ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

BY
Hasan MEHRJERDI

CONTROL AND COORDINATION FOR A GROUP OF MOBILE ROBOTS IN UNKNOWN ENVIRONMENTS

MONTRÉAL, DECEMBER 16 2010

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Maarouf Saad, Thesis Supervisor
Département de génie électrique à l'École de technologie supérieure

Mr. Guy Gauthier, President of the Board of Examiners
Département de génie de la production automatisée à l'École de technologie supérieure

Mr. Chakib Tadj, Board Member
Département de génie électrique à l'École de technologie supérieure

Mr. Jawhar Ghommam, External Member
Research Unit on Intelligent Control at National School of Engineers of Sfax, Tunisia

Mr. Wen-Hong Zhu, Independent External Member
Canadian Space Agency

THIS THESIS WAS PRESENTED AND DEFENDED

BEFORE A BOARD OF EXAMINERS AND PUBLIC

NOVEMBER 25 2010

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENTS

The last few years of my life have been the most challenging and rewarding because I truly had the opportunity to meet and understand people from many different countries who enriched my life by sharing a wide range of customs and cultural backgrounds. I am also deeply grateful to all the teachers that opened my eyes and guided my steps in the long and winding road to the fantastic world of robotics research.

Firstly, I would like to thank my advisor, Professor Maarouf Saad who was not only scientifically my mentor, but also socially my teacher. His strengths lie in his intelligence and consistency, straight-forwardness and compassion. These traits make him both a wonderful person and a great advisor. Undoubtedly, he was the most influential person during my PhD studies and I hope the bond that we established will continue for many years to come.

My committee members have been truly wonderful in their comments and creative ideas. I would like to thank Professor Guy Gauthier, Professor Wen-Hong Zhu and Professor Chakib Tadj for taking time to help me progress as a researcher and for having kindly agreed to participate in my thesis research and final PhD committees.

I would like to thank my primary collaborator, Professor Jawhar Ghommam, for his helpful comments, exciting discussions and tremendous insight on the many aspects of theoretical control. Although we were geographically far apart, our working relationship was always exciting and I am extremely grateful for his many interesting ideas.

I would like to give my special thanks to my Australian collaborator, Neil Burton. I have learned a lot from him and thank him for his motivation, enthusiasm and strong personality. Our work together was often a mix of fun and dedication, and his insights helped me continue the slow and sometimes arduous process of editing and refining the individual papers and thesis into a coherent presentation.

I thank Professor Paul Scerri at the Carnegie Mellon University for giving me the opportunity to visit their research facility and creating a pleasant and challenging research atmosphere that I truly appreciated.

I wish to thank my father, mother, brother and sisters, and nephew and nieces, Erfan, Viana and Armita, who although far away, gave me courage and strength through their understanding and moral support. Sadly, my father passed away, and no amount of words can truly thank him and all my family for making this dream possible.

I also express my gratitude to the Picard's family, Jean-Marc, Marie-Josée, Stéphane and Antoine, who played the role of my family and friends in Canada.

I would like to thank the former and current members who work with me in the Robotics Lab (ETS). In all the years we have shared an office, I have learned so much from them, and I have enjoyed working with them including Amel, Habib and his wife Lina, Raouf, Katia, Samuele, Mario, Samuel and Thierry.

Special thanks go to Sylvain and Jorge, who kept the robots running so my experiments could complete in time.

CONTRÔLE ET COORDINATION POUR UN GROUPE DE ROBOTS MOBILES DANS UN ENVIRONNEMENT INCONNU

Hasan MEHRJERDI

RÉSUMÉ

Il existe de nombreuses applications pour lesquelles la tâche requise sera atteinte beaucoup plus efficacement avec l'utilisation d'un groupe de robots mobiles comparativement à un robot unique. Les groupes de robots mobiles peuvent obtenir les résultats souhaités avec rapidité et précision, puisqu'il est possible de modifier chaque robot pour certaines tâches spécifiques, ils deviennent idéaux pour les applications telles que le sauvetage, l'exploration et le divertissement. Lorsqu'on compare l'issue de la mission d'un groupe de robots mobiles à celle d'un robot unique, il est facile de voir que les performances du groupe de robots mobiles améliorent la répartition des tâches spécifiques, la sécurité, la durée du temps nécessaire et l'efficacité du système pour atteindre le résultat souhaité. Cette thèse étudie le suivi de la trajectoire et le comportement coopératif pour un groupe de robots mobiles, fondés sur l'utilisation d'algorithmes non linéaires et intelligents.

Afin de créer l'algorithme de contrôle le plus efficace pour le suivi des trajectoires, nous présentons trois différentes techniques, soit la technique de Lyapunov, le contrôle intelligent (commande floue) et une version exponentielle du mode de glissement. Les algorithmes développés assurent la poursuite des trajectoires désignées tout en minimisant les erreurs. Les résultats expérimentaux en utilisant un seul robot mobile sont présentés pour démontrer le potentiel et la capacité des algorithmes développés.

Afin de coordonner un groupe de robots mobiles pour parvenir à un résultat commun, il est nécessaire de créer une architecture efficace et un algorithme de contrôle, ce qui leur permettra de travailler soit individuellement soit dans des formations robotiques organisées. Ceci est réalisé en employant des techniques de suivi de trajectoires et de commandes de coordination, en ayant une connaissance de la localisation des robots dans leur environnement. Dans cette thèse, trois contrôleurs hiérarchiques différents sont présentés, basés sur des techniques non linéaires et en se servant de l'asservissement par PID des roues des robots afin de concevoir un algorithme qui démontrera à la fois la coopération et la coordination d'une équipe de robots mobiles. Ces contrôleurs sont basés sur la technique de Lyapunov, le contrôle intelligent (commande floue) et une version exponentielle du mode de glissement.

Pour l'amélioration du suivi des trajectoires, chaque robot est muni de capteurs. Quand un obstacle est détecté par l'un des capteurs, l'algorithme permet une manœuvre incitant le robot à se déplacer autour de l'obstacle en changeant sa vitesse et sa direction. En plus de l'évitement d'obstacles, les contrôleurs permettent à tous les robots d'atteindre simultanément leur destination respective en ajustant la vitesse de chaque robot individuellement pendant que ceux-ci se déplacent le long de leur trajectoire prédéfinie.

Cela signifie que tout le groupe arrivera à destination en même temps, quelle que soit la longueur de chaque parcours individuel ou le nombre d'obstacles rencontrés.

Les résultats expérimentaux obtenus en utilisant trois robots mobiles montrent la performance de ces algorithmes de contrôle et le comportement de coopération et de coordination par un groupe de robots.

Mots-clés: Commande nonlinaire, Contrôle intelligent, Coopération, Coordination, Groupe de robots mobiles, Mode de glissement exponentiel, Suivi des trajectoires.

CONTROL AND COORDINATION FOR A GROUP OF MOBILE ROBOTS IN UNKNOWN ENVIRONMENTS

Hasan MEHRJERDI

ABSTRACT

This thesis studies the trajectory tracking and cooperative behavior for a team of mobile robots using nonlinear and intelligent algorithms to more efficiently achieve the mission outcome. There are many practical applications where specific tasks are more resourcefully achieved by using a group of mobile robots rather than a single robot. Mobile robots can subdivide and multi-task the mission with speed and accuracy and the ability to be individually modified for precise tasks makes them ideally suited for applications such as search and rescue, exploration or entertainment. When comparing the mission outcome of a group of multi mobile robots (MMR) to that of a single robot, we see that the performance of the MMR group improves the specific task allocation, safety, the time duration required and the system effectiveness to achieve the outcome.

In order to create the most effective control algorithm for trajectory tracking, we present three different techniques including Lyapunov technique, intelligent control (fuzzy control) and the exponential version of sliding mode. The developed algorithms instruct a robot to keep moving on their desired trajectory while simultaneously reducing tracking errors. The experimental results when using a single mobile robot are presented to demonstrate the potential and capability of the developed algorithms.

In order to coordinate a group of mobile robots to achieve a common outcome, the goal is to create efficient system architecture and a control algorithm that enables them to work both individually and in meaningful robot formations. This is achieved by employing coordination and trajectory tracking techniques with the knowledge derived by the localization of the robots from their environment. Three different hierarchical controllers are presented based on nonlinear and intelligent techniques in order to construct an algorithm that exhibits both group cooperation and coordination for a team of mobile robots. These controllers consist of Lyapunov technique, intelligent control (fuzzy control) and the exponential version of sliding mode.

For improved trajectory tracking, each robot is fitted with onboard sensors. When an obstacle is detected by any of the robots' sensors, they direct that robot to move around the obstacle by changing its velocity and direction. As well as obstacle avoidance, the controllers work to make the MMR group arrive concurrently at their target points by adjusting each of the individual robots' velocities as they move along their desired trajectories. This means the group will arrive at their destination within the same time duration, regardless of the length of each individual trajectory or number of obstacles that confront each robot.

The experimental results obtained using three mobile robots display the performance of these control algorithms in producing a cooperative and coordinated behavior for the robot group.

Key words: Coordination, Cooperative, Exponential sliding mode, Intelligent Control, Multi mobile robots, Nonlinear Control, Trajectory tracking

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS AND ACRONYMS

ANFIS	Adaptive Neuro-Fuzzy Inference System
API	Application Programming Interface
ASSM	Arbitrary Second-Order Sliding Mode
CSM	Conventional Sliding Mode
Digraph	Directed Graph
ESM	Exponential Sliding Mode
FIS	Fuzzy Inference System
LSE	Least Square Error
MANET	Mobile Ad-hoc Network
P2P	Peer-to-Peer
PDTD	Phase-Diffusion Time-Division
PID	Proportional Integral Derivative Controller
QSSM	Quasi-continuous Second-Order Sliding Mode
SMC	Sliding Mode Control

LIST OF SYMBOLS AND UNITS OF MEASURE

d	Distance between the geometric center of the robot and the center of mass, m
d_{RP}	Distance from the actual position of robot to the next desired position, m
I_c	Moment of inertia of the robot without the driving wheels, kg/m^2
I_m	Moment of inertia of each wheel and the motor rotor about a wheel diameter, kg/m^2
L	Distance between driving wheel and the axes of symmetry, m
m_c	Mass of the robot without the driving wheels, kg
m_ω	Mass of each driving wheel plus the motor rotor, kg
x, y, x_d, y_d	Position and desired position of the robot, m
r	Radius of each driving wheel, m
v, v_d	Linear and desired linear velocity of the mobile robot, m/s
ω, ω_d	Angular and desired angular velocity of the mobile robot, rad/s
α	Difference between the line joining the current position to the next desired position, rad
ψ, ψ_d	Orientation and desired orientation of the robot, rad
ω_r, ω_l	Angular velocities of the right and left wheels, rad/s
τ_r, τ_l	Right and left motor's torques, $N.m$

INTRODUCTION

The ability for a group of mobile robots to work cooperatively has been a challenging idea in artificial intelligence and robotics since the beginning of science fiction in the 1950's. Instead of building a powerful, single and elaborate robot, a MMR group can demonstrate flexibility and efficiency in performing the task required as well as making the system tolerant to possible individual robot faults. However, along with the benefits of skillfully completing a task, there is also the need to develop a behavioral approach within the MMR group similar to that found in any group of animals in nature. A synchronicity in the behavioral attitude of the group allows for a better outcome for the overall task to be accomplished.

As the world becomes more technologically complex, we see a diversity of applications that are either too complicated or dangerous for humans to perform and are therefore ideally suited for robots exhibiting intelligent group behavior. The ability to discern the surrounding environment and make informed decisions would bring robot behavior one step closer to mimicking human behavior allowing for the safe interrogation of unknown or dangerous environments. Some possible applications for a mobile robot group would be surveillance, exploration, underwater autonomous vehicles, a fleet of marines or an unmanned aerial vehicle.

Apart from the interest expressed by the robotics community in both the behavior and applications for MMR groups, there has also been excitement in the designing of control systems and algorithms needed to achieve these goals. This is especially relevant in the command and coordination of a team of mobile robots needing to work in known or cluttered environments. Theoretical control views of MMR behavior are divided between centralized and decentralized systems and different architectures and strategies employing both systems have been developed to control and coordinate a MMR group. These methods encompass behavior-based, virtual structure, leader follower, graph-based and potential field approaches.

The problem of exploring an unknown environment by a robot group can be facilitated by assigning each robot inbuilt location sensors which accumulate data and incrementally construct a map of the surrounding environment. In order to avoid obstacles at any moment when traveling on their trajectory, this information needs to be acted upon by each robot in the group. Individually they need to evaluate if they should stop to avoid a collision, and if so work out what the next target is to explore, and then program the viability of moving there safely. Artificial intelligence and nonlinear algorithms exploring collision avoidance for mobile robots has been well documented, but largely this work has been performed using only a single robot.

In this work, we focus on the problem of how to control and coordinate a group of mobile robots moving in group formation in an unknown environment that includes dynamic or static obstacles.
This thesis is organized as follows:

Chapter 1 presents an introduction to multi mobile robots and their applications. State of the art of the research in the field of coordination and control for a team of mobile robots is explored by evaluating current strategies and control algorithms.

Chapter 2 proposes the use of exponential sliding mode control to maximize the efficiency of trajectory tracking and reduce chattering on the control inputs of the trajectory tracking algorithm of a nonholonomic wheeled mobile robot. Additionally, as an efficient solution, Lyapunov technique and intelligent control techniques are introduced.

Chapter 3 explores the coordination and trajectory tracking for a team of mobile robots moving in a group formation using exponential sliding mode. The model chosen uses a combination of sliding technique and graph theory and the developed algorithm instructs the robots to keep moving on their desired trajectories while simultaneously reducing tracking errors to maintain the desired robot formation.

Chapter 4 considers the problem of creating a coordination algorithm for a team of mobile robots using a combination of the Lyapunov technique and graph theory embedded in the virtual structure. In this way, the knowledge derived by the localization of the robots assists the algorithm in creating efficient coordination and trajectory tracking commands for the group, which can then create useful robot formations. The experimental results obtained using three mobile robots display the performance of this control algorithm in creating efficient trajectory tracking and coordination.

Chapter 5 introduces the design and implementation of an intelligent cooperative algorithm for a team of multi mobile robots. The developed algorithm bases its characteristics on the efficiency of fuzzy logic, facilitating robots to cooperate as a group while simultaneously following their individual trajectories. This creates a scenario where all the robots in the MMR group will arrive within the same time duration, regardless of the length or shape of each individual trajectory.

Chapter 6 introduces an intelligent coordination algorithm to control a team of multi mobile robots being confronted with obstacles in an unknown environment. In this scenario, the developed algorithm processes the sensor's data being monitored from the environment by each robot, and this information is used to locate and avoid any obstacles in a robot's trajectory including other robots in the group. When there is a possibility of an imminent crash between any of the robots, or a robot and a static obstacle, the algorithm instructs these robots to avoid a collision while maintaining the group behavior and formation.

Finally, some concluding remarks and future plans will be presented.

CHAPTER 1

LITERATURE REVIEW

This chapter reviews the applications, recent strategies and developments of designed algorithms used for trajectory tracking, cooperative behavior and coordination for a team of mobile robots. The ability to control a group of mobile robots while permitting them to work cooperatively has long been a thought-provoking idea in robotics and artificial intelligence with the obvious advantages a robot group can bring to efficiently and speedily accomplish a mission. There are distinct advantages in using a group of homogeneous or heterogeneous mobile robots to accomplish a task rather than a more elaborately designed single robot. Experimental results indicate that when operating in the same environment, the overall mission performance of an MMR group compared to a single robot shows an improvement in task allocation, safety and performance, the time duration required and system effectiveness to achieve the mission outcome (Cao, Fukunaga et Kahng, 1997; Dudek, Jenkin et Wilkes, 1996; Gerkey et Mataric, 2004; Guzzoni *et al.*, 1997; Hagras et Colley, 2005; iguria et Howard, 2009; Nouyan *et al.*, 2009; Schneider-Fontan et Mataric, 1998; Viguria et Howard, 2009).

Instead of building a powerful single robot, a MMR group can provide flexibility in performing the task required, as well as making the system more tolerant to possible individual robot faults. The first critical step in designing the structure of a coordinated robot group is the ability to direct each robot to follow its individual desired trajectory. This concept originates from the trajectory tracking techniques used for single robots, which can be explored to create improved algorithms for the efficient and harmonious group coordination of multiple robots. Also, the technical advances in wireless communication, sensors and embedded computing have all played a part in developing the new generation of coordination algorithms for robot control including group formation strategies.

1.1 Group Coordination Definitions

Multi Mobile Robots (MMR) is a group of homogeneous or heterogeneous mobile robots that work in the same environment to accomplish a task, rather than a more sophisticated single robot. A cooperative multi robot is composed of individual robots that operate together to perform some global task. Coordination is a cooperative behavior within the MMR group in which the actions performed by each robotic agent take into account the actions executed by the other robotic agents (Farinelli, Iocchi et Nardi, 2004) where the robots have to communicate, exchange information or interact in some way to achieve an overall mission.

1.2 Multi mobile Robots Applications

MMR's can have a variety of applications depending on the working environments and the tasks to which they are assigned. In this section, we investigate the various applications that MMR's can be assigned, such as cooperative robot reconnaissance, surveillance, exploration satellite clustering, underwater autonomous vehicles and a fleet of marines, aerial vehicles and UAV.

The work in (Beard, Lawton et Hadaegh, 2001) presents the problem of coordinating multiple spacecraft subsuming leader-following, behavioral, and virtual-structure to fly in tightly formations. In (Encarna et Pascoal, 2001) marine craft trajectory tracking and trajectory tracking combination is developed based on maneuver modified trajectory tracking. They performed this algorithm for the coordinated operation of an autonomous surface craft and an autonomous underwater vehicle. The work in (Inalhan, Stipanovic et Tomlin, 2002) solves coordination problem of interconnected nonlinear discrete-time dynamic systems using a decentralized optimization method with multiple decision makers and applied it to a multiple unmanned air vehicle system. In (Huntsberger *et al.*, 2003) a coordination method of multiple robots is developed to perform a cliff traverse for science data acquisition and site construction operations that include grasping, hoisting, and the transport of extended objects such as large array sensors. The control of wireless mobile ad-

hoc networks (MANETs) is another collection of autonomous nodes that communicate with each other without using any fixed networking infrastructure (Wang, Zhou et Ansari, 2003).

In (Huang *et al.*, 2006) mobile robots are considered for highway safety applications where they automatically deploy and maneuver safety barrels commonly used to control traffic in highway work zones. The leader in the group performs the complex sensing and computation and the followers perform simple operations under the leader's guidance.

Finally, (Ghommam et Mnif, 2009) considers coordinating a group of underactuated ships along given trajectories using a combination of Lyapunov direct method, backstepping, and concepts from graph theory while holding a desired formation pattern for the group.

1.3 Trajectory Tracking

This section explores techniques to refine the trajectory tracking for mobile robots. Trajectory tracking can be explained as a tracking system that delivers a consistent high dynamic tracking performance to enable an efficient, smooth and continuous robot movement along a desired trajectory.

In recent years, the use of nonholonomic mobile robots has attracted the attention of researchers not only because of their practical applications, but also the theoretical challenges of both their nonholonomic characteristic and their nonlinearity modeling. However, the issues associated with nonlinearity modeling are unable to be solved by conventional linear control theory, and therefore other possibilities have therefore been explored by various researchers.

Several different nonlinear control strategies have been proposed for the trajectory tracking of a mobile robot such as the work of (Fierro et Lewis, 1995) which developed an 'adaptive backstepping method' with unknown parameters. (Fukao, Nakagawa et Adachi, 2000) propose the backstepping method for the dynamic model of a mobile robot.

In (Maalouf, Saad et Saliah, 2006) a robust fuzzy logic controller is presented for the trajectory tracking of a mobile robot based on controlling the robot at a higher level. The controller is highly robust and flexible and automatically follows a sequence of discrete waypoints, and no interpolation of the waypoints is needed to generate a continuous reference trajectory.

In (Antonelli, Stefano et Fusco, 2007) a trajectory tracking approach based on a fuzzy-logic set of rules which emulates the human driving behavior is proposed to achieve good tracking. The input and output of the fuzzy system are the approximate information concerning the next bend ahead of the vehicle and the cruise velocity respectively. The work of (Zhu et Yang, 2007) display a neurofuzzy-based approach with two behaviors, target seeking and obstacle avoidance, which are used for the combined coordination of the sensor information and the robot motion. To smooth the trajectory, a learning neural network technique tunes the parameters of membership functions. A state memory strategy is proposed for resolving the "dead cycle" problem.

The work in (Ferrara et Rubagotti, 2008) introduces a gradient-tracking approach to comply with the nonholonomic nature of the robot. Two control laws are designed by suitably transforming the system model into a couple of auxiliary second-order uncertain systems. Distributed sensor-network spaces are used in (Hwang et Chang, 2008) via fuzzy decentralized sliding-mode control for the trajectory tracking and obstacle avoidance of a car-like mobile robot. In (Park *et al.*, 2009) an ‘adaptive neural sliding mode control method’ is presented for trajectory tracking of mobile robots with model uncertainties and external disturbances. They use self recurrent wavelet neural networks for approximating arbitrary model uncertainties and external disturbances in the dynamic of a mobile robot.

1.4 Coordination Strategies

The coordination and control of MMR's can be performed in centralized or decentralized systems or a combination known as hybrid systems. However, in this section we consider the attributes of both centralized and decentralized algorithms.

1.4.1 Centralized Algorithms

In a centralized system, a powerful core unit makes decisions and communicates with the robots in the team. This core unit can optimize robot coordination, accommodate individual robots faults and monitor the accomplishment of the mission. However, it is possible that any faults in the core can facilitate a failure of the whole system. Centralized approaches do not scale well due to the fact that as the formation size increases, it incurs a large communication overhead. This is true even when employing the most advanced optimization solvers.

In (Mariottini *et al.*, 2005) a centralized feedback linearizing control strategy is used with an extended Kalman Filter to achieve a desired formation. In (Cruz et Carelli, 2006) a centralized control scheme and PD controller are proposed to ensure velocity tracking, but the derivative of the control velocity appears neglected. In (Huang *et al.*, 2006) a centralized localization and control method for long distance navigation is considered for follow-the-leader movement of a heterogeneous group of mobile robots. The algorithm allows the follower robots to be inexpensive by using minimal sensing and being computationally simple, and is therefore suited to missions with high-risk tasks. The work in (Harmati et Saffiotti, 2009) proposes a centralized collision free target tracking problem of a multi-agent robots' system by proposing a game theory concept using a semi-cooperative Stackelberg equilibrium point and a formation component in the individual cost functions.

In (Mehrjerdi, Saad et Ghommam, 2010c) an intelligent centralized coordination control and trajectory tracking algorithm is proposed and tested for a group of mobile robots using a fuzzy model. The designed fuzzy model employs two behaviors; being trajectory tracking

and coordination which work together to instruct the robots to move in formation travelling on specific forward trajectories.

1.4.2 Decentralized Algorithms

A decentralized system is a robotic team that lacks any centralized organization of the command structure. The robustness of the system to the failure or loss of individual member robots, scalability and being less affected by errors in communication or computation are all advantages of this type of approach. However, the failure of decentralized multiple-robot systems to make intelligent collective decisions is a significant obstacle to their deployment in the real world.

In (Keviczky, Borelli et Balas, 2006) a decentralized approach is presented where each robot optimizes locally for itself at each update, as well as for every neighbor. In (Keviczky *et al.*, 2008), the decentralized receding horizon planner is solved using mixed integer linear programming, where every robot is allotted a time slot in which to compute its own dynamically feasible collision free trajectory. In (Ren et Sorensen, 2008) a distributed formation control is proposed that accommodates an arbitrary number of group leaders and allows for arbitrary inter robot coupling on both the formation state estimation level and the vehicle control level. The vehicles come into agreement on the time varying position and orientation of the virtual center by using an extended consensus based estimation algorithm. In this algorithm, robustness against a single point of failure is improved by increasing the number of group leaders. In (Defoort *et al.*, 2009) a decentralized receding horizon motion planner is presented for a team of mobile robots subject to constraints imposed by sensors and the communication network. The advantage of the proposed algorithm is that each vehicle only requires local knowledge of its neighboring vehicles.

In (Lee et Chong, 2009) formations controls for a team of anonymous mobile robots based on a decentralized leader-referenced and neighbor-referenced are presented performing a task through cooperation. In (Ray *et al.*, 2009) a decentralized collision free coordination and

navigation is adapted where agents decide their own behaviors onboard depending upon the motion initiative of the master agent of the formation. In these approaches, any agent can estimate the behavior of other agents in the formation therefore reducing the dependency of an individual agent on other agents while it makes decisions. The reduction of communication burden on the formation where only the master agent broadcasts its motion status per sampled time is an advantage of this method.

1.5 Coordination and Control Algorithms

Different architectures and strategies have been developed in either centralized or decentralized methods in order to control and coordinate a team of mobile robots. These include: behavior-based, virtual structure, leader follower, graph-based and potential field approaches. In this section we give more explanation and references to each of these algorithms.

1.5.1 Virtual Structure

In the virtual structure approach, the entire formation is treated as a single entity. The desired motion is assigned to the virtual structure that traces out the trajectory for each member of the formation to follow. The main advantage of using this approach is that the behavior coordination of the robot group is relatively simple; however the main disadvantage is its inherent centralization which can lead a single point of failure for the whole system.

The work in (Beard, Lawton et Hadaegh, 2001) proposes a flying spacecraft formation based on a virtual structure where the coordination mechanism is specifically identified as the states of the formation control block and the states of the supervisor, and where the feedback to the formation is explicitly defined. In (Egerstedt et Hu, 2001) a model-independent coordination strategy is studied in combination with a desired reference trajectory for a nonphysical virtual leader. This method decouples the coordination problem into one planning problem. If the robots track their respective reference points perfectly, or if the tracking errors are bounded,

this method stabilizes the formation error. The work in (Ghommam *et al.*, 2010) presents a combination of virtual structure and trajectory tracking approaches to derive the formation architecture. A formation controller is proposed for the kinematic model of mobile robots. The approach is then extended to consider the formation controller by taking into account the physical dimensions and dynamics of the robots.

1.5.2 Behavior Based Methods

The behavior based model employs several behaviors for each robot and the final control is derived from a weighting of the relative importance of each behavior, but there is lack of modeling for the subsystems or robot surroundings. In (Balch et Arkin, 1998) a reactive behavior-based approach is introduced that implements formations integrated with navigational behaviors to enable a robotic team to reach navigational goals, avoid hazards and simultaneously remain in formation. In (Antonelli, Arrichiello et Chiaverin, 2009) a null-space-based behavioral control in presence of static and dynamic obstacles is introduced for a team of mobile robots. The NSB strategy is inherited from the singularity-robust task-priority inverse kinematics for industrial manipulators.

1.5.3 Leader Follower Approaches

In the leader follower method, one of the robots is designated as the leader, with the rest robots as followers. The follower robots need to position themselves relative to the leader and to maintain a desired relative position with respect to the leader. This method is characterized by simplicity and reliability, but its main disadvantage is that there is no explicit feedback from the followers to the leader.

The work in (Huang *et al.*, 2006) presents the control and localization of a heterogeneous group of mobile robots where inexpensive sensor-limited and computationally limited robots follow a leader robot in a desired formation over long distances. The proposed method is limited in that the leader needs to generally maintain line-of-sight contact with the followers.

The work in (Consolini et al., 2008) deals with leader–follower formations of nonholonomic mobile robots where the robots' control inputs are forced to satisfy suitable constraints that restrict both the leader's set of possible trajectories and the admissible positions of the follower with respect to the leader. In this algorithm, the follower position is not rigidly fixed with respect to the leader but varies in proper circle arcs centered in the leader's reference frame. The research in (Chen et al., 2009b) presents a receding-horizon leader follower control to yield a fast convergence rate of the formation tracking errors, and works to solve the formation problem of multiple nonholonomic mobile robots with a rapid error convergence rate. A separation–bearing–orientation scheme for two-robot formations and separation–separation–orientation scheme for three-robot formations is presented to maintain the desired leader–follower relationship. In (Gu et Wang, 2009) a leader-follower flocking system is introduced where only a few members are group leaders with knowledge of a desired trajectory, while the majority of the members are group followers who can communicate with neighbors but do not have the global knowledge. The followers do not have any idea who the leaders are in the group, and all group members estimate the position of flocking center by using a consensus algorithm via local communication in order to keep the flocking group connected.

1.5.4 Artificial Potential

In this section we review the research performed using artificial potential approaches.

In (Howard, Mataric et Sukhame, 2002) a distributed and scalable potential field based approach to deployment is presented, such that each node is repelled by both obstacles and by other nodes, thereby forcing the network to spread itself throughout the environment. In (Olfati-Saber et Murray, 2002), a natural potential functions obtained from the structural constraints of a desired formation is computed in a way that leads to a collision-free, distributed, and bounded state feedback law for each vehicle is used for the formation as well as the stabilization of multiple autonomous vehicles in a distributed fashion. The work in (Ogren, Fiorelli et Leonard, 2004) presents a stable control strategy for the coordination of groups of vehicles using virtual bodies and artificial potentials. In this scenario, each vehicle

in the group serves as a mobile sensor with the vehicle network being regarded as a mobile and reconfigurable sensor array.

In (Ge et Fua, 2005), a new concept is presented for artificial potential trenches that effectively control the formation of a group of robots. This method improves the scalability and flexibility of robot formations when the team size changes, while at the same time allowing the robot formations to adapt to obstacles. In (Barnes, Fields et Valavanis, 2009) an artificial potential field generated from normal and sigmoid functions is used for swarms of unmanned vehicles. The artificial potential functions and limiting functions are combined to control swarm formation, orientation, and swarm movement as a whole.

1.5.5 Graph theory Approaches

Graph theory is the study of graphs: mathematical structures used to model pair-wise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices or 'nodes' and a collection of edges that connect pairs of vertices. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another. Some research has been performed on the coordination of MMR using graph theory.

In (Moreau, 2005) a model of network of agents is presented with time-dependent communication links. In the model, each agent updates its present state based on the current information received from neighboring agents. The stability analysis is based on a blend of graph-theoretic and system-theoretic tools with the notion of convexity playing a central role. In (Ren et Beard, 2005) the problem of information consensus among multiple agents is introduced in the presence of limited and unreliable information exchange with dynamically changing interaction topologies. Both discrete and continuous update schemes are proposed for the information consensus. They claim that information consensus under dynamically changing interaction topologies can be achieved asymptotically if the unions of the directed interaction graphs have a spanning tree frequently enough. In (Ren, 2007) consensus

algorithms are presented with initially a constant reference state using graph theoretical tools, and then a time-varying reference state, to show the necessary and sufficient conditions under which consensus is reached on the time-varying reference state. In (Pereira, Kumar et Campos, 2008) the relative configurations of constraints between robots are modeled using a graph where each edge is associated with the interaction between two robots. This theory develops a decentralized motion control system that leads each robot to their individual goals while maintaining the constraints specified on the graph. The work of (Purvis, Astrom et Mhammash, 2008) investigates the problem of decentralized planning during the motion of a team of cooperating mobile robots subject to the constraints on the relative configuration imposed by the nature of the task they are executing. The constraints between robots are modeled using a graph where each edge is associated with the interaction between two robots and describes a constraint on the relative configurations.

1.5.6 Intelligent Control

Intelligent controllers simplify the computations used by the algorithm controller. They mimic the way the human brain makes decisions by grouping similar objects together, and so create faster and more accurate response times in the decision making process. This has distinct advantages in MMR modeling, where multiple robots are moving along designated trajectories and simultaneously being directed with rapid velocity changes. In (Gu et Hu, 2008) a fuzzy logic is proposed to the separation component where the Takagi-Sugeno rules and Gaussian membership functions are used. For fixed network flocking, a standard stability proof by using LaSalle's invariance principle is provided. For dynamic network flocking, a solution definition is given for non-smooth dynamics where stability is proved by a LaSalle's invariance principle. In (Dierks et Jagannathan, 2010) a combined kinematic/torque output feedback control law is developed for leader-follower-based formation control using backstepping to accommodate the dynamics of the robots and the formation in contrast with kinematic-based formation controllers. A neural network is introduced to approximate the dynamics of the follower and its leader using online weight tuning. Furthermore, a neutral network observer is designed to estimate the linear and angular velocities of both the

follower robot and its leader. Lyapunov theory shows that the errors for the entire formation are uniformly ultimately bounded while relaxing the separation principle. In (Jolly, Kumar et Vijayakumar, 2010) fuzzy neural network for task planning and action selection for a mobile robot in a robot soccer system is used. A five layer fuzzy neural network system is trained by an error back propagation learning algorithm to impart a strategy based action selection. In (Mehrjerdi, Saad et Ghommam, 2010c) a fuzzy coordination control and trajectory tracking algorithm is proposed and tested for a group of autonomous mobile robots. Hierarchical controllers have also been developed based on fuzzy and PID to instruct the robots to move in formation and on specific forward trajectories. The fuzzy rules applied to the robots are defined by the kinematic limitation which is bounded by both linear and angular velocities as well as the length and curvature of the trajectories.

1.6 Collision free Coordination

The behavioral attribute of discerning the surrounding environment brings robot behavior one step closer to mimicking the vast array of human behaviors. By incorporating this ability, robots can become the ideal vehicles for the safe interrogation of unknown or dangerous environments. In this section we review the current research on the coordination and cooperation between MMR's in unknown environments.

