Fuzzy Cooperative Control for a Team of Mobile Robots

by

Hussain Aftab

A THESIS

Submitted to

Department of Electronics and Power Engineering

Pakistan Navy Engineering College, Karachi

National University of Sciences and Technology, Islamabad

Supervised by

Dr. Bilal Kadri

Co-Supervised by

Cdr. Dr. Tariq Mairaj Khan

In fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

with specialization in Control Systems

August 2016

Title of Thesis:

Fuzzy Cooperative Control for a Team of Mobile Robots

Submitted by:

Hussain Aftab (Reg. No. NUST201260491MPNEC45012F)

Supervised by:

Dr. Bilal Kadri

Assistant Professor

Co-Supervised by:

Cdr. Dr. Tariq Mairaj Khan PN

Assistant Professor

Guidance and Examination Committee:

Cdr. Dr. Tariq Mairaj Khan PN

Assistant Professor

Dr. Bilal Muhammad Khan

Assistant Professor

Dr. Bilal Khawaja

Assistant Professor

بسم الله الرحمن الرحيم

This work is dedicated to my family…

ACKNOWLEDGEMENT

First and foremost I would like to thank ALLAH the almighty without whom all of this wouldn’t have been possible. I would like to express my gratitude to my supervisor Dr. Bilal kadri who’s continuous support and dedicated assistance helped my through my thesis. His patience, motivation and complete understanding of the subject has always been the source of inspiration for me. The moral support and encouragement he provided helped to complete this work with convenience.

I would also like to thank my co-supervisor Cdr. Dr. Tariq Mairaj Khan for his continuous support, guidance and encouragement which really helped me to go the extra mile and successfully complete my thesis.

I am very grateful to my guidance committee members for their reviews and ideas which has always helped me in improving my research.

I would like to thank National University of Science and Technology (NUST) and, my college Pakistan Navy Engineering College (PNEC) for the research environment and the highly qualified faculty they provided.

My earnest gratitude to Data Communication & Control Pvt Ltd (D.C.C), the Management and my team members who has always encouraged and supported me in completing my masters.

Finally I am grateful to my family members, without their support and motivation none of this could have ever happened.

ABSTRACT

The idea of controlling a team of robots is intriguing, that's why it has been extensively studied and researched for the past few decades. This work focuses on the formation control for a team of mobile robots. Path following and Cooperation seems an easy task from a human point of view, however for a robot it requires complex control algorithms. Therefore to mimic the human way of thinking, the proposed control algorithm is based on Fuzzy Logic which closely matches the human thought process. A Hierarchal control architecture is used with two levels of controllers which are Fuzzy based and Proportional Integral and Derivative (PID) based controllers. Fuzzy Controller is the high level controller and it has two task, Path Following and Cooperation between robots. Whereas the PID controller is responsible for accurate tracking of speed for the robot wheel motors. Every robot has its own path that is pre-determined by the trajectory planner, as the robot move along their respective path the control algorithm adjust their linear and angular velocities such that they move in formation and reach their destinations at same time regardless of the length and curvature of path. The proposed control scheme is implemented on simulation and the results are obtained.

Table of Contents

Table of Figures

1 Introduction

Formation control of multi mobile robots has been a topic of interest for past several years. A lot of research is being done on how to effectively control multiple mobile robots (MMR) to perform a collective task. The idea of formation control is universal and applies to almost all kinds of robots. For example a fleet of autonomous air vehicles use the formation control algorithms to test the aerodynamics of such vehicles. Control of Multi mobile robots have a vast range of applications such as unmanned air vehicles [1] – [3], autonomous ground vehicles [4], [5], mine sweepers [6] etc.

The primary motivation of using multi mobile robots is to improve the overall effectiveness of the system. MMRs can easily perform difficult task which a single robot or a team of independent (Non – Cooperative) robots cannot perform such as to carry a heavy object. Furthermore MMRs are cost effective in comparison of building one powerful robot several; identical small robots are much more cost effective and efficient.

Cooperative control of mobile robot is a challenging task, in this problem the robots are subjected to both path following and formation maintaining constraints. The idea of real time path following and formation control seems easy for humans, but for a robot it is a difficult task specially to decide between the individual and group goals.

The idea is to make such a model which can perform individual robot path following simultaneously maintaining an inter robot formation. The objective is that the robots should be able to follow their respective paths as well as maintain an inter robot formation. The robots may have different paths with different lengths and curvatures still the robots should track their path, maintain formation and reach the target at the same time regardless of the path length and their initial position.

Chapter # 2 deals with the Modeling and Literature Review. In this chapter existing research on this topic has been covered with references along with the mathematical models that are used in this thesis. Chapter # 3 discusses the Control architecture. The control problem is defined and both High Level Fuzzy Controller and Low Level PID Controller has been discussed in detail. Chapter # 4 define the simulation setup and conditions, it presents a detailed analysis of each simulation result. The results include the Speed PID Controller, Fuzzy Path Following and Cooperative Path Following. Chapter # 5 present the conclusion and future works.

2 Literature Review

2.1 Cooperative control of mobile robots

Cooperative control for a team of mobile robots is challenging task. The task is to develop a model that can perform cooperation between robots while following its own path. From human point of view, it seems like a trivial task but for a robot it is a difficult task and requires some complex control algorithms.

The goal was to make use of the human thought process for the cooperative behavior for this purpose several things needs to be considered. The robot should be able to track the path and send its location to the controller; therefore the robot must have localized feedback. Several approaches are being used for localization of a robot in its environment [7] – [9].

To achieve cooperation between robots three different models were studied to test their performance and applications.

- Behavior based model
- Leader follower method
- Virtual structure

In behavior based model the robot consist of several behaviors, each of which is responsible for a specific situation [10] – [12]. The problem with this type of model is that it does not support decomposition of task or the modeling of sub task. Furthermore it difficult to achieve precise cooperative control as it is mathematical analysis is difficult.

In the leader follower method one of the robots is the leader and the others are the followers [13]. In this scheme the leader is the one which responsible for the formation, all other robots are supposed to follow the leader and maintain a certain distance (relative position) from the leader. The disadvantage of this scheme is that there is no feedback available from the follower robots and hence if anyone of them encounters an obstacle then it is difficult to keep up with the formation.

In virtual structure approach the system consider the team of mobile robots as a rigid structure (virtual structure) [14], [15]. This approach is easier to implement as it all the robots are part of virtual structure hence the cooperation is easier, but it can be also a point of failure.

Due to its simplicity and ease of implementation the virtual structure approach was selected. However, to compensate for the weakness of this method a more streamlined control architecture was required.

Background research for the control algorithm reveals that the control algorithm used for cooperative control of mobile robots are based on Nonlinear Control techniques which often require complex mathematical equations to be implemented, which are difficult to be realized in a microcontroller environment, therefore Fuzzy control was selected as it mimics human logic and it is easier to implement and it has fast decision making process which makes it an ideal candidate for multiple mobile robots scenario.

The model used in this thesis is based on the virtual structure with a fuzzy controller. There has been a lot of work being done artificial intelligence for path following of mobile robots. The fore study of path following techniques tells that Neural Network and Fuzzy Logic Controllers are commonly used for path following of mobile robots for both “Known” and “unknown” environments [16] – [19]. There are researches which exploited both these techniques also known as neurofuzzy controllers [20], [21]. However, most of these researches were carried out on single mobile robots but with a little modification these can be implemented in a cooperative control environment.

It was decided to implement a Hierarchal Control structure in which there are two different hierarchies namely High level Fuzzy Controller and Low Level PID controller. The Fuzzy Controller is the main controller which is responsible for the cooperation, path following and localization of robots. While Low level controller is a PID controller which is responsible for the speed tracking for each of the robot's wheel motor.

2.2 Differential drive robot

Differential drive robot comprises of two separately powered wheels mounted on either side (left and right) of the robot along with an optional caster wheel for balancing purpose. The direction of the differential drive robot can be controlled by changing the relative speed between the rotation wheels as described below.

- If both wheels of the robot are rotating in the same direction with equal angular velocity then the robot moves in straight line.
- If both wheels are rotating the same direction with the left wheel rotating faster than the right one, then the robot will move in curved path towards right and vice versa.
- If both wheels are rotating with equal speed but in opposite direction, then the robot will rotate on its axis.

Figure 1 shows the above cases.

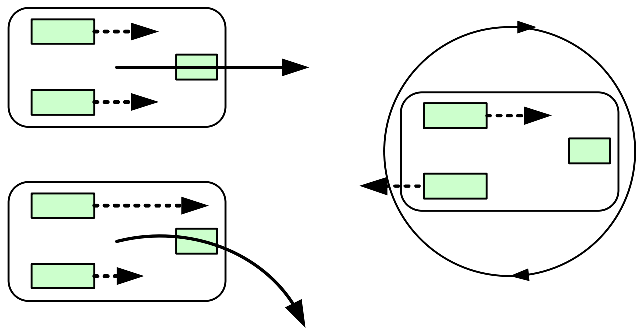

Figure 2-1 Differential Drive Robot

Differential Drive robots are often used for control purposes because of their relatively simple mathematical model and ease of implementation. In this thesis we used Differential Drive robot models for testing our path following and cooperative control algorithms.

2.3 Permanent magnet dc motor

Permanent magnet dc motor is a very commonly used type of dc motor. These motors are used in wide range applications including gear motors, heaters, pumps air conditioners of car and personal computers etc. These motors doesn't require any excitation current as there is no field winding, rather the permanent magnet provides the magnetic flux.

A PMDC consist of two parts stator and rotor. The stator is fixed with permanent magnets while the armature is the rotating part. The working principle of the dc motor is simple, whenever a current carrying conductor (armature) is placed inside a magnetic field it will experience a force (Torque). The magnitude of the force is directly proportional to the amount of the current flowing through the conductor while the direction of the force is determined by the direction of current through the conductor.

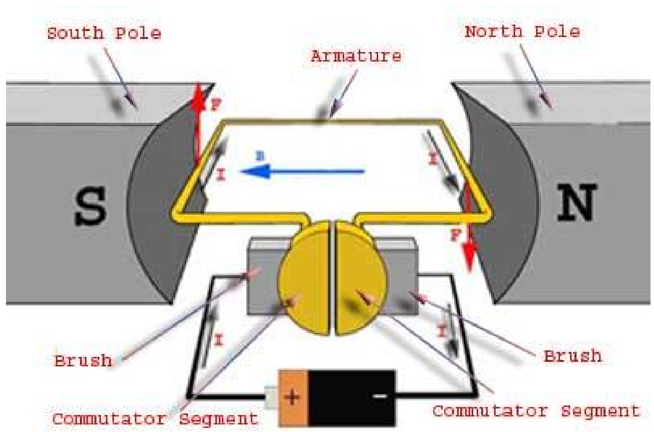

Figure 2-2 Permanent magnet dc motor

3 Modeling

To simulate the robots and test the cooperative control scheme we must first have the mathematical models of the system. In this chapter we discuss the modeling of the differential drive robot and the permanent magnet dc motor along with the system constraints.

3.1 Modeling of differential drive robot

Figure 2-1 represents a simple uni cycle type differential drive robot, describing the two different frame of references. $\{X_m, Y_m\}$ Being the moving frame or the robots frame of reference while $\{X_b, Y_b\}$ being the base frame or world frame of reference.

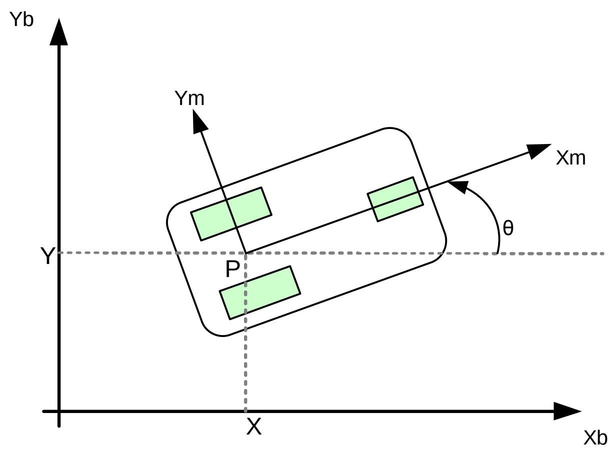

Figure 3-1 Frame of reference

The robots position in the base frame is given by$P = [X, Y, \varphi]^T$. The rotation matrix R is used to express the orientation of the base frame with respect to the moving frame.

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The kinematic model of a unicycle type mobile robot can be expressed by the following set of nonlinear equations [].

$$\begin{cases} \dot{x}(t) = v(t)\cos\varphi(t) \\ \dot{y}(t) = v(t)\sin\varphi(t) \\ \dot{\varphi}(t) = \omega(t) \end{cases} \tag{3.2}$$

Where $v(t)$ is the linear velocity of the robot and $\omega(t)$ is the angular velocity of the robot. $v(t)$ and $\omega(t)$ are the control variables. To derive the relationship of these two control variables for a differential drive type robot let us consider the figure 2-3.

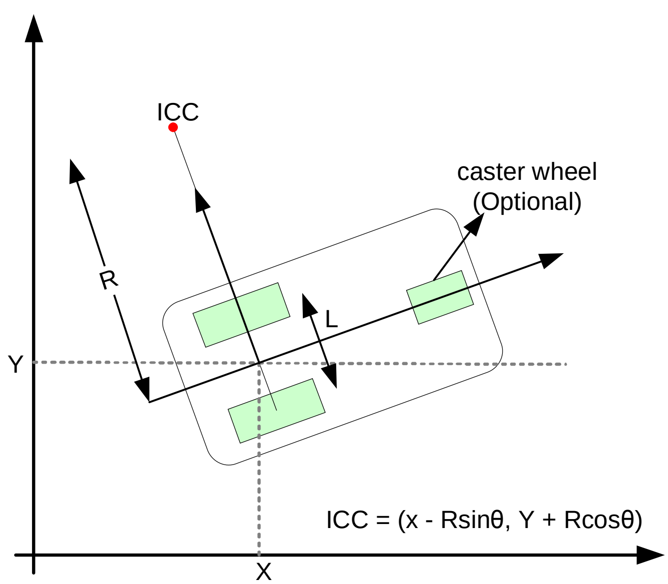

Figure 3-2 Instantaneous center of curvature ICC

Figure 2-3 shows a differential drive robot with two wheels mounted on a common axis. If the wheels are rotating considering there is no lateral slip then there exist a point ICC (instantaneous center of curvature) provided that ($v_r \neq v_l$). Where v_r and v_l are the linear velocities of the right and left wheels respectively. By varying v_r and v_l we can change the ICC and hence we can control the movement of the robot. R is the distance from the ICC to the center of the two wheels which is also considered as the center of the robot. L is the distance between the two wheels. And θ is the orientation of the robot from x axis.

