a logical deduction that the AGV controller will work equally as well at any slower velocity.

iii. The time step, $\delta t$, is determined to be equal to 0.05 seconds, so that the AGV will not be able to travel long distances between time steps. The small time step will increase the possibility that the AGV will reach the target.

The first trial that is seen in Figure 29 demonstrates the AGV controllers ability to keep the vehicle at a safe distance away from the obstacle while
Figure 29 Vehicle Path with Variable Vehicle Starting Orientations with an Obstacle following a path that can possibly be considered a near minimum length path around the obstacle. It also can be seen that the AGV controller resists the tendency to oscillate as it travels to the target compared to the unoptimized membership sets as explained in Chapter 4.
In the next trial, the environment contains several obstacles that are closely positioned and a large concave obstacle near the target, Figure 30. The AGV controller makes the appropriate path planning decisions for all situations. Once again, the AGV controller demonstrates its ability to keep the vehicle at a safe distance from any obstacle. The results of Figure 30 show that the vehicle followed almost the same path with the exception of the 45 degree case. In this particular case, the robot sensed the obstacle nearest to the starting point to be to the right of the vehicle. Therefore, it followed the other side of the obstacle as compared to all the other trials.
Figure 31 Vehicle Path with Variable Starting Vehicle Orientation with a Series of Long Slender Obstacles

The next trial, as seen in Figure 31, demonstrates the AGV controller's ability to travel between and around multiple obstacles. The AGV travels toward the target without using procedure three. The results of Figure 31 show that the vehicle follows almost the same path for all trials except for the case when the vehicle's initial orientation, $\theta$, is 90 degrees. For this case the AGV travels on the right side of the obstacle due to the vehicle's orientation when the obstacle was first sensed.
The environment of Figure 32 provides the controller with a more challenging test. The AGV controller will first have to negotiate a complex dead-end alley composed of two segments. As the AGV exits the second segment of the dead-end alley the controller will draw an imaginary wall in its internal memory to ensure that this branch will not be entered again. Once procedure three is turned off, the controller guides the AGV outside the dead-end alley. The vehicle follows the outside contour of the dead-end alley. At the first corner, procedure three is activated again. When the AGV stops sensing the wall on the right, a second imaginary wall is constructed.
A model of a four-wheeled vehicle has been developed for an AGV. The AGV travels along a circular arc during each time step. A control system has been developed to guide the AGV through an unknown environment. The AGV controller has a path planning algorithm that will determine the desired correction to the vehicle orientation. The path planning algorithm uses an ultrasonic sensor to detect obstacles. This ultrasonic sensor is mounted on the front of the vehicle and is rotated to produce a sensed area that the path planning algorithm uses to determine the presence of obstacles. An objective of the new path planning algorithm is to reduce the computational time of the search for a feasible path. This is done by dividing the algorithm into three separate procedures to reduce the amount of area that has to be scanned. The first procedure is used to guide the AGV when there is no obstacle in front of it. The second procedure is used to guided the AGV away from an obstacle or it can also be used to travel between obstacles. The final procedure used when the entire ultrasonic sensor sweep is saturated. This saturation would occur when the AGV enters an area such as a dead-end alley. Procedure three is capable of solving this problem. The path planning algorithm is used in conjunction with the fuzzy-logic controller because the amount of change in the steering depends on the current steering angle as well as the desired path.
A novel method for the design of a fuzzy-logic controller was proposed. This method is separated into two parts. In the first part, fuzzy inference rules of the controller are automatically generated based on the user's observations and the guidelines proposed. The objective of these guidelines is to ensure smooth transition between various neighboring activated inference rules by adding numerical weights to all membership sets. In the second part, the number and the shape of trapezoidal linguistic membership sets that describe the inputs and the outputs of the controlled system are varied to optimize a performance index using a modified form of Hooke-Jeeves nonlinear programming method. These ideas are applied to the development of a fuzzy-logic controller for an AGV.

Testing the controller is composed of two different stages. The first stage of the test is done by varying the time step, velocity, and the range of the ultrasonic sensor. The second stage involved further testing of the controller using the best results of the first stage of the testing.

The research presented in this thesis has produced a reliable controller for an autonomous guided vehicle. The path planning algorithm could be expanded to allow for more situations that may arise such as having the target in a long dead-end alley and the vehicle on the other side of the wall but close to the target. In this situation the controller would have to be capable of travelling in a direction that contradicts procedure one and procedure two. The path planning algorithm should also be expanded to be able to calculate a desired path with moving obstacles. This is a complex problem but it is important for any practical
intelligent vehicle  The design of a fuzzy-logic controller that will control that acceleration pedal angle change could easily be implemented to control the velocity. This would alleviate the problems that occurs when the velocity of the vehicle was the cause when crashing. To allow the vehicle to move across terrain with varying elevations (true 3-Dimensional terrain), a velocity and steering fuzzy-logic controller will have to be integrated with the path planning algorithm to ensure that the vehicle does not roll over.
REFERENCES


H. Berenji (a), "An Architecture for Designing Fuzzy Controllers Using Neural Networks", *Journal of Approximate Reasoning*, vol. 6, pp. 267-292, 1992

H. Berenji and P. Khedar (b), "Learning and tuning fuzzy logic controller through reinforcements", *IEEE Transactions on Neural Networks*, vol. 3, 1992


C. Isik, "Identification and Fuzzy Rule-Based Control of a Mobile Robot Motion", *IEEE Conference on Robotics and Automation*, pp. 94-99, 1987


F. Pin and Y. Watanabe, "Driving a Car with Custom-Designed Fuzzy Inferencing VLSI Chips and Boards", *Proceedings of the Third International Workshop on Neural Networks and Fuzzy Logic*, vol. 2, pp. 331-341, 1992