In (Hollinger, Djugash et Singh, 2007) non-line-of-sight range measurements are used to define a framework for finding a non-adversarial target in cluttered environments using multiple robotic searches. They present two Bayesian methods for updating the expected location of a mobile target, and integrating these updates into planning. In (Cruz et Carelli, 2008) an obstacle avoidance method based on the concept of impedance with fictitious forces is introduced where a dynamic perimeter enclosing the formation is used to help the robot group efficiently avoid an obstacle. The fictitious forces are applied on the perimeter rather than to any single robot, and thus the group formation is maintained whilst being deviated. The perimeter changes its shape dynamically to allow the formation to successfully avoid an obstacle. The work of (Ko, Seo et Simmons, 2008) proposes a method to coordinate the

motion of multiple heterogeneous robots; including obstacle avoidance. They use a prioritization technique where priority is assigned to each robot and a robot with lower priority avoids any robots of higher priority. To avoid collision with other robots, the concepts of elastic force and potential field force are used. In (Peasgood, Clark et McPhee, 2008) obstacle-free trajectories for robots are created by the use of a graph and spanning tree representation. The work in (Yang *et al.*, 2008) considers a suboptimal model using predictive formation control and obstacle avoidance for a group of nonholonomic mobile robots. They use a potential function to define the terminal state penalty term, and a corresponding terminal state region is added to the optimization constraints. In (Chang *et al.*, 2009) a robot-deployment algorithm is explained that overcomes unpredicted obstacles and employs full-coverage deployment with a minimal number of sensor nodes. Without the location information, node placement and spiral movement policies are proposed for the robot to deploy sensors efficiently to achieve power conservation and full coverage. The research in (Do, 2009) propose dynamic cooperative controllers using potential as well as bump functions that force a group of mobile robots with limited sensing ranges to perform desired formation tracking as well as a guarantee there are no collisions between the robots.

The work of (Harmati et Saffiotti, 2009) proposes a collision free target tracking problem for a multi-agent robot system. The convergence of target tracking is improved by a new game theoretic concept using a semi-cooperative Stackelberg equilibrium point and a new formation component in the individual cost functions. To enhance the robustness, a PD like fuzzy controller tunes the cost function weights directly for the game theoretic solution and helps to achieve a prescribed value of cost function components. In (Hu, Zhao et Wang, 2009) a decentralized vision-based for target-tracking and collision-avoidance of autonomous robotic fish capable of 3-D locomotion is developed, and considers underwater applications with the uncertainties and complexity inherent in a hydro environment. In (Li, Yang et Seto, 2009) a neural network approach is proposed for a multi-robot system with moving obstacles. In (Skrjanc et Klancar, 2010) a Bernstein_Bézier curve is introduced for cooperative collision avoidance of MMR's. They use model predictive trajectory tracking to drive the robots along the obtained reference trajectories.

1.7 Communication

The diversity of the MMR's applications and their potential communication protocols has challenged the existing literature on artificial intelligence communication. The less coordination a multi-robot system requires, then the better it should scale to large numbers of robots.

In (Klavins, 2003) the scalability of multi-robot algorithms is studied and formalism, called CCL, is defined for specifying multi-robot systems and algorithms. In (Fax et Murray, 2004) the problem of decentralized cooperation among a collection of vehicles is presented. These vehicles perform a shared task with inter vehicle communication using algebraic graph theory in modeling the communication network and relating its topology to formation stability. The information flow can thus be rendered highly robust to changes in the graph, enabling tight formation control despite limitations in inter vehicle communication capability. In (Sepulchre, Paley et Leonard, 2007) a design methodology to stabilize isolated relative equilibria in a model of all-to-all coupled identical particles is presented moving in the plane at unit speed. In (Takahashi *et al.*, 2009) a unified system of fully distributed meshed sensor network composed of static wireless nodes is studied to show mobile robot cooperation that serves as a sink node. This system is capable of fully distributed peer-to-peer (P2P) ad hoc communication with ZigBee based protocol. A novel communication timing control employing coupled-oscillator dynamics is proposed, named phase-diffusion time-division method (PDTD). This aims at the realization of an ad hoc collision-free wireless communication network. The basic PDTD is extended so that it can exhibit flexible topological reconfiguration according to the moving sink node (robot). A mobile robot will function as a sink node and access the mesh network from an arbitrary position.

1.8 Contribution

In this thesis we focus on the problem of how to control and coordinate a group of mobile robots moving in group formation in an unknown environment that includes static or dynamic obstacles. The goal is for all the robots in the group to simultaneously reach their desired target points while maintaining an overall group formation. In order to achieve this outcome, a method is developed in which each robot is equipped with inbuilt sensors to provide environmental information to a designed algorithm. This algorithm uses a combination of hierarchical controllers based on nonlinear and PID techniques which efficiently instruct the robots to maneuver along their designated trajectories and also avoid any obstacles they encounter. Additionally, we have carried out a variety of simulation experiments to prove the validity of the designed algorithm to achieve this result.

In this research we focus on the question of how to facilitate smooth and efficient trajectory tracking, cooperative group behavior and obstacle avoidance for a MMR group. This thesis explores the use of exponential sliding mode control, Lyapunov technique and the ability to create an intelligent control system for a MMR group working in an unknown environment. Within the context of this research, an unknown environment may comprise both static and dynamic obstacles. Static obstacles are foreign stationary objects observed in the environment, while each robot can be considered as a potential dynamic obstacle for other members of the group.

By exchanging information over the communication network, the robots use control laws to track their desired trajectories while simultaneously adjusting their speed profiles on their individual trajectories. The direction angle that each robot moves is determined not in isolation, but by the desired trajectories and overall group formation required of all the robots. The control algorithms are implemented on multiple homogeneous mobile robots called EtsRos.

A crash avoidance behavior is designed so that the robots within the group are assigned a priority status. If there is an imminent collision between robots, this behavior selects the one(s) that must stop to avoid a crash while allowing the priority robot to move forward or change its desired trajectory. Once the priority robot moves away and the crash danger passes, the other robots are allowed to continue moving forward in group formation.

The core of designed algorithm uses fuzzy logic which tries to imitate the way the human brain processes and responds. A robust fuzzy coordination algorithm is developed so that if any of the robots lose their coordination, they can resume it again once the problem is solved. For example, if there is the chance of an imminent crash amongst any of the robots forcing them to temporarily abandon their place in the formation, the coordination algorithm will then re-acquire their correct positions once the crash has been overcome. We also propose a fuzzy logic approach to the monitoring and deciphering of the information sourced from the surrounding environment. This environmental information is gathered using a combination of lateral and longitudinal sonar sensors mounted on the robots. A fuzzy model coordination construct manipulates the robots into different formations, and requires them to follow their desired trajectories in either a clear environment or one that contains obstacles.

The choice of using of a fuzzy coordination algorithm for robot control gives both flexibility and adaptability to the individual robots when maintaining group coordination. The robots must be able to dynamically change their velocity or desired trajectory when confronting obstacles or when they lose coordination with the rest of the group. The algorithmic model precisely guarantees that all the robots within the group will reach their target points both individually and in formation.

CHAPTER 2

TRAJECTORY TRACKING CONTROL OF A NONHOLONOMIC MOBILE ROBOT

This chapter explores techniques to improve upon previous trajectory tracking methods enabling a mobile robot to reach its desired target point. Trajectory tracking can be defined as a tracking system that delivers a consistent high dynamic tracking performance to enable an efficient, smooth and continuous robot movement along a desired trajectory.

Nonlinear control strategies for trajectory tracking proposed by other authors were reviewed to consider the best techniques for a single mobile robot which ultimately could be utilized for all robots within a MMR group. The research of (Yang et Kim, 1999) presents a robust sliding mode tracking control for a nonholonomic mobile robot. This uses a feedback linearized by the computed-torque method for the dynamic equation of the robot with its position being calculated by polar coordinates. The research of (Corradin et Orland, 2002) proposes the 'trajectory tracking problem' for a mobile robot by considering the presence of uncertainties in the dynamical model. Their proposed solution is based on a discrete sliding mode control which ensures both robustness and the implement ability of the controller. The works by (Kim *et al.*, 2003) and then (Chwa, 2004) propose a 'position and heading direction controller' using the sliding mode control technique for mobile robots. However, model uncertainties in the dynamics of the mobile robot were not considered, and it was assumed that the bounds of external disturbances were already known. The work by (Coelho et Nunes, 2005) proposes a 'robust control method' for the dynamic model with parametric uncertainties and external disturbances. The work by (Das et Kar, 2006) designs and implements an adaptive fuzzy logic for wheeled mobile robots. The work by (Hwang et Chang, 2007) proposes the concept of trajectory tracking and obstacle avoidance of a car-like mobile robot within an intelligent space via mixed H_2/H_∞ decentralized control using two distributed charge-coupled devices. In (Chen *et al.*, 2009a), the authors present an adaptive sliding mode dynamic controller for trajectory-tracking of wheeled mobile robots that

considers system uncertainties and disturbances. The work by (Lee *et al.*, 2009) proposes a 'sliding mode control' to asymptotically stabilize a mobile robot to a desired trajectory.

Upon evaluation of all these previous methodologies, this chapter proposes and tests different trajectory tracking methods for a single mobile robot by using three different control procedures that consist of:

- Exponential Sliding Mode Control
- Fuzzy Control
- Lyapunov Control

In principal, these techniques are combined by the use of a low level PID controller. This combination of using a low level PID controller and a high level algorithm controller creates an efficient nonlinear controller for the dynamic model of trajectory tracking with a mobile robot. This two level architecture control, based on the use of nonlinear and PID controllers, has been developed so that a mobile robot will exhibit a smooth tracking movement while moving forward on a predefined trajectory. The low level PID controller adjusts the speed of the left and right front wheel motors, while the high level controller coordinates the speed and movement of the robot by using a feedback controller. Experimental tests are performed in a laboratory environment employing different variations of robot trajectory tracking to show the ability and efficiency of these developed controllers.

2.1 Dynamic Tracking Control of a Nonholonomic Mobile Robot using Exponential Sliding Mode

Sliding Mode Control (SMC) developed in the 1950's took the attention of the authors because it employs a nonlinear control strategy that uses a high speed switching control law with a discontinuous property (Slotine et Li, 1991). The work of (Hung, Gao et Hung, 1993; Hwang, 2004; Utkin, 1977) evaluate SMC for the control of mobile robots, showing it has merit to deliver a consistent high dynamic tracking performance due of its simple structure, fast response, good transient performance, and robustness with regard to parameter

variations. However, the most problematic issue in SMC applications is chattering, that is the high frequency finite amplitude control signal, which due to its interference makes it impossible for use with real physical systems (Levant, 1993; Park, Chwa et Hong, 2006; Sankaranarayanan et Mahindrakar, 2009). There are different causes of chattering which can exist in many situations such as the presence of fast actuators and sensors (Boiko, 2005; Fridman, 2003), parasitic dynamics, time delay and hysteresis (Utkin, Guldner et Shi, 1999), and the discontinuity of the sign function on the sliding manifold (Utkin, 1977).

In order to attenuate and / or eliminate chattering in SMC systems, the following proposals have been studied by other authors. These are the use of a continuous smooth approximation (Levant, 1993; Shtessel, Shkolnikov et Brown, 2003; Slotine et Li, 1991) replacing the 'sign' function with a 'boundary layer' function (Shima, Idan et Golan, 2006), or the use of applied fuzzy logic to adjust the boundary layer function (Choi et Kim, 1997). Although these methods measurably decrease control chattering, they also unfortunately reduce the robustness and create an increase in the steady-state error.

In (Bartolini et al., 2003; Ferrara et Rubagotti, 2008; Levant, 2003; Riachy et al., 2008; Slotine et Li, 1991) the use of high-order sliding-mode approaches are also considered. These have the advantage of a higher accuracy of motions, as well as chattering reduction and finite-time convergence for systems with relative degree two (Bartolini *et al.*, 2001; Bartolini, Pisano et Usai, 2001). Also linked to this method, (Bartolini, Ferrara et Usai, 1998) proposes time optimal bang–bang control based on second order sliding mode to avoid chattering. Chattering is also mitigated by mixing second order and first order sliding mode control by (Bartolini et al., 2000), relying on the establishment of a hierarchy in the reaching phase. The work in (Boiko *et al.*, 2007a) also develops a chattering analysis with second-order sliding-mode controllers caused by the presence of fast actuators. Still, the main disadvantage of using the high-order sliding model is in its approach, due to the fact that the implementation becomes too complex along with the higher order sliding surface.

Observer-based approaches are also effective in the presence of unmodeled dynamics, for example (Boiko *et al.*, 2007b) analyze the effect of introducing a fast actuator into a nonlinear system driven by the generalized suboptimal algorithm.

However, all of these methodologies designed to mitigate chattering essentially involve low-pass filters and thus sacrifice steady-state errors. There are other methods which try to reduce chattering; the work of (Boiko *et al.*, 2007b) proposes a 'sliding algorithm' which does not require the use of observers and differential inequalities, and (Xu, 2008) proposes a 'nonlinear robust control algorithm' with better performance in terms of chattering free and saturation protection with asymptotic stability. These previous methodologies have drawbacks in achieving efficient and smooth robot movement along a desired trajectory.

Upon consideration of these theories, the solution adopted in this thesis to create an efficient and smooth robot movement builds upon the technique presented by (Fallah, 2007), where the authors consider an exponential reaching law for multivariable systems. In this chapter, we propose and evaluate the exponential sliding mode method for the dynamic trajectory tracking of a mobile robot (Mehrjerdi et Saad, 2010). Furthermore, we describe experiments in which we analyze a comparison of exponential sliding mode with both conventional and second order sliding modes to show its performance on chattering reduction and trajectory tracking for a mobile robot.

2.1.1 Dynamic and Kinematic Modeling of Mobile Robot

Figure 2.1 shows the general model of the mobile robot which consists of two driving wheels mounted on the same axis at the front while the two back wheels can freely rotate.
In this figure, r is the radius of each driving wheel, L is the distance between driving wheel and the axes of symmetry, C_c is the center of mass of the mobile robot, $C_0 - XY$ is the coordinate system fixed to the mobile robot, C_0 is the origin of the coordinate system $C_0 - XY$ and is the center point between the right and left driving wheels, and d is the

distance from C_0 to C_c. . $q = [x, y, \psi]^T$ denotes the position and orientation vector of the robot, and $q_d = [x_d, y_d, \psi_d]^T$ represents the desired trajectory.

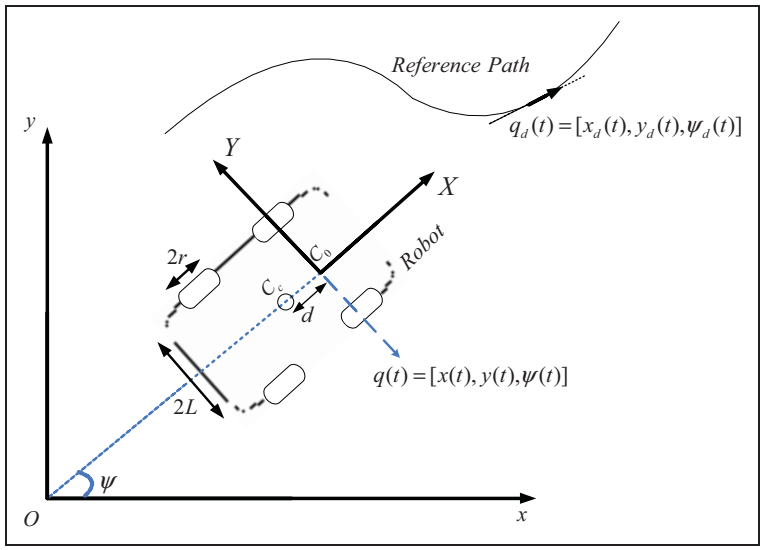


Figure 2.1 Mobile robot.

Figure 2.2 illustrates the block diagram which used to track a trajectory for a mobile robot using exponential sliding mode control. There are two levels of the controller, being low and high respectively. The low level controller is designed and implemented to adjust the right and left wheel velocities, whereas the higher level controller, which is an exponential sliding mode controller, is designed to follow a generated trajectory. Posture sensors are used to localize the robot, and a control algorithm uses this information to direct the robot along a desired trajectory.

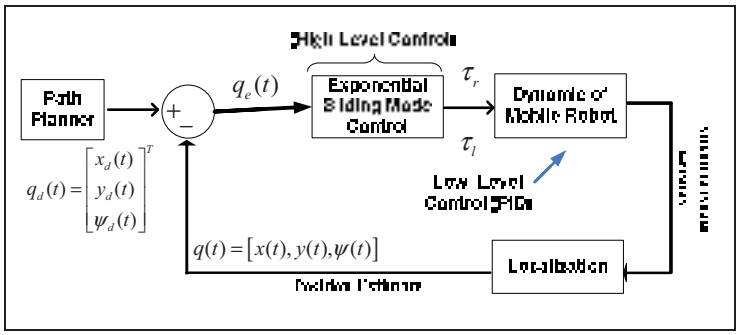


Figure 2.2 Infrastructure of control.

For this mobile robot, the general dynamic equation is described by (Fukao, Nakagawa et Adachi, 2000):

$$M(q)\ddot{q} + c(q,\dot{q})\dot{q} + G(q) = B(q)\tau + A^T(q)\lambda \tag{2.1}$$

where $\tau \in \mathbb{R}^r = [\tau_r, \tau_l]^T$ is the input vector and consists of motors' torques τ_r and τ_l which act on the right and left wheels, $\lambda \in \mathbb{R}^m$ is the vector of constraint forces, $M(q) \in \mathbb{R}^{n\times n}$ is a symmetric and positive-definite inertia matrix, $c(q,\dot{q}) \in \mathbb{R}^n$ is the centripetal and coriolis vector, $G(q) \in \mathbb{R}^n$ is the gravitational vector, $B(q) \in \mathbb{R}^{n\times r}$ is the input transformation matrix, and $A(q) \in \mathbb{R}^{m\times n}$ is the matrix associated with the constraints. We consider that the robot is moving on flat terrain and therefore conclude that $G(q) = 0$. Equation (2.1) can be adapted as:

$$\begin{bmatrix} m & 0 & -2m_\omega d\sin\psi \\ 0 & m & 2m_\omega d\cos\psi \\ -2m_\omega d\sin\psi & 2m_\omega d\cos\psi & I \end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{\psi}\end{bmatrix} + \begin{bmatrix} -2m_\omega d\,\dot{\psi}^2\cos\psi \\ -2m_\omega d\,\dot{\psi}^2\sin\psi \\ 0\end{bmatrix}$$
$$= \frac{1}{r}\begin{bmatrix} cos\,\psi & cos\,\psi \\ sin\,\psi & sin\,\psi \\ L & -L\end{bmatrix}\begin{bmatrix}\tau_r\\ \tau_l\end{bmatrix} + \begin{bmatrix} sin\,\psi \\ -cos\,\psi \\ -d\end{bmatrix}\lambda \tag{2.2}$$

and

$$\begin{aligned} m &= m_c + 2m_\omega \\ I &= I_c + 2m_\omega(d^2 + L^2) + 2I_m \end{aligned} \tag{2.3}$$

Where m_c is the mass of the robot without the driving wheels, m_ω is the mass of each driving wheel plus the motor rotor, I_c is the moment of inertia of the platform without the driving wheels and I_m is the moment of inertia of each wheel and the motor rotor about a wheel diameter; the kinematic constraints can be denoted as:

$$A(q)\dot{q} = 0 \tag{2.4}$$

$\dot{x}\sin\psi - \dot{y}\cos\psi - d\dot{\psi} = 0$.

It is clear that the matrix $R(q) = \begin{bmatrix} c(L\cos\psi + d\sin\psi) & c(L\cos\psi - d\sin\psi) \\ c(L\sin\psi - d\cos\psi) & c(L\sin\psi + d\cos\psi) \\ c & -c \end{bmatrix}$ satisfies

$A(q)R(q) = 0$, where $c = \frac{r}{2L}$. And therefore, we have:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}c(L\cos\psi+d\ sin\ \psi) & c(L\cos\psi-d\ sin\ \psi)\\ c(L\sin\psi-d\ cos\ \psi) & c(L\sin\psi+d\ cos\ \psi)\\ c & -c\end{bmatrix}\begin{bmatrix}\omega_l\\ \omega_r\end{bmatrix} \tag{2.5}$$

where ω_r , ω_l represent the angular velocities of the right and left wheels. If we consider v , ω as the linear and angular velocities of the mobile robot, the relation between v , ω and ω_r , ω_l can be explained as:

$$\begin{bmatrix}\omega_l\\ \omega_r\end{bmatrix}=\begin{bmatrix}\frac{1}{r} & \frac{L}{r}\\ \frac{1}{r} & -\frac{L}{r}\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}\Rightarrow v=H\ v_1 \tag{2.6}$$

From (2.5) and (2.6) it is clear that:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}cos\ \psi & d\ sin\ \psi\\ sin\ \psi & -d(cos\ \psi)\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}=R_1v_1 \tag{2.7}$$

The derivative of equation (2.7) gives:

$$\dot{q}=R_1v_1\Rightarrow\ddot{q}=\dot{R}_1v_1+R_1\dot{v}_1\Rightarrow\ddot{q}=\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{\psi}\end{bmatrix}$$

$$=\begin{bmatrix}-\dot{\psi}\ sin\ \psi & \dot{\psi}\ d\ cos\ \psi\\ \dot{\psi}\ cos\ \psi & d\dot{\psi}\ sin\psi\\ 0 & 0\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}+\begin{bmatrix}cos\ \psi & d\ sin\ \psi\\ sin\ \psi & -d(cos\ \psi)\\ 0 & 1\end{bmatrix}\begin{bmatrix}\dot{v}\\ \dot{\omega}\end{bmatrix} \tag{2.8}$$

By simplification of equation (2.8) we get:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{\psi}\end{bmatrix}=\begin{bmatrix}-\dot{\psi}\ v\ sin\ \psi+\dot{\psi}\ d\ \omega\cos\psi+\dot{v}\ cos\ \psi+\dot{\omega}\ \mathrm{d}\sin\psi\\ sin\ \psi(\dot{v}+\dot{\psi}\ \omega d)+\ cos\ \psi(\dot{\psi}\ v-\dot{\omega}d)\\ \dot{\omega}\end{bmatrix} \tag{2.9}$$

From combining relations (2.2) and (2.9) and in our case setting d to zero we can conclude that $(q,\dot{q})=0$ and:

$$\dot{v}=\frac{\tau_r+\tau_l}{mr}\quad,\qquad\dot{\omega}=\frac{L(\tau_r-\tau_l)}{Ir} \tag{2.10}$$

2.1.2 Exponential Sliding Mode Control

In this section, we propose the theoretical structure of the developed exponential sliding mode technique for the dynamic trajectory tracking of a mobile robot. We first define the control algorithm by conventional and second order sliding modes, and then modify the algorithm to evaluate the response using exponential sliding mode.

Substituting (2.8) for relation (2.1) and multiplying by $R_1^T(q)$ and considering $d = 0$ gives (Chen *et al.*, 2009a):

$$\widehat{M}(q)\dot{v}_1 + \hat{C}v_1 = \hat{B}(q)\tau \tag{2.11}$$

where $\widehat{M} = R_1^T M R_1 \in \mathbb{R}^{2\times2} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}$, $\hat{C} = R_1^T M \dot{R}_1 \in \mathbb{R}^{2\times2} = 0$ and $\hat{B} = R_1^T B = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}$.

Equation (2.11) can be therefore be rewritten as:

$$\dot{v}_1 = \widehat{M}^{-1}(q)\,\hat{B}(q)\tau = \Gamma.\tau \tag{2.12}$$

where $\Gamma = \widehat{M}^{-1}(q)\,\hat{B}(q) = \frac{1}{mrI}\begin{bmatrix} I & I \\ Lm & -Lm \end{bmatrix}$.

Velocity tracking error and its derivative can be defined as:

$$\begin{cases} e = v_d - v \\ \dot{e} = \dot{v}_d - \dot{v} \end{cases} \tag{2.13}$$

The first step in the sliding mode control is to choose the switching function s in terms of the tracking error. The general switching function for sliding mode control can be considered as:

$$S = e + \lambda \int_0^t e(\tau)\, d\tau \tag{2.14}$$

When the sliding surface is reached, the tracking error converges to zero as long as the error vector stays on the surface. The convergence rate is in direct relation with the value of λ.

If we consider Lyapunov function as:

$$V = \frac{1}{2}S^T S \quad \Rightarrow \dot{V} = S^T \dot{S} \tag{2.15}$$

To make the system stable, we consider $\dot{S}$ as:

$$\dot{S} = -K\ sign(S)\ ,\ \forall t, K > 0\ \Rightarrow\ \dot{V} < 0 \tag{2.16}$$

The derivative of equation (2.16) gives:

$$\dot{S} = \dot{e} + \lambda e = -K\ sign(S) \tag{2.17}$$

By using equations (2.12) and (2.17) we get:

$$\dot{S} = [\dot{v}_d - \Gamma.\tau] + \lambda e = -K\ sign(S) \tag{2.18}$$

The control law can be defined as:

$$\tau = \Gamma^{-1}[\dot{v}_d + \lambda e + K\ sign(S)] \tag{2.19}$$

where $\Gamma^{-1} = -\frac{r}{2L}\begin{bmatrix} -Lm & -I \\ -Lm & I \end{bmatrix}$,

Arbitrary-order sliding controllers with finite-time convergence have been demonstrated in the work of Levant (Levant, 2003). An example of an arbitrary second order sliding mode (ASSM) can be considered as:

$$\dot{S} = -K\ sign\left(\dot{S} + \left|S^{\frac{1}{2}}\right| sign(S)\right) \tag{2.20}$$

Another form of higher-order sliding mode is called quasi-continuous higher-order sliding mode control (Levant, 2005). An example of quasi-continuous second-order sliding mode (QSSM) can be considered as:

$$\dot{S} = -K\ \left(\left|\dot{S}\right| + |S|^{\frac{1}{2}}\right)^{-1}\left(\dot{S} + \left|S^{\frac{1}{2}}\right| sign(S)\right) \tag{2.21}$$

The exponential sliding mode proposed in this section is given by:

$$\dot{S} = \frac{-K}{Q(S)}\ sign(S) = \dot{e} + \lambda e \tag{2.22}$$

where $Q(S) = \mu_0 + \left(1 - \mu_0\right)e^{-\alpha|S|}$, then equation (2.22) can be rewritten as:

$$\dot{S} = \dot{e} + \lambda e = \frac{-K}{\mu_0 + (1 - \mu_0)e^{-\alpha|S|}}\ sign(S) \tag{2.23}$$

where μ_0 is strictly positive offset less than 1 ($0 < \mu_0 < 1$), and α is strictly positive. With proposed exponential sliding mode if we consider the Lyapunov function as:

$$V = \frac{1}{2} S^T S \quad \Rightarrow \dot{V} = S^T \dot{S} \;\Rightarrow\; \dot{V} < 0 \tag{2.24}$$

This proof stability of system because $Q(S)$ is always strictly positive. If μ_0 is selected as 1, the equation (2.23) becomes as equation (2.16) which shows conventional sliding mode is subset of exponential sliding mode. In equation (2.23) with increasing $|S|$, $Q(S)$ tend to μ_0 , and consequently $K/Q(S)$ converges to K/μ_0 , which is greater than K. This means that $K/Q(S)$ increases in the reaching phase, and accordingly the attraction of the sliding surface will be faster. On the other side, with decreasing S , $Q(S)$ tend to 1 and then $K/Q(S)$ converges to K. This means that when the system approaches the sliding surface, $K/Q(S)$ gradually decreases and consequently reduces the chattering. The proposed exponential sliding mode will therefore dynamically adapt to the variations of the switching function by letting $K/Q(S)$ vary between K and K/μ_0. Figure 2.3 shows the switching function with the exponential sliding control for different values of K and μ_0. As can be seen in this figure, if we select $\mu_0 = 1$ the exponential sliding control will be the same as the conventional sliding mode control.

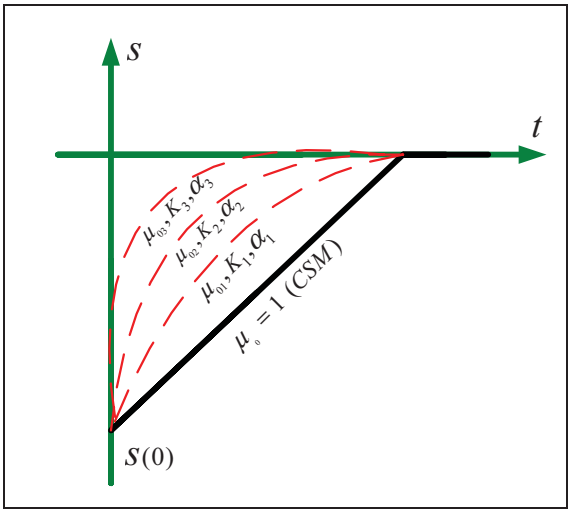


Figure 2.3 The switching function with the exponential sliding control for different values of K and μ_0 .

2.1.3 Experimental Results

The purpose of the experiments is to demonstrate the stability and the performance characteristics inferred from the theoretical development. Before presenting the experimental results, we briefly discuss the experimental setup, communication and design of the two level algorithm controllers being used for the trajectory tracking of an EtsRo mobile robot.

2.1.3.1 Experimental Setup

Figure 2.4 displays the structural design for the control and trajectory planning of the 'EtsRo' nonholonomic mobile robot used in the experimental tests. EtsRo is a mobile robot with two actuated wheels, with the front wheels being equipped with two DC motors using $7.5\,\text{Volt}, 175\,\text{rpm}$ which are installed on the right and left front wheels. The incremental encoders are mounted on the motors counting with a resolution of $6000\ Pulses/Turn$. The wheels have a radius of $r = 4.5\,cm$. The length, width and height of EtsRo are 23, $20\ and\ 11\ cm$ respectively. The total weight of the robot is around 2.3 kg. The maximum linear velocity is $1.12\ m/sec$ and the maximum angular velocity is $5.74\,rad/s$. The experimental tests on robot are performed in a laboratory using a flat terrain with a work area of 4×7 meters. By using the following equation, we can obtain both the linear and angular velocities of the individual robots:

$$v = \frac{v_R(t) + v_L(t)}{2} \quad , \quad \omega = \frac{v_R(t) - v_L(t)}{L} \tag{2.25}$$

Where $v_R(t)$ and $v_L(t)$ denote the right and left velocities and L shows the distance between the two actuated wheels.

ZigBee modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of a low-cost, low-powered wireless sensor network being used for communication. The modules operate within the 2.4 GHz frequency band, and the module range in the indoor

environment is up to 100 meters while the RF data rate is 250,000 $bits/s$. The PC communicates with the robots through a serial port with a modem working in API mode.

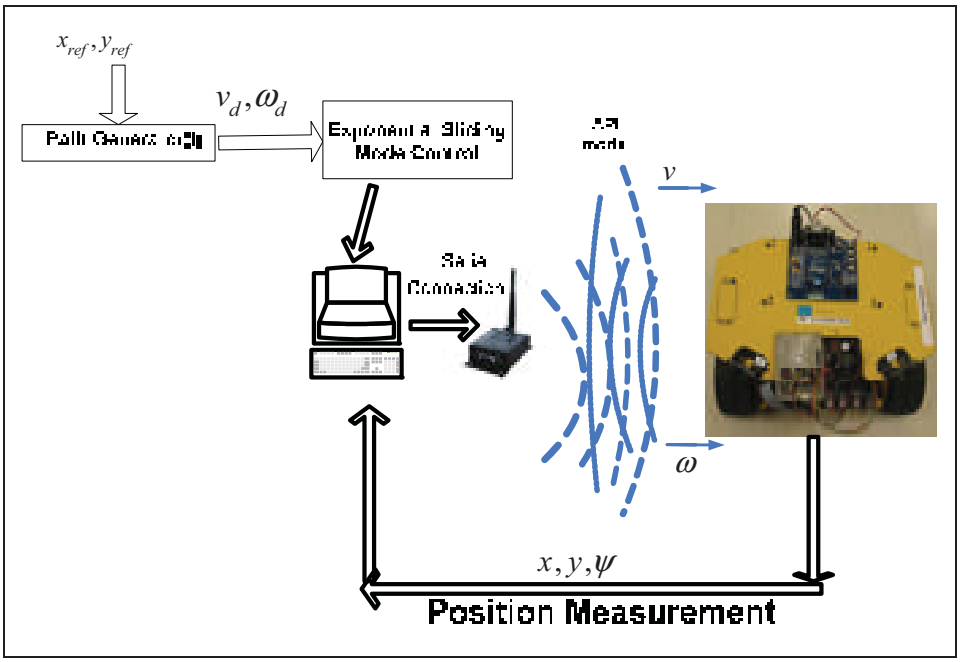


Figure 2.4 EtsRo and Control structural design.

The velocity of ZigBee modem is set to 9600 $bits/s$. API is used for serial-to-RF packetization and the frame-based API extends the level to which a host application can interact with the networking capabilities of the module. The robot has two second-order filters to eliminate sparks on the velocities. Two low level PID controllers are designed to achieve the best accuracy for the right and left motors with a sampling time of $T_s = 10\,ms$ which is designated by a microcontroller mounted on the robot. The high-level controller, which is a non linear controller, is designed in the real-time Simulink (Matlab) with a sampling time of $T_s = 50\,ms$. Figure 2.5 demonstrates the control and coordination architecture for the mobile robot.