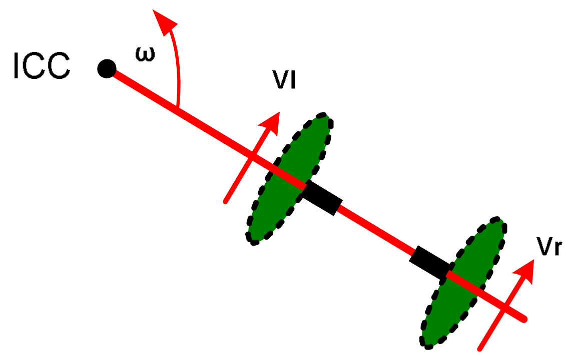

Figure 3-3 Differential drive

The speed of the wheel (Linear speed) $v = 2\pi R/T$ where T is the time to complete one rotation around ICC. Similarly the angular speed ω is given by $\omega = 2\pi/T$ which

$$\omega = \frac{2\pi}{T}$$

$$\omega = \frac{2\pi}{T}\frac{R}{R}$$

$$\omega = \frac{v}{R} \quad or \quad v = R\omega$$

Note that plugging in R(distance from ICC) and v results in the same ω for both wheels or else the two wheels will move relative to each other, hence the following equations holds. Refer to figure 3-3.

$$v_r = (R + \frac{L}{2})\omega$$

$$v_l = (R - \frac{L}{2})\omega$$

Now solving for R and ω we get the following equations.

$$R = \frac{L}{2}\frac{(v_r + v_l)}{(v_r - v_l)}$$

$$\omega = \frac{(v_r - v_l)}{L} \tag{3.3}$$

By using the relationship $v = r\omega$ we get.

$$v = \frac{(v_r + v_l)}{2} \tag{3.4}$$

By using equation (3.3) and equation (3.4) and solving for v_r and v_l we get the following relations.

$$v_r = \frac{(2v + \omega L)}{2} \tag{3.5}$$

$$v_l = \frac{(2v - \omega L)}{2} \tag{3.6}$$

Note that the velocities v_r and v_l are linear velocities of the right and left wheel respectively.

3.2 Modeling of Permanent Magnet DC Motor (PMDC)

In order to develop a mathematical model of a PMDC motor both electrical and mechanical dynamics must be considered. Figure 3-4 shows an equivalent electrical circuit of a PMDC motor where V_a is the armature voltage (applied voltage), $R_a$is the equivalent series resistance of the motor winding, $L_a$is the inductance of the motor winding and V_c is the induced voltage.

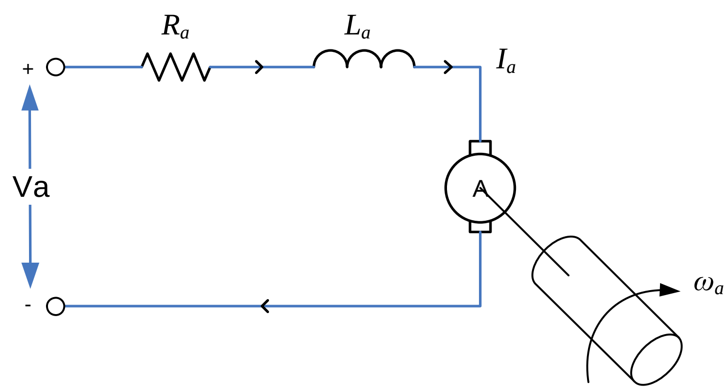

Figure 3-4 Equivalent circuit of PMDC motor

From figure a differential equation can be derived using Kirchhoff's voltage law which is as follows.

$$V_a - V_{Ra} - V_{La} - V_c = 0 \tag{3.7}$$

The voltage across the resistor can be found using ohm's law

$$V_a = i_a * R_a$$

We know that the voltage across the inductor is directly proportional to the change in current therefore.

$$V_{La} = L_a \frac{d}{dt} i_a$$

Finally the induced voltage V_c or back emf can be written as

$$V_c = K_e * \omega$$

Where K_e is the electrical time constant of the motor. Substituting these values in equation (3.7) we get the following differential equation.

$$V_a - i_a R_a - L_a \frac{d}{dt} i_a - K_e \omega = 0 \tag{3.8}$$

To model the mechanical subsystem we must perform an energy balance of the system i.e. the sum of all torques must be zero.

$$T_e - T_{\dot{\omega}} - T_{\omega} - T_L = 0 \tag{3.9}$$

T_e is the electromagnetic torque or induced torque, $T_{\dot{\omega}}$ is the torque due to the rotational acceleration of the rotor, T_{ω} is the torque due to the rotational speed of the rotor and T_L is the load torque.

The electromagnetic torque is proportional to the current through the winding.

$$T_e = K_t * i_a$$

Where K_t is the torque constant. $T_{\dot{\omega}}$ can be written as.

$$T_{\dot{\omega}} = J \frac{d}{dt} \omega$$

Where J is the moment of inertia of the rotor and equivalent mechanical load. The velocity torque can be written as.

$$T_{\omega} = B * \omega_a$$

Where B is the damping coefficient of the motor. Substituting these values in equation (3.9) we get the following differential equation

$$K_t i_a - J\frac{d}{dt}\omega - B\omega - T_L = 0 \tag{3.10}$$

Rearranging the equations (3.8) and (3.10) we get the final model of the pmdc motor.

$$\begin{cases} \frac{d}{dt}i_a = -\frac{R_a}{L_a}i_a - \frac{K_e}{L_a}\omega + \frac{V_a}{L_a} \\ \frac{d}{dt}\omega = -\frac{B}{J}\omega + \frac{K_t}{J}i_a - \frac{T_L}{J} \\ \frac{d}{dt}\theta = \omega \end{cases} \tag{3.11}$$

3.3 Constraints

The differential drive robot is subject to non-holonomic constraints such as it cannot move latterly along its axel as shown in figure.

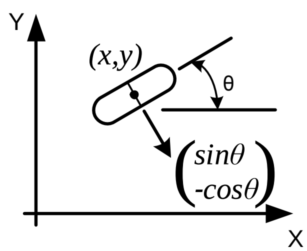

Figure 3-5 Non-holonomic constraints

Let $P = [x, y, \theta]^T$ define the position of the robot then $\dot{P} = [\dot{x}, \dot{y}, \dot{\theta}]^T$ from above figure we can see that the robot has one constraint. Let w represent the conditions of the constraint.

$$w = \begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix}$$

The constraint can be written as

$$wP = 0$$

$$\begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0$$

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{3.12}$$

Similarly the Linear and angular velocities of the robot are also subject to following constraints:

$$|v(t)| \le v_{\max} \qquad |\omega(t)| \le \omega_{\max}$$

$$|\dot{v}(t)| \le \dot{v}_{\max} \qquad |\dot{\omega}(t)| \le \dot{\omega}_{\max}$$

4 Cooperative Control of Mobile Robots

In the previous chapter we have defined the mathematical models of DC motor model and the Differential Drive robot. This chapter focuses on the control scheme used in this thesis, it defines the hierarchical control architecture, sets the control objective, explains in detail the Fuzzy Path Following and Cooperative Controller and Trajectory Planner.

4.1 Cooperative Control Structure

The control structure used for the cooperative control of multi mobile robots is shown in figure 4-1. The control strategy has been divided into two parts

1. High level Fuzzy Cooperative Control
2. Low Level PID Control

The trajectory planner generates the trajectories for each robot and gives the reference points to the High level fuzzy controller which then generates an output signal for the Low level PID controllers. The Low level PID controller is responsible for accurate tracking of Robot's Left and Right motor speeds while the High Level controller is responsible for the cooperation of robots as well as path following of individual robots. The feedback is collected from the Robot's wheel motors in the form of speed and then a localization algorithm is used to locate each robot's position and orientation which is then sent back to the Fuzzy controller.

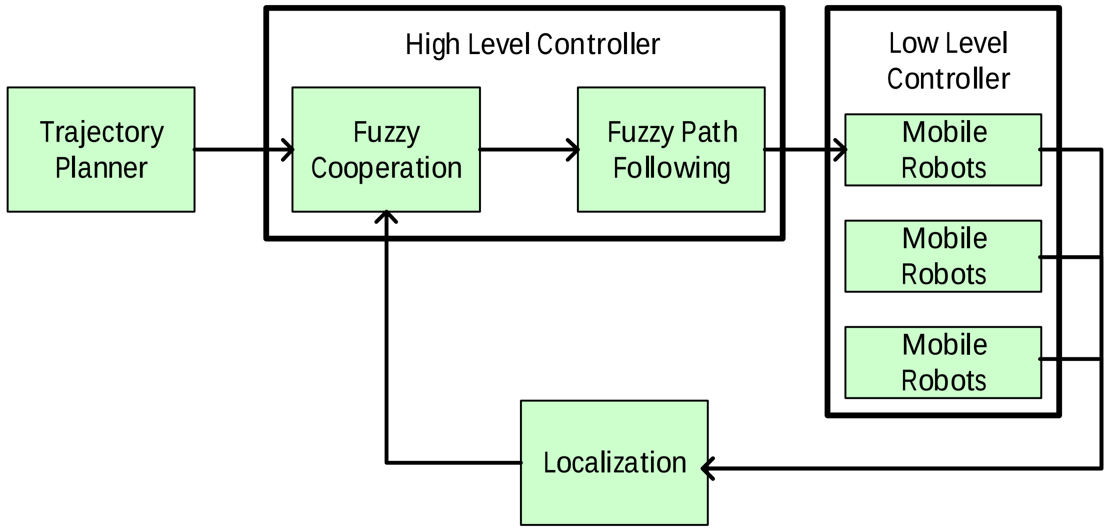

Figure 4-1 Multi Mobile Robot Cooperative Control Structure

4.1.1 High Level Fuzzy Cooperative Control

The High Level Fuzzy Controller is a Takagi sugeno type fuzzy controller which is responsible for the path following of the individual mobile robots as well as the group cooperation between the robots. The defuzzification scheme used in the fuzzy controller is weighted average. Figure 4-2 shows the Fuzzy Control structure.

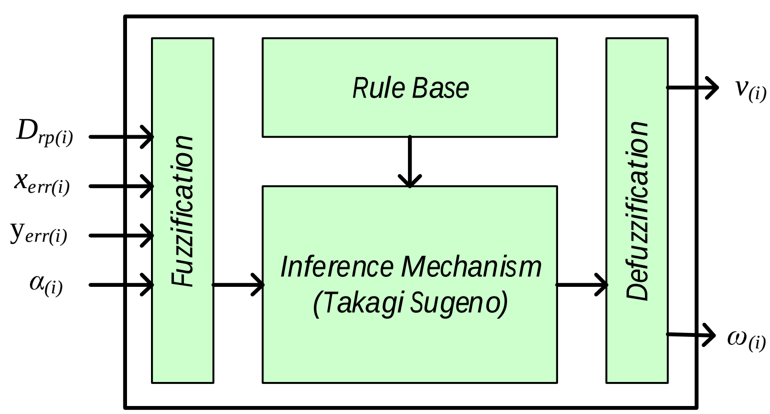

Figure 4-2 Fuzzy Control Structure

To explain the inputs to the fuzzy controller refer to figure 4-3. From figure $i = 1, \ldots, k$ is the robot number. The position and orientation of the robot can be described by the vector $P_i = [x_i, y_i, \varphi_i]^T$.The path to be followed by the robots is divided into n discrete set of points, where $n = 0, \ldots, f$. Each point on the path can be described by the vector$Q_{di(n)} = [x_{di(n)}, y_{di(n)}, \xi_{di(n)}]^T$. Where $Q_{di(0)}$ being the starting point of the path and $Q_{di(f)}$ being the final point. Whereas$Q_{di(n)}$ represent the nth sample point on the path.

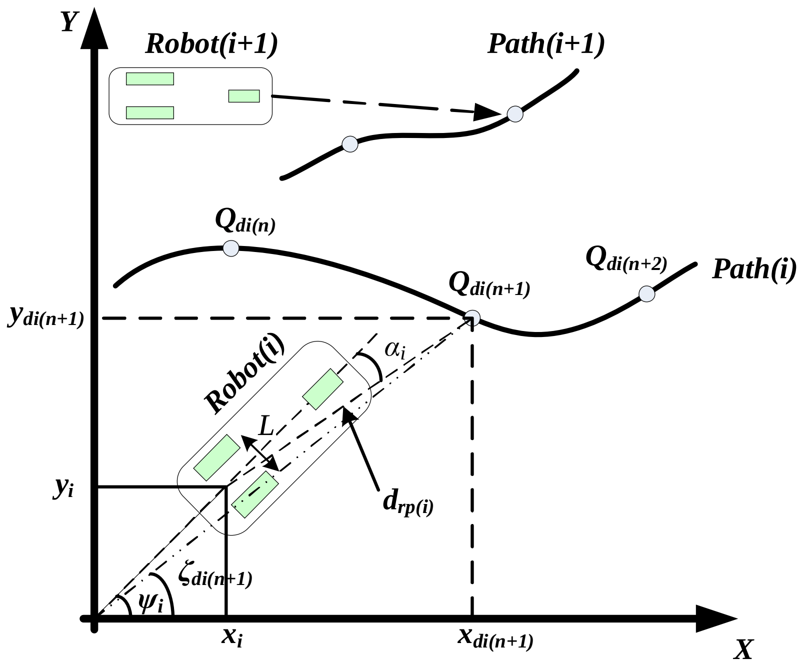

Figure 4-3 Robot path following parameters

The Inputs to the Fuzzy Controller are $D_{rp(i)}$, $x_{err(i)}$, $y_{err(i)}$ and $\alpha_{(i)}$. Where

- $D_{rp(i)}$ is the distance of the actual position of the ith robot from its next desired point on the path.
- $x_{err(i)}$ is the error of the ith robot's position the X direction.
- $y_{err(i)}$is the error of the ith robot's position in the Y direction.
- $\alpha_{(i)}$ is the error in orientation of ith robot from the next desired point.

$D_{rp(i)}$ is calculated using the distance formula as shown in the following equation.

$$D_{rp(i)} = \sqrt{(x_{di} - x_i)^2 + (y_{di} - y_i)^2} \tag{3.13}$$

Where x_{di}, y_{di} are the coordinates of the next desired point on the path while x_i, y_i are the coordinates of the actual position of the robot.

$x_{err(i)}, y_{err(i)}$ and $\alpha_{(i)}$ are in the robot's reference frame and are calculated from the following equation. Refer to equation (3.1) for the rotation matrix.

$$\begin{bmatrix} x_{err(i)} \\ y_{err(i)} \\ \alpha_{(i)} \end{bmatrix} = \begin{bmatrix} \cos\varphi_{(i)} & \sin\varphi_{(i)} & 0 \\ -\sin\varphi_{(i)} & \cos\varphi_{(i)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{di} - x_i \\ y_{di} - y_i \\ \zeta_{di} - \varphi_i \end{bmatrix} \tag{3.14}$$

Where φ_i is the robots orientation which is adjusted to $\pm\,\pi$ radians. ζ_{di} is the orientation of the next desired point which can be calculated via using the following equation.

$$\zeta_{di} = \tan^{-1}\left(\frac{y_{di} - y_i}{x_{di} - x_i}\right) \tag{3.15}$$

The Fuzzy controller generates the required linear and angular velocities v_i, ω_i of the robot. The output of the fuzzy controller is used as a set point the low level PID controller.