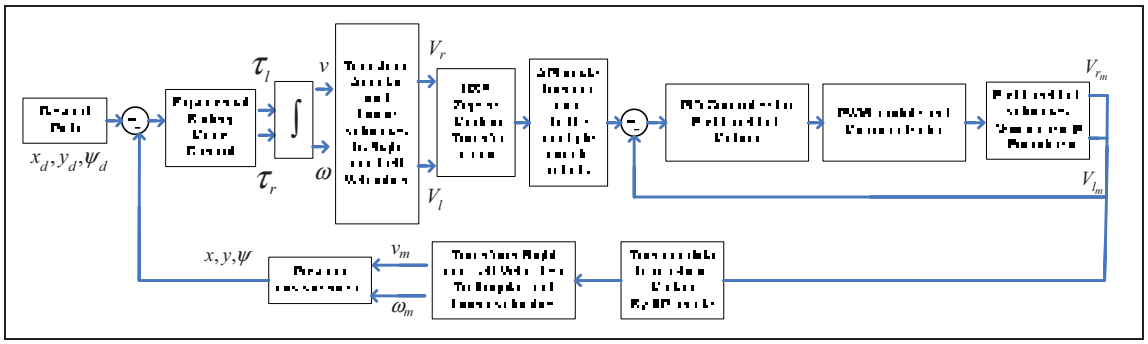


Figure 2.5 Control architecture of EtsRo.

2.1.3.2 Experimental Tests

In this section, we discuss the results of the trajectory tracking missions involving the use of an EtsRo mobile robot. Experimental tests are carried out to evaluate the performance of the trajectory tracking algorithm used for the EtsRo robot and are designed to illustrate the effectiveness of the proposed exponential sliding mode technique compared to the conventional and second-order modes. The dynamic parameters for the EtsRo robots are considered as the following:

$$m_c = 2.3\ kg\ , \qquad m_\omega = .28\ kg\ , \qquad I_c = .01\frac{kg}{m^2}, \qquad I_m = .0021\frac{kg}{m^2}$$
$$r = .04\ m\ , \qquad L = .1\ m, \qquad d = .02\ m$$

The experiment tests were performed on a sinusoidal-shape trajectory. In the first scenario, a CSM using control gains of $K = 1\ ;\ \lambda = 10$ is considered. In the second and third scenario, an ASSM and a QSSM with control gains $K = .8\ ;\ \lambda = 10$ are considered. In the last scenario, ESM with control gains $K = .2\ ;\ \lambda = 10\ ; \mu_0 = 0.1\ ; \alpha = 3$ is considered. The initial position of the robot is $[x(t_0), y(t_0), \psi(t_0)]^T = [0,1,0]^T$. Figure 2.6 presents the reference and actual robot trajectories. This figures show the robot moving effectively along its trajectory, indicating the exponential sliding mode to be experimentally successful. The linear and angular velocities of the robot in different scenarios are plotted in Figure 2.7. As can be seen in these figures, the CSM has the maximum chattering on input controls. The trajectory tracking errors $\boldsymbol{y_{err}}$ and $\boldsymbol{x_{err}}$ are shown in Figure 2.8.

The results obtained by the proposed exponential sliding mode and conventional sliding method are compared. Table 2.1 shows the results obtained by these methods with the trajectory tracking errors in which $x_{err-total}$, $y_{err-total}$, $\psi_{err-total}$, MSE and $Chatter$ can be calculated by:

$$x_{err-total} = \frac{\sum_{n=0}^{f} |x_{err(n)}|}{f} \qquad y_{err-total} = \frac{\sum_{n=0}^{f} |y_{err(n)}|}{f} \qquad \psi_{err-total} = \frac{\sum_{n=0}^{f} |\psi_{err(n)}|}{f}$$

$$MSE = \frac{\sum_{n=0}^{f} \sqrt{x_{err(n)}^2 + y_{err(n)}^2 + \psi_{err(n)}^2}}{f} \quad Chatter = \frac{\sum_{n=0}^{f} |v_{d(n)} - v_n|}{f} = \frac{\sum_{n=0}^{f} |v_{err(n)}|}{f} \tag{2.26}$$

Table 2.1 Comparison between CSM, ASSM, QSSM and ESM

Total errors	x_{err}	y_{err}	ψ_{err}	MSE	$Chatter$
Conventional Sliding Mode	0.0195	0.0367	0.0532	.0780	0.0898
Arbitrary Second-Order Sliding Mode	0.0089	0.0180	0.0579	0.0682	0.0352
Quasi Second-Order Sliding Mode	0.0191	0.0372	0.0287	0.0577	0.0479
Exponential Sliding Mode	0.0100	0.0324	0.0154	0.0418	0.0233

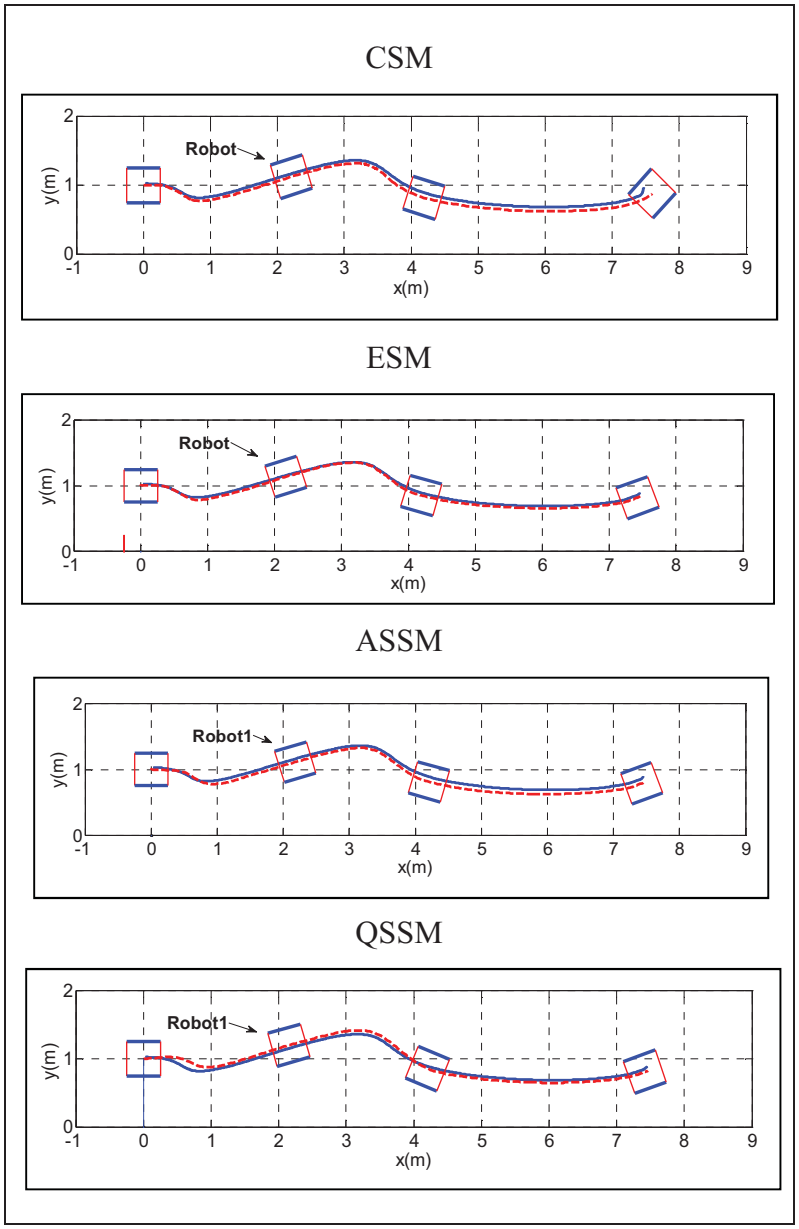
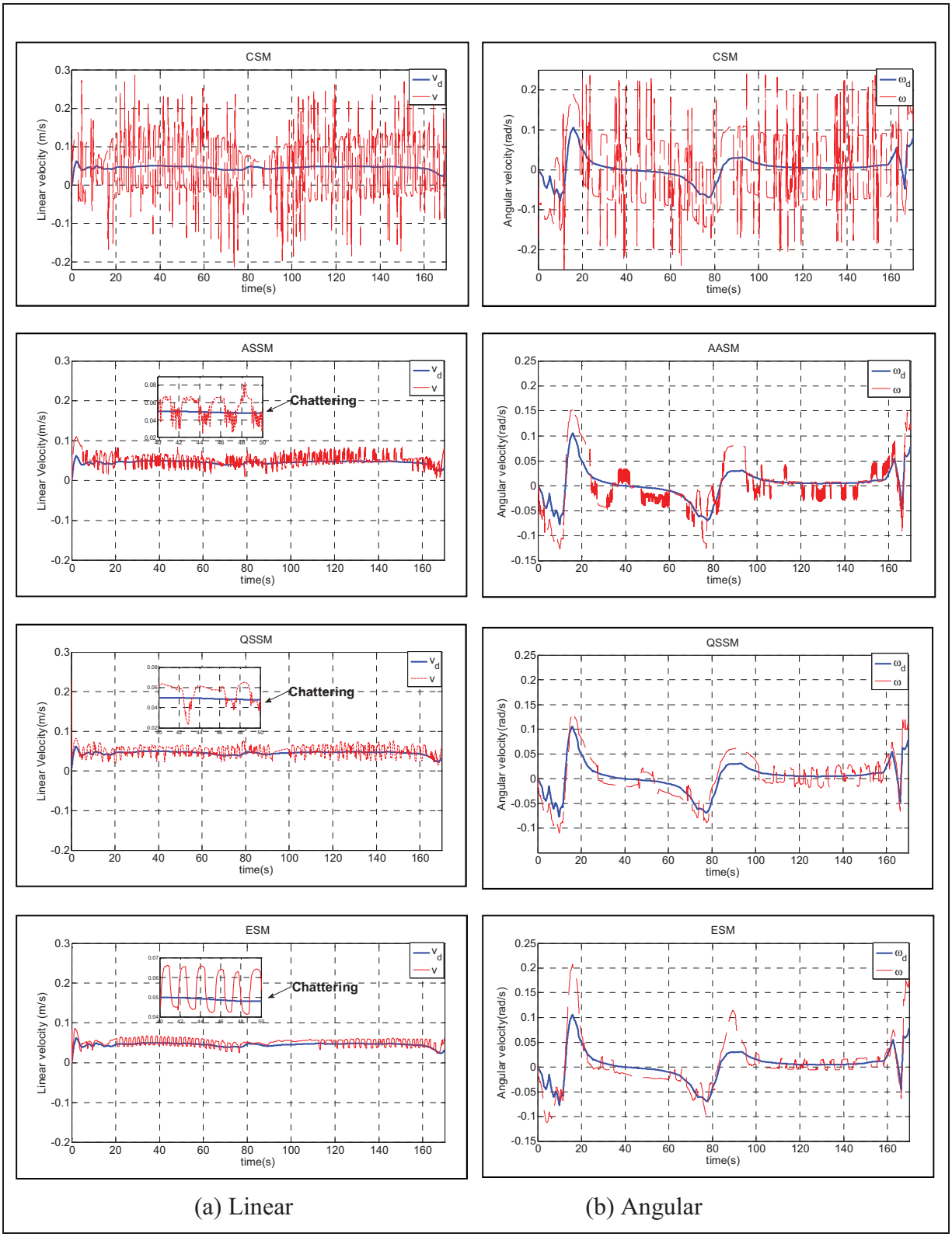


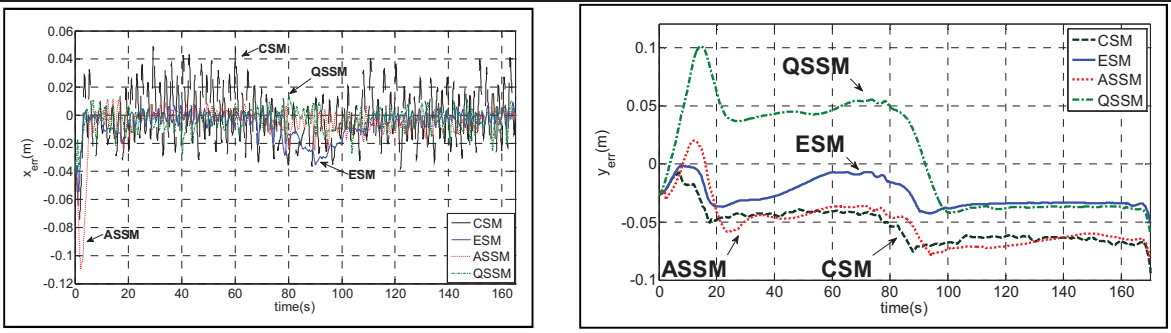
Figure 2.6 Reference and real robot trajectories.

MSE is defined as the mean square error, f is the number of the sampling points and $Chatter$ is defined as the chattering on the linear velocity of the robot. As this table shows, when the robot moves on its desired trajectory, the ESM produces a smaller mean square error compared to the CSM, AASM and QSSM, which therefore allows the robot follow its desired trajectory more precisely. Furthermore, the chattering present in ESM is less than the chattering produced by other methods.



(a) Linear (b) Angular

Figure 2.7 Velocities of robot.



(a) x_{err} (b) y_{err}

Figure 2.8 Trajectory tracking error.

2.2 Trajectory Tracking Based on Lyapunov Technique

In this section, a trajectory tracking approach is explained based on a Lyapunov function (Mehrjerdi, Saad et Ghommam, 2010a). The kinematic equation of this robot can be explained as:

$$\begin{cases} \dot{x} = v\cos(\psi) \\ \dot{y} = v\sin(\psi) \\ \dot{\psi} = \omega \end{cases} \tag{2.27}$$

where $P = [x, y, \psi]^T$ denotes the position and orientation vector of the robot; v and ω are linear and angular velocities respectively. The errors in posture with respect to a local frame of reference of the robot are given by:

$$e = \begin{bmatrix} x_{err} \\ y_{err} \\ \psi_{err} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \psi - \psi_d \end{bmatrix} \tag{2.28}$$

The time derivative of the posture errors take into account the constraint (2.4) which yields:

$$\dot{e} = \begin{cases} \dot{x}_{err} = v - v_d \cos(\psi_{err}) + y_{err}\omega \\ \dot{y}_{err} = v_d \sin(\psi_{err}) - x_{err}\omega \\ \dot{\psi}_{err} = \dot{\psi}_d - \dot{\psi} = \omega - \omega_d \end{cases} \tag{2.29}$$

The aim of a control law that executes the trajectory tracking task is to reduce the posture errors so they converge to zero. The proposed control inputs v and ω of the control law are to account for the velocity assignment of the robot along the trajectory as given by (Maalouf, Saad et Saliah, 2006):

$$v = K_x x_{err} + v_d \cos(\psi_{err}) \tag{2.30}$$

$$\omega = \omega_d + v_d\, K_y y_{err} + K_\psi \sin(\psi_{err}) \tag{2.31}$$

where $K_x\,, K_y\ and\ K_\psi$ are the control gains and positive constants.
By substituting v and ω in the errors of (2.29), we get

$$\begin{bmatrix} \dot{x}_{err} \\ \dot{y}_{err} \\ \dot{\psi}_{err} \end{bmatrix} = \begin{bmatrix} -K_x x_{err} + y_{err}\big(\omega_d + v_d(K_y y_{err} + K_\psi \sin(\psi_{err}))\big) \\ v_d \sin(\psi_{err}) - x_{err}\big(\omega_d + v_d(K_y y_{err} + K_\psi \sin(\psi_{err}))\big) \\ -v_d\big(K_y y_{err} + K_\psi \sin(\psi_{err})\big) \end{bmatrix} \tag{2.32}$$

Let the candidate Lyapunov function be defined as

$$V = \frac{1}{2}(x_{err}^2 + y_{err}^2) + \frac{1 - \cos(\psi_{err})}{K_y} \tag{2.33}$$

Differentiating relation (2.33) along the solutions of (2.32), we get

$$\dot{V} = -K_x x_{err}^2 - \frac{K_\psi}{K_y} v_d \sin^2(\psi_{err}) \tag{2.34}$$

Given that $K_x, K_y\ and\ K_\psi$ are all positive constants then $\dot{V} \leq 0$ and the system with the defined control law is stable.

2.2.1 Experimental Results

In this section, we discuss the experimental test results of the trajectory tracking missions of an EtsRo mobile robot in order to illustrate the performance of the proposed Lyapunov technique. These tests were performed firstly on a sinusoidal-shape trajectory and finally on a circle trajectory. In the first test, a sinusoidal-shape trajectory is considered. In this scenario, the initial position of robot is:

$[x(t_0), y(t_0), \psi(t_0)]^T = [-1,0,0]^T$ and nonlinear control gains are considered as:

$K_x = 1\,; K_y = 1\,; K_\psi = 1.$

Figure 2.9 presents the reference and actual robot trajectories while the trajectory tracking errors y_{err} and x_{err} are shown in Figure 2.10. Both these figures show the robot moving effectively along its trajectory. The linear and angular velocities of the robot in different scenarios are plotted in Figure 2.11.

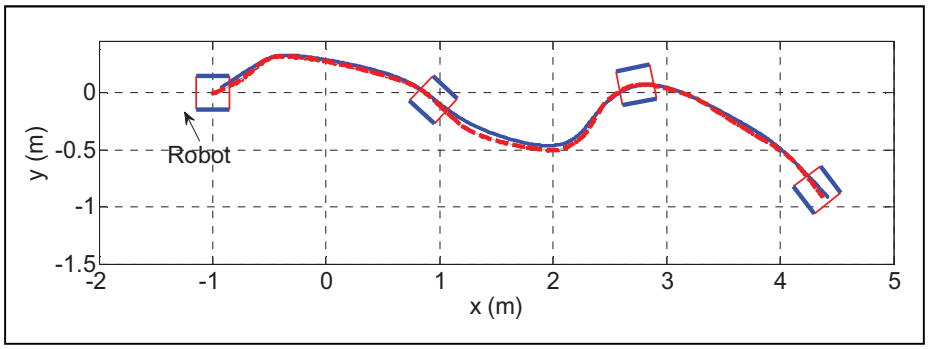
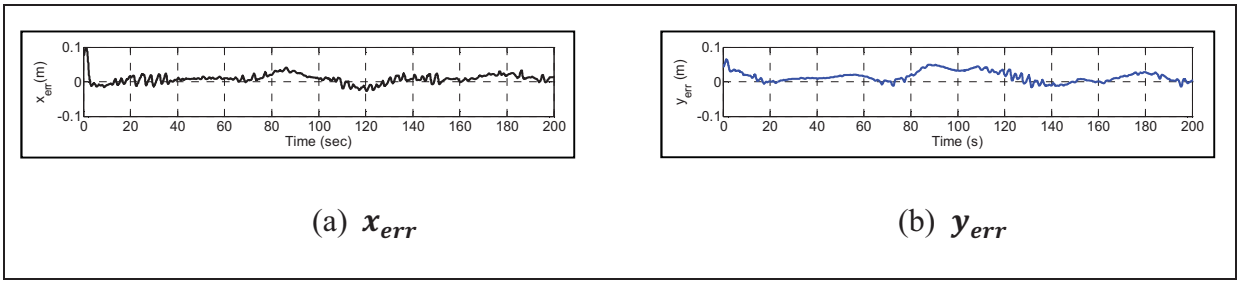


Figure 2.9 References and real robot trajectory.



(a) x_{err} (b) y_{err}

Figure 2.10 Trajectory tracking errors.

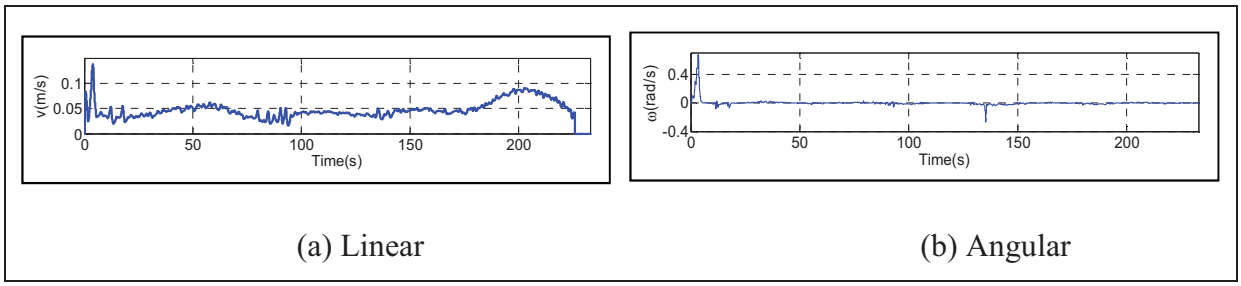


Figure 2.11 Velocity of robot.

In the second test, a circle trajectory is considered. In this scenario, the initial position of robot is: $[x(t_0), y(t_0), \psi(t_0)]^T = [0,1.5,0]^T$ and non-linear control gains are considered as: $K_x = 1\,; K_y = 1\,; K_\psi = 1.$

Figure 2.12 presents the reference and actual robot trajectories while the trajectory tracking errors y_{err} and x_{err} are shown in Figure 2.13. Both these figures show the robot moving effectively along its trajectory. The linear and angular velocities of the robot in different scenarios are plotted in Figure 2.14.

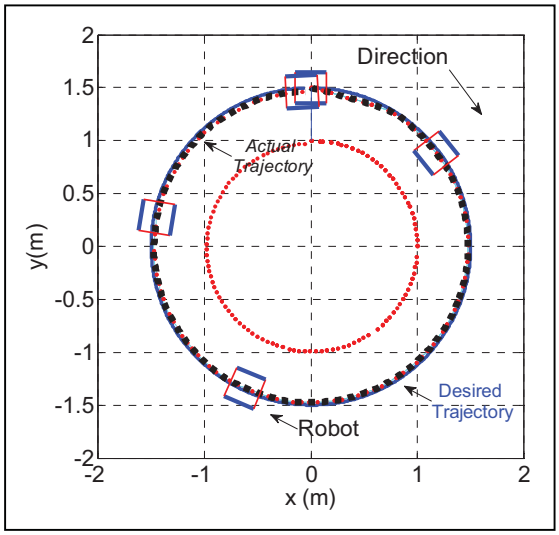


Figure 2.12 References and real robot trajectories.

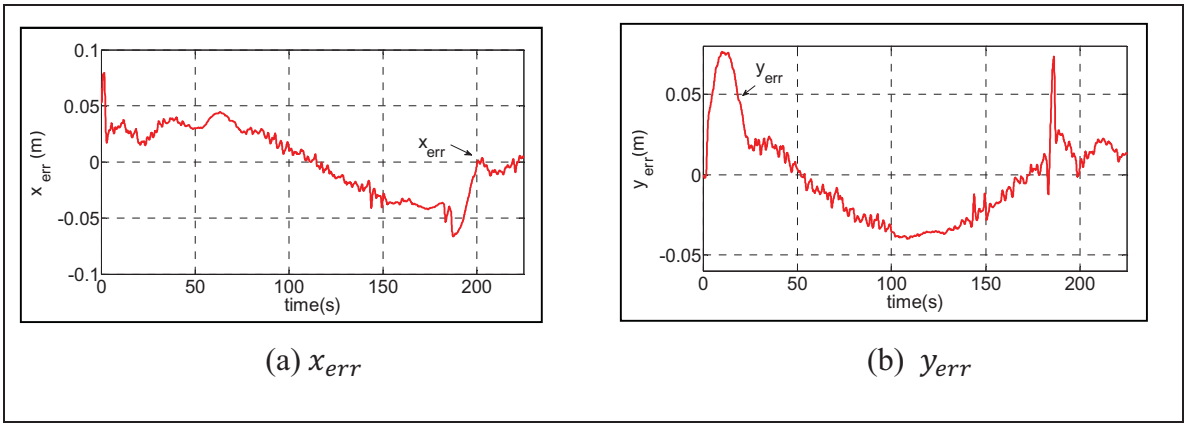


Figure 2.13 Trajectory tracking errors.

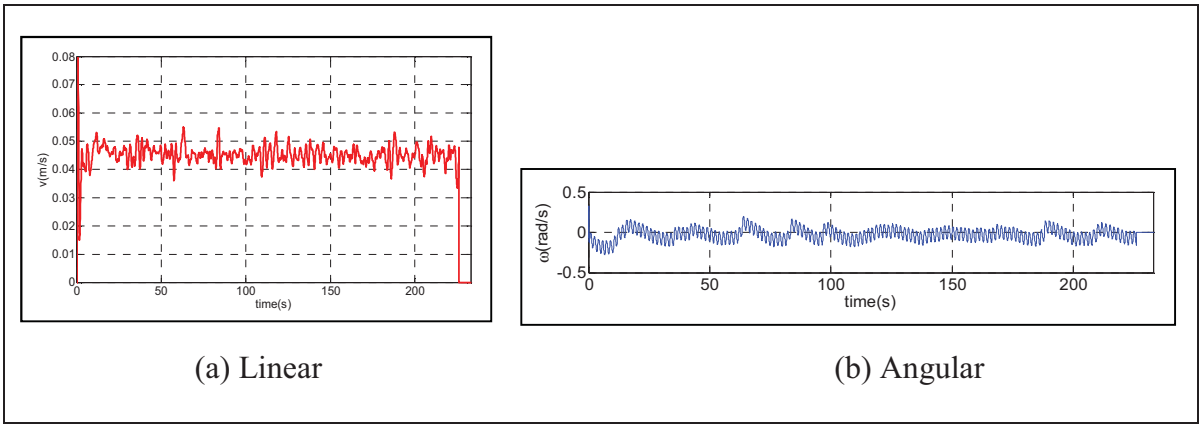


Figure 2.14 Velocity of robot.

2.3 Trajectory Tracking Based on Fuzzy Control

In this section, a trajectory tracking approach based on fuzzy control (Mehrjerdi *et al.*, 2010a) is used to implement a trajectory tracking controller. The core of the designed algorithm uses fuzzy logic which tries to imitate the way the human brain processes and responds to problems. Up until recently, computers could only process via logical patterns of programming. People however, think not only logically but can also recognize similarities or 'shades of grey'. This means that they can recognize that something possesses the qualities of several things at the same time and this mode of thinking is mimicked by the use of fuzzy logic. A two level hierarchical architecture control based on the fuzzy model and PID are

used for indoor multiple robots. A set of linguistic fuzzy rules are extended to implement expert knowledge under a variety of situations. The output of the fuzzy controller determines the linear and angular velocities of individual robots. Experimental tests are performed using an EtsRo mobile robot to test the ability of the controller's effectiveness. All control and trajectory planning in this section are implemented on the EtsRo. Figure 2.15 shows the general kinematic model of the mobile robot and a discretized trajectory. In this figure, $P = [x, y, \psi]^T$ denotes the position and orientation vector of the robot. $Q_{d(n)} = \left[x_{d(n)}, y_{d(n)}, \xi_{d(n)}\right]^T$ represents the coordination of n^{th} sample point on the trajectory where $n = 0, \ldots, f$. The trajectory is described by a set of discrete node positions $Q_{d(0)}$ to $Q_{d(f)}$ linked to each other starting from the initial position to the final desired position. $Q_{d(n)}$ is n^{th} discretized sample on the trajectory.

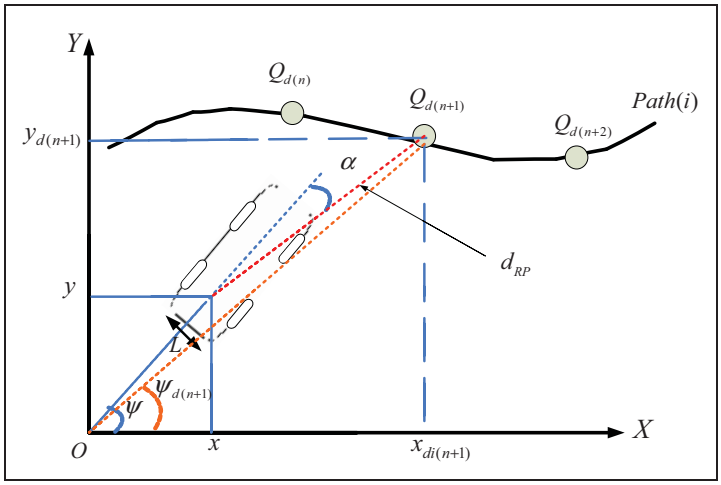


Figure 2.15 Kinematic model of robot.

Figure 2.16 shows the fuzzy system structure with the desired number of inputs and outputs. Inputs of fuzzy controller are d_{RP} , α , x_{err} and y_{err} where d_{RP} is the distance from the actual position of robot to the next desired position, α is the difference between the line joining the current position to the next desired position and the actual heading of the robot. The output of the fuzzy controller determines the linear and angular velocities of individual robots.

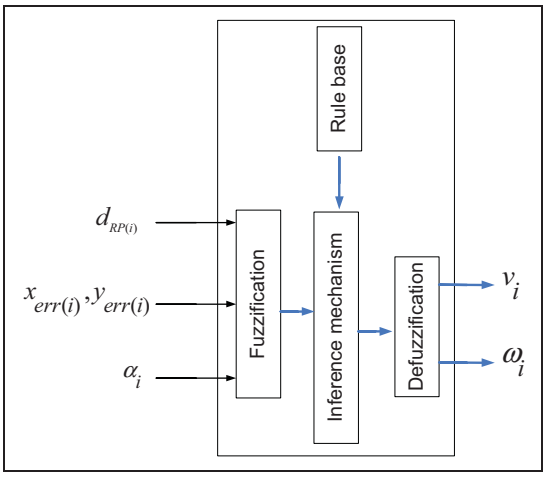


Figure 2.16 Fuzzy control structure.

The algorithm controller has two hierarchical levels of operation being low and high respectively. The high level controller is a fuzzy controller designed to follow generated EtsRo robot trajectories that are modeled by a fifth order polynomial and divided into segments for analysis. The idea to solving the problem of propelling a robot along a continuous desired trajectory is to divide the trajectory into discrete segments and consider the robot moving between discontinuous sampling points.

2.3.1 Trajectory Tracking Problem

The task of the fuzzy trajectory tracking controller is to command the robots to follow the trajectories in a smooth and continuous manner with the best possible precision. In order to achieve this goal, it is not necessary for the robots to pass exactly through specific sampling points on their trajectories, but they must at least pass within a proximity to them while reaching their final destination. For our purposes, trajectory tracking is categorized in two different groups:

1) The robot is placed on its predefined trajectory. In this scenario, which is shown in Figure 2.17, the robot tries to follow its trajectory and stay on its desired trajectory.

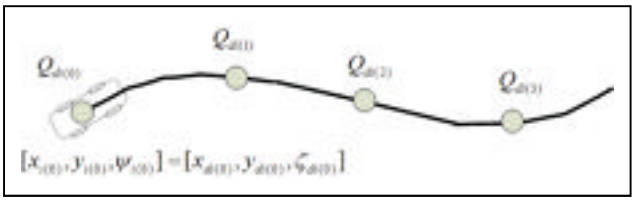


Figure 2.17 Robot is on the desired trajectory.

2) In the second modelling, the robot is not placed on its predefined trajectory. In this scenario, which is shown in Figure 2.18, the robot tries to move forward to reach and follow its desired trajectory.

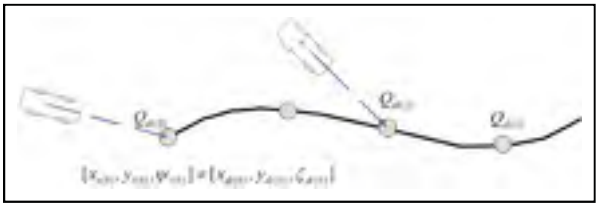


Figure 2.18 Robot is not on the desired trajectory.

We consider P and Q_d to represent the position of robot and the desired trajectory discretized. We let v_d, ω_d denote the desired linear and angular velocities assignment for the robot. The purpose of trajectory tracking is to make the robot's velocity $u = [v, \omega]^T$ track a desired velocity reference $u_d = [v_d, \omega_d]^T$ which means $\|u - u_d\| \to 0$ and also $\|P - Q_d\| \to 0$.

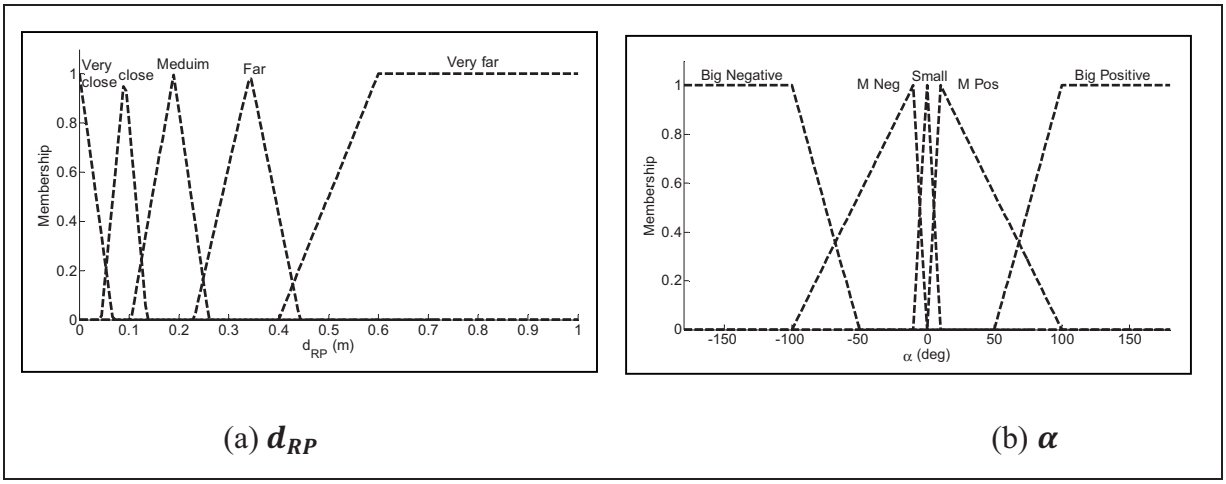
2.3.2 Fuzzy Trajectory Tracking

The form of the control law equation for trajectory tracking and cooperation is as follows:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} f_1(d_{RP}, \alpha, x_{err}, y_{err}) \\ f_2(d_{RP}, \alpha, x_{err}, y_{err}) \end{bmatrix} \tag{2.35}$$

The functions f_1 and f_2 are the control laws of a Sugeno type fuzzy controller. Sugeno controllers take in fuzzy inputs and outputs. The task of the trajectory following program is to make the robot passes in proximity of the sampling points in a continuous and smooth

manner. The behavior of the controller is such that if the discrete points are close to each other, then a higher precision of trajectory tracking is achieved, but the robots will move at a lower speed due to the processing of this higher number of sampling points. If less precision is required, the discrete points can be selected further apart and the robot will therefore move at higher speed. Here close sampling points with high precision are considered. The membership functions of inputs d_{RP} and α are shown in Figure 2.19.



(a) $\boldsymbol{d_{RP}}$ (b) $\boldsymbol{\alpha}$

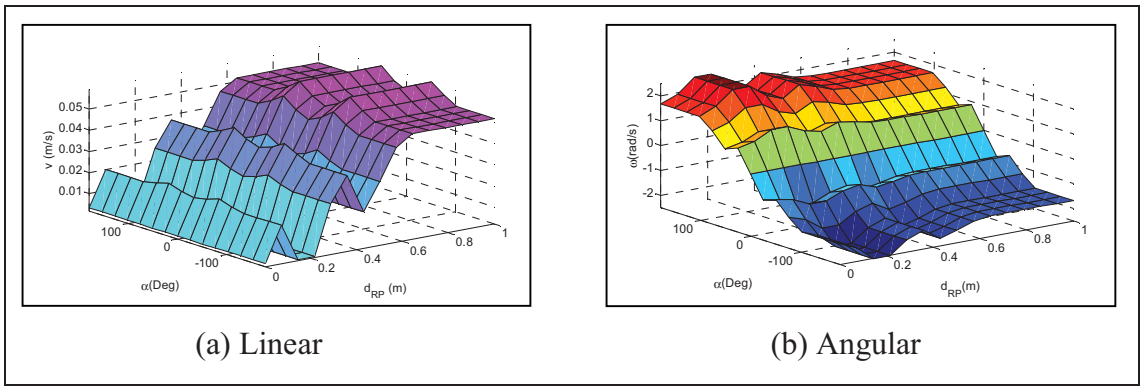
Figure 2.19 Membership functions.

The linear and angular velocities obtained by the fuzzy controller related to inputs d_{RP}, α are shown in the Figure 2.20. As can be seen in this figure, when the robot is far from the trajectory, or the distance between the actual position of robot and the next ahead sampling point is large, then its linear velocity will be increased. When the robot is close to the trajectory, or the distance between actual position of robot and the next ahead sampling point is small, the velocity will be decreased and the robot will move slower. In this figure, when the angle between the robot and the trajectory is large, the robot will have more angular velocity to decrease this error.

The reason for using x_{err} as an input to the fuzzy system in trajectory tracking can be described as follows: If we suppose that a robot should travel from $Q_{d(n)}$ to $Q_{d(n+1)}$, this means that it has to catch the targeting point $Q_{d(n+1)}$. However, if the robot passes points

$Q_{d(n+1)}$ and then $Q_{d(n+2)}$ in the robot reference, then when moving to the next step, the targeting point $Q_{d(n+1)}$ falls behind the actual position of the robot, which means $(x > x_d) \Rightarrow x_{err} < 0$, To solve this issue, we add the following rule to the fuzzy controller,

- If $x > x_d$ then robot will stop, until condition $x \leq x_d$ is fulfilled. Figure 2.21 depicts this issue,



(a) Linear (b) Angular

Figure 2.20 Velocities obtained by fuzzy controller.

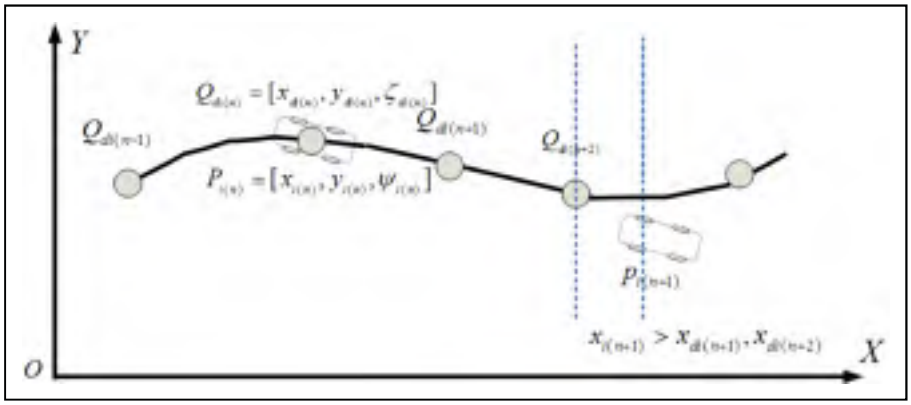


Figure 2.21 Robot passes targeting point ahead.

If robot is not on the trajectory, or the vertical position of robot in x-y coordination (y) is different from the next vertical sampling point on the trajectory, then y_{err} will be used as an extra input to the fuzzy controller. This input is used to help the robot turn toward the trajectory and catch it. Figure 2.22 shows the necessity of using the y_{err} as an input to the

fuzzy controller. If the robot is not on the trajectory and there is an error between the heading angle of the robot and the trajectory, the robot will turn to minimize this error. However, when this error is zero ($\alpha = 0$) , the robot will move to a trajectory parallel to the actual trajectory and therefore will never reach it. To solve this problem an angle is added to α which is λ. Using the y-coordination, the further the robots are from the trajectory the larger this angle becomes and will only reduce as the robots move closer to the trajectory. Once the robot catches the trajectory, λ will be finally zero.

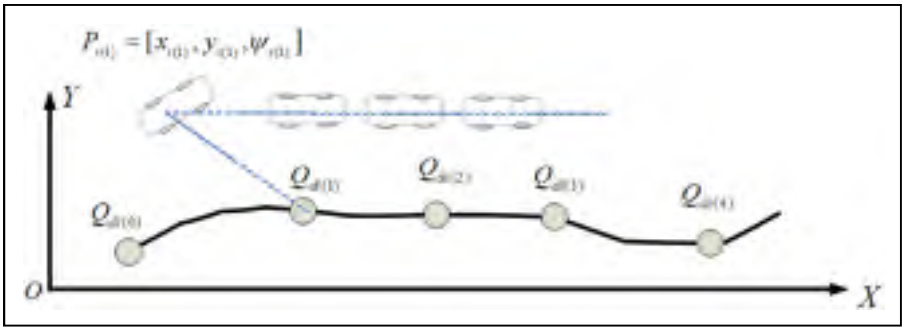


Figure 2.22 Robot is not on the same y-coordination as trajectory.

$$if\ |y_{err}| > 0\ \Rightarrow \alpha(new) = \lambda + \alpha \tag{2.36}$$

The membership functions of this input is shown in the Figure 2.23,

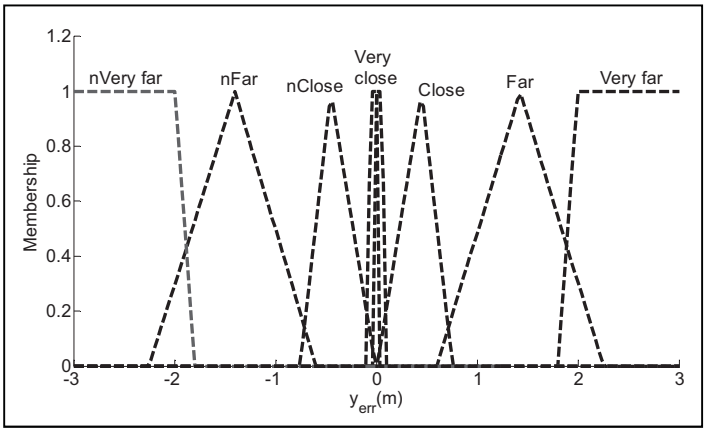


Figure 2.23 Membership function of y_{err}.

Figure 2.24 shows the change of λ related to change of y_{err},

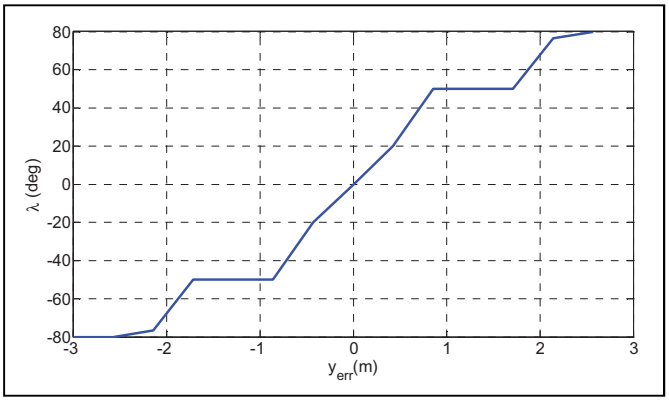


Figure 2.24 λ Obtained by fuzzy controller and y_{err}.

2.3.3 Stability Proof of Trajectory Tracking Algorithm

The available controllers to steer the robot to its trajectory are v, ω. d_{RP} is defined as $d_{RP} = \sqrt{x_{err}^2 + y_{err}^2}$, where the derivation of d_{RP} along (2.29) gives:

$$\dot{d}_{RP} = \frac{-v\, x_{err}}{d_{RP}} + v_d \cos(\alpha + \gamma) \tag{2.37}$$

where $\gamma = -\tan^{-1}\frac{y_{err}}{x_{err}}$. The state error can be written as: $\dot{X} = f(d_{RP}, \alpha, x_{err}, y_{err}, \delta)$ where $\delta = [v, \omega]$.

The idea of the fuzzy controller is to linearize $\dot{X}_i$ about a number of operating points depending on the linguistic rules that are defined. Since in trajectory tracking the robots are required to move tangentially on their desired trajectories with a given velocity, we redefine the desired velocity to be tracked as $v_d = \overline{v}_d \dot{\varsigma}$ where ς is a trajectory parameter characterizing a desired trajectory. Linearizing (2.37) about an operating point using Taylor series would give the Takagi-Sugeno fuzzy model for M^j linguistic rule as (Wang, Tanaka et Griffin, 1996):

$$if\ X\ is\ (X_{eq}, \delta_{eq}) \quad then \quad \dot{\tilde{X}} = A^j \tilde{X} + B^j \tilde{\delta} \tag{2.38}$$

where $\tilde{X} = X_i - X_{eq}$, $\tilde{\delta}_{eq} = \delta - \delta_{eq}$. The matrices A^j and B^j can be found as:

$A^j = \frac{\partial f}{\partial X^j}\big|X_{eq}, \delta_{eq}$, $B^j = \frac{\partial f}{\partial \delta^j}\big|X_{eq}, \delta_{eq}$, this definition gives the following:

$$A^j = \begin{bmatrix} -\bar{v}_d(\bar{\omega}^{eq})' \cos(\alpha^{eq} + \gamma^{eq}) + \bar{v}_d \bar{\omega}^{eq} \sin(\alpha^{eq} + (\gamma^{eq})') & \bar{v}_d \bar{\omega} \sin(\alpha^{eq} + (\gamma^{eq})') \\ 0 & 0 \end{bmatrix} \tag{2.39}$$

$$B^j = \begin{bmatrix} \cos\gamma^{eq} & 0 \\ 0 & 1 \end{bmatrix}$$

where $(\cdot)' = \frac{\partial}{\partial d_{RP}}$. Let μ_j be the membership function of the inferred system set which corresponds to each operating point, and therefore the linearized system would be written as:

$$\dot{\tilde{X}} = \frac{\sum_{j=1}^{r}(\mu_j \tilde{X} + B^j \tilde{\delta}^j)}{\sum_{j=1}^{r} \mu_j} \tag{2.40}$$

where $j = 1, \dots, r$ and r is number of rules. Then for each model, we select a fuzzy state feedback controller of the form $\tilde{\delta}_j = -K^j(X - X_{eq})$.

The structure of the inferred controller is then

$$\tilde{\delta} = \frac{\sum_{j=1}^{r} \mu_j K^j \tilde{X}^j}{\sum_{j=1}^{r} \mu_j} \tag{2.41}$$

The inferred closed loop fuzzy system has the following form

$$\dot{X} = \frac{\sum_{K=1}^{r}\sum_{j=1}^{r} \mu_K \mu_j (A^j - B^j K^j) X}{\sum_{K=1}^{r}\sum_{j=1}^{r} \mu_K \mu_j} \tag{2.42}$$

The matrix K^j is chosen such that the matrix $\left(A^j - B^j K^j\right)$ is Hurwitz. Stability for the equilibrium points of the fuzzy system (2.42) are known and reduce to find an appropriate Lyapunov function such that there exists a common positive matrix P verifying

$(A^j - B^j K^j)^T P + P(A^j - B^j K^j) < 0$, $\forall\, j = 1, \dots, r$ *and*

$Q^{jk^T} P + P Q^{jk^T} < 0^T$ where $Q^{jk} = 0.5(A^j - B^j K^j)^T + (A^k - B^k K^k)$

From this we can show that the equilibrium of the fuzzy system (2.42) is globally asymptotically stable.

2.3.4 Experimental Results

In this section, we discuss the results of different trajectory tracking experiments using an EtsRo mobile robot. These are:

1) where the robot is placed on the trajectory
2) where the robot is not placed on the trajectory

In the first test initial position of robot is $[x(t_0), y(t_0), \psi(t_0)]^T = [0, -1, 0]^T$. Figure 2.25 shows the reference and the actual robot trajectories in the first scenario. The trajectory tracking error y_{err} is shown in Figure 2.26. As can be seen in these figures, the robot travels along its trajectory with negligible errors and the trajectory tracking is experimentally successful.

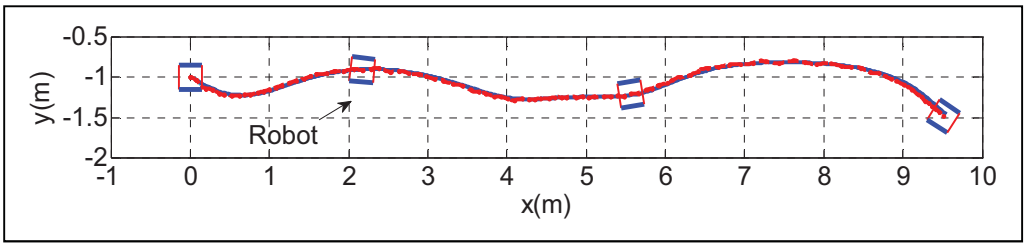


Figure 2.25 Reference and real robot trajectories.

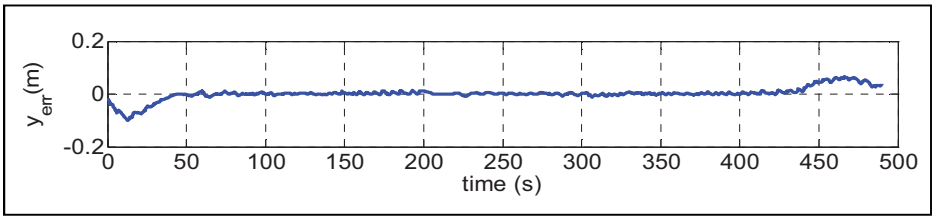


Figure 2.26 Trajectory tracking error y_{err}.

The linear and angular velocities of the robot are plotted in Figure 2.27 and 2.28.

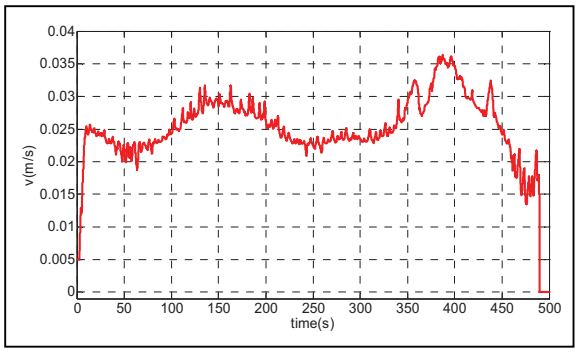


Figure 2.27 Linear velocity of robot.

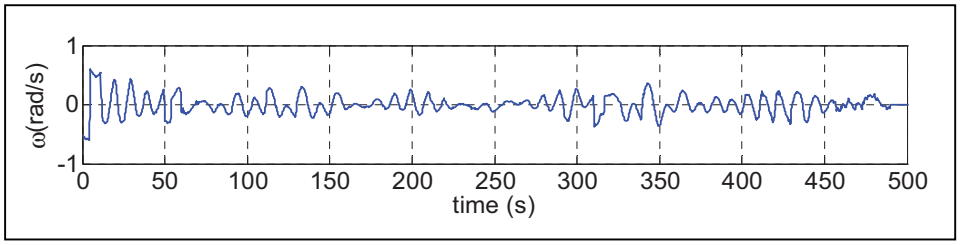


Figure 2.28 Angular velocity of robot.

In the second test initial position of robot is: $[x(t_0), y(t_0), \psi(t_0)]^T = [-1, -2, 0]^T$. Figure 2.29 shows the reference and the actual robot trajectories in the second scenario. The trajectory tracking error y_{err} is shown in Figure 2.30. As can be seen in these figures, the robot travels along its trajectory with negligible errors and the trajectory tracking is experimentally successful.

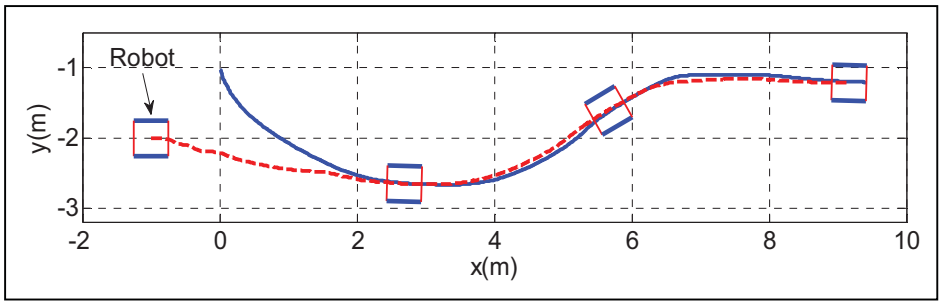


Figure 2.29 Reference and real robot trajectories.

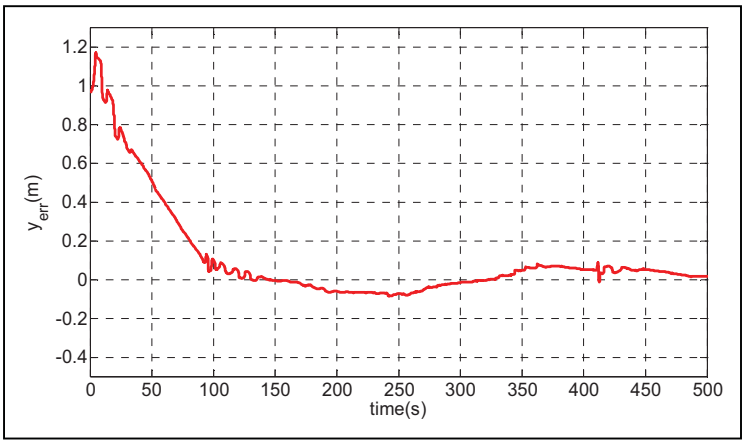


Figure 2.30 Trajectory tracking error y_{err} .

The linear and angular velocities of the robot are plotted in Figure 2.31 and 2.32.

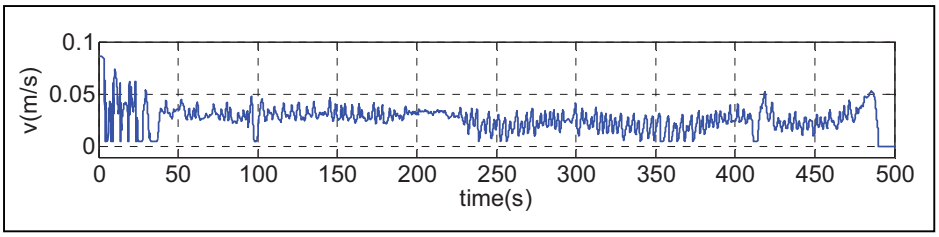


Figure 2.31 Linear velocity of robot.

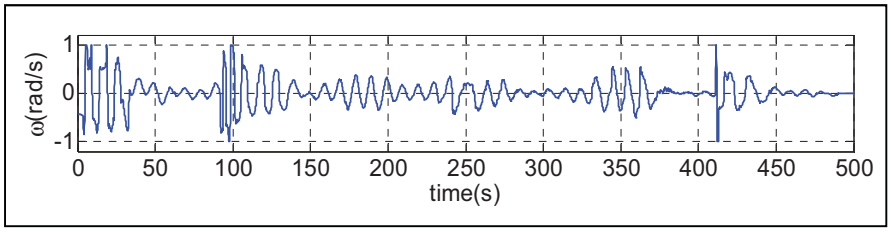


Figure 2.32 Angular velocity of robot.

2.4 Conclusion

In this chapter, in order to improve upon previous trajectory tracking methods enabling a nonholonomic EtsRo mobile robot to reach its desired target point on a trajectory, efficient dynamic tracking controls have been developed based on exponential sliding mode, Lyapunov technique and fuzzy control. A two level controller is designed incorporating a low level PID controller for the right and left motors, and a high level controller to control the speed and movement of the robot. The high level controller uses a feedback controller utilizing the nonlinear functions. The developed exponential sliding mode control reduces chattering on the control input compared to conventional sliding modes, and delivers a high dynamic tracking performance in a steady state mode. The experimental results obtained using an EtsRo mobile robot show the effectiveness of the theoretical outcomes.

CHAPTER 3

COORDINATION CONTROL FOR A NONHOLONOMIC TEAM OF MOBILE ROBOTS USING EXPONENTAL SLIDING MODE

This chapter considers the problem of creating a coordination and trajectory tracking algorithm using exponential sliding mode for the control of a MMR group. This is facilitated by the robots utilizing knowledge derived from onboard sensors that reference their localization in the surrounding environment.

Exponential sliding mode has distinct advantages for use as a coordination and trajectory tracking algorithm, not only due to its robustness but also its inherent ability to reduce the issue of chattering on the control inputs. In this section we investigate the navigation and coordination of a MMR group working in a two-dimensional environment. To do this, a two level architecture control based on the ESM and PID are used for indoor multiple robots achieving different formations. A high level controller coordinates the speed and movement of the robot group by using a feedback controller employing the ESM function. The output of the feedback controller determines the linear and angular velocities of individual robots. Each robot has its own goal or trajectory to be navigated individually and in formation with other robots. By attaching wheeled encoder sensors to each robot in the MMR group, information about the local environment can be input and interpolated to create an efficient trajectory tracking and coordination by the group. Exponential sliding technique is used in conjunction with graph theory, and together they efficiently define the formation controller for the dynamic model of mobile robots. The direction angle that each robot moves along its trajectory is determined not in isolation, but by the desired trajectories and formations of all the robots in the group.

Experimental tests are performed in a laboratory environment in different robot formations to show the ability and efficiency of our controller and coordination algorithm.

3.1 Coordination Algorithm

Figure 3.1 shows the general model of MMR's.

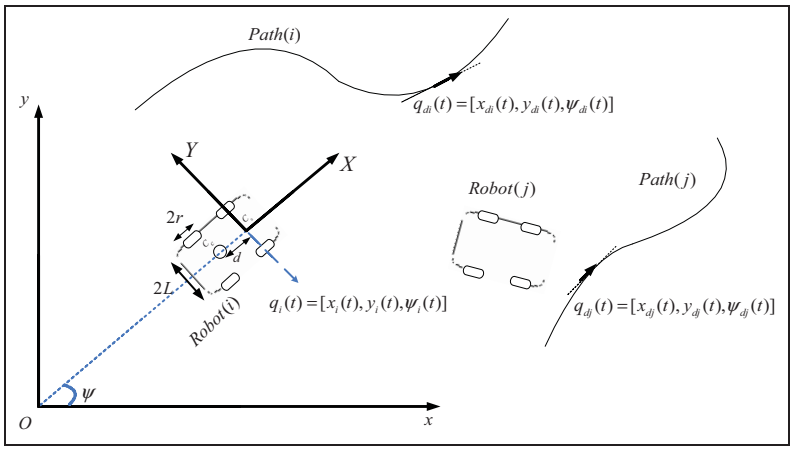


Figure 3.1 Mobile robots and desired trajectories.

Figure 3.2 illustrates the block diagram used to describe the sliding mode control for a group of mobile robots. There are two levels of the controller, being low and high respectively. The higher level controller, which is an exponential sliding mode controller, is designed to follow a generated trajectory.

Posture sensors are used to localize the robot, and a control algorithm uses this information to direct the robot along a trajectory in a desired group formation.

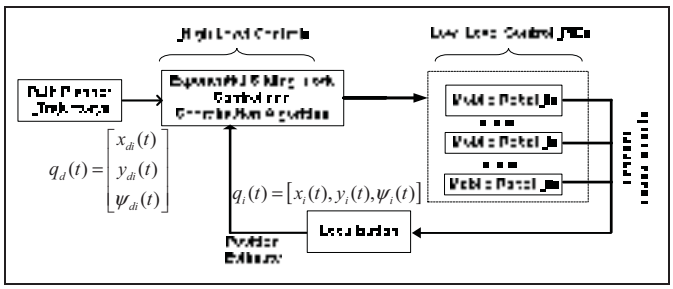


Figure 3.2 Infrastructure of control.

To manage the coordination problem for a group of mobile robots that must move in formation toward their desired target points, the developed strategy for trajectory tracking

that is described in chapter 2 must be modified. For this reason, information exchange between robots can be modeled by graph G. There are two kinds of graphs which can help coordination problem; directed and undirected graphs. A directed graph (digraph) consists of a pairs (N, ϵ), where N is a finite non-empty set of nodes and $\epsilon \in N \times N$ denotes a set of ordered pairs of nodes, called edges. An edge (i, j) in a digraph denotes that robot j can obtain information from robot i, but not necessarily vice versa. In contrast, the pairs of nodes in an undirected graph are unordered, where an edge (i, j) denotes that robots $i \; and \; j$ can obtain information from each other. The following Lemma will be used to solve the coordination problem.

Lemma 3.1 (Godsil et Royle, 2001): Let D and A be degree matrix and adjacency matrix of an undirected graph G , respectively. If $\breve{M}$ is the incidence matrix of G, then

$$L = [l_{ij}] \in \mathbb{R}^{n \times n} = D - A = \breve{M} \, \breve{M}^{\mathrm{T}} \tag{3.1}$$

and matrix L satisfies the following conditions:

$$\begin{cases} l_{ij} \leq 0 \;\; , i \neq j \\ \sum_{j=1}^{n} \iota_{ij} = 0 \; , \; i = 1, \cdots, n \end{cases} \tag{3.2}$$

For an undirected graph, L is called the Laplacian matrix, which is symmetric positive semi-definite.

Assumption 3.1: We assume that G is a connected graph, i.e. a trajectory exists between every two distinct nodes of G. We assume that each mobile robot has to track a given desired trajectory where each trajectory is defined with respect to a virtual smooth trajectory $\Omega_0(t)$. Considering a mass point on $\Omega_0(t)$ which position is denoted by $q_0(t)$ a desired trajectory for vehicle i is defined by:

$$\mathrm{q_{id}} = \Omega_0(t) + l_i \tag{3.3}$$

where $l_i = T(\psi_d)q_0(t)$ and $\psi_d = \tan^{-1}\frac{\dot{y}_d}{\dot{x}_d}$. Figure 3.3 shows formation setup of robots related to virtual trajectory.

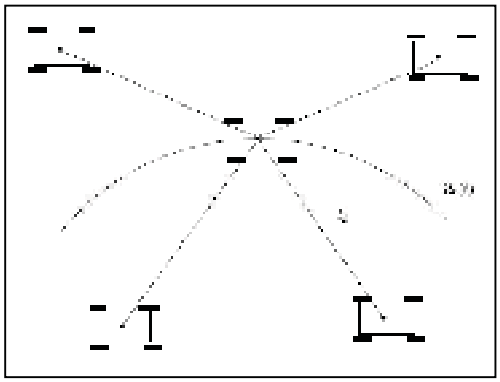


Figure 3.3 Formation setup.

To ensure formation along a desired trajectory, we need information exchange among robots about their positions. Since each robot has to converge to its desired trajectory, the inter-vehicle communication will encompass not only positions but also the error with respect to the desired trajectory. The torque input controller in equation (2.19) will remain the same except that the sliding surface will undergo a slight modification to include the coordination terms. We propose the following sliding surface:

$$S_i = \dot{e}_i + \lambda_d e_i + \gamma_i \tag{3.4}$$

where γ_i is defined as follows:

$$\gamma_i = \varepsilon_i + \lambda_p \int_{t_0}^{t} e_i(\tau)d\tau \quad , \qquad \varepsilon_i = \sum_{i \neq j} K_{ij}(e_i - e_j) \tag{3.5}$$

where K_{ij} is a diagonal matrix which its components are the elements of the Laplacian graph matrix that captures the communication topology between robots.

Theorem 3.1: Let G be a strongly connected graph, that captures the communication topology between robots. Define the sliding surface as in equation (3.4), then applying the

torque input (2.19) to each robot. The trajectory tracking and the coordination problems are resolved which means $\lim_{t\to\infty} e_i = 0$ and $\lim_{t\to\infty} e_i - e_j = 0$, $\forall\, i \neq j$.

Proof: From equation (2.23), it is clear that the sliding surface S is exponentially converging to zero. To analyze convergence to the desired trajectory, we will consider the manifold $\breve{M}$ where $(S, \dot{S}) = (0,0)$. In this manifold using (3.4) we can obtain:

$$\ddot{e}_i = -\lambda_d \dot{e}_i - \lambda_p e_i - \sum_{i\neq j} K_{ij}(\dot{e}_i - \dot{e}_j) \tag{3.6}$$

We will first show that in $\breve{M}$ the errors $(e_i, \dot{e}_i) = (0,0)$. Equation (3.6) in a matrix form considering the following vector errors $e = [e_1^{\mathrm{T}}, \dots, e_n^{\mathrm{T}}]^{\mathrm{T}}$, the matrices $\bar{\lambda}_p = diag(\lambda_p, \lambda_p, \dots, \lambda_p), \bar{\lambda}_d = diag(\lambda_d, \lambda_d, \dots, \lambda_d)$, and $\bar{L} = I \otimes L$ where $\otimes$ is the Kronicker product and can be rewritten as:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\bar{\lambda}_p & -(\bar{\lambda}_d + \bar{L}) \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{3.7}$$

Since the graph G is strongly connected, then $\bar{L}$ is a symmetric nonnegative definite matrix, which has one 0-eigenvalue and all the rest are strictly positive. Since we have the freedom to choose the matrix $\bar{\lambda}_d$ such that it is dominant over the eigenvalues of $\bar{L}$ then the matrix $\begin{bmatrix} 0 & I \\ -\bar{\lambda}_p & -(\bar{\lambda}_d + \bar{L}) \end{bmatrix}$ is stable. Consequently, system (3.6) is exponentially stable in $\breve{M}$, and therefore $(e_i, \dot{e}_i) = (0,0)$ in M. It is clear then, on the manifold M, the trajectory tracking is solved since $e_i = 0$. To show that formation of the robots is maintained on the manifold $\breve{M}$, we consider the following equation that holds on M

$$\dot{e}_i = -\lambda_d e_i - \lambda_p \int_{t_0}^{t} e_i(\tau) d\tau - \sum_{i\neq j} K_{ij}(e_i - e_j) \tag{3.8}$$

In a matrix form (3.8) rewrites as:

$$\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\bar{\lambda}_p & -(\bar{\lambda}_d + \bar{L}) \end{bmatrix} \begin{bmatrix} \int_{t_0}^{t} e_i(\tau)d\tau \\ e \end{bmatrix} \tag{3.9}$$

It is easy to check that the matrix $\begin{bmatrix} 0 & I \\ -\bar{\lambda}_p & -(\bar{\lambda}_d + \bar{L}) \end{bmatrix}$ is stable, and consequently $\left(\int_{t_0}^{t} e(\tau)d\tau, e\right)$ converges asymptotically to zero on M. It is obvious then from equation (3.8) that the term $\sum_{i \neq j} K_{ij}(e_i - e_j)$ converges to zero as time goes to infinity. This ends the proof.

3.2 Experimental Results

In this section, we discuss the results of experimental tests performed on the trajectory tracking missions involving a group of EtsRo mobile robots which illustrate the performance of the proposed exponential sliding mode trajectory tracking and coordination algorithm.

3.2.1 Experimental Setup

Figure 3.4 shows the structural design of the control, trajectory planning and coordination for the group of mobile robots being used in the experimental tests.