4.1.2 Low Level PID Control

The Low level control is a PID controller which is responsible for the accurate tracking of the Left and right wheel motors velocities $\omega_{r(i)}, \omega_{l(i)}$. Figure 4-5 shows the structure of the low level PID controller.

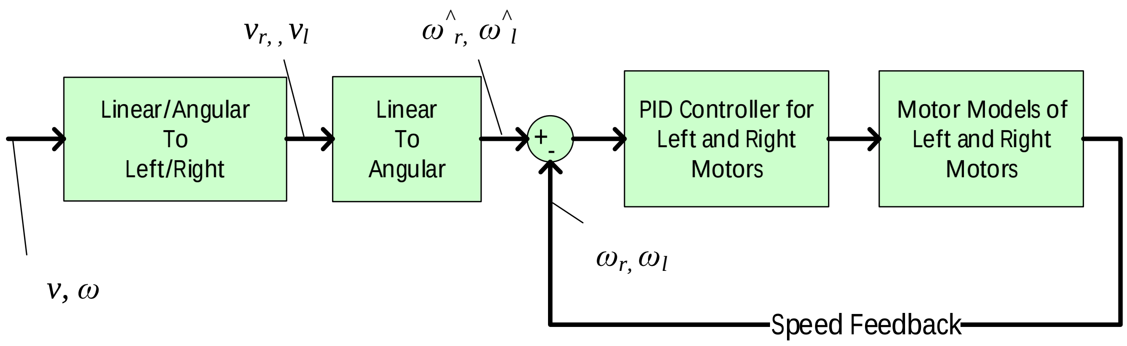

Figure 4-4 Low Level PID Control Structure

The Output from the Fuzzy Controller is in the form of Linear and Angular velocities v_i, ω_i of Robot therefore the velocities must first be converted to Left and Right wheel motor velocities $v_{r(i)}, v_{l(i)}$. This conversion is carried out using equation (3.5) and (3.6).

The above conversion gives the linear velocities of the left and right wheel but the PID controller needs the angular velocity as reference. Therefore the angular velocities $\omega_{r(i)}, \omega_{l(i)}$ of the right and left motors can be obtained using following conversion.

$$v_i = r\omega_i$$

Where r is the radius of the right and left wheels of the robot.

The velocities $\omega_{r(i)}, \omega_{l(i)}$ are given to the PID controller as reference signal. The PID controller tracks these velocity references with desired accuracy and within certain time limits. The whole functionality of the High Level Fuzzy Cooperative Control is dependent on the assumption that the inner PID controller accurately tracks these velocity references.

4.2 Path Following and Cooperation Problem

The two main tasks of the robot is to follow a desired path as well as maintain a formation with other robots along the journey. The main idea behind the robot following a continuous path is to analyze the path in a set of discrete points. Each point serves as an intermediate destination to the robot, tracking each point will make the robot appear to be moving smoothly in a continuous path. The paths are being generated by the trajectory generator, the paths are modeled using a fifth order polynomial to generate reasonably difficult curved paths for the robots to follow.

The task of the robot is to move smoothly in a continuous path with best precision possible, passing through every point is not necessary but the robot should pass within the vicinity of a sampling point. The path following is divided into two categories.

1. When the robot is initially placed on the predefined path it then follows the path.
2. When the robot is initially placed away from the predefined path, it then moves forward to reach the path and track it.

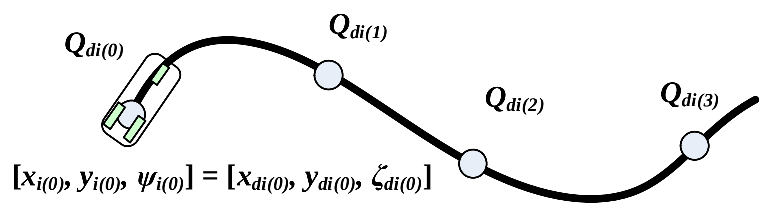

Figure 4-5 Robot Placed on the Path

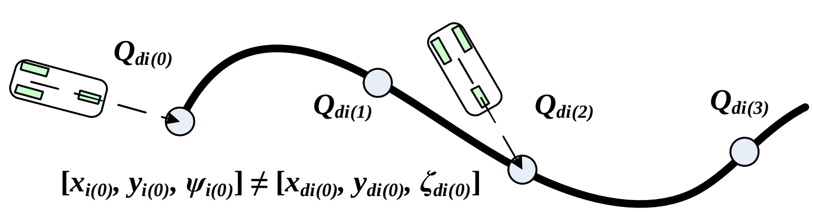

Figure 4-6 Robot not placed on path

As mentioned previously P_i being the Robot's position and Q_{di} being the next desired point we have considered v_i and ω_i as robot's linear and angular velocities. The objective of the path following controller is to generate an output such that the robots velocity$u_i = [v_i, \omega_i]^T$tracks the desired reference velocity$u_{di} = [v_{di}, \omega_{di}]^T$.As the robot's velocity u_i tracks the desired reference velocity $u_{di}$the error minimizes i.e. $\|u_i - u_{di}\| \to 0$ which will eventually make $\|P_i - Q_{di}\| \to 0$

Now consider a team of mobile robots each of which having its own path following controller like the one described above, hence every individual robot will follow its desired path. For the cooperation problem every robot must follow their respective path in such a way that they maintain an inter robot formation along their journey as well as they must reach their final goal at the same time regardless of their path lengths.

4.3 Fuzzy Path Following and Cooperative Controller

The Fuzzy Controller of Figure 4-2 is responsible for the path following and cooperation of multiple mobile robots it has two outputs v_i being the linear velocity and ω_i being the angular velocity for the robot to track. The fuzzy controller is based on Takagi - Sugeno Fuzzy Model. The control law equations are of the form.

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} f_1(D_{rp(i)}, x_{err(i)}, y_{err(i)}, \alpha_{(i)}) \\ f_2(D_{rp(i)}, x_{err(i)}, y_{err(i)}, \alpha_{(i)}) \end{bmatrix}$$

As mentioned previously, the task of the fuzzy controller is to make the robot pass within the vicinity of the desired sampling point if not through it. The controller is designed such that if the sampling points are placed close to each other, then the robot will move at a slower speed but with higher precision. However if the sampling points are placed far from each other, then the robot movement will be less precise but with higher speed.

4.3.1 Membership Function $D_{rp(i)}$

The membership functions of $D_{rp(i)}$is shown in figure 4-7.$D_{rp(i)}$ is the distance of the robots actual position to the next desired point, the input has five membership functions namely **VeryClose**,**Close**, **Medium**, **Far** and **VeryFar**. it is measured in meters (m). The membership functions **VeryClose**, **Close**, **Medium** are placed close to each other this helps the robot maneuver more precisely but with lower speed when the desired sampling point is in close vicinity, however the **Far** and **VeryFar** membership functions covers large region and are placed far from each other this helps the robot move at higher speed but with less precision. Note that all the membership functions have overlapping regions this helps the robot to move smoothly during the transition from one membership function to another.

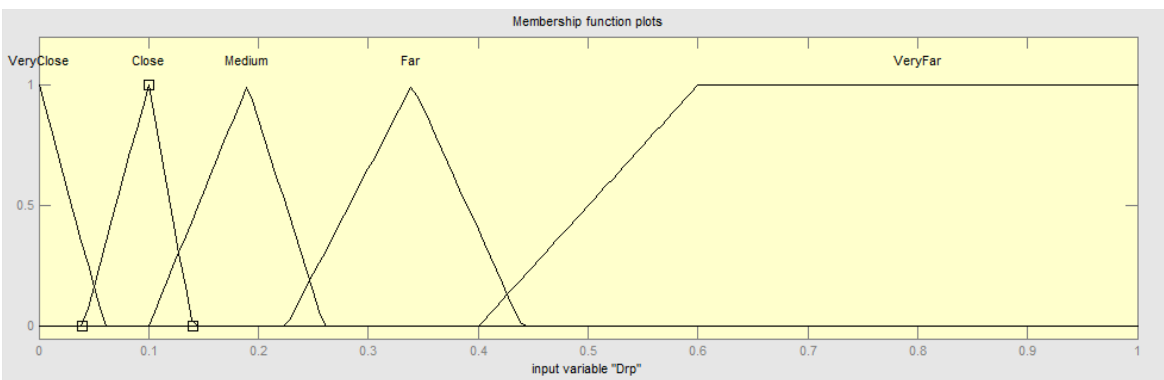

Figure 4-7Membership Function of $D_{rp(i)}$

4.3.2 Membership Functionα_i

The membership function of α_i is shown in figure 4-8. α_i is the error in orientation of the robot to that of the next sampling point. It has five membership functions namely **BigNegative**, **mNeg**, **Small**, **mPos**, **BigPositive** it is measured in degrees. The membership functions **mNeg**, **Small** and **mPos** are placed closely to make fine adjustments to the robots orientation when the error is small, while **BigNegative** and **BigPositive** are placed far to compensate for the large errors in orientation and quickly overcome the orientation error. Here also the membership functions have overlapping regions to have a smooth transition between membership functions.

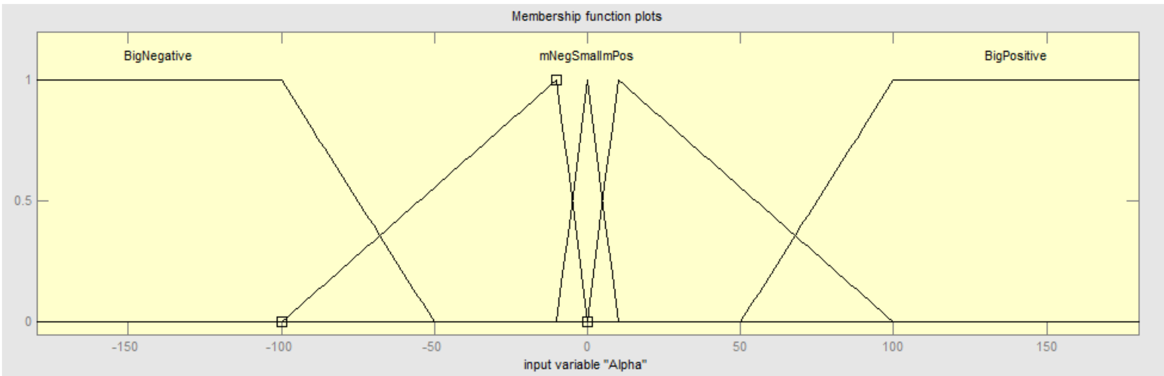

Figure 4-8 Membership Function of α_i

4.3.3 Linear and Angular Velocity obtained from $D_{rp(i)}$ and α_i

The linear and angular velocities obtained from the fuzzy controller related to $D_{rp(i)}$ and α_i are shown in figure 4-9 and 4-10 respectively. The linear velocity is measured in m/s (meter per second) while angular velocity is measured in degrees/sec. From figure 4-9 it can be seen that the linear velocity increase as the distance between the robot and the next sampling point $D_{rp(i)}$ increases and vice versa. This way the robot moves faster when the sampling point is far and moves slower as it comes closer. Notice that the error in orientation α_i has minimal effect on the linear velocity and rightly so. However form figure 4-10 it can be seen that the angular velocity profile is dominated by the α_i input, as the error in orientation increases the robot angular velocity increases and vice versa, This helps the robot make turns quickly and slowly as the error in orientation increases and decreases respectively.

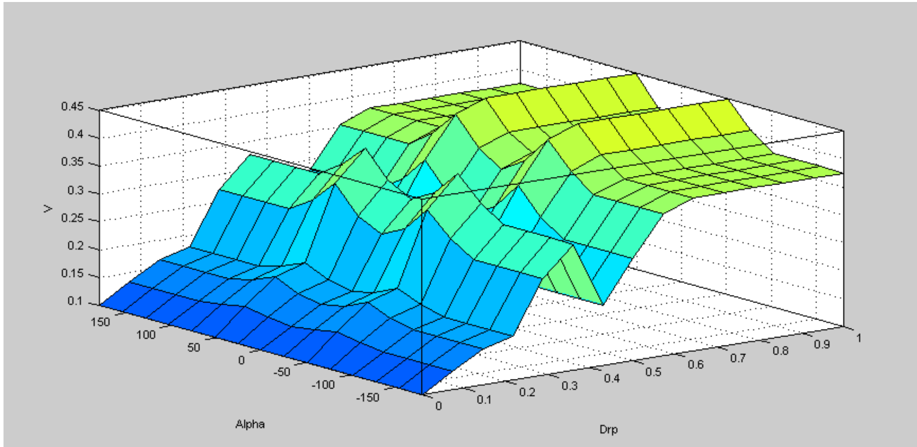

Figure 4-9 Linear Velocity obtained by Fuzzy Controller for inputs $D_{rp(i)}, \alpha_i$

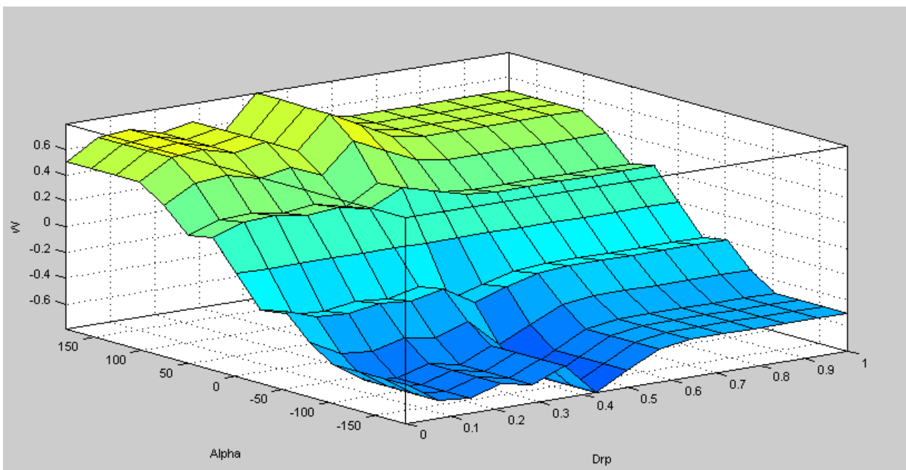

Figure 4-10 Angular Velocity obtained by Fuzzy Controller for inputs $D_{rp(i)}, \alpha_i$

4.3.4 Robot ahead of its Path Problem

One of the problems of cooperative path following is that all robots should reach their final goal at the same time. Let us consider a case where the robot is at position $P_i = [x_i, y_i, \varphi_i]^T$ has to move from point $Q_{di(n)} = [x_{di(n)},\ y_{di(n)},\ \xi_{di(n)}]^T$ to $Q_{di(n+1)}$ this means that the robot next desired sampling point is $Q_{di(n+1)}$. If the robot passes the point $Q_{di(n+1)}$ and $Q_{di(n+2)}$ thenwhen moving to next step the $Q_{di(n+1)}$ is left behind from the actual position of robot or mathematically ($x_i > x_{di} \Rightarrow\ x_{err(i)} < 0$). This probelm is explained in figure4-11.To avoid the condition of robot moving backwards we use the input $x_{err(i)}$. The fuzzy rule base contains the following rule to avoid the the above mentioned condtion.

- **Rule:** $if\ x_i > x_{di}\ then\ Robot\ will\ stop, untill\ the\ condition\ x_i < x_{di}\ is\ met.$

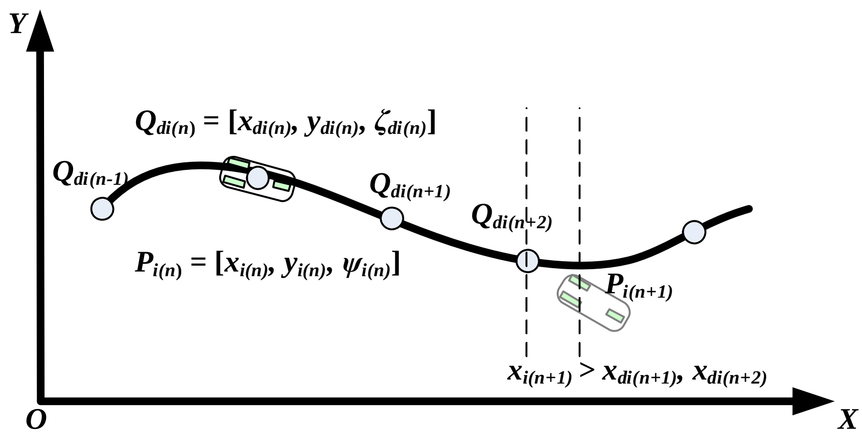

Figure 4-11 Robot ahead of the Target Point

4.3.5 Robot moving Parallel to its Path Problem

Another problem related to path following that needed to be addressed,caused the addition of the input $y_{err(i)}$.If there is an error in the vertical position of the robot $y_{err(i)}$but not in the orientation then the robot will traval parallel to the desired path and will never reach it. This can be seen from figure 4-12. The input $D_{rp(i)}$ has minimal effect on the Angular velocity therfore it cannot turn the robot towards the target point there need to be an error in orientation $\alpha_i$to turn the robot towards the target point.

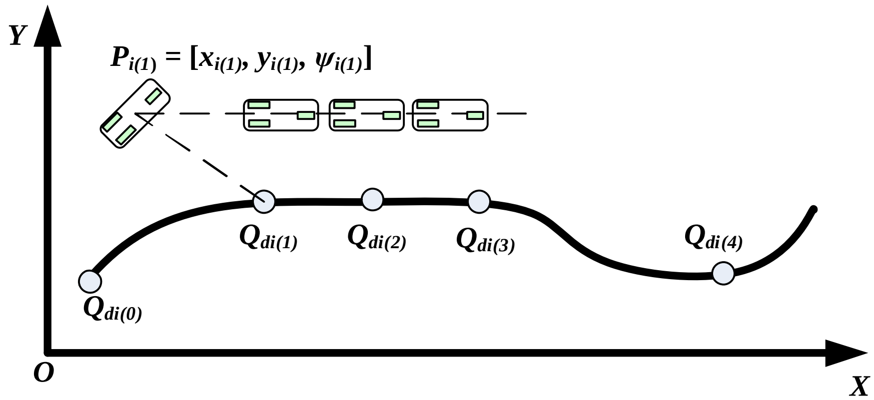

Figure 4-12 Robot travelling Parallel to the path

In order to address the above mentioned problem we introduced a new variable λ_i which is angle measured in degrees. λ_i is added to α_i if there is an error in y coordination. it will help the robot to turn towards the target point even if the $\alpha_i = 0$, it will tell the robot to turn towards the target point and will decrease as the $y_{err(i)}$decreases and will be zero when robot catch the desired path.

$$if\left|y_{err(i)}\right| > 0 \Rightarrow \alpha_i(new) = \lambda_i + \alpha_i$$

The membership function of $y_{err(i)}$ is shown in figure 4-13. It is divided into seven membership functions namely **nVeryFar**, **nFar**, **nClose**, **VeryClose**, **Close**, **Far**, **VeryFar** distributed symetrically from -3 to 3, it is measured in meters. The relationship between $y_{err(i)}$ and λ_i is shown in figure 4-14, it can be seen that λ_i increases as $y_{err(i)}$ increases and vice versa. Hence it makes sure that robot turn towards the target point if there is an error in the ycoordination.

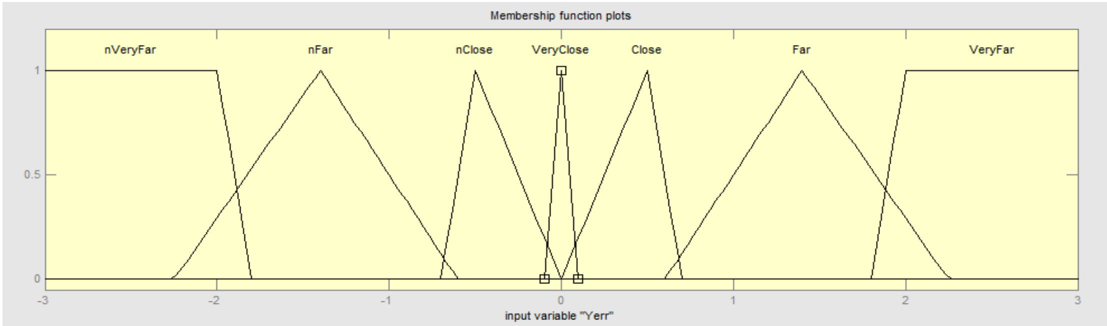

Figure 4-13 Membership function of $y_{err(i)}$

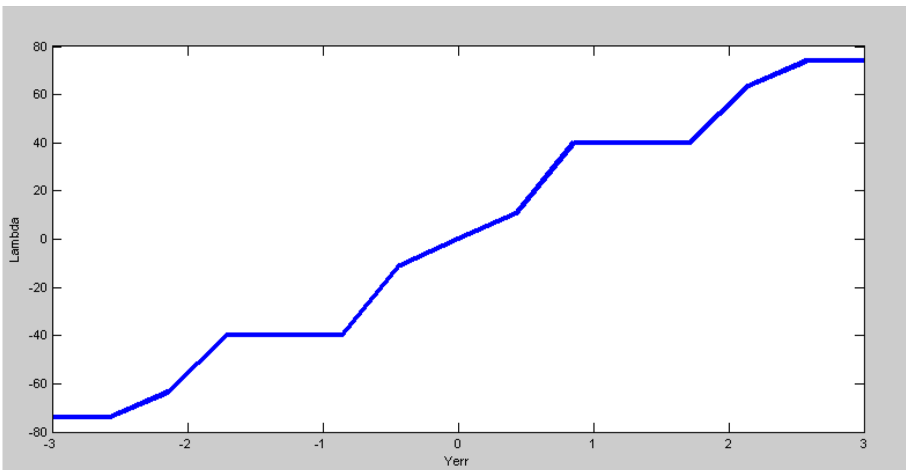

Figure 4-14 Lambda λ_i obtained form $y_{err(i)}$

4.3.6 Cooperation Using X-Coordination

The path of each robot is divided into equal number of small segment and the parameter ς_i will keep track of the current segment the robot is executing. The robots are required to maintain $\varsigma_i = \varsigma_j$ for all i, j. For cooperation it is necessary that all the paths are divided into equal number of segments regardless of their shape and length. If the path is longer than the sampling points will be far from each other or the distance between sampling point will increase and vice versa. Hence the robot will move faster or slower if the sampling point are close or far from each other respectively. Therefore if each robot track their respective sampling points then they will be moving in a cooperative behavior and will reach the final point at the same time. if a robot is ahead of its path then it will wait for other robots to catch up and then start following its path. This is achieved by x coordination error input i.e if $x_{err(i)} < 0$ then robot will stop. The effect of $x_{err(i)}$ and $D_{rp(i)}$on linear velocity $v_i$is shown in figure 4-15 and the obtained angular velocity$\omega_i$from the input$x_{err(i)}$ and $\alpha_i$is shown in figure 4-16.

It can be seen from figure 4-15 and 4-16 that if $x_{err(i)}$ is negative than both linear and angular velocities are zero.Hence if the robot is ahead of its path then it will not move and wait until the condition $x_{err(i)} < 0$ is present. However if the$x_{err(i)}$ is positive then the robot will move normally and track the desired path. However if the robot has fallen behind its desired path then it will move faster to catch the other robots once it catches the path then it will normally with the desired path as can be seen from figure 4-15.

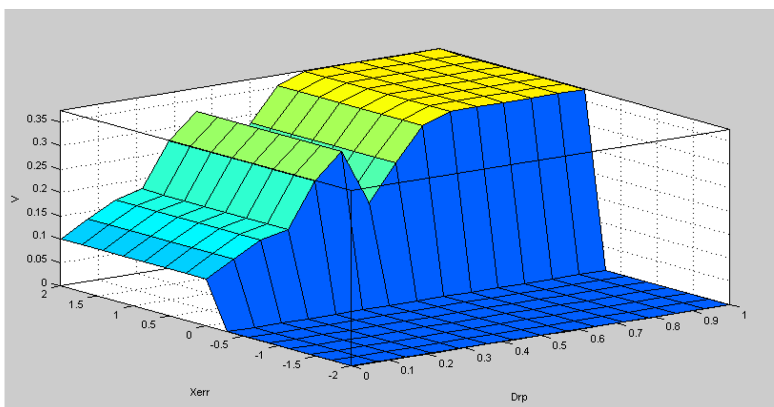

Figure 4-15 Linear Velocity obtained from $x_{err(i)}$ and $D_{rp(i)}$

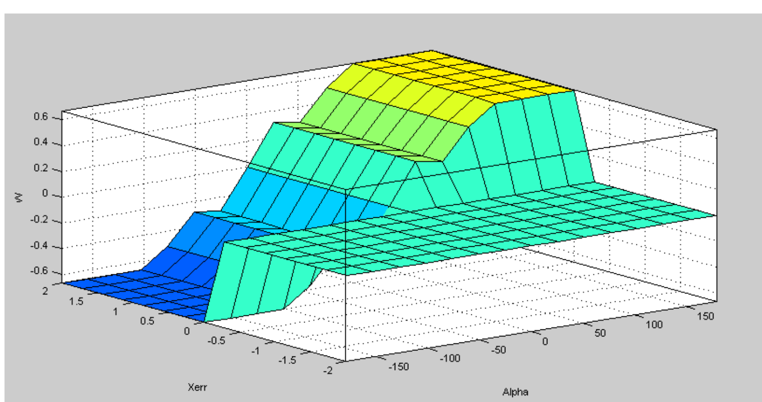

Figure 4-16 Angular Velocity obtained from $x_{err(i)}$ and α_i

4.3.7 Fuzzy Controller Rule Base

The Rule Base of the fuzzy controller is shown in Table 4-1. On the basis of these rules the fuzzy controller evaluates the inputs and generates the output. All the rules are very much self-explanatory like if the distance between the next sampling point and robot $D_{rp(i)}$ increase the fuzzy controller increases the linear velocity output so that the robot can catch its target. Similarly if the orientation error α_i increases then the fuzzy controller increases the angular velocity output to overcome the orientation error and correct the robot heading direction. Also if the robot is ahead of its path the Fuzzy controller will make both linear and angular velocity outputs zero to stop the robot.

Table 4-1:

Rule No	Fuzzy Rule Base
1	*If* $D_{rp(i)}$ *is* ***VeryClose*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***VeryVerySlow***, ω_i *is* ***Zero***
2	*If* $D_{rp(i)}$ *is* ***Close*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***VerySlow,***ω_i *is* ***Zero***
3	*If* $D_{rp(i)}$ *is* ***Medium*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Slow,***ω_i *is* ***Zero***
4	*If* $D_{rp(i)}$ *is* ***Far*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***Fast,***ω_i *is* ***Zero***
5	*if* $D_{rp(i)}$ *is* ***VeryFar*** *and* $X_{err(i)}$ *is* ***Positive*** *then* v_i *is* ***VeryFast,***ω_i *is* ***Zero***
6	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***BigNegative*** *then* v_i *is* ***VeryVerySlow,***ω_i *is* ***BigNegative***
7	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***MediumNegative*** *then* v_i *is* ***VeryVerySlow,***ω_i *is* ***Negative***
8	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***Small*** *then* v_i *is* ***VeryVerySlow,***ω_i *is* ***Zero***

9	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***MediumPositive*** *then* v_i *is* ***VeryVerySlow,*** ω_i *is* ***Positive***
10	*if* $X_{err(i)}$ *is* ***Positive*** *and* α_i *is* ***BigPositive*** *then* v_i *is* ***VeryVerySlow,*** ω_i *is* ***BigPositive***
11	*if* $X_{err(i)}$ *is* ***Negative*** *then* v_i *is* ***Zero,*** ω_i *is* ***Zero***

-

4.4 Trajectory Planner

The Idea of robot following a continuous path is to realize the path in set of discrete points. Each of the point are used as a destination for the robot one by one. Placing the points sufficiently close to each other makes the robot looks like it is following a continuous smooth path, this is the job of the Trajectory Planner.

The trajectory planner has pre-defined paths, it divides each path into same number of pieces regardless of their length and shape. Each of these pieces serves as the intermediate destinations for the robots. The points are sent to the robot periodically and since the number of points are always the same hence the robots will always reach their destinations at the same time.

5 Simulation Results

In this chapter the proposed high level Fuzzy Cooperative Controller and Low Level PID Controllers results are presented and discussed. The simulation setup is similar to the block diagram shown in figure 4-1. The differential drive robot model described in section 3.1 is used for simulations. The specification of the robot and DC Motor model are given below.

Robot model and DC motor model parameters.

S.No	Parameter	Value	Unit
Robot Model Parameters			
1	L_i	0.2	m
2	r	0.045	m
DC Motor Model Parameters			
1	R	3.07	$ohms$
2	L	0.04	mh
3	J	0.0294	$Kg.m^2$
4	B	0.0141	$N.m.s/rad$
5	K_e	0.65	$N.m/Amp$
6	K_t	0.65	$V.s/rad$

The simulation is carried out in MATLAB Simulink with a fixed step size of 1e-2 secs. To see the complete simulation diagrams refer to appendix.

The Low Level PID Controller is tested in a separate simulation with the motor model and quantizer to simulate the effects of a low resolution digital encoder the results are presented and discussed. The Fuzzy Path Following Controller is tested using a single robot model and trajectory generator, the controller is tested under different conditions for which the results are observed. Finally the Fuzzy Cooperative controller is tested using three identical robot models and trajectory generator, two different experiments are carried out to test the cooperative behavior of mobile robots the results are presented and discussed.