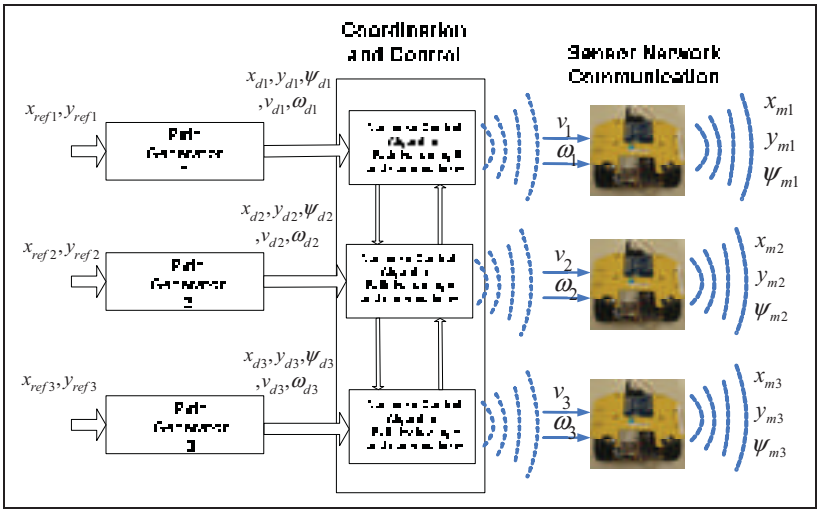


Figure 3.4 Structural design for MMR coordination.

3.2.2 Experimental Tests

In this test, trajectories which have different lengths are considered. In this scenario, the initial positions and the length of the trajectories are defined as:

$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [1,3,0]^T, [x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [0,0,0]^T, [x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [-1,-2,0]^T$
$L_2 = 4.90\ m\ , L_1 = L_3 = 10.41\ m$

Figure 3.5 displays the reference trajectories and the actual robot trajectories in first scenario. The trajectory tracking errors y_{err} and ψ_{err} are shown in Figure 3.6. As can be seen in these figures, the robots travel along their trajectories, and as soon as robots reach their desired trajectories, trajectory tracking errors reduce to zero and the formation is experimentally successful.

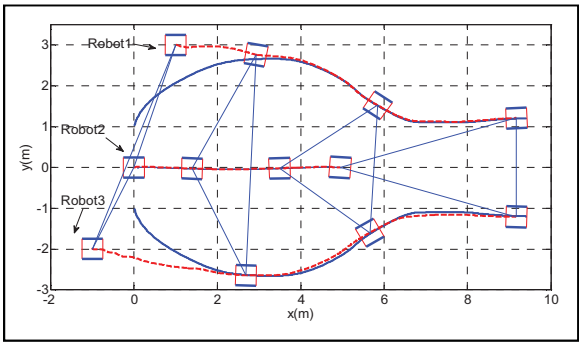
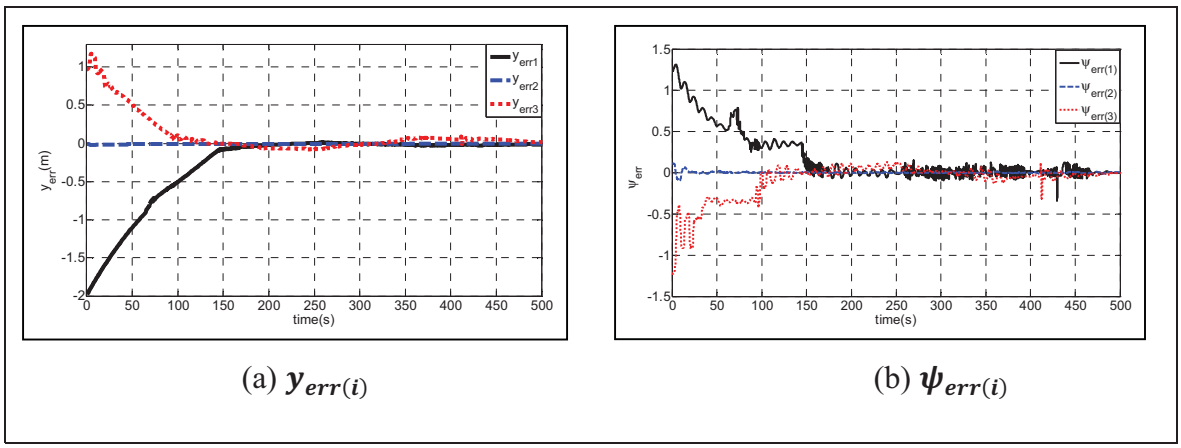


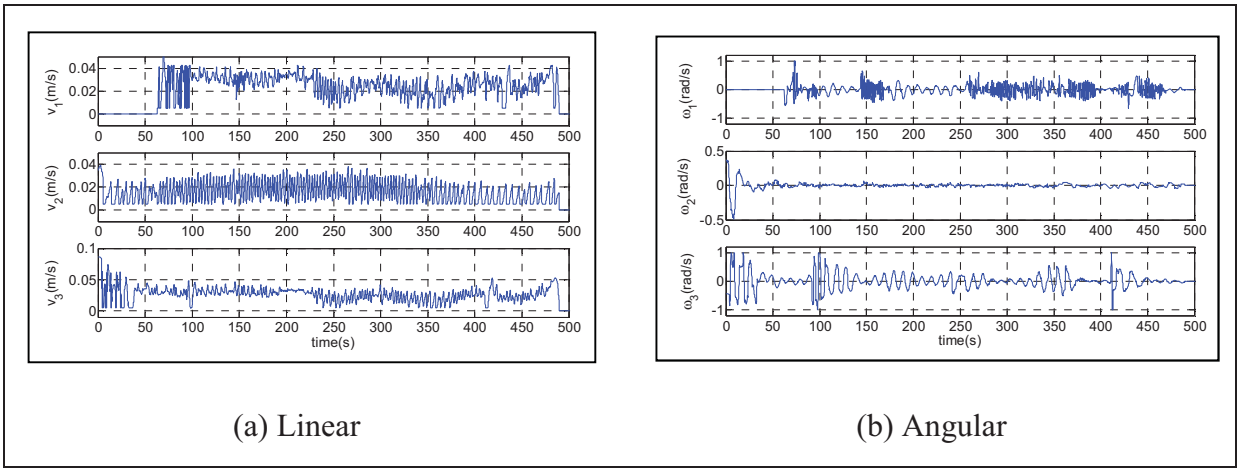
Figure 3.5 Reference and real robots' trajectories.



(a) $\boldsymbol{y_{err(i)}}$ (b) $\boldsymbol{\psi_{err(i)}}$

Figure 3.6 Trajectory tracking errors.

The linear and angular velocities are plotted in Figure 3.7. These figures show that the robots travel with different velocities related to the length of the trajectory travelled and their initial positions. We observe that at the start robot 3 has the highest velocity and robot 1 has a velocity of zero. Robot 2 has the lowest velocity as its length is the shortest.



(a) Linear (b) Angular

Figure 3.7 Velocity of robots.

3.3 Conclusion

In this chapter, a control algorithm and efficient coordination architecture is proposed for a group of mobile robots enabling them to work both individually and in meaningful robot formations. The developed technique combines exponential sliding mode with graph theory to create an efficient coordination algorithm. A designed two level controller incorporates a low level PID controller for the right and left motors, and a high level controller to coordinate the speed and movement of the robot group. The high level controller utilizes a feedback controller performing the exponential sliding mode function to give a consistent high dynamic tracking performance to each robot in the group. The experimental results performed on a multi-robot platform demonstrate the effectiveness of the theoretical result.

CHAPTER 4

NONLINEAR COORDINATION CONTROL USING A VIRTUAL STRUCTURE

In this chapter we move forward from our previous work to create an efficient method to enable a team of mobile robots with inbuilt localization sensors to travel in group formation along desired trajectories and simultaneously reach their individual target positions. The work in chapter 2 considered Lyapunov technique to provide an efficient trajectory tracking technique for a single robot utilizing sensor localization. Chapter 3 then explored an efficient coordination architecture employing exponential sliding mode for the control algorithm architecture therefore enabling a group of robots to move in formation to their desired target positions.

In this chapter we explore the robustness of the system against noise and loss of signal to the individual robots, while looking at ways to improve the velocity of the robots in the MMR group to speedily reach their target locations. We now review the success of the algorithm controller described in chapter 2 as being a suitable model for creating a more efficient feedback controller. This controller must now be modified to work with multiple robots and utilizes the Lyapunov function in a two level architecture employing nonlinear and PID controllers which enable the robots travel in formation along predefined trajectories. The high level feedback controller uses the Lyapunov function to coordinate the speed and movement of the robot group and the low level PID controller adjusts the speed of the left and right front wheel motors.

This design creates an elegant way of employing a single system to convert the formation control problem into a stabilization problem, and therefore bypasses the difficulty when using constrained motion control involving multiple systems. However, for this design to be functional, the issues related to the control of trajectory tracking need to be individually solved for each mobile robot by allowing them access to local measurements of the environment including the positions of the other robots. However, as each robot in the group

has access to their individual localization data from their posture sensors, this information can shared amongst all the robots allowing for a more efficient individual trajectory tracking and coordination.

By introducing the concept of a virtual target that moves progressively along a desired trajectory, new control laws have been designed and implemented to drive the mobile robots. By exchanging information among robots, they use these control laws to track the moving virtual target while simultaneously adjusting their speed profiles on their individual trajectories.

In this chapter we perform experimental tests in a laboratory environment using different robot formations to verify the ability of our designed Lyapunov controller and coordination algorithm to operate more efficiently. We also test the robustness of the system under various types of signal loss or failure. If there is a temporary loss of coordination due to short communication failures or sensor noise on the position measurement, then individual robots may lose group formation and divert from their designated trajectories. The results show that our algorithm controller will allow for this temporary loss of robot coordination, and that once the communication signal is restored and their coordination is resumed, the robots continue with the group formation on their predefined trajectories.

4.1 The Lyapunov Control and Coordination Algorithm

Figure 4.1 illustrates the block diagram used to control and coordinate a group of mobile robots. The low level controller is designed and implemented to adjust the right and left wheel velocities, whereas the higher level controller, which is a non-linear controller, is designed to follow a generated trajectory and coordinate the robots. Posture sensors are used to localize the robots, and a coordination algorithm uses this information to direct the robot group along trajectories in a desired formation.

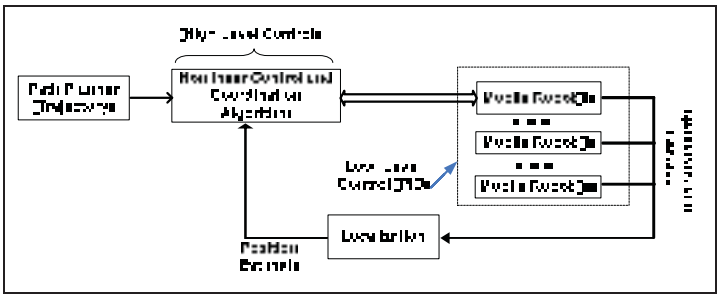


Figure 4.1 Infrastructure of multi robot coordination.

For our purposes, we consider a fleet of $n \geq 2$ mobile robots. To obtain coordination between robots, each trajectory is parameterized in term of parameter s_i. Robots keep coordination if $s_i = s_j$ for all i, j. This parameter is defined as $s_i = \frac{\zeta_i}{L_i}$, where ζ_i is signed curvilinear abscissa and L_i is the length of trajectories. For $i = 1, \ldots, n$ we let $p_i(t) = [x_i, y_i, \psi_i]^T$ and $p_{di}(s_i) = [x_{di}, y_{di}, \psi_{di}]^T$ denote the position of robot i and its assigned (desired) trajectory parameterized in terms of a generalized variable $s_i \in \mathbb{R}$, We further let $v_{di}(s_i), \omega_{di}(s_i) \in \mathbb{R}$ denote the desired linear and angular velocities assignment for robot i . We decompose the motion-control problem into two loops. The first loop is an inner-loop kinematic task, which consists of making the robot's velocity $u_i = [v_i, \omega_i]^T$ track a desired velocity reference $u_{di} = [v_{di}, \omega_{di}]^T$, where v_{di} and ω_{di} are given in terms of the generalized variable s_i time derivative as (Ghommam *et al.*, 2010):

$$v_{di} = \sqrt{\dot{x}_{di}(s_i)^2 + \dot{y}_{di}(s_i)^2}\, \dot{s}_i = \bar{v}_{di}\dot{s}_i \tag{4.1}$$

$$\omega_{di} = \frac{\dot{x}_{di}(s_i)\, \ddot{y}_{di}(s_i) - \ddot{x}_{di}(s_i)\dot{y}_{di}(s_i)}{\dot{x}_{di}(s_i)^2 + \dot{y}_{di}(s_i)^2}\dot{s}_i = \bar{\omega}_{di}\dot{s}_i \tag{4.2}$$

where x_{di}, y_{di} denote the desired position for robot i. The second loop is an outer-loop dynamic task, which assigns the reference velocity to the robots so as to achieve a convergence to the trajectory.

4.2 Coordination Problem

Consider a group of mobile robots each with a local controller for trajectory tracking. To achieve coordination between the elements of the group, a common velocity profile v_L has to be assigned to all the trajectories, so that the robots move along their trajectories while holding a desired inter-robot formation pattern. The update for s_i can be seen as a coordination state such that coordination exists between two mobile robots i and j if and only if $s_i(t) - s_j(t) = 0$. The key idea in designing the coordination controller is to introduce a control variable in form of a correction term $\tilde{v}_{di}(t)$ added to the defined common velocity profile, to obtain reference velocities for individual robots that is:

$$v_{di}(t) = v_L + \tilde{v}_{di}(t) \tag{4.3}$$

As the robots always move forward on their desired trajectories, then $v_L > 0$. To solve the coordination problem, a trajectory which is called Ω_0, is needed with trajectory parameter of s_0. We suppose that there is a virtual leader with no dynamic equation that moves along this trajectory with a desired velocity $\dot{s}_0 = v_L$. We require all the robots to synchronize with this virtual leader, which means $s_i, i = 1, \dots, N$ equal to s_0. Therefore, a control law for the correction velocity $\tilde{v}_{di}$ is derived such that for all $i, j \in I$ the coordination error $\lim_{t\to\infty}|s_i - s_0| \to 0$ and the formation velocity error $\lim_{t\to\infty}|\dot{s}_i - v_L| \to 0$.

The time derivative of the posture errors (2.28), using the relations in (4.1), (4.2) and taking into account the nonholonomic constraints

$$\begin{cases} \dot{x}_{err(i)} = v_i - \bar{v}_{di}\dot{S}_i \cos(\psi_{err(i)}) + y_{err(i)}\omega_i \\ \dot{y}_{err(i)} = \bar{v}_{di}\dot{S}_i \sin(\psi_{err(i)}) - x_{err(i)}\omega_i \\ \dot{\psi}_{err(i)} = \bar{\omega}_{di}\dot{s}_i - \omega_i \end{cases} \tag{4.4}$$

The velocity error is defined as $\dot{\tilde{S}}_i$ where

$$\dot{\tilde{S}}_i = \dot{s}_i - v_l \tag{4.5}$$

The kinematic error (4.4) rewrites by substituting v_{di} and ω_{di} by their expressions in (4.1), (4.2) as

$$\begin{cases} \dot{x}_{err(i)} = -v_i + \bar{v}_{di} v_l \cos(\psi_{err(i)}) + y_{err(i)} \omega_i + \dot{\bar{S}}_i \mu_{1i} \\ \dot{y}_{err(i)} = \bar{v}_{di} v_l \sin(\psi_{err(i)}) - x_{err(i)} \omega_i + \dot{\bar{S}}_i \mu_{2i} \\ \dot{\psi}_{err(i)} = \bar{\omega}_{di} \dot{\bar{S}}_i - \omega_i + \bar{\omega}_{di} v_l \end{cases} \tag{4.6}$$

where $\mu_{1i} = \bar{v}_{di} \cos(\psi_{err(i)})$ and $\mu_{2i} = \bar{v}_{di} \sin(\psi_{err(i)})$. By substituting (2.30) and (2.31) in the error of (4.6), we get

$$\begin{bmatrix} \dot{x}_{err(i)} \\ \dot{y}_{err(i)} \\ \dot{\psi}_{err(i)} \end{bmatrix} = \begin{bmatrix} -K_{xi} x_{err(i)} + y_{err(i)} \left(\bar{\omega}_{di} v_l + \bar{v}_{di} v_l (K_{yi} y_{err(i)} + K_{\psi i} \sin(\psi_{err(i)})) \right) + \dot{\bar{S}}_i \mu_{1i} \\ \bar{v}_{di} v_l \sin(\psi_{err(i)}) - x_{err(i)} \left(\bar{\omega}_{di} v_l + \bar{v}_{di} v_l (K_y y_{err(i)} + K_{\psi i} \sin(\psi_{err(i)})) \right) + \dot{\bar{S}}_i \mu_{2i} \\ -\bar{v}_{di} v_l \left(K_y y_{err(i)} + K_{\psi i} \sin(\psi_{err(i)}) \right) + \bar{\omega}_{di} \dot{\bar{S}}_i \end{bmatrix} \tag{4.7}$$

Let the candidate Lyapunov function be defined as

$$V_i = \frac{1}{2} \left(x_{err(i)}^2 + y_{err(i)}^2 \right) + \frac{1 - \cos(\psi_{err(i)})}{K_{yi}} \tag{4.8}$$

Differentiating the relation (4.8) along the solutions of (4.7), we get

$$\dot{V}_i = -K_{xi} x_{err(i)}^2 - \frac{K_{\psi i}}{K_{yi}} \bar{v}_{di} v_l \sin^2(\psi_{err(i)}) + \Lambda_i \dot{\bar{S}}_i \tag{4.9}$$

where $\Lambda_i = x_{err(i)} \mu_{1i} + y_{err(i)} \mu_{2i} + \frac{\sin(\psi_{err(i)})}{K_{yi}} \bar{\omega}_{di}$.

4.3 Coordination Solution

The coordination problem is solved as follows:

Considering the real situation, where $\dot{\bar{S}}_i$ is not zero. We need to find a coordinating controller that assures the synchronization of the trajectory parameters with the virtual trajectory parameter s_0. Let the coordinating controller be defined like (Hong, Hu et Ao, 2006)

$$\dot{S}_i = v_L - \kappa \left[\sum_{i \neq j} \iota_{ij} (s_i - s_j) + b_i (s_i - s_0) \right] + \eta_i \tag{4.10}$$

$$\dot{\eta}_i = -\gamma\kappa\left[\sum_{i\neq j} \iota_{ij}(s_i - s_j) + b_i(s_i - s_0)\right] \tag{4.11}$$

where b_i denotes the non-zero connection between robot i with the center of the virtual structure. We suppose that the center of the virtual structure moves along the desired trajectory Ω_0 with an assigned dynamic $\dot{s}_0 = v_L$. Define $\tilde{s} = [s_1, s_2, \ldots, s_n]^T - s_0 1$ and $\eta = [\eta_1, \eta_2, \ldots, \eta_n]^T$. Equations (4.10) and (4.11) in vector form can be written as

$$\dot{s} = v_L 1 - \kappa(L + B)s + \kappa B s_0 1 + \eta \tag{4.12}$$

$$\dot{\eta} = -\gamma\kappa(L + B)s + \gamma\kappa B s_0 1 \tag{4.13}$$

where $1 = [1]_{n\times 1}$ and B is the diagonal matrix whose i^{th} diagonal element is b_i. Using the fact that $L1 = 0$, equations (4.12) and (4.13) can be written in linear matrix form as:

$$\begin{bmatrix}\dot{\tilde{s}}\\ \dot{\eta}\end{bmatrix} = \begin{bmatrix}-\kappa(L+B) & I\\ -\gamma(L+B) & 0\end{bmatrix}\begin{bmatrix}\tilde{s}\\ \eta\end{bmatrix} \tag{4.14}$$

Theorem 4.1: Assume that the graph that captures the communication topology between the n mobile robots is connected and then there exists constants k and γ such that the coordinated controller (4.10) and (4.11) yields the exponential convergence of $\lim_{t\to\infty}|s_i - s_0| \to 0$, $\lim_{t\to\infty}|\dot{s}_i - v_L| \to 0$.

Proof: To show that the state $x = (\tilde{s}, \eta)^T$ of the linear system (4.14) is exponentially stable, a Lyapunov function $V = X^T P x$ for this system is constructed where the matrix P is a positive definite matrix and can be chosen as in (Hong, Hu et Ao, 2006) as follows

$$P = \begin{bmatrix} I & -\gamma I\\ -\gamma I & I\end{bmatrix} \tag{4.15}$$

The time derivative of the Lyapunov function V along the solutions of (4.12) and (4.13) is

$$\dot{V} = -x^T\begin{bmatrix}2\kappa(1-\gamma^2)\mathrm{H} & -I\\ -I & 2\gamma I\end{bmatrix}x \tag{4.16}$$

where $H = L + B$. By Lemma 3 of (Hong, Hu et Ao, 2006), the matrix $L + B$ is positive definite as we assume that the graph that captures the communication topology is connected,

then for the matrix Q to be positive definite we select $k > \frac{1}{4\gamma(1-\gamma^2)\lambda_{max}}$ with λ_{max} is the maximum eigenvalue of H. It follows, that there is a constant β such that $\dot{V}(x) \leq -2\beta V(x)$ or equivalently $v(x) \leq v\big(x(0)\big)e^{-2\beta(t-t_0)}$ which implies that $x(t) \leq \alpha x(t_0)e^{-\beta(t-t_0)}$ and finally $\lim_{t\to\infty}|s_i - s_0| \to 0$ and $\lim_{t\to\infty}|\dot{s}_i - v_L| \to 0$.

4.4 Interconnection of the Trajectory Tracking Subsystem and the Coordination Subsystem

Now the objectives that we originally defined are satisfied, if we consider that the trajectory tracking and the coordination are two separated subsystems that have to be dealt with independently. However this is not the case, as it can be seen in equations (4.7) and (4.9). We can regard equation (4.7) as a nonlinear system that can be written as

$$\dot{\xi}_i = f(\xi_i) + g(\xi_i, \dot{\tilde{S}}_i) \tag{4.17}$$

with the features that the nominal system $\dot{\xi}_i = f(\xi_i)$ and its solutions are asymptotically stable and the perturbing term $g\left(\xi_i, \dot{\tilde{S}}_i\right)$ linearly dependent of $\dot{\tilde{S}}_i$ which exponentially vanishes to zero. We will need a lemma to prove that the overall system converges to zero as time goes to infinity.

Lemma 4.1: Suppose that there exists a Lyapunov function for the nominal system $\dot{\xi}_i = f(\xi_i)$, satisfying the following conditions

Condition 1: $c_1\|\xi_i\|^2 \leq V_i \leq c_2\|\xi_i\|^2$

Condition 2: $\left\|\frac{\partial V_i}{\partial \xi_i}\right\| \leq c_3\|\xi_i\|$

Condition 3: $\left\|\frac{\partial V_i}{\partial \xi_i} f(\xi_i)\right\| \leq -c_4\|\xi_i\|^2$

and the perturbing term satisfies the exponential growth condition

Condition 4: $\left\|g(\xi_i, \dot{\tilde{S}}_i))\right\| \leq c_5\|\xi_i\|e^{-\beta(t-t_0)}$

Then the solution $\xi(t)$ asymptotically converges to zero as time goes to infinity.

Proof: The time derivative of V along the solutions of (4.16) satisfies

$$\dot{V}_i = \frac{\partial \dot{V}_i}{\partial \xi_i} f(\xi_i) + \frac{\partial \dot{V}_i}{\partial \xi_i} g\left(\xi_i, \dot{\tilde{S}}_i\right) \leq -c_4 V_i + c_5 \frac{c_3}{c_1} V_i e^{-\beta(t-t_0)} \leq -\left(c_4 - c_5 \frac{c_3}{c_1} e^{-\beta(t-t_0)}\right) V_i \tag{4.18}$$

Consider the differential equation:

$$\dot{\delta} = -\left(c_4 - c_5 \frac{c_3}{c_1} e^{-\beta(t-t_0)}\right) \delta \tag{4.19}$$

Solution of (4.19) can be given as:

$$\delta(t) = \delta(t_0) e^{-c_4(t-t_0)} e^{c_5 \frac{c3}{c_1 \beta}\left(1-e^{-\beta(t-t_0)}\right)} \tag{4.20}$$

Now apply the comparison lemma yields

$$V_i(t) \leq V_i(t_o) e^{-c_4(t-t_0)} e^{c_5 \frac{c3}{c_1 \beta}\left(1-e^{-\beta(t-t_0)}\right)} \tag{4.21}$$

This implies that there exists a positive constants c_6 and c_7 such that

$$\xi(t) \leq c_6 \xi(t_0) e^{-c_7(t-t_0)} \tag{4.22}$$

which completes the proof.

All we need to do is to verify conditions 1-4 to our Lyapunov function given in equation (4.8) in the section. For condition 1, using the fact that $|1 - \cos\psi_i| \leq |\psi_i|$ and the fact that we restrict ψ_i to vary in the range of $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ we can easily find that $min\left\{0.5, \frac{1}{K_{yi}}\right\} \|\xi_i\|^2 \leq V_i \leq max\left\{0.5, \frac{1}{K_{yi}}\right\} \|\xi_i\|^2$. For condition 2 we again use the property that $|\sin\psi_i| \leq |\psi_i|$, then $\left\|\frac{\partial V_i}{\partial \xi_i}\right\| \leq \sqrt{2 + \frac{1}{K_{yi}^2} \|\xi_i\|}$. Condition 3 is easily verified for $c_4 = max\left\{1, K_{xi}, \frac{K_{\psi i}}{K_{yi}}\right\}$. For condition 4, the term $g\left(\xi_i, \dot{\tilde{S}}_i\right) = \Lambda \dot{\tilde{S}}_i$, since $\dot{\tilde{S}}_i$ is exponentially convergent this implies that $\left\|g\left(\xi_i, \dot{\tilde{S}}_i\right)\right\| \leq max\left\{\bar{v}_{di}, \frac{\bar{\omega}_{di}}{K_{yi}}\right\} \|\xi_i\| e^{-\beta(t-t_0)}$. All conditions are satisfied.

4.5 Experimental Tests

To illustrate the performance of the proposed coordination and control scheme, some tests are performed using three mobile robots. The purpose of the experiments is to demonstrate the stability and the performance characteristics inferred from the theoretical development. Figure 4.3 shows the EtsRos used in the experimental tests.

Figure 4.2 EtsRos.

If we assume robot 1 is allowed to communicate with robot 2 and 3 according to the graph of Figure 4.2, then we can find Laplacian matrix as follow, $L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$

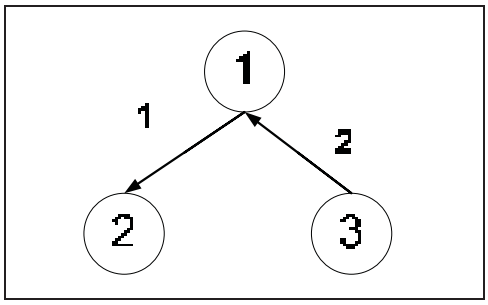


Figure 4.3 Robots communication topology.

In this section, we discuss the results of symmetric and asymmetric coordinated trajectory tracking missions involving three EtsRo mobile robots. In the first test, circle trajectories which are concentric and have different radius are considered. In this scenario, robots are placed on a common vertical line with the initial position of:

$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [0, .5, 0]^T, [x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [0,1,0]^T, [x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [0,1.5,0]^T$

and non-linear control gains are considered as: $K_{xi} = 1$; $K_{yi} = 1$; $K_{\psi ei} = 1$.

Figure 4.4 presents the reference and the actual robots trajectories in first scenario. The trajectory tracking errors $x_{err(i)}$ and $y_{err(i)}$ are shown in Figure 4.5. These figures show that the robots travel along their individual trajectories with different velocities to arrive together at the target point and the formation is therefore experimentally successful.

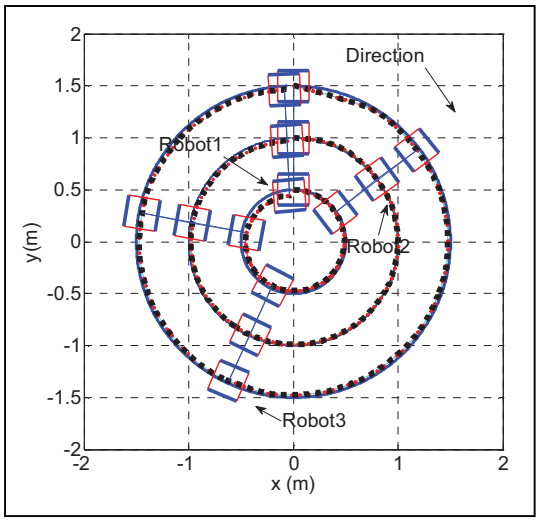
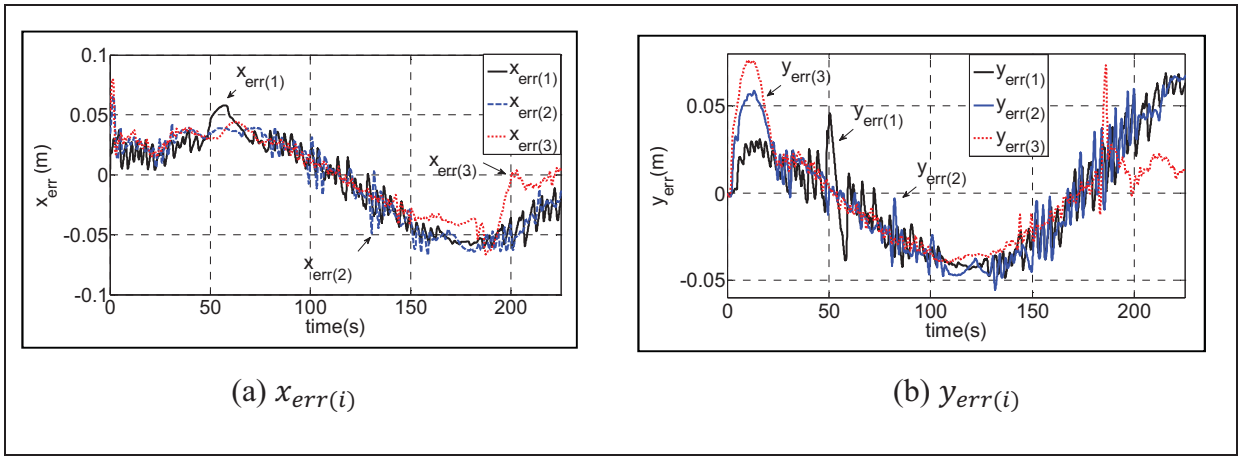


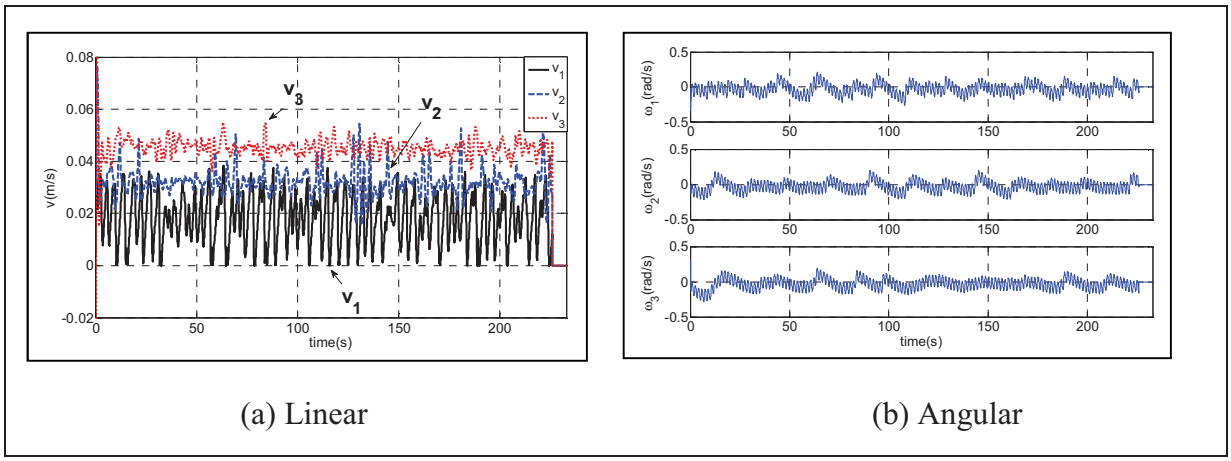
Figure 4.4 References and real robots' trajectories.



(a) $x_{err(i)}$ (b) $y_{err(i)}$

Figure 4.5 Trajectory tracking errors.

The linear and angular velocities are plotted in Figure 4.6. As can be seen, the third robot moves the fastest because of the largest circumference trajectory while the first robot has the slowest velocity due to the smallest circumference trajectory. The robots will adjust their speed along their individual trajectories to achieve an overall group coordination. Figure 4.7 represents coordination state error. These errors reduce to zero, as is expected by the theory explained for all the robots.



(a) Linear (b) Angular

Figure 4.6 Velocity of the robots.

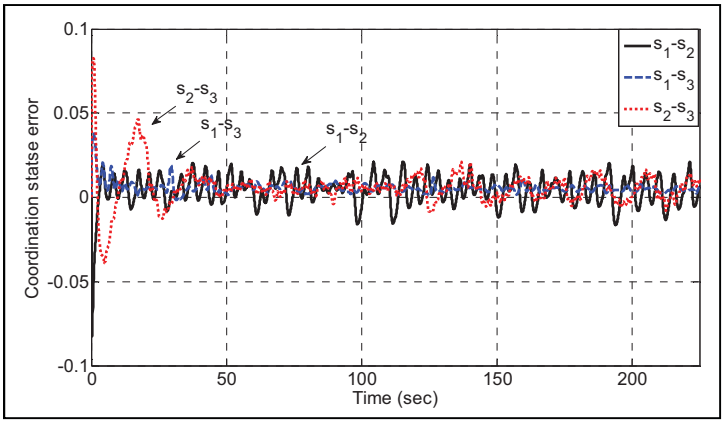


Figure 4.7 Coordination state error $s_i - s_j$.