5.1 Speed PID Control of DC Motor

The DC motor speed control is essential part of the system. The fuzzy controller is working under the assumption that the output it is generating (v_i, ω_i) is tracked by the low level PID controller. The output of the Fuzzy Controller is the linear and angular velocities v_i, ω_i of the Robot, these velocities are converted to Left and Right Linear velocities v_r, v_l for the left and right wheel motors and finally these Left and Right velocities are converted to Angular Left and Right Velocities which is used as a reference command for the PID Controller. for details of the calculation see section 4.1.2

In this section we first present the open loop response of the PMDC motor, then the PID Speed controller for the DC motor result are presented and discussed.

Figure 5-1 shows the open loop response of the DC motor. There is no load torque applied to motor. The first graph shows Reference vs. Actual Speed, the motor passes 1 rad/sec reference and settle to around 1.4 rad/sec value and then maintains that speed because of no external disturbances. The second graph shows the current as it can be seen that there is an initial peak in current, this is due to the fact that motor has to break the inertia to start moving after that the current remains steady as there are no further disturbances applied.

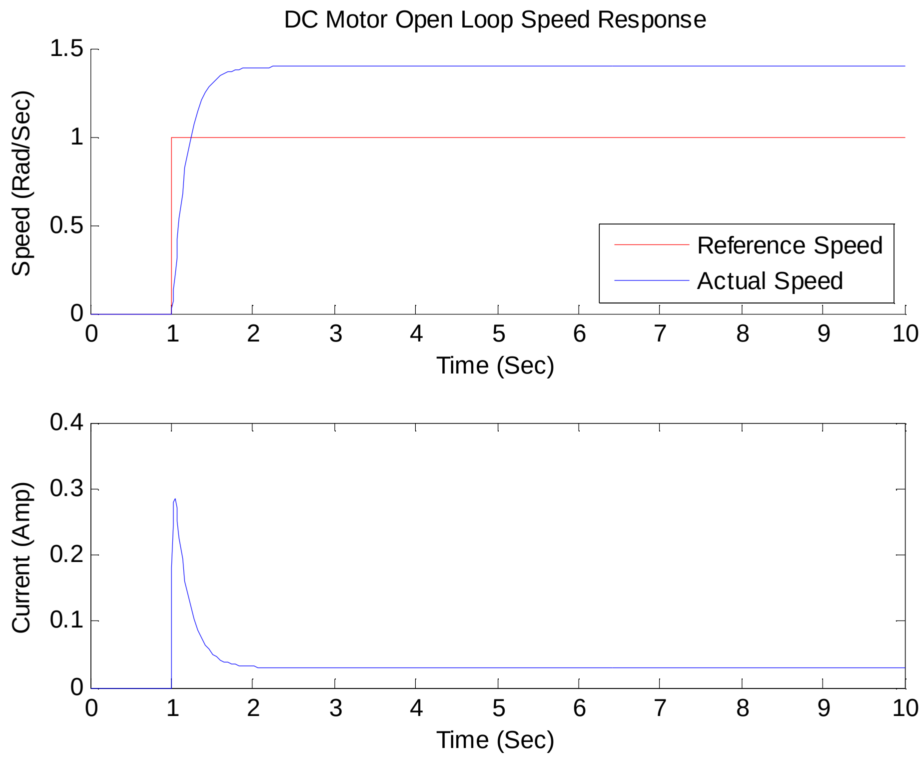

Figure 5-1 DC Motor Open Loop Response

Figure 5-2 shows the Speed PID controller response. The first graph compares the reference and actual speed. As can be seen that the PID controller quickly achieves the desired reference, the response time of the PID controller is less than 200 ms while the settling time is less than 500 ms. To check the controllers robustness against external disturbances we give step change in load torque at t = 5 sec. from figure it can be seen that the PID controller very quickly overcomes the external disturbance and converges to the reference value. In the current graph the initial peak as discussed above is due to the inertia, however the step at t = 5 sec is because of the load torque step. The controller increase the effort i.e. Current to maintain the reference speed. The current then settles at a higher value, the same speed is regulated with a higher current value because of the external load torque.

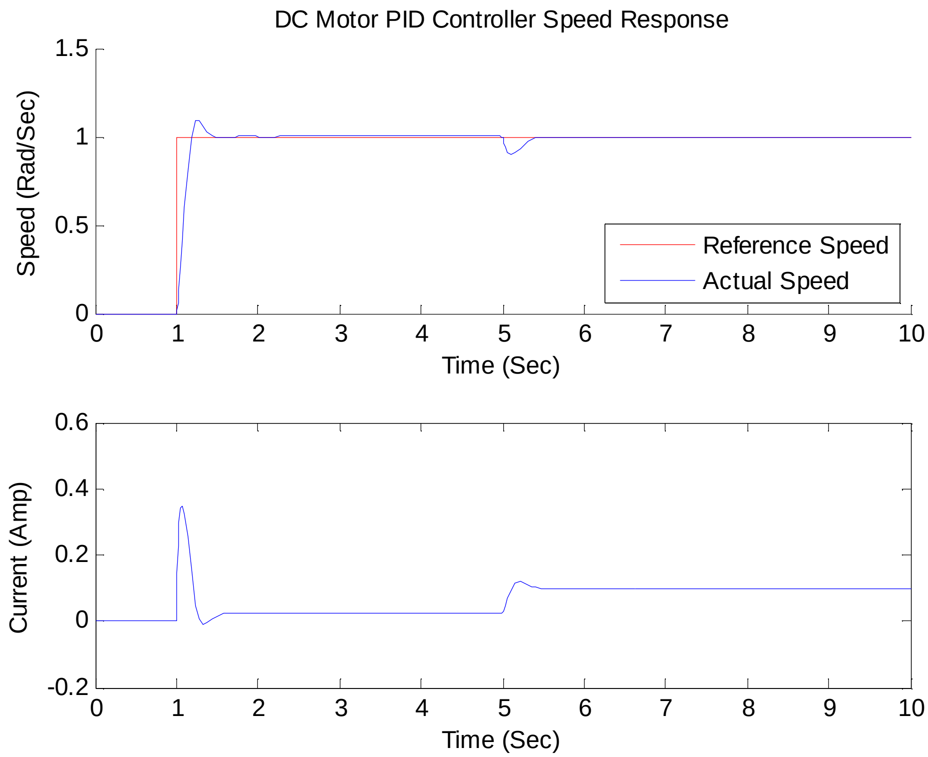

Figure 5-2 DC Motor Speed PID Controller Response

5.2 Fuzzy Path Following

One of the objective of the robot is to follow its own path, this section discuss the results of the mobile robot individual path following. For cooperative path following a necessary condition is that, the robot should be able to follow its own path therefore first we need to test the fuzzy controller for individual path following. We tested the Fuzzy controller with different scenarios in individual path following. First the robot is given a straight line path and then a curved path. In these tests we further changed the initial conditions of the robot being placed on the path or away from the path and collected the results. The results of interest are the robots trajectory, Linear and Angular velocities and the y coordination error.

5.2.1 Straight Line path

In this test the robot is required to follow x-axis (straight line) the robot is initially placed away from its desired path $P_i = [x_i, y_i, \varphi_i]^T = [1, -1.5, 0]$. First the robot needs to catch the desired path and then follow it. Figure 5-3 shows the trajectory of the robot, it can be seen that the robot very quickly converges to the desired path and it follows a continuous smooth trajectory, after reaching the desired path it follows the path with minimal error.

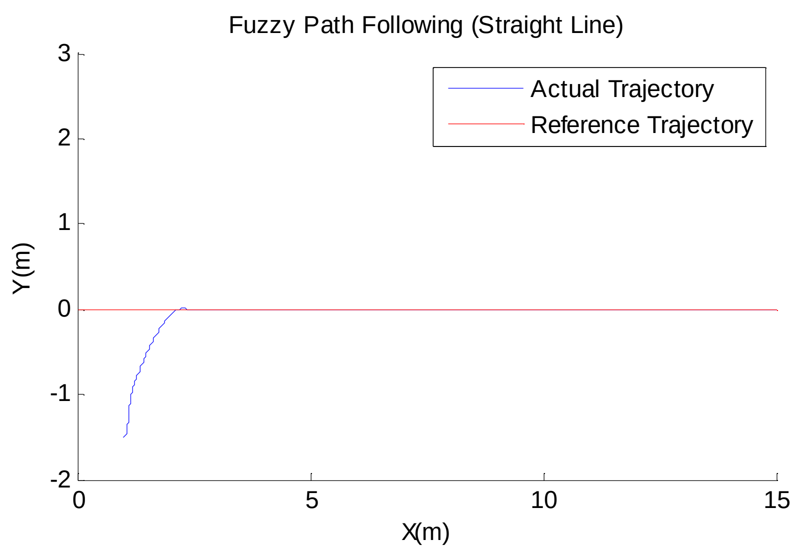

Figure 5-3 Fuzzy Path Following Straight Line Robot's Trajectory

Figure 5-4 shows the actual linear velocity of the robot. From Figure 5-3 it can be seen that initially the robot didn't move because the robot was placed ahead of its desired path. This will be discussed in detail in cooperative path following. The point at which the desired path crosses the robot's x position the robot now starts moving at a higher speed to overcome the distance it is lagging as soon as the robot catches the desired path it moves slowly tracking the path.

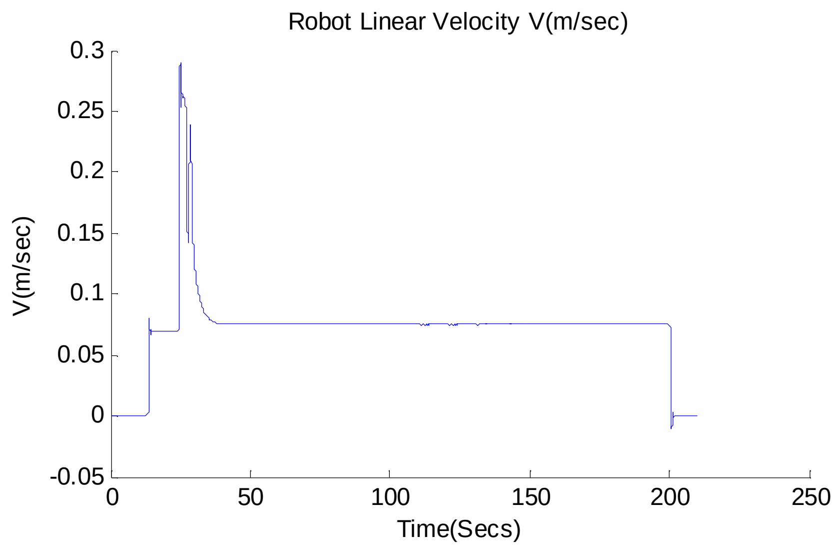

Figure 5-4 Fuzzy Path Following Straight Line Linear Velocity

Figure 5-5 shows the angular velocity of the robot following a straight line path. It can be seen that the robot takes a quick left turn then slowly turns right towards the path this can also be seen in Figure 5-3. This is due to the fact that robot's initial orientation is parallel to x- axis, therefore it must turn towards x-axis to reach it. After reaching the path angular velocity becomes zero because there is no error in orientation.

Figure 5-6 shows the error in y-coordination y_{err} initially it remains constant as the robot was stationary, once the robot starts moving it quickly overcomes the y-coordination error and after it reaches the desired path y_{err} becomes almost zero as it moves forward along the path.

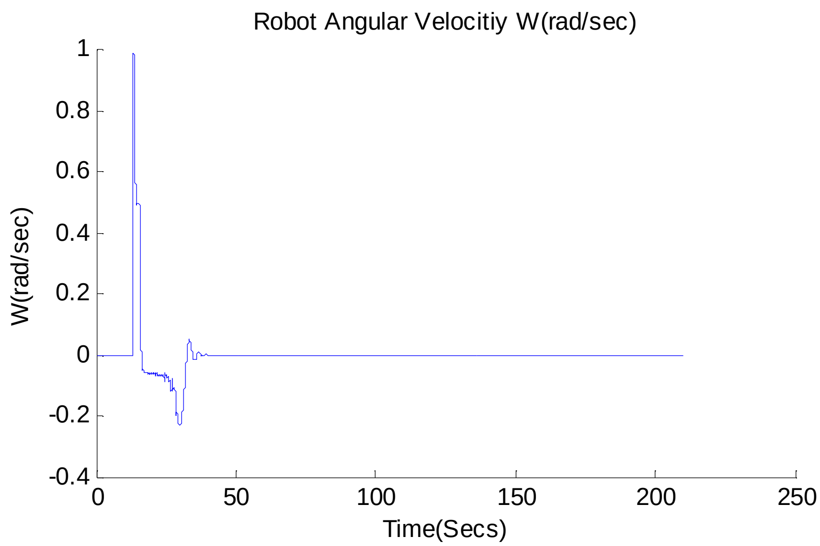

Figure 5-5 Fuzzy Path Following Straight Line Angular Velocity

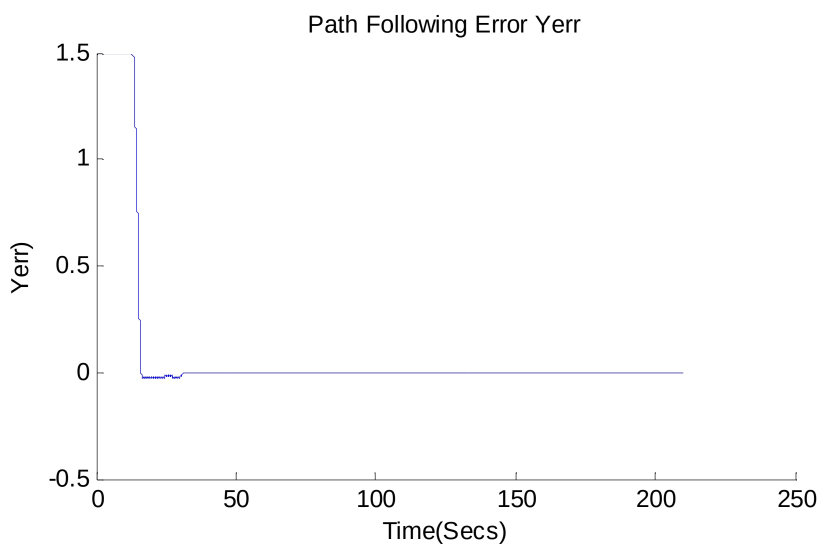

Figure 5-6 Fuzzy Path Following Straight Line y_{err}

5.2.2 Curved path

The second test puts the robot on a curved path. The curved path is generated using a fifth order polynomial to generate reasonably difficult curved path. This test is performed with two different initial conditions but same path, at first the robot is placed on the path secondly the robot is placed away from the path. Results are collected for the two cases and are discussed below.

5.2.2.1 Robot on path

In this experiment the robot is initially placed on the path. The position of the robot is given as $P_i = [x_i, y_i, \varphi_i]^T = [0.01, 1.5837, 0]^T$. Figure 5-7 shows the reference and actual trajectory of the robot , since the robot is already placed on the path therefore it doesn't have to travel long to catch the reference trajectory except for the small initial orientation error for which it compensate for very quickly, after that the robot tracks the trajectory with minimal error.

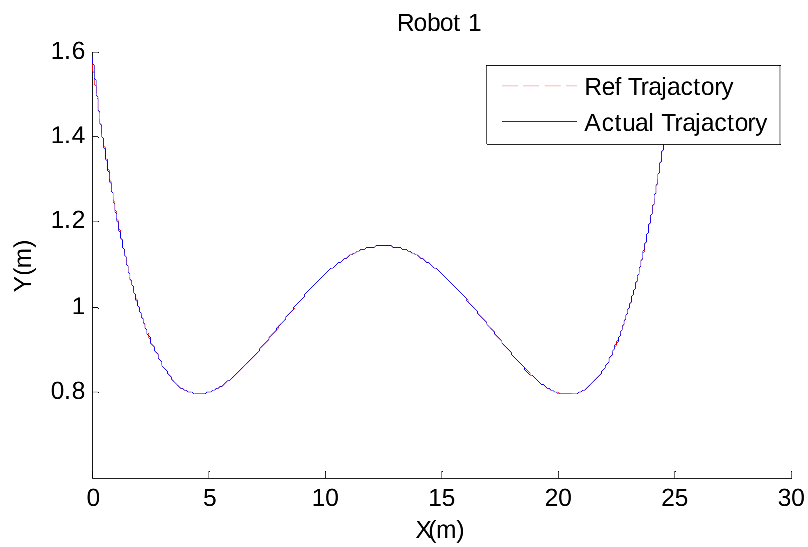

Figure 5-7 Fuzzy Path Following Curved Path Robot's Trajectory

Figure 5-8 shows the linear velocity v_i of the robot, as can be seen that throughout the journey the linear velocity of the robot remains almost constant around 0.126 m/sec. This is because the robot was on the track for the whole time and it does not have to catch the trajectory, hence no need to travel faster.

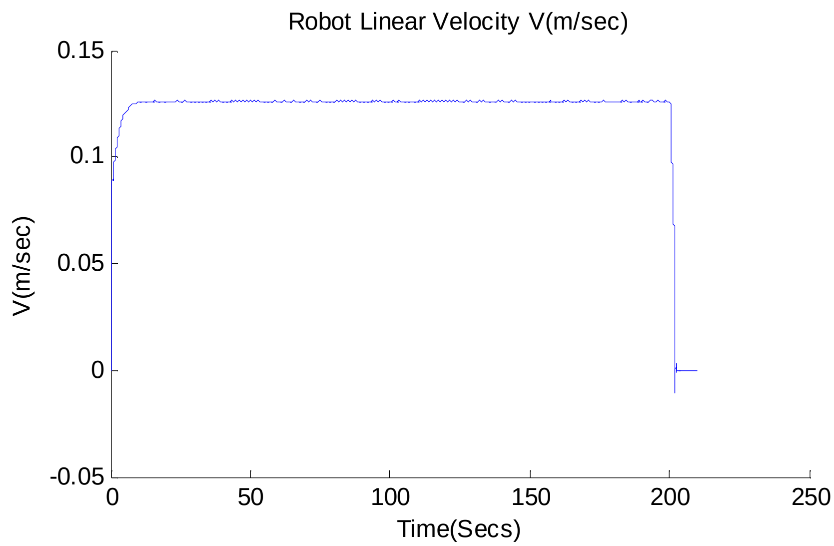

Figure 5-8 Fuzzy Path Following Curved Path Linear Velocity

Figure 5-9 shows the angular velocity ω_i of the robot. The initial positive and negative peaks shows that the robot makes a high speed left turn and a right turn immediately after it, this is to compensate for the initial orientation error of the robot and the trajectory. Once the robot achieves its desired orientation it follows the path with minimal orientation error. The positive and negative values represents the left and right turn.

Figure 5-10 show the y-coordination errory_{err}. This graph is somewhat similar to the angular velocity one because of the fact that angular velocity is being dominated by the y coordination error just as linear velocity is dominated by the input D_{rp} or the error in distance.The robot starts with an initial y_{err}, the fuzzy controller makes a counter effort to compensate for the error by increasing the angular velocity as can be seen in Figure 5-9, by the time it achieves the orientation the desired track has moved forward and its desired orientation is again changed so the controller again compensates for the error and this time it reaches the desired orientation and then follows the track with minimal error.

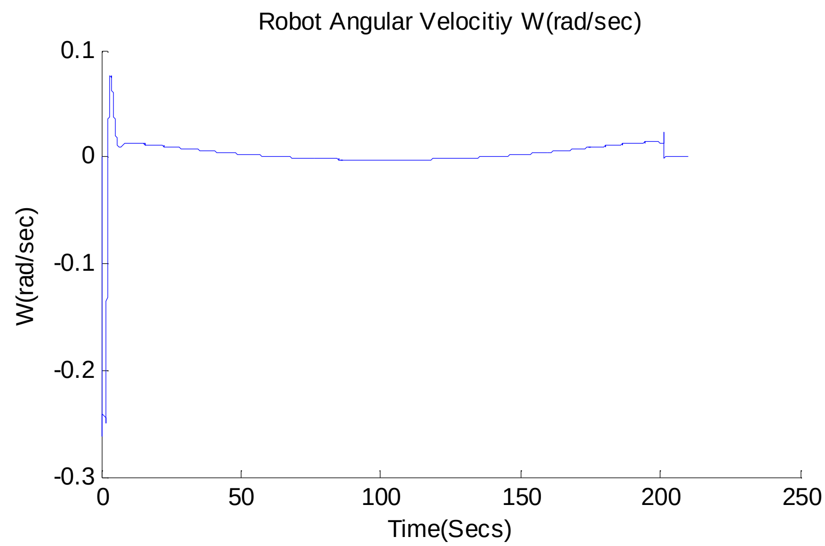

Figure 5-9 Fuzzy Path Following Curved Path Angular Velocity

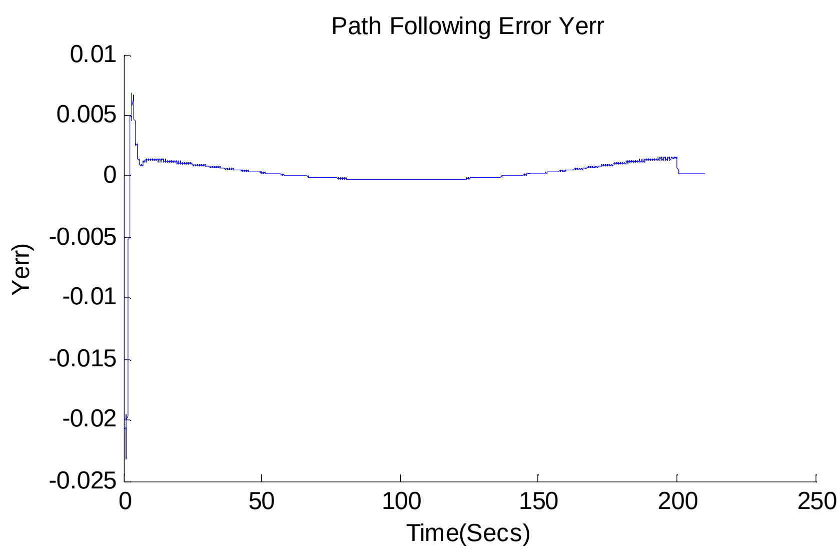

Figure 5-10 Fuzzy Path Following Curved Path Y_{err}

5.2.2.2 Robot not on path

In this experiment the robot is initially placed away from the desired path, however then path remains same. The initial position of the robot is given as $P_i = [x_i, y_i, \varphi_i]^T = [2.0, 0.9, 0]^T$. From Figure 5-11 it can be seen that the robot starts with a delay waiting for the desired trajectory to move ahead of its position. At start the robot turns towards the path and catches it, once it catches the path it tracks the path with minimal error as can be seen form figure.

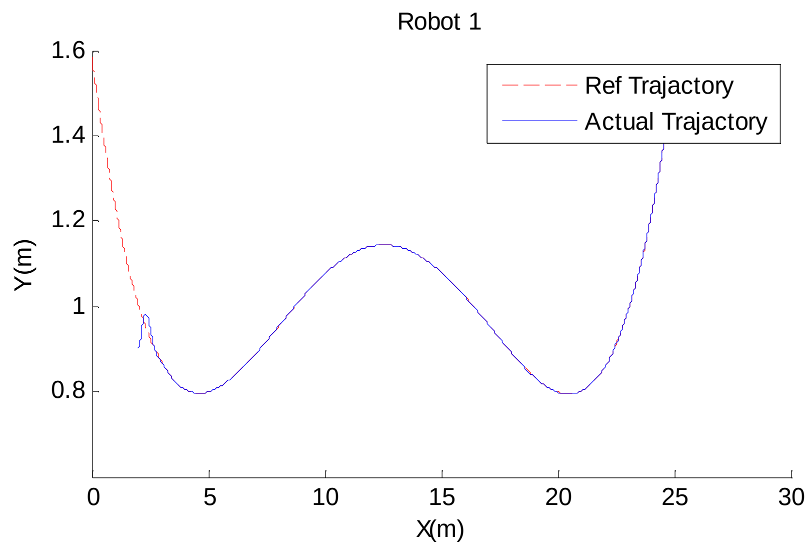

Figure 5-11 Fuzzy Path Following Curved Path Robot's Trajectory

Figure 5-12 shows the linear velocity v_i as can be seen initially the linear velocity is zero as the robot is waiting for the trajectory to move forward its starting position. After the trajectory moves ahead of the robot, it starts moving and since the robot is not placed on the path therefore it moves faster to catch the trajectory. As the robot comes close to the trajectory its linear velocity gradually slows down and becomes almost constant after reaching the trajectory.

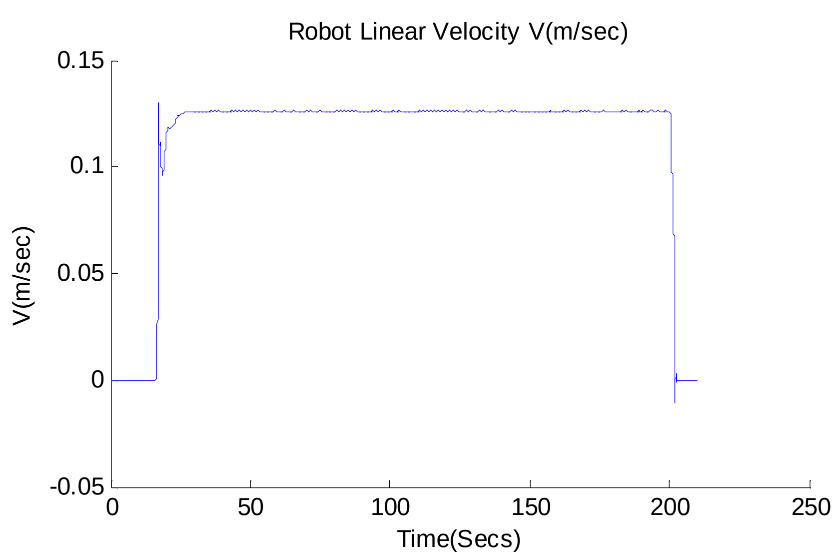

Figure 5-12 Fuzzy Path Following Curved Path Linear Velocity

Figure 5-13 shows the angular velocity $\omega_i$of the robot. Initially the angular velocity is zero because of the robot being stationary. Once the robot starts moving it makes a high speed left turn then immediately after that it makes a right turn and then again a right turn but this time it turns gradually as the robot is close to trajectory and there is not much y-coordination error left to produce big angular velocity output.

Figure 5-14 shows the Y-coordination errory_{err}, it can be seen that initially y_{err}only depends on the trajectory since the robot is stationary. As the robot starts moving y_{err} decreases rapidly. After that it grows in the other direction this is because of the difference in orientation. Once the robot catches the trajectory y_{err} becomes almost zero as can also seen from the robots trajectory in Figure 5-11.

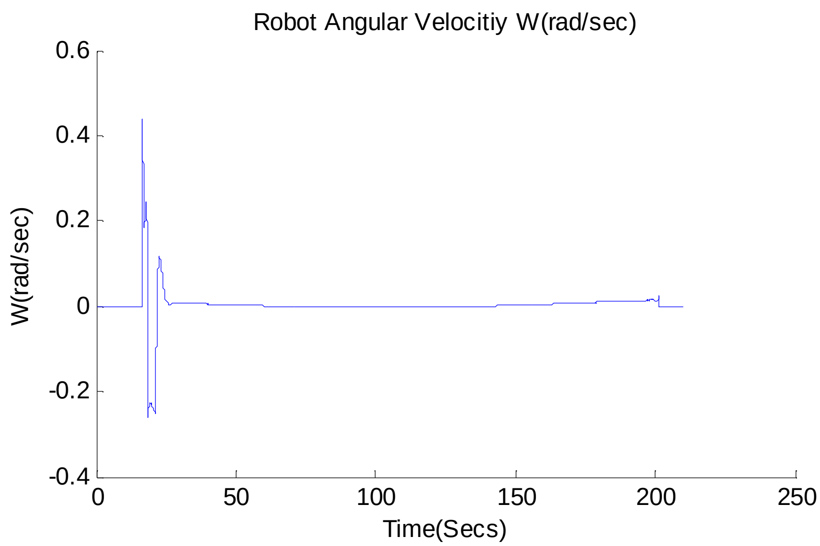

Figure 5-13 Fuzzy Path Following Curved Path Angular Velocity

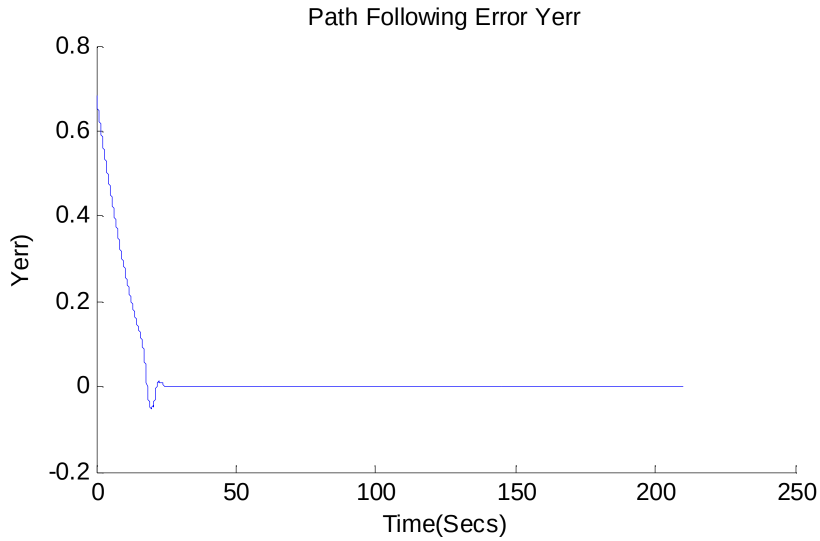

Figure 5-14 Fuzzy Path Following Curved Path Y_{err}

5.3 Cooperative Path Following

The main objective of the thesis is cooperative path following of a team of mobile robots. This section discusses the results of cooperative path following experiments. To test the cooperation problem we use a team of three identical mobile robot models, two different experiments were designed to test the cooperative algorithm. First the robots are tested with paths of different lengths but placed on the path initially, then the robots are given same path but are placed away from the paths, result are collected and discussed below.