In the second test, the robots are placed on a common vertical line, but the trajectories are asymmetric and interconnect in three different points and have different lengths. In this scenario the initial positions of robots are considered as:

$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [0,2,0]^T, [x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [0,1,0]^T, [x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [0,-2,0]^T$

Figure 4.8 presents the reference and the actual robots trajectories in second scenario. The trajectory tracking errors $x_{err(i)}$ and $y_{err(i)}$ are shown in Figure 4.9. These figures show that the robots travel along their individual trajectories and the formation is experimentally successful.

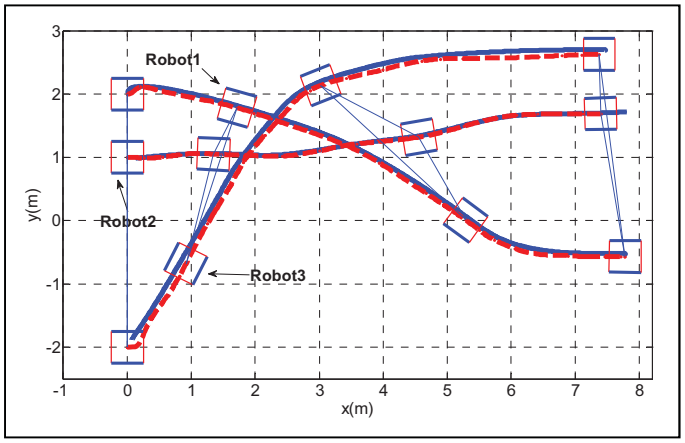
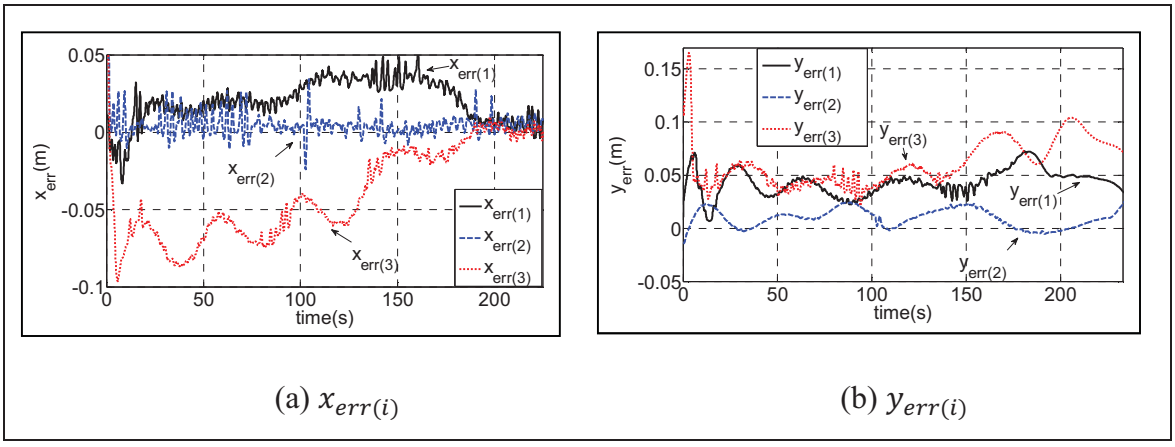


Figure 4.8 References and real robots' trajectories.

The linear and angular velocities are plotted in Figure 4.10. As can be seen in these figures, the robots have different shapes of velocities because they move on different trajectories and so they will adjust their velocities to reach the target point in the same time duration.



(a) $x_{err(i)}$ (b) $y_{err(i)}$

Figure 4.9 Trajectory tracking errors.

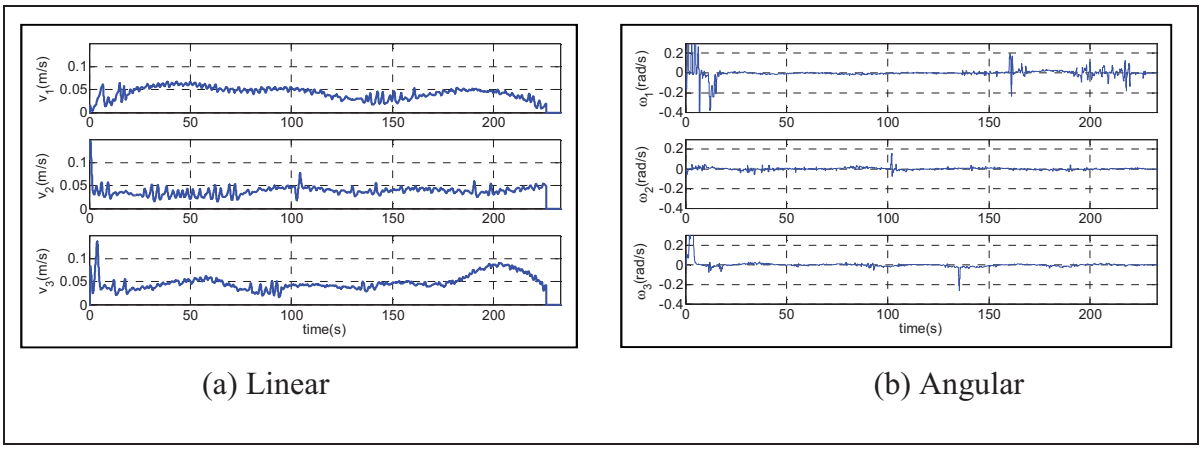


Figure 4.10 Velocities of robots.

Figure 4.11 depicts the coordination state error which reduces to zero as predicted by the theory.

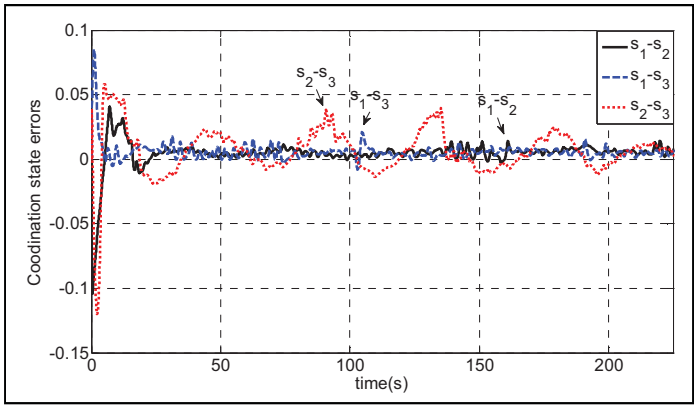


Figure 4.11 Coordination state error $s_i - s_j$.

In the last scenario, tests are devised to deliberately add noise into the measuring position and communication system to create either delays or failures in the signal. In this test, the trajectories have different lengths and the initial positions of the robots are:

$$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [0,1,0]^T, [x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [0,0,0]^T, [x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [0,-1,0]^T$$

In this scenario, a white noise with disturbance of ten percent is intentionally added to the position measurement for robot 1, a communication failure is implied for robot 2 and lastly for robot 3 we consider a communication delay. Figure 4.12 shows the references trajectories

and the real trajectories of the robots and the results of the errors that we impose upon the system. These are a 30 sec position measurement for robot 1, and errors of a 10 sec communication failure for robot 2, and a 10 sec delay for robot 3. This figure shows that regardless of these short time communication problems or noise on position measurement, the robots still keep coordination and follow their desired trajectories.

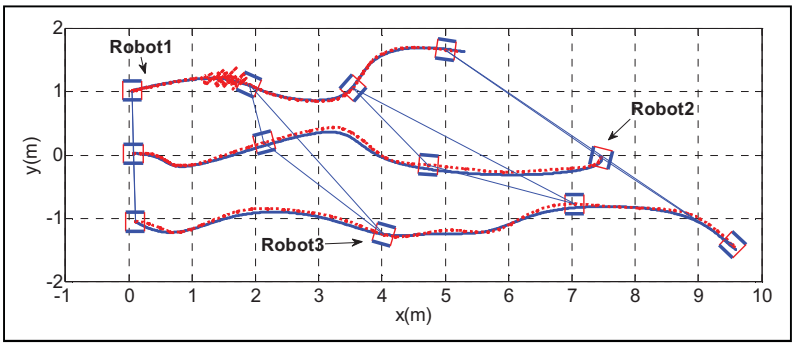
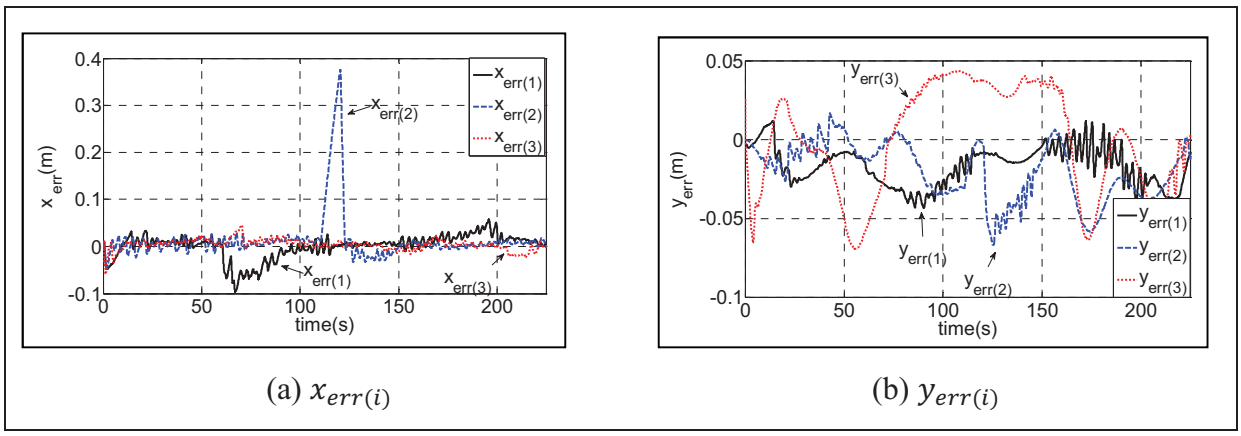


Figure 4.12 References and real robots' trajectories with position measurement noise and temporary communication failures or delays.

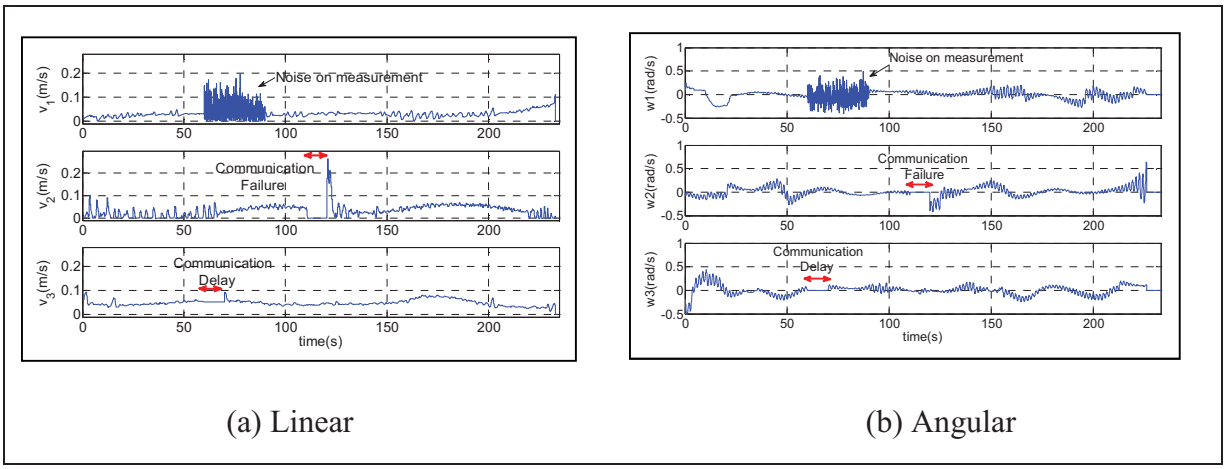
The trajectory tracking errors $x_{err(i)}$ and $y_{err(i)}$ are shown in Figure 4.13. As can be seen in Figure 4.12(a), when robot 2 has a failure in communication there is a visible error on $x_{err(i)}$ and it loses its desired position. When the communication resumes, the coordination algorithm forces robot 2 to follow its desired trajectory and the trajectory following error reduces to zero.



(a) $x_{err(i)}$ (b) $y_{err(i)}$

Figure 4.13 Trajectory tracking errors.

The linear and angular velocities are plotted in Figure 4.14. As can be seen in these figures, when there is a communication failure, the linear and angular velocities of the robot reduces to zero and the robot will stop moving. However, when there is a communication delay, the robot will continue with the same angular and linear velocities as it had before the delay. Therefore a communication failure will stop a robot from moving, but if there is only a communication delay, the robot will still continue to move until the communication is restored. As well, these figures show that the presence of noise on the position measurement of robot 1 also creates noise on the linear and angular velocities of the robot.



(a) Linear (b) Angular

Figure 4.14 Velocities of the robots.

Figure 4.15 depicts the coordination state error. This figure shows that even with the existence of a temporary loss of coordination (up to 15 sec) due to short communication failures or delays, the robots can still maintain their formation. The results indicate that the coordination algorithm allows for this temporary loss of coordination, and once the communication signal is restored and coordination is re-established, the robots will continue with the group formation on their predefined trajectories.

However, if the loss of coordination is of significant time duration (more than 15 sec), then the robots lose group formation and will divert from their designated trajectories. Figure 4.16

shows this example with an extended communication failure for robots 2 and a delay of 20 *sec* for robot 3.

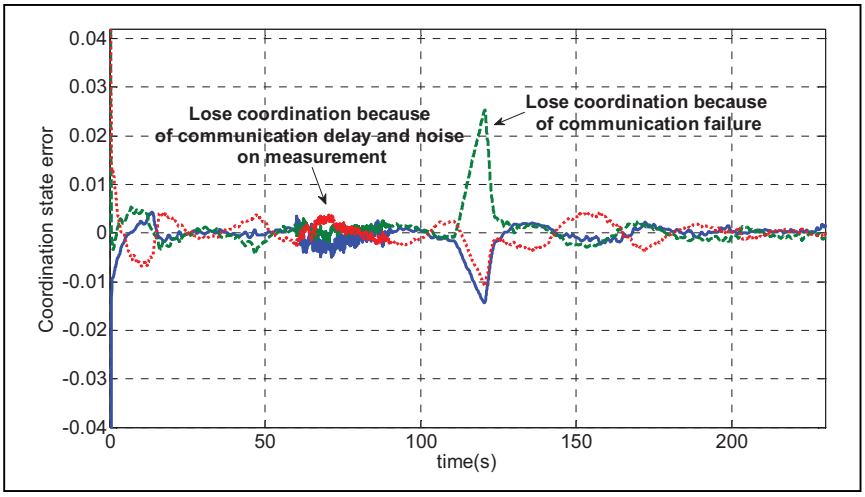


Figure 4.15 Coordination state error $s_i - s_j$.

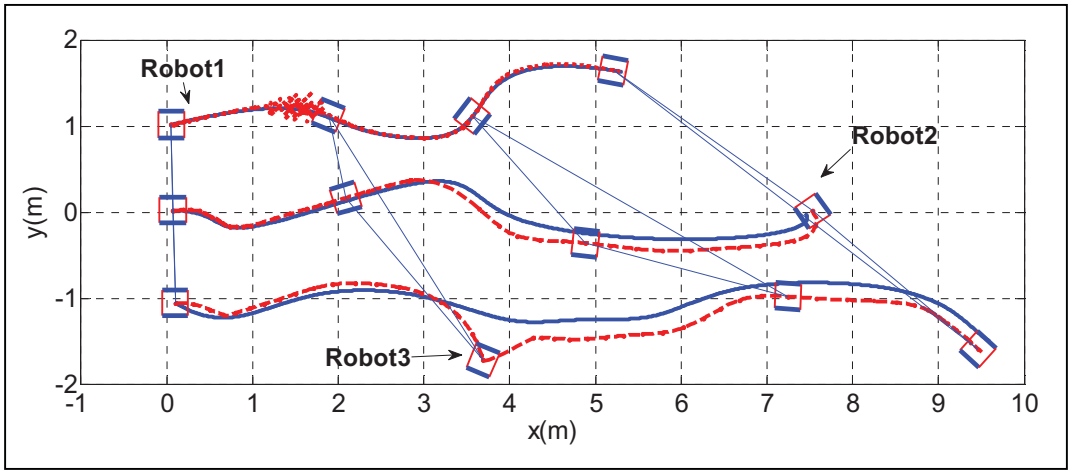


Figure 4.16 References and real robots' trajectories with extended communication failures or delays.

Figure 4.17 presents the coordination state error that exists in these tests. This figure shows that with the existence of a temporary loss of coordination for $20\,sec$ due to short communication failures or delays, the robots can start losing formation.

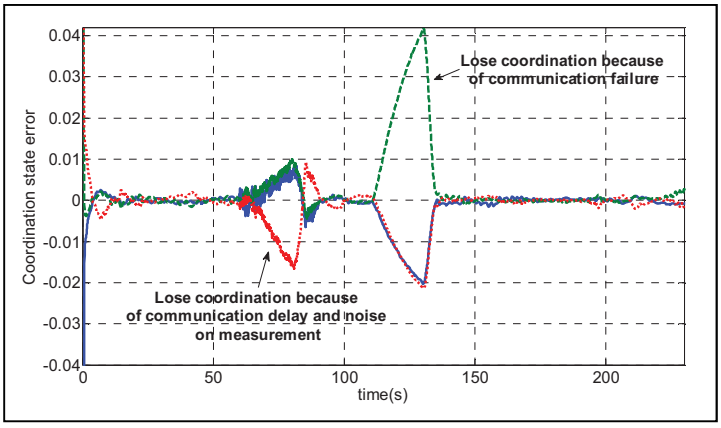


Figure 4.17 Coordination state error $s_i - s_j$.

4.6 Conclusion

In this chapter, a control algorithm and efficient coordination architecture is proposed for a group of mobile robots enabling them to work both individually and in meaningful robot formations. The developed approach uses a combination of Lyapunov technique with graph theory embedded in a virtual structure. A two level controller is designed incorporating a low level PID controller for the right and left motors, and a high level controller to coordinate the speed and movement of the robot group. The high level controller also utilizes a feedback controller employing the Lyapunov function. The communication is developed with Zigbee technologies enabling communication with multiple mobile robots. The experimental results on a multi-robot platform show the effectiveness of the theoretical result and the performance of the system against short term communication loss or failure, as well as position measurement errors.

CHAPTER 5

HIERARCHICAL FUZZY COORDINATION CONTROL FOR A TEAM OF MOBILE ROBOTS

The algorithms designed to create cooperation between multiple mobile robots often use complex mathematical methods which make them difficult to implement. In this chapter, we present an algorithm that coordinates a group of mobile robots utilizing 'fuzzy logic' to simplify the computations used by the algorithm controller. This proposed method has distinct advantages in MMR modeling, where multiple robots are moving along designated trajectories and simultaneously being directed to perform rapid velocity and direction angle changes.

Artificial intelligence research is well documented for the trajectory following of a mobile robot with the evaluation of a neural network approach as well as fuzzy logic controllers in "known" and "unknown" environments (Antonelli, Stefano et Fusco, 2007; Chang et Chen, 2000; Geracioti *et al.*, 1989; Li, Chang et Tong, 2004; Maalouf, Saad et Saliah, 2006). Some of this work uses combinations of neural network and fuzzy, termed 'neurofuzzy' applications (Mbede, Huang et Wang, 2003; Zhu et Yang, 2007). Though single robots have been the primary reference for much of the artificial intelligence research, these same methods can also be used to solve the cooperative problems for a group of multiple mobile robots. The cooperation between robots working as a group is a key issue to be solved where the robots can construct various formations while traveling on separate trajectories that can have differing lengths.

In this chapter, the fuzzy model used for the algorithm controller consists of a two level hierarchical structure, where 'fuzzy logic' is the prime controller and performs the tasks of trajectory following, localization and group cooperation (Mehrjerdi, Saad et Ghommam, 2010c; Mehrjerdi *et al.*, 2010b) . The direction angle of each robot is determined by the desired trajectories and the formations of the whole MMR group. The proposed cooperative method, derived from the use of fuzzy logic and a low level PID controller, empowers the

robots to move, follow and coordinate their velocity and direction on trajectories that may have differing trajectory lengths.

5.1 Modeling of the Mobile Robot

Figure 5.1 shows the general kinematic model of mobile robots and the discretized trajectories. In this figure, $i = 1, \dots, k$ represents the number of individual mobile robots and $P_i = [x_i, y_i, \psi_i]^T$ denotes the position and orientation vector of the i^{th} robot of the group. $Q_{di(n)} = \left[x_{di(n)}, y_{di(n)}, \xi_{di(n)}\right]^T$ represents the coordination of n^{th} sample point on the i^{th} trajectory where $n = 0, \dots, f$. The trajectory is described by a set of discrete node positions $Q_{di(0)}$ to $Q_{di(f)}$ linked to each other starting from the initial position to the final desired position. $Q_{di(n)}$ is the n^{th} discretized sample on the trajectory.

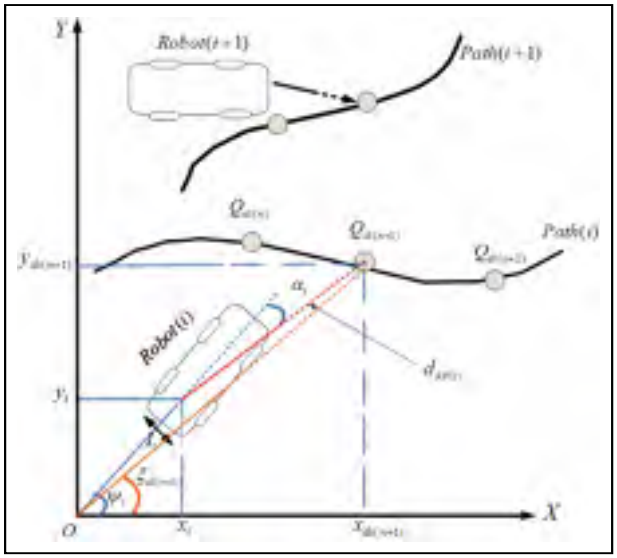


Figure 5.1 Kinematic model of the mobile robots.

The Linear and angular velocities of mobile robots are subject to following bounds:

$$\begin{aligned} |v_i(t)| \le v_{i\,max} \quad |\omega(t)| \le \omega_{i\,max} \\ |\dot{v}_i(t)| \le \dot{v}_{i\,max} \quad |\dot{\omega}(t)| \le \dot{\omega}_{i\,max} \end{aligned} \tag{5.1}$$

In discrete time, equation (5.2) can be explained as follows:

$$\begin{cases} x_i(t_{n+1}) = x_i(t_n) + Tv_i(t_n)\cos\psi_i(t_n) \\ y_i(t_{n+1}) = y_i(t_n) + Tv_i(t_n)\sin\psi_i(t_n) \\ \psi_i(t_{n+1}) = \psi_i(t_n) + T\psi_i(t_n) \end{cases} \tag{5.2}$$

where T and t_n are the sampling time and the expressions of time in n^{th} sample point respectively. The desired robot position and orientation in sample time t_n can be defined as:

$$P_i(t_n) = [x_i(t_n), y_i(t_n), \psi_i(t_n)]^T \tag{5.3}$$

5.2 Optimized Neuro-Fuzzy Coordination for Multiple Mobile Robots

Linear velocity $v_i(t)$ and the orientation angle $\theta_i(t)$ with respect to t can be calculated as:

$$v_i(t) = \sqrt{\dot{x}_i^2(t) + \dot{y}_i^2(t)} \tag{5.4}$$

$$\theta_i(t) = \tan^{-1}(\dot{y}_i(t), \dot{x}_i(t)) \tag{5.5}$$

and angular velocity $\omega_i(t)$ is calculated by deriving of orientation with respect to t as:

$$\omega_i(t) = \frac{1}{1 + (\dot{y}_i(t)/\dot{x}_i(t))^2} \frac{\ddot{y}_i(t)\dot{x}_i(t) - \ddot{x}_i(t)\dot{y}_i(t)}{\dot{x}_i^2(t)} \tag{5.6}$$

Figure 5.2 shows the general model of the system which is used to coordinate multiple mobile robots. Posture sensors mounted on motors of the leader robot can be used to localize it so that its position can be estimated. Trajectory tracking and coordination between leader and follower robots will be given by a neuro-fuzzy algorithm.

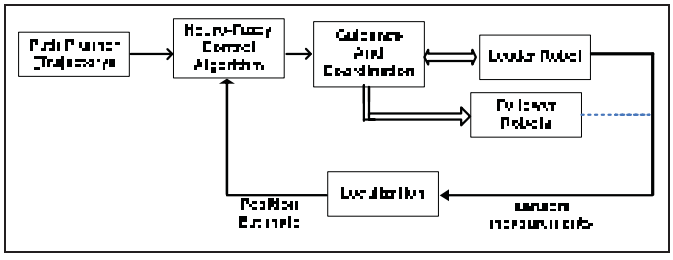


Figure 5.2 Kinematic model for mobile robot.

5.2.1 Generation of Training Data

In this section the generation of training data which are based on the position and trajectory of the leader robot and the initial positions of followers is explained in detail. In order to plan trajectory following and coordination for a group of mobile robots, the principal idea is to construct a mapping between the posture of the leader robot, and the linear and angular velocities of the leader and follower robots. We consider sine trajectories which show a smooth and feasible trajectory that is suitable for the robots to easily follow. The first step is the generation of an appropriate training data set. The input-output data pairs have been obtained by adopting a sine wave trajectory and discretizing it to 400 sampling points. The offline generation of the adopted trajectory is based on knowing trajectory and the initial position of the robots. The training data set has been generated for the five different formations corresponding to the position of the leader and the follower mobile robots. These five formations can be considered as follows:

1) The leader robot is very far back in the group compared to the follower robots. In this situation the leader robot moves very fast to reach the followers, as the follower robots move with their desired velocity.
2) The leader robot is a large distance behind the follower robots, and in this situation the leader robot moves fast to reach the follower robots.
3) The leader robot is in the same position (x-reference) as the follower robots, and in this situation the leader robot moves at the same velocity as the followers on their desired trajectory.
4) The leader robot is a large distance ahead of the follower robots, and in this situation the leader robot stops and the follower robots move until they catch the leader.
5) The leader robot is very far ahead of the follower robots, and again in this situation, the leader robot stops and the follower robots move until they catch the leader.

Figure 5.3 shows these different formations which are considered for training. In this figure F_1 and F_2 denote the follower robots. Coordination between the leader and follower robots is considered in this training procedure.

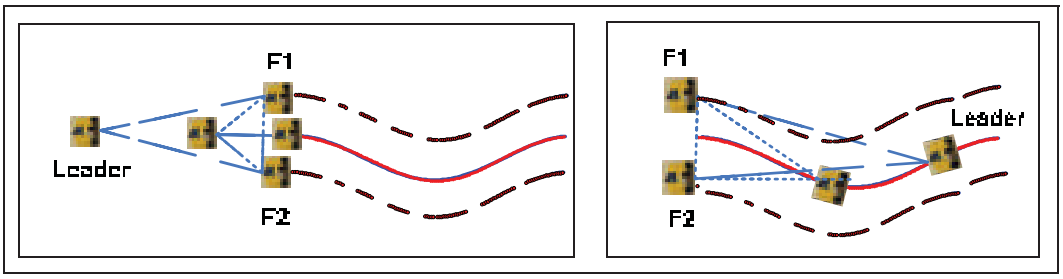


Figure 5.3 Different formations of mobile robots.

Figure 5.4 shows the position of the leader robot in x, y *and* θ coordination considering the training with five different formations. Figures 5.5(a) and 5.5(b) show the obtained linear and angular velocities used for the leader and follower robots in training.

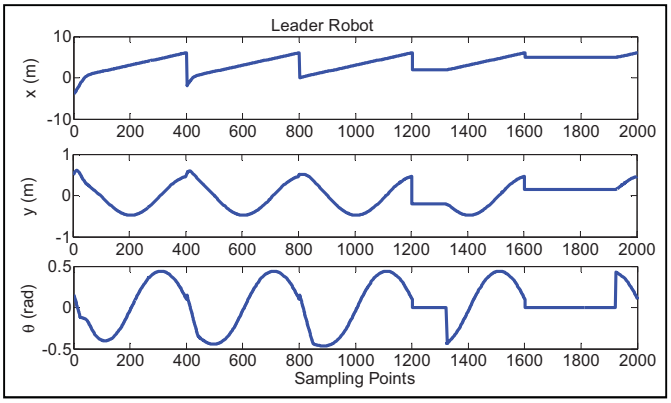
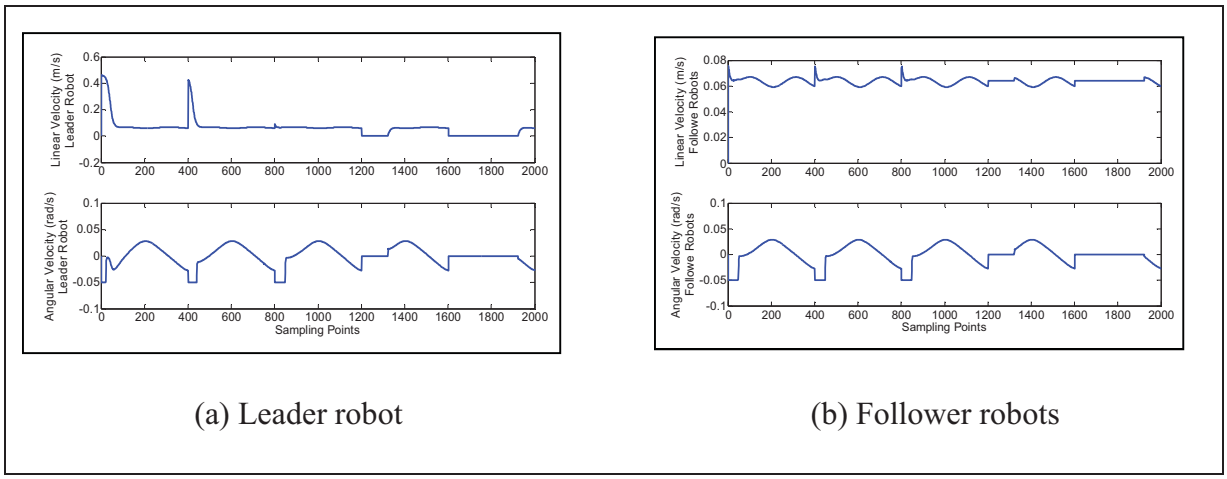


Figure 5.4 Position of the leader robot in training.



(a) Leader robot (b) Follower robots

Figure 5.5 Linear and angular velocity of the robots.

Each formation in training has been sampled to 400 points and five different formations are considered. Consequently the training data is a matrix with dimension of 2000×3 as an input and four matrices which dimension of each is 2000×1 as outputs. The four output data are the linear and angular velocities for the leader and followers robots. We consider all follower robots as a unit, and then we imply the same linear and angular velocities to them. The subtractive clustering with Sugeno-type is made to train these input and output sets.

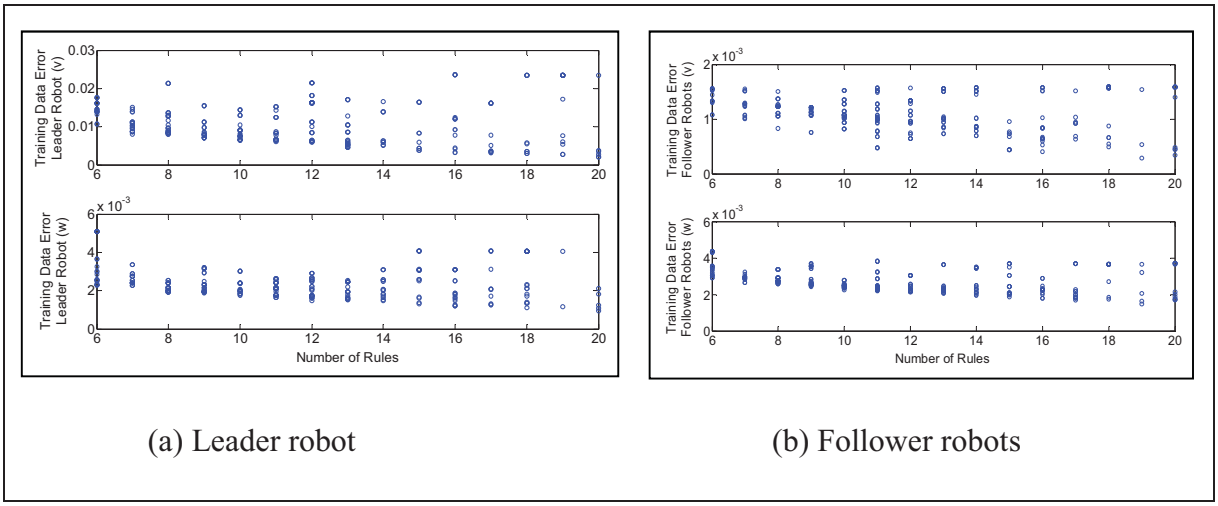
5.2.2 Neuro-Fuzzy Controller Design

An effective solution to the online coordination and trajectory tracking for multiple mobile robots relies on the fuzzy logic based approach. The position of leader robot gives all necessary information to follow a desired trajectory and to establish coordination between the robots. Fuzzy networks can produce the linear and angular velocities for the leader and follower robots which result in the correct steering for the robots. We have used the following type of first order Sugeno rules which are:

$$\begin{gathered} R_i: IF\ x_\iota\ is\ \ A_{1i}\ and\ y_\iota\ is\ A_{2i}\ and\ \theta_\iota\ is\ A_{3i} \\ then \\ v_{\iota i} = P_{0i} + P_{1i} \times x_l + P_{2i} \times y_l + P_{3i} \times \theta_\iota \\ \omega_{\iota i} = Q_{0i} + Q_{1i} \times x_l + Q_{2i} \times y_l + Q_{3i} \times \theta_\iota \\ v_{fi} = R_{0i} + R_{1i} \times x_l + R_{2i} \times y_l + R_{3i} \times \theta_\iota \\ \omega_{\iota i} = S_{0i} + S_{1i} \times x_l + S_{2i} \times y_l + S_{3i} \times \theta_\iota \\ for\ i = 1, \cdots, M \end{gathered} \tag{5.7}$$

where M is the number of rules. The inputs to the fuzzy network controller are $x_l, y_l\ and\ \theta_l$ which present the position of leader robot in each point. These positions are obtained by the incremental encoder sensors mounted on the right and left motors on the wheels of leader robot. $v_{li}, \omega_{li}\ and\ v_{fi}, \omega_{fi}$ denote outputs of fuzzy networks which are the linear and angular velocities for leader and follower mobile robots. These fuzzy controllers provide the trajectory tracking and formation to the robots. Depending on the position of leader robot, the linear and angler velocities (v_l, ω_l) are calculated and sent to the leader robot, while at the

same time the linear and angular velocities (v_f, ω_f) are calculated and sent as a command to the follower robots. Four different 3-input 1-output training data has been prepared (from the different formations). Fuzzy rule antecedent membership functions have been identified by using the subtractive clustering algorithm with enumerative parametric search. The consequent parameters of the rules are optimized by least square estimation. To find the appropriate number of rules and minimum least square error (LSE), an optimization is done with different number of rules and the related LSE are obtained. Figures 5.6(a) and 5.6(b) show the result obtained for these four different fuzzy training sets. These figures present the error of fuzzy clustering in reference to the number of rules for the leader and the follower robots.