5.3.1 Experiment # 1

In this experiment three mobile robots are used each with a different path (different lengths). Robot 1 has the smallest path and Robot 3 has the longest path. The robots are initially placed on the path the initial positions of the robots and path lengths are given below.

$$P_1 = \begin{bmatrix} x_1 & y_1 & \varphi_1 \end{bmatrix}^T = \begin{bmatrix} 0.01 & 11.4303 & -\frac{\pi}{4} \end{bmatrix}^T, L_{path(1)} = 18.7372m$$

$$P_2 = \begin{bmatrix} x_2 & y_2 & \varphi_2 \end{bmatrix}^T = \begin{bmatrix} 0.01 & 6.7207 & -\frac{\pi}{4} \end{bmatrix}^T, L_{path(2)} = 21.2581m$$

$$P_3 = \begin{bmatrix} x_3 & y_3 & \varphi_3 \end{bmatrix}^T = \begin{bmatrix} 0.01 & 2.0489 & -\frac{\pi}{4} \end{bmatrix}^T, L_{path(3)} = 25.3926m$$

The robots are required to follow their respective paths as well as maintain a formation along the journey also they must reach the final destination at the same time regardless of the path lengths.

Figure 5-15 shows the cooperation of the mobile robots, as can be seen that robot 1 has the shortest path whereas Robot 3 has the longest path. The round markers are used to highlight the position of each robot at a given time. The first markers on each path will indicate the initial position of the robots att_0, the second marker will indicate the position of each robot at t_1 and so on. These markers are distributed equally with respect to time. As can be seen that the robot are placed vertically above each other initially, the successive markers shows that the robots are moving along their path but are also maintaining an inter robot formation throughout the journey, and reach their final destination at the same time.

Figure 5-16 shows the individual trajectories of each robot, as can be seen from the figure that the robots track their respective paths with minimal error.

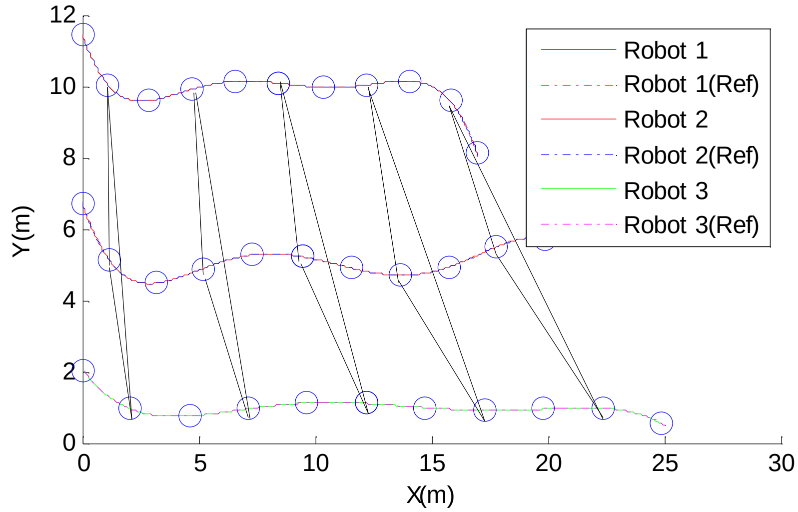

Figure 5-15 Cooperative Path Following Robot Formation (Different Paths)

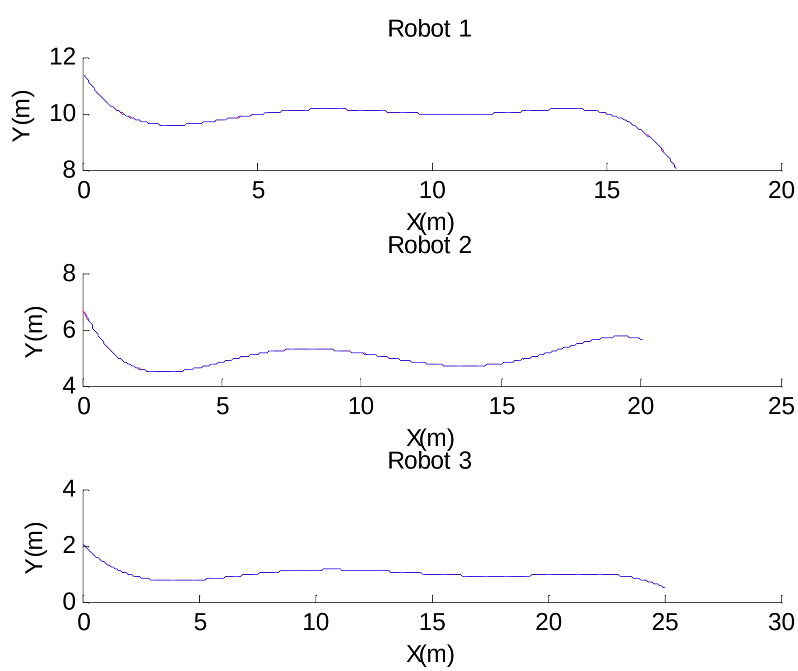

Figure 5-16 Cooperative Path Following Individual Trajectories (Different Paths)

Figure 5-17 shows the linear velocities of the robot. Since robot 3 has the longest path therefore it travel at a higher speed as can be seen from the figure V3 is continuously greater than V1 and V2. The trajectory of robot 3 is not very curvaceous with respect to the other two robots therefore the linear velocity of robot 3 dominates angular velocity and it moves faster. However for robot 1 and 2 the trajectory is more curved and less flat therefore these two robots tends to move slowly.

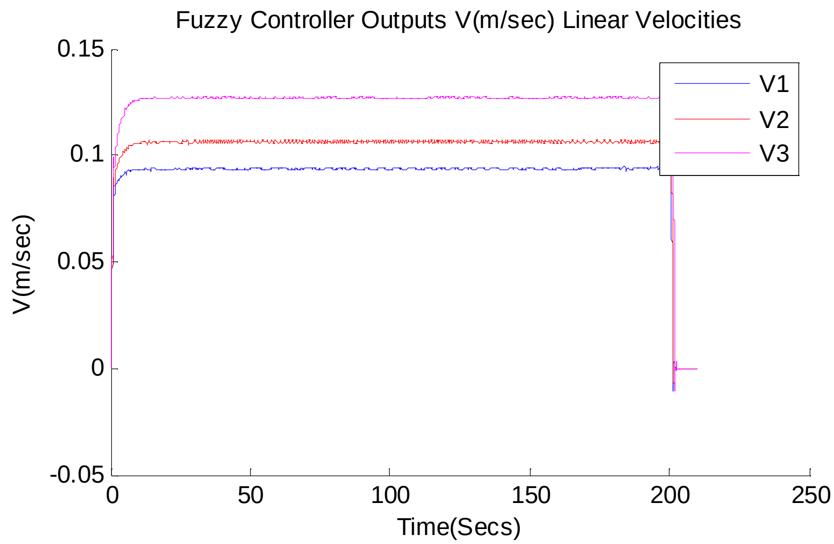

Figure 5-17 Cooperative Path Following Linear Velocities (Different Paths)

Figure 5-18 shows the angular velocities of the robots. The initial peaks represent the controller effort to compensate for the initial orientation error. It can be seen that robot 1 and 2 have higher angular velocity peaks this is because their paths are more curved, hence the fuzzy controller increases the angular velocity output to accurately track the curves. However for robot 3 it is the opposite case.

Figure 5-19 shows the y-coordination error$y_{err(i)}$. The initial peaks as discussed above are due to initial orientation error. It can be seen that once the robot track their respective path $y_{err(i)}$ becomes minimal. Robot 3 has the largest $y_{err(i)}$ because of the fact that it has the longest path it has to move fast, hence with lesser precision. But still tracks the path with sufficient precision as can be seen from Figure 5-16.

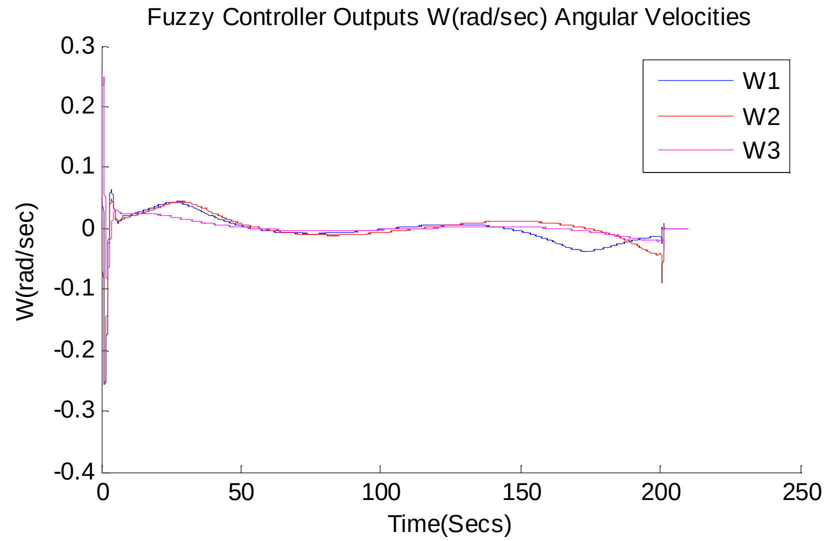

Figure 5-18 Cooperative Path Following Angular Velocities (Different Paths)

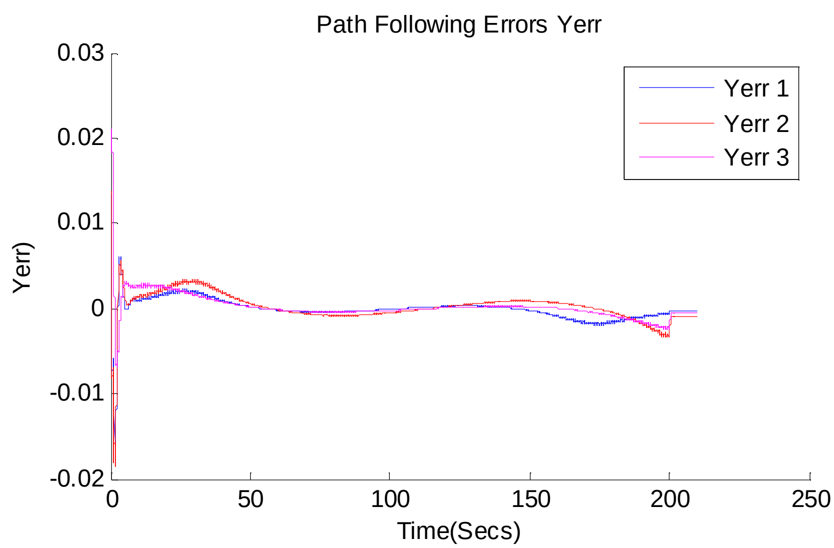

Figure 5-19 Cooperative Path Following y_{err} (Different Paths)

5.3.2 Experiment # 2

In this experiment three robots are used, each of which is provided with the same path, but they are placed away from their respective path initially. Robot 1 and robot 3 are placed ahead of their trajectory while Robot 1 is placed behind its trajectory. The robots initial positions are given below.

$$P_1 = \begin{bmatrix} x_1 & y_1 & \varphi_1 \end{bmatrix}^T = \begin{bmatrix} 1.0 & 11.0 & 0 \end{bmatrix}^T, L_{path(1)} = 19.5288m$$

$$P_2 = \begin{bmatrix} x_2 & y_2 & \varphi_2 \end{bmatrix}^T = \begin{bmatrix} -0.5 & 8.0 & 0 \end{bmatrix}^T, L_{path(2)} = 19.5288m$$

$$P_3 = \begin{bmatrix} x_3 & y_3 & \varphi_3 \end{bmatrix}^T = \begin{bmatrix} 1.5 & -1.0 & 0 \end{bmatrix}^T, L_{path(3)} = 19.5288m$$

The robots are required to follow their respective as well as maintain an inter robot formation. Since the robots are not placed on their respective paths, therefore first they need to catch the trajectory, and they must reach their respective end points at the same time.

Figure 5-20 shows the cooperation of robots in this experiment. it can be seen that Robot 1 and Robot 3 are placed ahead of their respective paths, while Robot 2 is placed behind the trajectory. Note that Robot 2 starts immediately as soon as the simulation starts while Robot 1 and 3 remains idle waiting for the trajectory to move forward. This is because of the assumption that if a robot is ahead of its path then it will wait for the other robots to catch up. However if the robot has fallen behind then it will move faster and catch the trajectory as discussed in **section 3.6.6**.

The markers on the paths shows that around the 3 meter mark, the robots achieve synchronization (formation), then maintains the formation as they move forward along their respective paths and reach their final destination at the same time.

Figure 5-21 shows the individual trajectories of the robots. It can be seen that all the robots follow their respective paths with minimal error.

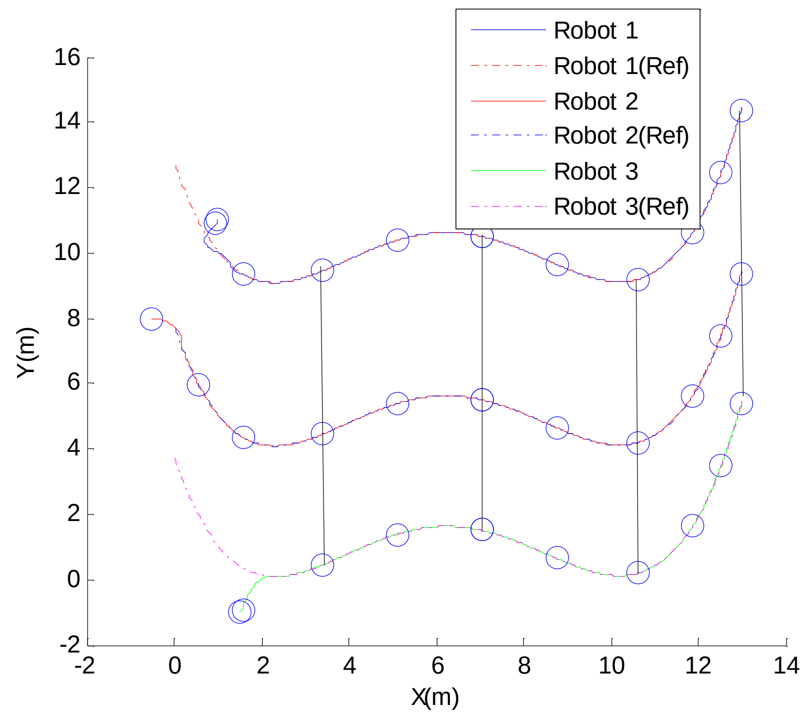

Figure 5-20 Cooperative Path Following Robot Formation (Same path)

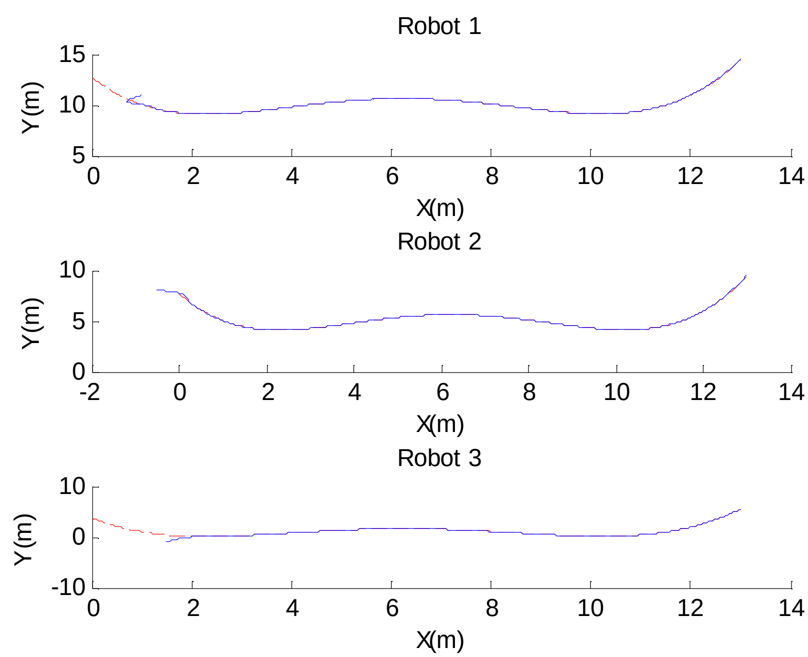

Figure 5-21 Cooperative Path Following Individual Trajectories (Same path)

Figure 5-22 shows the linear velocities of the robots. From figure it can be seen that initially only Robot 2 (RED) is moving (has non zero linear velocity). This is because, Robot 2 is placed behind its trajectory so it has to move fast to catch the trajectory. After about 60 secs Robot 2 starts moving as its trajectory passes its initial position, similarly Robot 3 starts moving at about 90 secs. All three robots have to cover some distance as soon as they start moving to catch the trajectory therefore an initial peak can be seen for all three linear velocities. After catching the trajectory all the robots linear velocities becomes same, this is due to the fact that all the robots have same path similarly all the robots stop (Linear velocity zero) at the same time.

Figure 5-23 shows the angular velocities of the robots. As discussed above initially only Robot 2 is moving (non zero angular velocity). The initial peaks in angular velocities represent the controller effort to correct the heading of the robot with respect to the trajectory. Once the robots achieve formation their angular velocities become similar because of same paths.

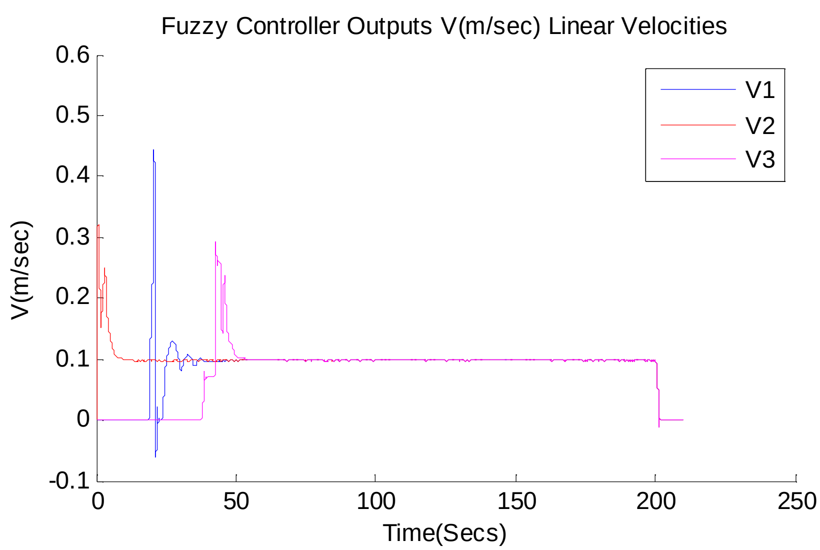

Figure 5-22 Cooperative Path Following Linear Velocities (Same path)

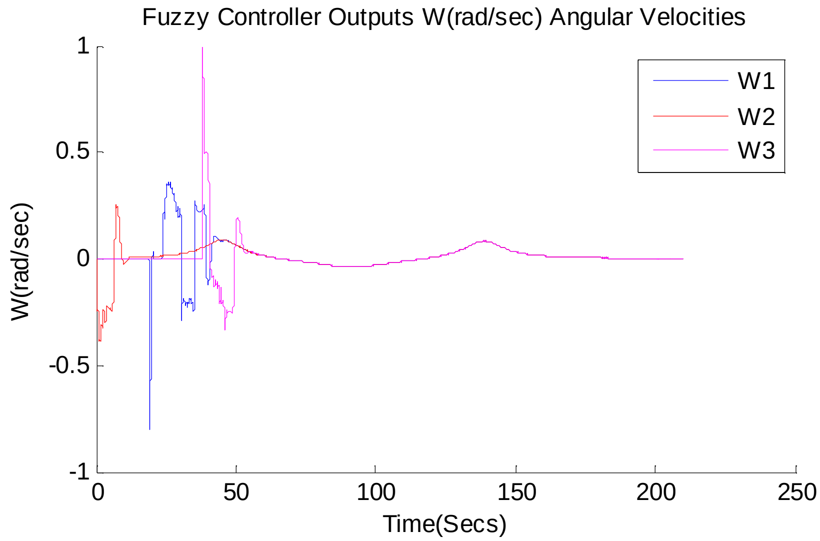

Figure 5-23 Cooperative Path Following Angular Velocities (Same path)

Figure 5-24 shows the y-coordination error of the robots. Initially y_{err} of Robot 1 and 3 is a function of the trajectory as the robots are standing still. However Robot 2 quickly overcomes its y-coordination error. It can be seen that as soon as the robot starts moving they overcomes the y_{err} very quickly. Once the robots achieve formation they follow their respective paths with minimal y-coordination error.

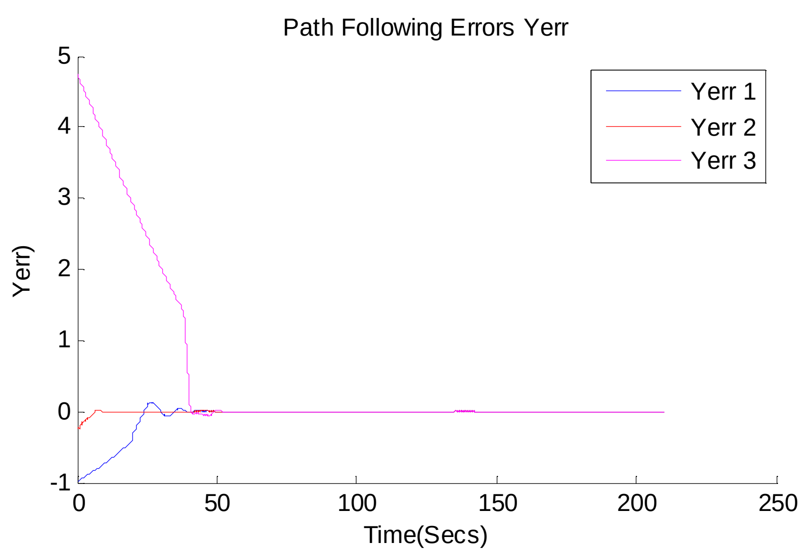

Figure 5-24 Cooperative Path Following y_{err} (Same path)

6 Conclusion

In the field of robotics and control systems, formation control of multi mobile robots is currently one of the most important topics and there is a lot of research is being carried out in this field. The primary motivation of using multiple mobile robots is to improve the overall efficiency of the system. It has been established that a group of robots working cooperatively can perform much better than an individual robot and can also perform much complex tasks.

Formation control of MMRs is a difficult task as each robot has to decide between individual and group goals. It requires a control scheme that can not only make the robot follow its own path but also maintain an inter robot formation.

6.1 Concluding Remarks

This work covers the formation control of mobile robots. The proposed design consist of a hierarchal control architecture a Low Level PID controller and High Level Fuzzy Cooperative Controller. The PID Controller tracks the reference speed of the robot wheel motors, it was tuned for the motor parameters to effectively track the reference speed with minimum tracking time and steady state error. The Result of the PID speed controller shows that it converges very quickly and tracks the speed reference with minimal steady state error.

The Fuzzy Controller is where the path following and cooperation is done. It was implemented in steps initially only straight line tracking was tested, after that a point tracking was established in which the robot was made to reach out a certain point in the plane. Then a path tracking was achieved in which the robot tracks a series of points that make a complete path and finally the cooperative behavior was test in which a team of three identical robots were used. The robots were tested in different experiments in which they are subjected to different initial conditions and they are supposed to follow their respective paths and maintain a formation amongst them. The results shows that the Fuzzy Cooperative Controller effectively achieves its goals of individual path following and group cooperation, it has been show that the robots tracks their respective paths with minimal error and maintain the formation while travelling.

6.2 Recommendations for Future Work

This work is a small effort towards the Formation control of mobile robots. There are still several areas left which were beyond the scope of this work, but are of significant importance from present research's perspective. Some of the areas that are recommended for future work are as follows.

- Obstacle detection is an important aspect of mobile robotics. A control algorithm should be developed that can detect any obstacle in the path and avoid them.
- The robot should be able to move in known and unknown environments. Where there is unknown obstacles.
- The proposed work should be implemented practically on a hardware robot and the result should be verified. Following are the considerations for practical implementation.
    - Localization sensor for robot e.g. Encoder, Camera, GPS etc.
    - Controller/DSP to implement the control algorithms
    - Wireless communication medium e.g. RF, GSM, Bluetooth, Wi-Fi etc.
    - DC Motors
    - Power Supply

7 References

[1] R. W. Beard, J. Lawton, and F. Y. Hadaegh, “A coordination architecture for spacecraft formation control,” IEEE Trans. Control Syst. Technol., vol. 9, no. 6, pp. 777–790, Nov. 2001.

[2] B. Seanor,Y.Gu, M. R. Napolitano, G. Campa, S. Gururajan, and L. Rowe, “3-aircraft formation flight experiments,” presented at the 14th Mediterr. Conf. Control Autom., Ancona, Italy, Jul. 2006.

[3] R.W. Beard, T.W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, “Decentralized cooperative aerial surveillance using fixed- wing miniature UAVs,” Proc. IEEE, vol. 94, no. 7, pp. 1306–1324, Jul. 2006.

[4] N.Wu andM. Zhou, “Modeling and deadlock control of automated guided vehicle systems,” IEEE/ASME Trans.Mechatronics, vol. 9, no. 1, pp. 50– 57, Mar. 2004.

[5] A. Howard, L. E. Parker, and G. S. Sukhatme, “Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection,” Int. J. Robot. Res., vol. 25, no. 5, pp. 431–447, May 2006.

[6] Healey, Anthony J. "Application of formation control for multi-vehicle robotic minesweeping." Decision and Control, 2001. Proceedings of the 40th IEEE Conference on. Vol. 2. IEEE, 2001.

[7] D. M. Bevly and B. Parkinson, “Cascade Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles,” IEEE Trans. Control Syst. Technol., vol. 15, no. 2, pp. 199–208, Mar. 2007.

[8] S. Han, H. S. Lim, and J. M. Lee, "An efficient localizations scheme for a differential-driving mobile robot based on RFID system," IEEE Trans. Ind. Electron., vol. 54, no. 6, pp. 3362–3369, Dec. 2007.

[9] K. B. Purvis, K. J. Astrom, and M. Mhammash, "Estimation and optimal configurations for localization using cooperative UAVs," IEEE Trans. Control Syst. Technol., vol. 16, no. 5, pp. 947–958, Sep. 2008.

[10] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," IEEE Trans. Robot. Autom, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[11] H. Li and S. X. Yang, "A behavior-based mobile robot with a visual landmark-recognition system," IEEE/ASME Trans. Mechatronics, vol. 8, no. 3, pp. 390–400, Sep. 2003.

[12] D. Gu and H. Hu, "Integration of coordination architecture and behavior fuzzy learning in quadruped walking robots," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 37, no. 4, pp. 670–681, Jul. 2007.

[13] J. Huang, S. M. Farritor, A. Qadi, and S. Goddard, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," IEEE/ASME Trans. Mechatronics, vol. 11, no. 2, pp. 205–215, Apr. 2006.

[14] M. A. Lewis and K. H. Tan, "High precision formation control of mobile robots using virtual structures," Auton. Robots, vol. 4, no. 4, pp. 387–403, 1997.

[15] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," Robot. Auton. Syst., vol. 58, no. 5, pp. 727–736, 2010.

[16] Y. C. Chang and B. S. Chen, “Robust tracking designs for both holonomic and nonholonomic constrained mechanical systems: Adaptive fuzzy approach,” IEEE Trans. Fuzzy Syst., vol. 8, no. 1, pp. 46–66, Feb. 2000.

[17] T. H. S. Li, S. J. Chang, and W. Tong, “Fuzzy target tracking control of autonomous mobile robots by using infrared sensors,” IEEE Trans. Fuzzy Syst., vol. 12, no. 4, pp. 491–501, Aug. 2004.

[18] E. Maalouf, M. Saad, and H. Saliah, “A higher level path tracking controller for a four-wheel differentially steeres mobile robot,” Robot. Auton. Syst., vol. 54, pp. 23–33, 2006.

[19] G. Antonelli, S. Stefano, and G. Fusco, “A fuzzy-logic-based approach for mobile robot path tracking,” IEEE Trans. Fuzzy Syst., vol. 15, no. 2, pp. 211–221, Apr. 2007.

[20] J. B. Mbede, X. Huang, and M. Wang, “Robust neuro-fuzzy sensor-based motion control among dynamic obstacles for robot manipulators,” IEEE Trans. Fuzzy Syst., vol. 11, no. 2, pp. 249–261, Apr. 2003.

[21] A. Zhu and S. X. Yang, “Neurofuzzy-based approach to mobile robot navigation in unknown environments,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 37, no. 4, pp. 610–621, Jul. 2007.