(a) Leader robot (b) Follower robots

Figure 5.6 Training data errors and rule numbers.

In view of the number of rules and LSE, eight rule numbers are considered for all four fuzzy training networks. Table 5.1 shows the fuzzy clustering data for these fuzzy networks. These data consist of Radius, Squash factor, LSE, accept and reject ratios.

The structure of the fuzzy inference system with 8 rules is depicted in figure 5.7. In this figure, the inputs $x_l, y_l \; and \; \theta_l$ are measured by the position of the leader robot and the outputs are the linear and angular velocities.

Table 5.1 Fuzzy clustering data

	Fuzzy Network 1	Fuzzy Network 2	Fuzzy Network 3	Fuzzy Network 4
Rule Number	8	8	8	8
Trading Data Error	7.997e-003	2.011e-003	8.287e-004	2.757e-003
Radius	.9	.9	.5	.9
Squash factor	.5	.5	.1	.5
Accept ratio	.5	.5	.9	.5
Reject ratio	.1	.1	.7	.3

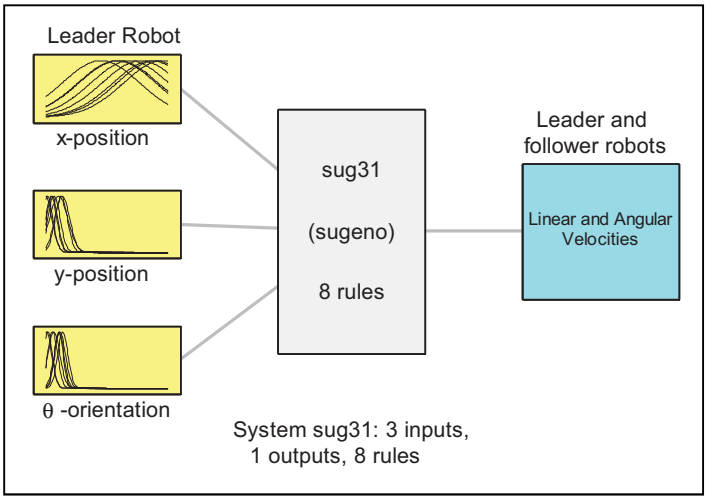
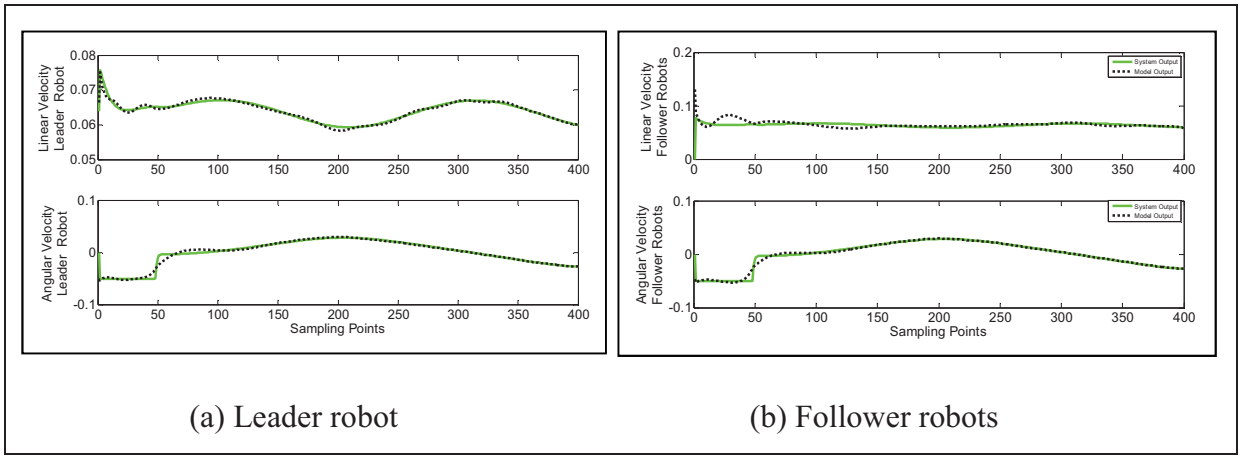


Figure 5.7 Fuzzy inference structure.

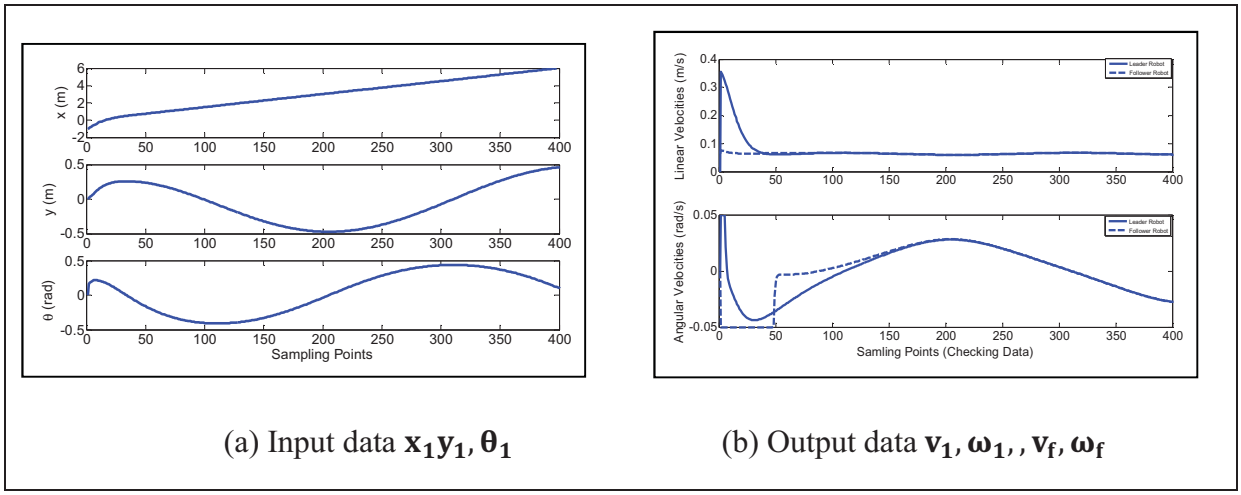
Figures 5.8(a) and 5.8(b) show the result obtained for output velocities after training is done. In this test, the initial position of the leader robot is considered as $[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [-1,0.5,0]^T$ and the initial position of the follower robots are $\left[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)\right]^T = [0,1.5,0]^T$ and $\left[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)\right]^T = [0,-0.5,0]^T$. In this section we explain an adaptive neuro-fuzzy inference system (ANFIS) which constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone or in combination with a least squares type of method. The parameters associated with the membership functions changes through the learning process. To overcome overfitting, we consider the training data set by model. For validation the model, we consider a set of inputs and outputs which are sufficiently distinct from the training data.



(a) Leader robot (b) Follower robots

Figure 5.8 Linear and angular velocities after subtractive clustering.

In principle, the model error for the checking data set tends to decrease as the training takes place up to the point that overfitting begins, and then the model error for the checking data suddenly increases. Figures 5.9(a) and 5.9(b) show the set of inputs and outputs used to validate ANFIS. A different formation from training is considered for checking validation. In this situation the initial position of leader robot is considered as $[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [-1,0.5,0]^T$ and the initial position of follower robots are $\left[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)\right]^T = [0,1.5,0]^T$ and $\left[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)\right]^T = [0,-0.5,0]^T$.



(a) Input data $\mathbf{x_1 y_1, \theta_1}$ (b) Output data $\mathbf{v_1, \omega_1, , v_f, \omega_f}$

Figure 5.9 Checking data.

The membership functions of the input variables identified by ANFIS are shown in figure (5.10).

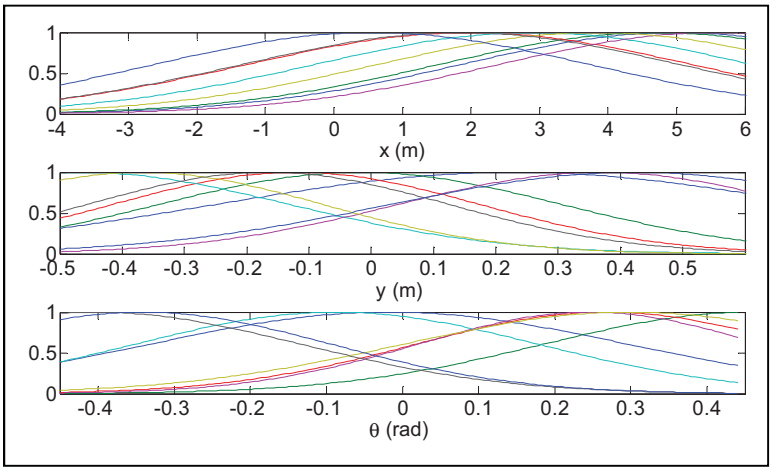


Figure 5.10 Antecedent membership functions for the derived fuzzy model (fuzzy network 1).

The premise parameters which are identified by the cluster center c and the standard deviations σ of the extracted Gaussian membership function parameters are shown in Tables 5.2 and 5.3 for all four fuzzy networks.

Table 5.2 Antecedent parameters (leader robot)

	Fuzzy Network 1 (Linear Velocity)			Fuzzy Network 2 (Angular Velocity)		
	Input 1 $[c, \sigma]$	**Input 2 $[c, \sigma]$**	**Input 3 $[c, \sigma]$**	**Input 1 $[c, \sigma]$**	**Input 2 $[c, \sigma]$**	**Input 3 $[c, \sigma]$**
Rule 1	[3.159 5.006]	[0.4799 0.2307]	[0.3132 -0.01601	[3.201 4.999]	[0.3777 0.2068]	[0.2781 0.08375]
Rule 2	[3.177 4.703]	[0.3243 -0.01734]	[0.2728 0.4576]	[3.197 4.666]	[0.3385 -0.009894]	[0.234 0.1455]
Rule 3	[3.251 1.979]	[0.2905 -0.1226]	[0.2502 0.2665]	[3.23 1.992]	[0.3569 -0.2098]	[0.234 0.1455]
Rule 4	[3.195 2.906]	[0.335 -0.4667]	[0.2654 -0.08668]	[3.19 5.451]	[0.3604 0.3317]	[0.2451 0.3057]
Rule 5	[3.172 5.569]	[0.3231 0.3682]	[0.2276 0.2465]	[3.197 3.704]	[0.3419 -0.3881]	[0.3357 0.2373]
Rule 6	[3.194 3.825]	[0.2902 -0.3674]	[0.298 0.2993]	[3.195 2.731]	[0.2556 -0.3129]	[0.3605 -0.2413]
Rule 7	[3.201 1.852]	[0.2909 -0.1657]	[0.2697 -0.4027]	[3.19 1.913]	[0.3058 -0.214]	[0.342 -0.3712]
Rule 8	[3.184 0.5505]	[0.3902 0.4227]	[0.2462 -0.3412]	[3.182 0.3796]	[0.3895 0.3296]	[0.1725 -0.06248]

Table 5.3 Antecedent parameters (follower robots)

	Fuzzy Network 1 (Linear Velocity)			Fuzzy Network 2 (Angular Velocity)		
	Input 1 $[c, \sigma]$	Input 2 $[c, \sigma]$	Input 3 $[c, \sigma]$	Input 1 $[c, \sigma]$	Input 2 $[c, \sigma]$	Input 3 $[c, \sigma]$
Rule 1	[3.159 5.006]	[0.4799 0.2307]	[0.3132 -0.01601	[3.201 4.999]	[0.3777 0.2068]	[0.2781 0.08375]
Rule 2	[3.177 4.703]	[0.3243 -0.01734]	[0.2728 0.4576]	[3.197 4.666]	[0.3385 -0.009894]	[0.234 0.1455]
Rule 3	[3.251 1.979]	[0.2905 -0.1226]	[0.2502 0.2665]	[3.23 1.992]	[0.3569 -0.2098]	[0.234 0.1455]
Rule 4	[3.195 2.906]	[0.335 -0.4667]	[0.2654 -0.08668]	[3.19 5.451]	[0.3604 0.3317]	[0.2451 0.3057]
Rule 5	[3.172 5.569]	[0.3231 0.3682]	[0.2276 0.2465]	[3.197 3.704]	[0.3419 -0.3881]	[0.3357 0.2373]
Rule 6	[3.194 3.825]	[0.2902 -0.3674]	[0.298 0.2993]	[3.195 2.731]	[0.2556 -0.3129]	[0.3605 -0.2413]
Rule 7	[3.201 1.852]	[0.2909 -0.1657]	[0.2697 -0.4027]	[3.19 1.913]	[0.3058 -0.214]	[0.342 -0.3712]
Rule 8	[3.184 0.5505]	[0.3902 0.4227]	[0.2462 -0.3412]	[3.182 0.3796]	[0.3895 0.3296]	[0.1725 -0.06248]

Table 5.4 Consequent parameters (leader robot)

	Fuzzy Network 1 Leader Robot (Linear Velocity)				Fuzzy Network 2 Leader Robot (Angular Velocity)			
	P_{1i}	P_{2i}	P_{3i}	P_{0i}	Q_{1i}	Q_{2i}	Q_{3i}	Q_{0i}
Rule 1	-0.01651	-0.0319	-0.0223	0.0771	-0.2398	-5.147	-6.918	3.265
Rule 2	0.4022	-0.02227	-0.7788	-1.318	0.221	0.9656	2.945	-3.645
Rule 3	-0.1351	-1.2	1.519	1.13	-0.9262	-6.303	7.207	-1.14
Rule 4	-0.1884	0.2303	0.3745	0.53	-0.1634	-3.077	-3.415	4.804
Rule 5	-0.833	-0.3675	2.264	2.431	0.4762	2.905	0.868	-1.297
Rule 6	-0.0098	-1.203	-0.109	0.7862	0.249	3.408	3.207	4.1
Rule 7	1.476	-2.942	-1.121	-6.367	-0.2407	0.8054	-3.621	-2.523
Rule 8	0.00814	-0.03152	0.05042	0.00193	0.004623	0.01106	-0.189	-0.02402

Table 5.5 Consequent parameters (follower robots)

	Fuzzy Network 3 Leader Robot (Linear Velocity)				Fuzzy Network 4 Leader Robot (Angular Velocity)			
	P_{1i}	P_{2i}	P_{3i}	P_{0i}	Q_{1i}	Q_{2i}	Q_{3i}	Q_{0i}
Rule 1	1.504	32.85	34.58	-20.4	-0.01683	-6.836	-1.102	3.607
Rule 2	1.57	-8.551	-2.619	-15.97	0.04864	-0.4284	0.0635	0.3286
Rule 3	5.219	62.63	-27.95	5.955	-0.5693	-6.271	2.212	-2.352
Rule 4	-0.573	3.701	2.056	8.091	-0.329	0.7379	3.19	2.722
Rule 5	0.6148	7.805	15.09	-19.11	-0.7134	-1.494	3.19	-0.6362
Rule 6	-2.372	-2.137	4.933	16.81	1.704	3.518	2.952	-0.3323
Rule 7	4.199	9.684	8.577	0.4569	-1.407	-2.424	-2.939	0.3328
Rule 8	0.3871	-5.17	2.612	5.771	0.04004	-0.3503	0.2926	0.3574

Figure 5.11 shows a general model for the adaptive neuro-fuzzy used for the modeling of our system. The inputs are the position of the leader robot and the outputs are the velocities of the leader and follower robots.

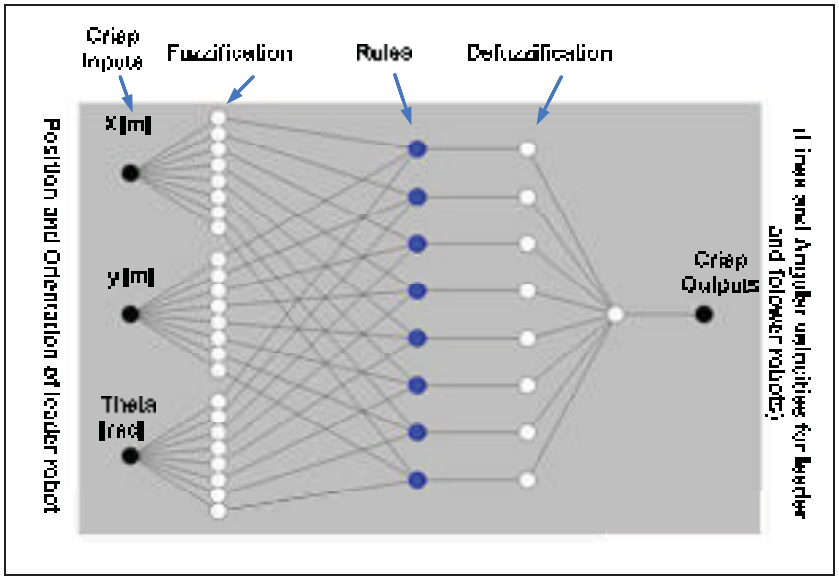
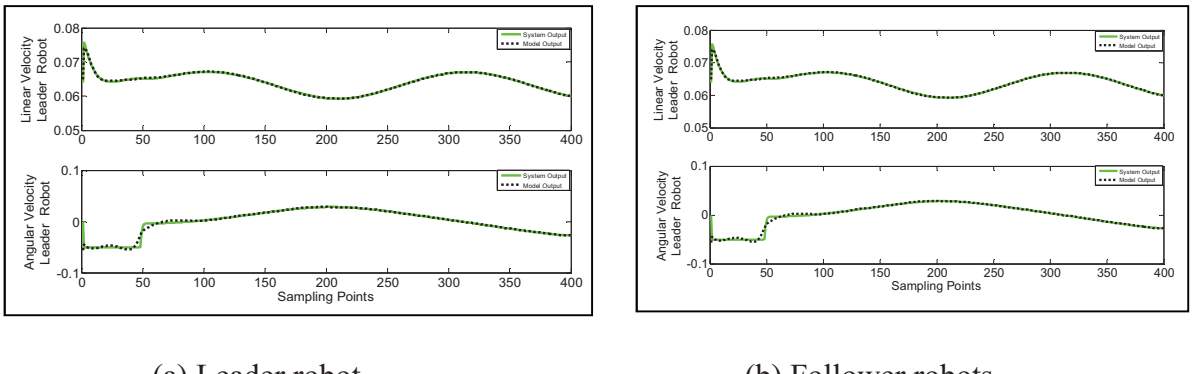


Figure 5.11 Adaptive neuro-fuzzy model (fuzzy network 1).

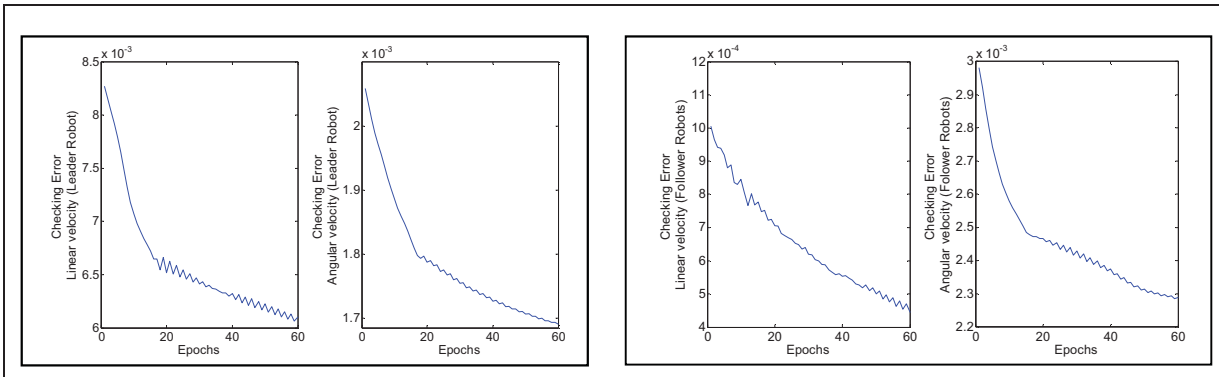
Figures 5.12(a) and 5.12(b) show the result obtained for output velocities after our input-output data sets are trained by ANFIS. In this situation the initial position of the leader robot is considered as$[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [-1,0.5,0]^T$ and the initial position of the follower robots are $\left[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)\right]^T = [0,1.5,0]^T$ and $\left[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)\right]^T = [0,-0.5,0]^T$.



(a) Leader robot (b) Follower robots

Figure 5.12 Linear and angular velocities after ANFIS training.

Figure 5.13(a) and 5.13(b) show the LSE's obtained by checking the input-output set and ANFIS training with 60 epochs.



(a) Leader Robot (b) Follower robots

Figure 5.13 Checking error versus epoch number.

5.2.3 Experimental Results

Figure 5.14 shows a general view of the experimental setup and Figure 5.15 shows the leader-followers architecture for MMR.

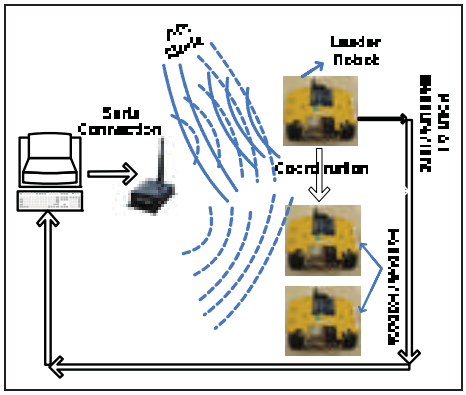


Figure 5.14 The general view of experimental setup.

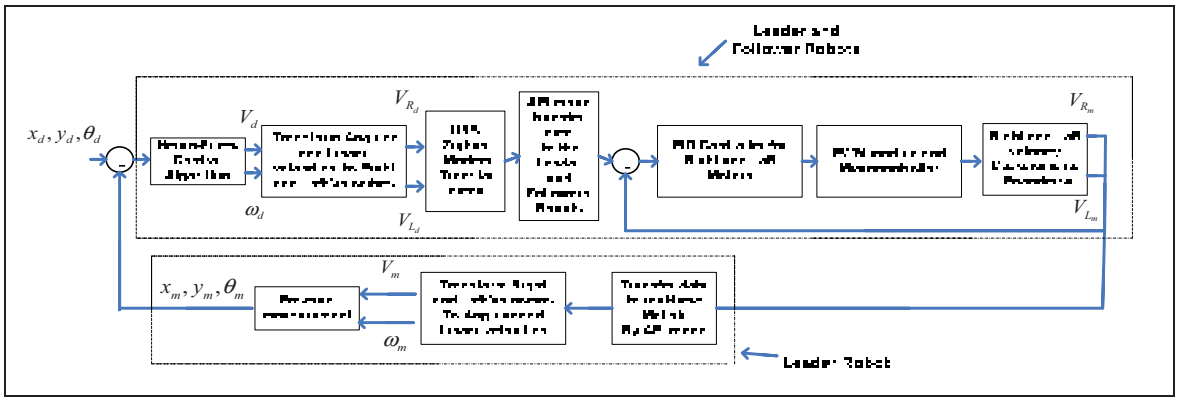


Figure 5.15 Intelligent coordination and trajectory tracking architecture for multiple mobile robots.

The Cartesian coordination of the mobile robots can be obtained as follows:

$$x = v\cos(\theta)\ , \qquad y = v\sin(\theta)\ , \qquad \theta = \int \omega\, dt \tag{5.8}$$

In the first experimental test, the leader robot is placed far back behind the follower robots with the initial position of $\left[x_1(t_0), y_1(t_0), \theta_1(t_0)\right]^T = [-3, 0.5, 0]^T$.

The initial position of the follower robots are $[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)]^T = [0,1.5,0]^T$ and $[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)]^T = [0,-0.5,0]^T$ accordingly. Figure 5.16 shows the results obtained in $the\ x, y$ coordinates which presents the position of the mobile robot in trajectory. As we can see in this figure, the leader moves fast to catch the follower robots as it sends commands to the followers to move at a slower velocity. As soon as the leader robot catches the followers, all the robots continue to move on the sine trajectory with the same velocity.

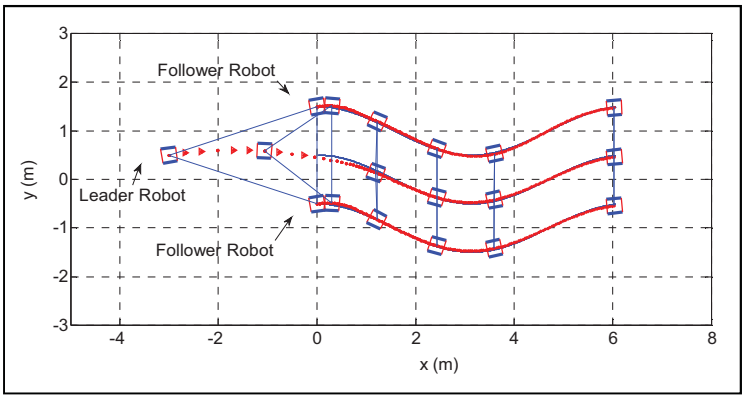
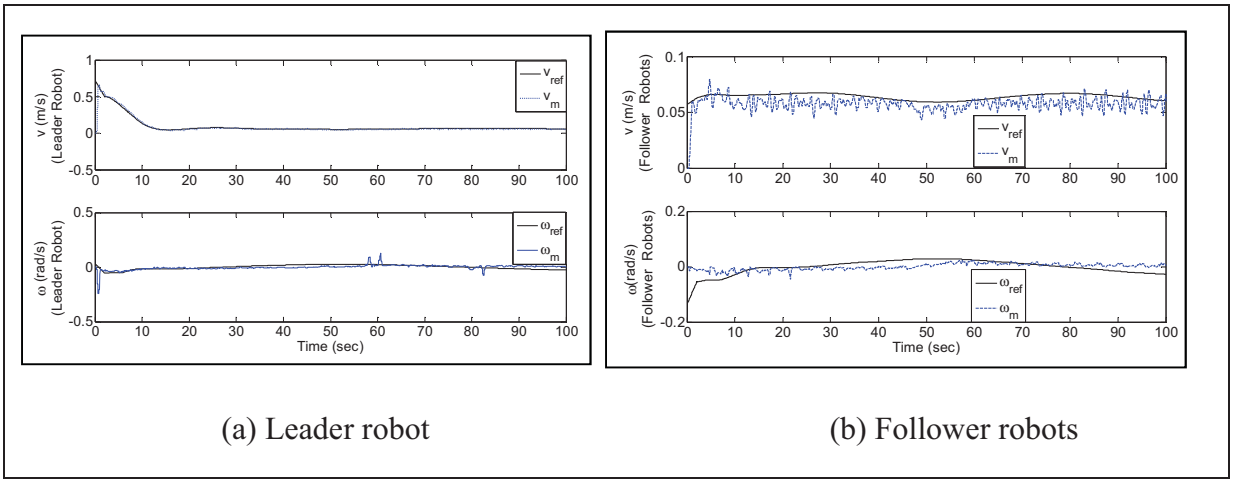


Figure 5.16 Coordination and trajectory tracking for multiple robots.

Figures 5.17(a) and 5.17(b) show the reference and measured linear and angular velocities obtained by the fuzzy networks and given to the leader and follower robots.



(a) Leader robot (b) Follower robots

Figure 5.17 Linear and Angular velocities for robots.

In the second experimental test, the leader robot is placed far ahead of the follower robots with an initial position of $[x_1(t_0), y_1(t_0), \theta_1(t_0)]^T = [3, -0.5, 0]^T$ and the initial position of follower robots are: $[x_{f1}(t_0), y_{f1}(t_0), \theta_{f1}(t_0)]^T = [0, 1.5, 0]^T$ and $[x_{f2}(t_0), y_{f2}(t_0), \theta_{f2}(t_0)]^T = [0, -0.5, 0]^T$. Figure 5.18 shows the results obtained in the x, y coordinates. As we can see in this figure, the leader robot stops and sends commands to the follower robots to move until all the robots arrive at the same point (x-reference). As soon as the follower robots catch the leader, all the robots continue to move on the sine trajectory.

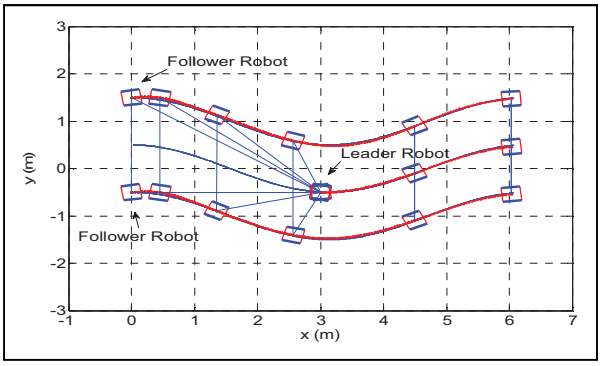
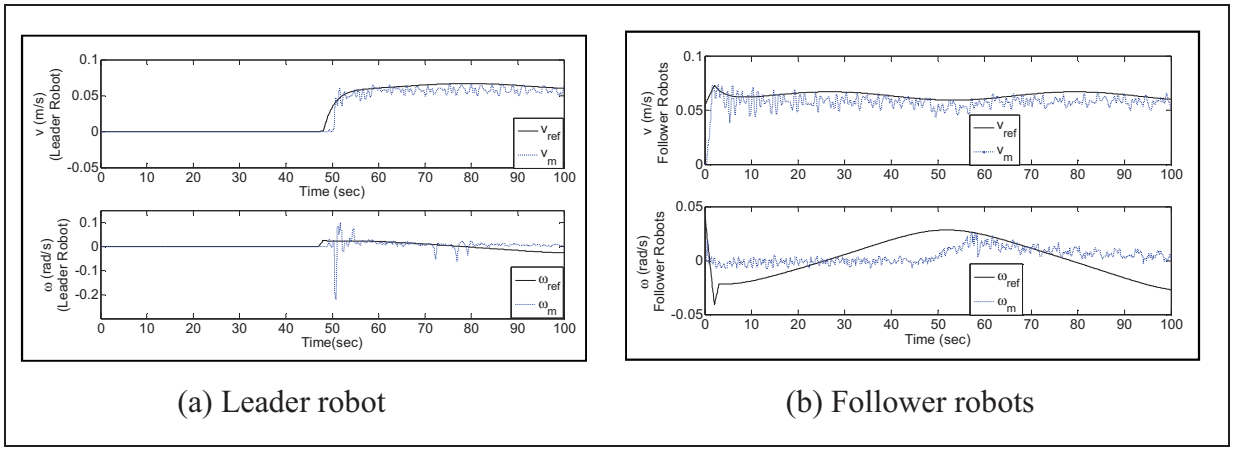


Figure 5.18 Coordination and trajectory tracking for multiple robots.

Figures 5.19(a) and 5.19(b) illustrate the reference and measured linear and angular velocities obtained by the fuzzy networks and given to the leader and follower robots.



(a) Leader robot (b) Follower robots

Figure 5.19 Linear and Angular velocities for the robots.

5.3 Hierarchical Fuzzy Cooperative Control and Trajectory Tracking for a Team of Mobile Robots

Figure 5.20 illustrates the block diagram used to control and synchronize a group of mobile robots. There are two Hierarchical levels of controller being low and high respectively. The high level controller, which is a fuzzy controller, is designed to coordinate the group formation as well as instructing the robots to follow their generated trajectories. Figure 5.21 shows the fuzzy system structure with the desired inputs and outputs.

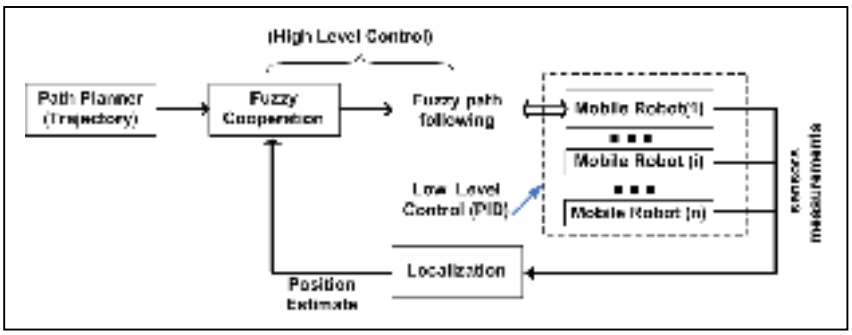


Figure 5.20 Infrastructure of multi mobile control and cooperation.

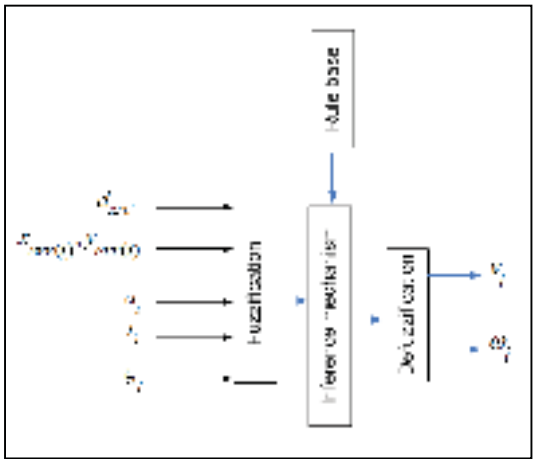


Figure 5.21 Fuzzy control structure.

The inputs of the fuzzy controller are $d_{RP(i)}$, α_i , L_i , ς_i , $x_{err(i)}$ and $y_{err(i)}$, where $d_{RP(i)}$ is the distance from the actual position of i^{th} robot to the next desired position, α_i is the difference between the line joining the current position to the next desired position and the actual

heading of the robot. L_i is the length of trajectories and ς_i denotes the trajectory parameter being used to synchronize the individual robot within the group formation. $x_{err(i)}$, $y_{err(i)}$ and α_i are calculated by the following equations in robot reference as (B. Siciliano et Khatib, 2008):

$$\begin{bmatrix} x_{err(i)} \\ y_{err(i)} \\ \psi_{err(i)} \end{bmatrix} = \begin{bmatrix} \cos\psi_i & sin\,\psi_i & 0 \\ -\sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{di} - x_i \\ y_{di} - y_i \\ \zeta_{di} - \psi_i \end{bmatrix} \tag{5.9}$$

The output of the fuzzy controller determines the linear and angular velocities of individual robots.

5.3.1 Fuzzy Trajectory Tracking and the Cooperative Controller

The form of the control law equation for trajectory tracking and cooperation is as follows:

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} f_1(d_{RP(i)}, \alpha_i, x_{err(i)}, y_{err(i)}, L_i, \zeta_i) \\ f_2(d_{RP(i)}, \alpha_i, x_{err(i)}, y_{err(i)}, L_i, \zeta_i) \end{bmatrix} \tag{5.10}$$

Rule bases for trajectory tracking and cooperation are shown in Table 5.6. In this table the inputs 1, 2 and 3 are $d_{RP(i)}$, α_i and $x_{err(i)}$ and outputs 1 and 2 are v_i, ω_i respectively.

Consider a group of mobile robots each with a local controller for trajectory tracking. In order to have cooperation across the whole robot group, they are required to move along their trajectories while maintaining a desired inter-robot formation pattern, as well as reaching their final goals at the same time regardless of their trajectory lengths or their initial positions. To obtain group cooperation between the robots, each trajectory is parameterized in terms of parameter ς_i. The robots will keep in cooperation if $\varsigma_i = \varsigma_j$ for all i, j .

This parameter is defined as $\varsigma_i = \frac{S_i}{L_i}$, where s_i is signed curvilinear abscissa of sampling points $Q_{di(n)}$ along the trajectory. For group cooperation, all trajectories are discretized to the same number of sampling points. As the robot group needs to reach their destination at the same time, if a trajectory is longer, then the distance between the sampling points will

increase, thus forcing the individual robot to move faster in order to complete the same number of sampling points within the allocated time. However, if the trajectory length is smaller, then the distance between the sampling points will decrease and the robot will move slower with respect to the other robots in the group. To improve the cooperation between the robots travelling in formation, the following rules are applied when they are ahead or behind to their desired trajectory position:

Table 5.6 Rule base for trajectory tracking

Rule1	if (*input 1 is Very close*) and (*input 2 is No matter what*) and *(input3 is Poserr*) then (*output 1 is Very very slow*) (*output2 is Zero*)
Rule2	if *(input 1 is Close*) and (*input 2 is No matter what*) and (*(input 3 is Poserr*) then (*output 1 is Very slow*) (*output 2 is zero*)
Rule3	if (*input 1 is Medium*) and (*input 2 is No matter what*) and *(input 3 is Poserr*) then (*output 1 is Slow*) (*output 2 is Zero*)
Rule4	if (*input 1 is Far*) and (*input 2 is No matter what*) and *(input 3 is Poserr*) then (*output 1 is Fast*) (*output 2 is Zero*)
Rule5	if (*input 1 is Very far*) and (*input 2 is No matter what*) and (*(input 3 is Poserr*) then (*output 1 is Very fast*) (*output 2 is Zero*)
Rule6	if (*input 1 is No matter what*) and *(input 2 is Big negative*) and (*input 3 is Poserr*) then (*output 1 is Very very slow*) (*output2 is Big negative*)
Rule7	if (*input 1 is No matter what*) and (*input 2 is Negative*) and *(input 3 is Poserr*) then (*output 1 is V ery very slow*) (*output 2 is Negative*)
Rule8	if (*input 1 is No matter what*) and (*input 2 is Straight*) and (*(input3 is Poserr*) then (*output 1 is Very very slow*) (*output2 is zero*)
Rule9	if (*input 1 is No matter what*) and (*input 2 is Positive*) and (*(input 3 is Poserr*) then (*output 1 is Very very slow*) (*output2 is Positive*)
Rule10	if (*input 1 is No matter what*) and (*input 2 is Big positive*) and (*(input 3 is Poserr*) then (*output 1 is Very very slow*) (*output 2 is Big positive*)
Rule11	if (*input 1 is No matter what*) and (*input 2 is No matter what*) and (*input 3 is Negerr*) then (*output 1 is Zero*) (*output2 is Zero*)

- *If the robot is ahead of the desired trajectory ($x_{err(i)} < 0$), then it will stop until the condition $\varsigma_i = \varsigma_j$ is fulfilled for all robots.*
- *If the robot is behind the desired trajectory ($x_{err(i)} \geq 0$), then the robot will go faster to catch the trajectory and the other robots.*

Figure 5.22 and Figure 5.23 show the linear and angular velocities obtained by the inputs $d_{RP(i)}$, $x_{err(i)}$. As can be seen in these figures, if $x_{err(i)} < 0$ then the linear and angular velocities will be zero and the robot will stop.

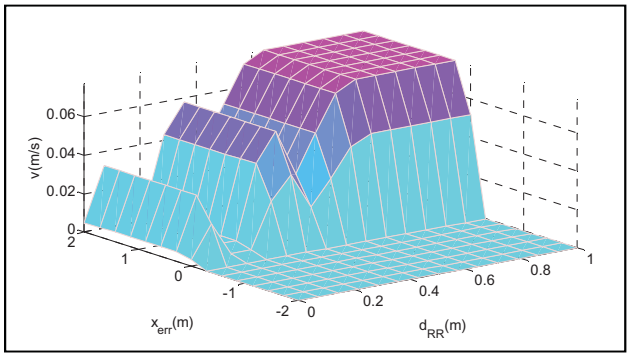


Figure 5.22 Linear velocity obtained by the fuzzy controller and inputs $d_{RP(i)}$, $x_{err(i)}$.

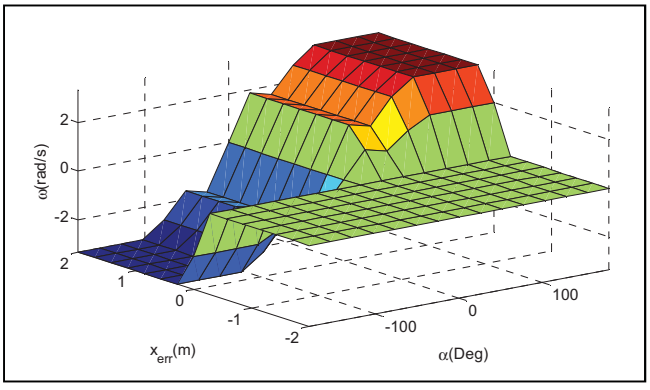


Figure 5.23 Angular velocity obtained by the fuzzy controller and inputs α_i, $x_{err(i)}$.

Figure 5.24 represents the trajectory tracking and the cooperation flow chart algorithm.

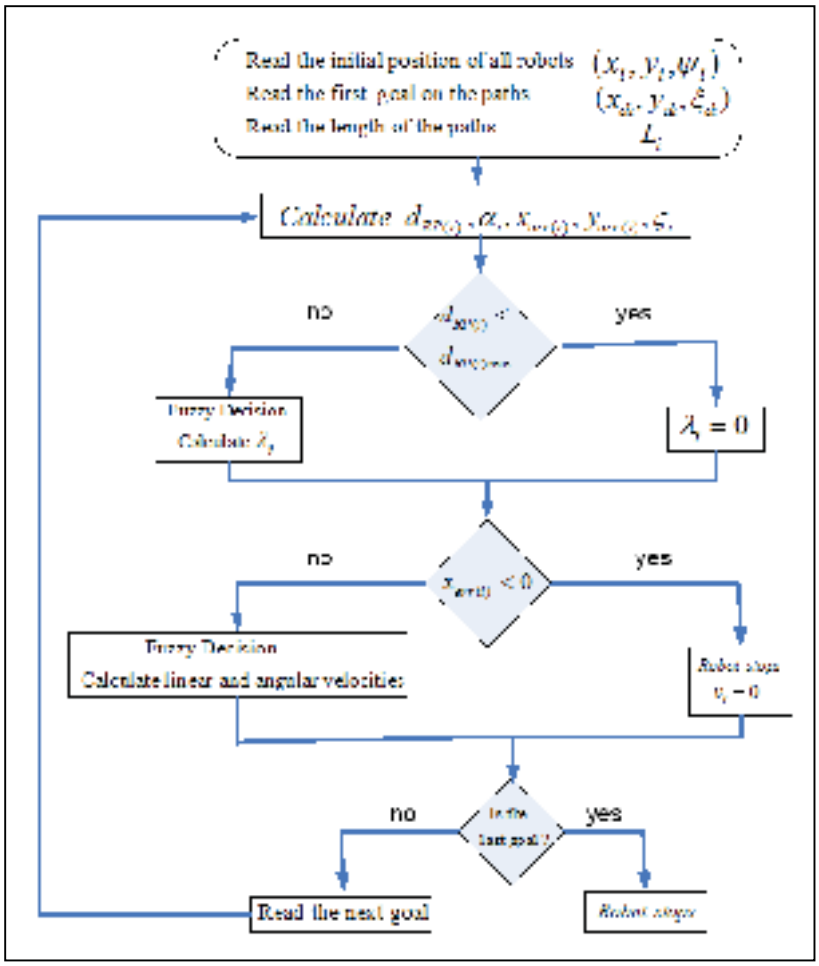


Figure 5.24 Intelligent flow chart of trajectory tracking and cooperation.

5.3.2 Stability Proof of the Cooperative Algorithm

To consider the cooperation between the robots, we make use of the virtual structure approach. $\dot{\tilde{\varsigma}}_i$ will be a corrective signal to ensure the synchronization of the robots along with the center of the virtual structure of the rigid formation. The fuzzy system (2.42) considering the corrective term $\dot{\tilde{S}}_i$ is rewritten as follows:

$$\dot{X}_i = \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j\left(A_i^j - B_i^j K_i^j\right)X_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j} + \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j}\dot{\tilde{\zeta}}_\iota \tag{5.11}$$

where H_i^j is defined as

$$H_i^j = \begin{bmatrix} -\bar{v}_{di}\sin\left(\alpha_i^{eq} + (y_i^{eq})'\right) & \bar{v}_{di}\sin(\alpha_i^{eq} + (y_i^{eq})') \\ 0 & 0 \end{bmatrix} \tag{5.12}$$

We define the Lyapunov function for this system as:

$$V = 0.5\sum_{i=1}^{n}\tilde{X}_i^T P_i \tilde{X} + (\zeta_i - \zeta_0)^2 \tag{5.13}$$

The differentiation of this Lyapunov function gives:

$$\dot{V} = -\sum_{i=1}^{n}\tilde{X}_i^T Q_i \tilde{X}_\iota + (\zeta_i - \zeta_0)(\dot{\tilde{\zeta}}_i + \bar{\omega}_i\left(d_{RP(i)}\right) - \dot{\zeta}_0 + \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j}\dot{\tilde{\zeta}}_i \tag{5.14}$$

If we choose $\bar{\omega}_i(d_{RP(i)}) = \dot{\varsigma}_0 = \bar{\omega}_i(t)(1 - k_1 e^{k_2(t-t_0)})(1 - \tanh(d_{RP}{}^T d_{RP}))$

where $d_{RP} = \left[d_{RP(1)}, d_{RP(2)}, ..., d_{RP(n)}\right]^T$, this yields:

$$\dot{V} = -\sum_{i=1}^{n}\tilde{X}_i^T Q_i \tilde{X}_\iota + \frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j} + (\zeta_i - \zeta_0))\dot{\tilde{\zeta}}_i \tag{5.15}$$

If we choose: $\dot{\tilde{\zeta}}_\iota = -\delta_i\frac{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j H_i^j \tilde{X}_i^j}{\sum_{K=1}^{r}\sum_{j=1}^{r}\mu_k\mu_j} + (\zeta_i - \zeta_0)) = -\delta_i n_i$

Then from (5.15) we will have:

$$\dot{V} = -\sum_{i=1}^{n}\tilde{X}_i^T Q_i \tilde{X}_i - \delta_i n_i^2 \tag{5.16}$$

which is negative definite. This shows by Barbalet's lemma that $\tilde{X}_i \to 0$ and $\eta_i \to 0$. It is easy to see then that by construction we have $\varsigma_i - \varsigma_j \to 0$.

5.3.3 Experimental Results

To illustrate the performance of the proposed cooperation and control scheme, some tests are performed using different trajectories on three mobile robots. Figure 5.25 shows the structural design of the control, trajectory planning, and cooperative behavior for the group of mobile robots being used in the experimental tests.

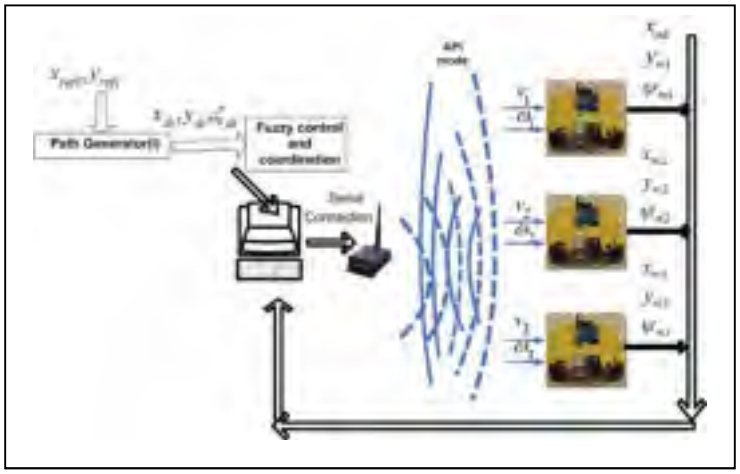


Figure 5.25 General view of the experimental setup.

In this section, we discuss the results of using different formations of robots. These formations are:

1) where robots travel on trajectories with differing lengths;
2) where robots travel on trajectories with the same lengths, but the robots are not placed on the trajectories.

In the first test, the trajectories which have different lengths are considered. In this scenario, the robots are placed on a common vertical line with the initial position and lengths as:

$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [0,1,0]^T, [x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [0,0,0]^T, [x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [0,-1,0]^T$
$L_1 = 5.41\ m\ , L_2 = 7.75\ m, L_3 = 9.83\ m$

Figure 5.26 shows the reference and the actual robots trajectories in the first scenario. The trajectory tracking error y_{err} is shown in Figure 5.27. As can be seen in these figures, the robots travel along their trajectories with negligible errors and the formation is experimentally successful.

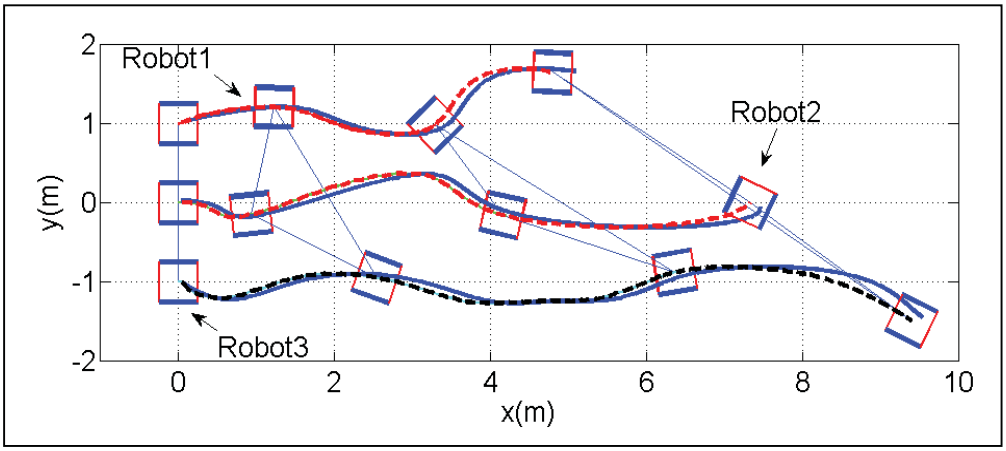


Figure 5.26 Reference and real robots' trajectories.

The linear and angular velocities are plotted in Figure 5.28. These figures show that the robots travel with different velocities relative to the length of the trajectory on which they travel. We observe that robot 3 has the highest velocity (longest trajectory) and robot 1 the lowest velocity (shortest length).

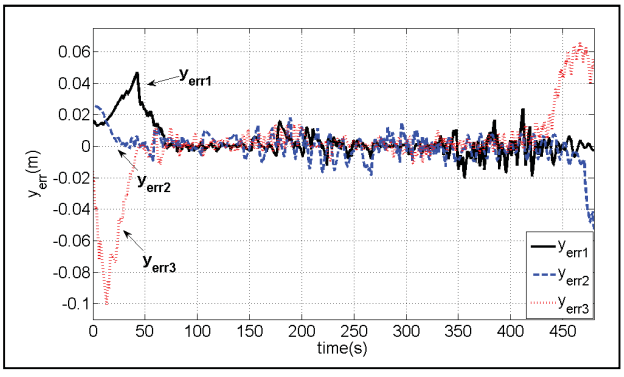
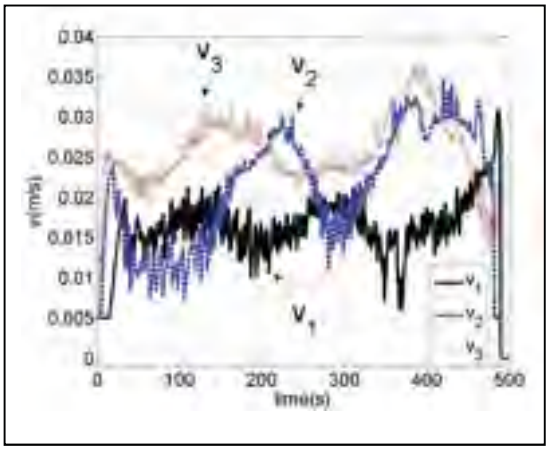
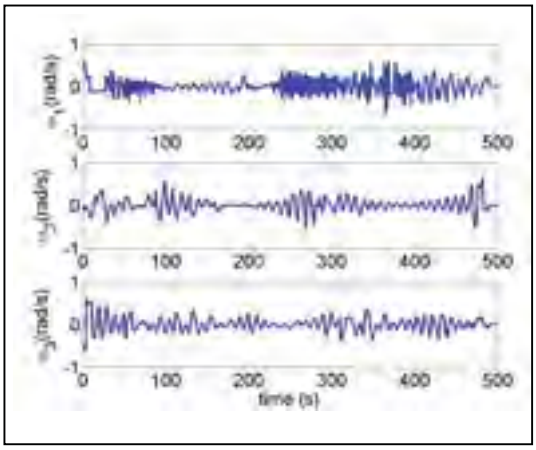


Figure 5.27 Trajectory tracking errors $y_{err(i)}$.





(a) Linear (b) Angular

Figure 5.28 Velocity of robots.

In the second test, trajectories with the same lengths are considered, but the robots are not placed on the trajectories. As can be seen in Figure 5.29, robots 1 and 3 are ahead of their trajectories and robot 2 is behind its trajectory. In this scenario, all trajectories have the same length as $L_1 = 7.76\ m$ and the initial positions of robots are defined by:

$$[x_1(t_0), y_1(t_0), \psi_1(t_0)]^T = [1,2,0]^T, [x_2(t_0), y_2(t_0), \psi_2(t_0)]^T = [-1.5,1,0]^T, [x_3(t_0), y_3(t_0), \psi_3(t_0)]^T = [2,-2,0]^T$$

Figure 5.29 shows both the reference and the actual robots trajectories in this scenario. This figure shows that to keep the formation and cooperation between the MMR group, robots 1 and 3 come to a stop and allow robot 2 to travel faster to reach them. As soon as robot 2 arrives at the same vertical point as robots 1 and 3, they adjust their velocities so that all three robots arrive at the desired sampling points at the same time. The trajectory tracking error y_{err} is shown in Figure 5.30. As can be seen in this figure, the errors reduce to zero as the robots catch their predefined trajectories.

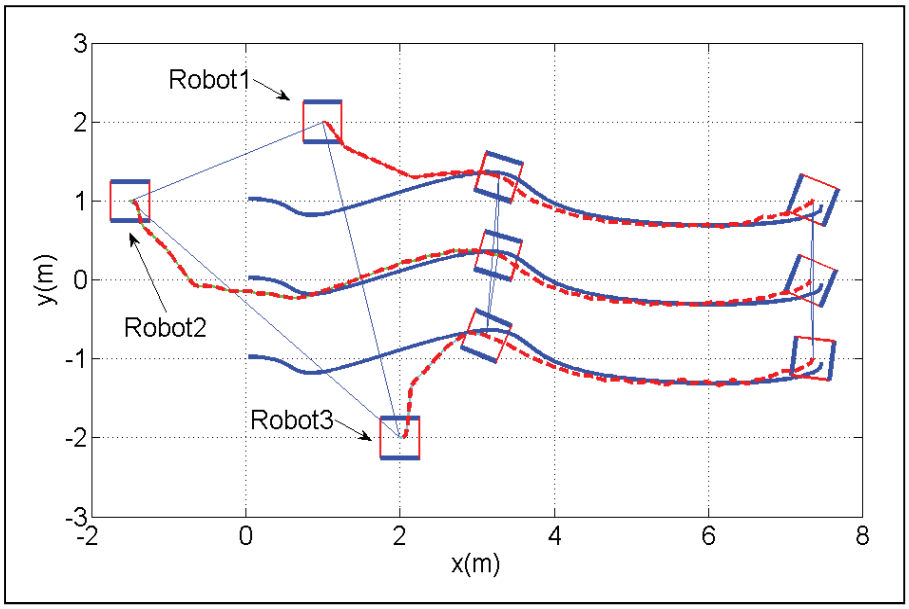


Figure 5.29 Reference and real robots' trajectories.

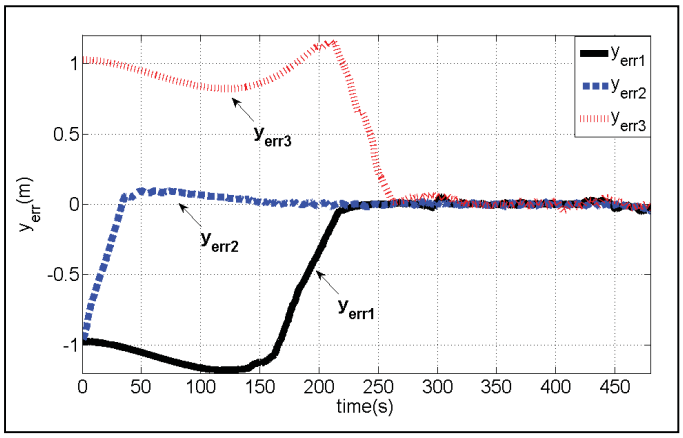
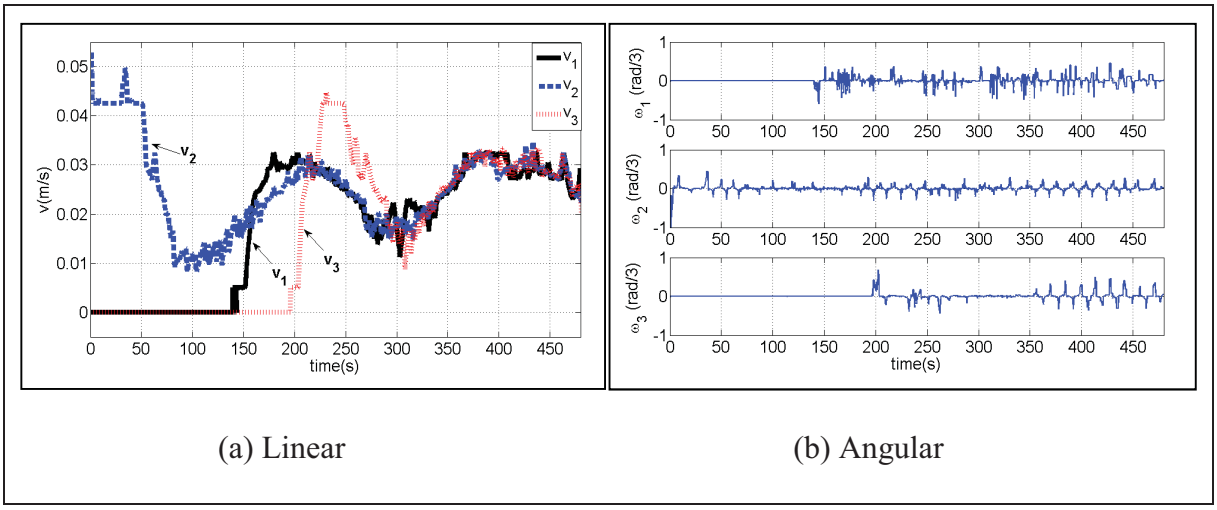


Figure 5.30 Trajectory tracking errors $y_{err(i)}$.

The linear and angular velocities are plotted in Figure 5.31. These figures show that robots 1 and 3 stop and robot 2 moves faster to catch the trajectory.



(a) Linear (b) Angular

Figure 5.31 Velocity of the robots.

5.4 Results Comparison of both the Fuzzy and the Nonlinear Method

In this section, we show a comparison of the results obtained by the proposed fuzzy algorithm and the nonlinear control method proposed in chapter 4. Tables 5.7 and 5.8 show the results obtained by these methods with the trajectory following errors compared for tests 1, 2 and 3 in which $x_{err-total}$, $y_{err-total}$, α_{total} and MSE can be calculated by:

$$x_{err-total} = \frac{\sum_{n=0}^{f}|x_{err(n)}|}{f} \quad , \quad y_{err-total} = \frac{\sum_{n=0}^{f}|y_{err(n)}|}{f} \tag{5.17}$$

$$\psi_{err-total} = \frac{\sum_{n=0}^{f}|\alpha_{(n)}|}{f}$$

$$MSE = \frac{\sum_{n=0}^{f}\sqrt{x_{err(n)}^2 + y_{err(n)}^2 + \alpha_{(n)}^2}}{f}$$

MSE is defined as the mean square error. As fuzzy logic mimics the way the human brain solves problems by grouping like things together, this approach simplifies the computations needed by the algorithm controller. Tables 5.7 and 5.8 show that as the robots travel on their desired trajectories, the fuzzy controller produces a smaller mean square error compared to the nonlinear method thus allowing the robots to follow their trajectories more precisely.

However, when the initial positions of the robots are not on the desired trajectories (test 2), the nonlinear control is faster at guiding them to their desired trajectories.

Table 5.7 Comparisons between the nonlinear and fuzzy controller for test 1

	Nonlinear Control				**Fuzzy Control**			
Total errors	x_{err}	y_{err}	α	MSE	x_{err}	y_{err}	α	MSE
Robot1	-.080	.00013	-.0002	.0891	.0216	.0015	-3.5e-005	.0325
Robot2	-.116	-.0032	.0057	.1345	.2408	-.0016	-.0054	.0493
Robot3	-.150	-.0025	.0055	.1589	.2842	.0018	.27e-005	.0056

Table 5.8 Comparisons between the nonlinear and fuzzy controller for test 2

	Nonlinear Control				**Fuzzy Control**			
Total errors	x_{err}	y_{err}	α	MSE	x_{err}	y_{err}	α	MSE
Robot1	-.0993	.2920	-.2728	.4981	0.040	-0.531	0.060	0.6858
Robot2	-.1979	.0334	-.0527	.2309	.110	-0.201	.091	0.2619
Robot3	-.0084	.0749	-.1106	.2962	-0.437	0.592	-0.138	.8985

5.5 Conclusion

In this chapter, a fuzzy logic controller is proposed to facilitate the smooth and efficient control for the trajectory following and cooperation of a group of mobile robots. A two-level hierarchical controller is presented to facilitate the best performance for the control algorithm. A high-level fuzzy controller is used to perform the tasks of trajectory following, localization and cooperation. This cooperation method, derived from fuzzy logic and PID, empowers the robots to move, follow and coordinate trajectories in different MMR group formations. The outputs of the fuzzy module controller are the linear and angular velocities of the individual robots. The experimental results obtained from three different trajectory scenarios demonstrate the effectiveness of the proposed algorithm.

CHAPTER 6

COORDINATION FOR MULTI MOBILE ROBOTS IN UNKNOWN ENVIRONMENTS

In this chapter, an algorithm is proposed for the behavioral control of MMR's using fuzzy logic techniques to create coordination, crash avoidance and obstacle avoidance among a MMR group. The design of this fuzzy logic system exhibits three behaviors: firstly trajectory tracking (covered in chapter 2), then group coordination (covered in chapter 5), and now in this chapter, collision avoidance between the robots and static obstacles or between the robots themselves (robots consider each other as dynamic obstacles). The coordination of the robot group is designed so that they finish their trajectories within same time duration regardless of the number of crash avoidance incidences between them, the length of the individual robot trajectory or the number of obstacles confronted along their paths.

6.1 Intelligent Crash Avoidance between Multi Mobile Robots

A crash avoidance behavior is designed so that the robots within the group are assigned a priority status (Mehrjerdi, Saad et Ghommam, 2010d). If there is a chance of an imminent collision between robots, this behavior selects the one(s) that must stop to avoid a crash while allowing the priority robot to move forward or change its desired trajectory. Once the priority robot moves away and the danger of a crash has past the other robots are then allowed to continue moving forward in the group formation. The core of designed algorithm uses fuzzy logic which imitates the way the human brain processes and responds to problems. In this section, the fuzzy control looks only at those collisions which may occur between robots where they consider each other as dynamic obstacles which can potentially cause a crash within the MMR group (Mehrjerdi, Saad et Ghommam, 2010b).

Here, we investigate the navigation of multiple mobile robots working in a two-dimensional environment using a crash avoidance algorithm. To do this, a three level hierarchical architecture control based on the fuzzy model and PID is used to facilitate multiple mobile

robots to work together in different formations. This section focuses on achieving crash avoidance between the robots while they are working in a group formation to reach their individual targets. A set of linguistic fuzzy rules are extended to implement expert knowledge under a variety of situations. The output of the fuzzy controller determines the linear and angular velocities of the individual robots, where each robot has its own goal or individual trajectory to be navigated while also maintaining a group formation with the other robots.

A robust fuzzy coordination algorithm is developed so that when robots lose their coordination, they can resume it once the problem is solved. For example, if there is the chance of an imminent crash among any of the robots forcing them to temporarily abandon their place in the formation, the coordination algorithm will then re-acquire their correct positions once the crash has been overcome. Within a virtual model, this architecture achieves different group formations by the robots, while facilitating trajectory planning, localization, coordination and crash avoidance behavior. The distance between the robots is crucial in the mapping of the direction of robot movement. The direction angle that each robot moves is determined not in isolation, but by the desired trajectories and formations of all the robots.

If there is the chance of an impending crash among any of the robots, the algorithm works out which possible virtual trajectory is best for each robot in the formation in order to avoid these collisions. Control laws adjust the linear and angular velocities of each robot involved in a possible collision therefore altering their trajectories and avoiding a robot crash.

When there are imminent crash points on the trajectories, the robots must stop or change their designated route according to a priority status, and either of these actions will cause the robots lose their coordination with the rest of the group. The fuzzy algorithm allows for this temporary loss of coordination, and instructs the robots to recalculate their next move whilst avoiding a collision with any dynamic obstacle (in this case another robot). Once a robot has stopped or changed its desired trajectory, and there is now no danger of a collision, it is allowed to resume its trajectory and continue with the group formation. This cycle of

allowing a loss of coordination during an imminent collision, and then re-establishing it once the danger has past, allows for a smooth and efficient crash avoidance behavior.

The distance between robots is calculated by the following equation and is limited to:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad , \; i \neq j \; , Crash\ Zone : d_{ij}(t) \leq d_{ij(min)} \tag{6.1}$$

where $d_{ij(min)}$ is fixed according to the physical dimension of robots. Figure 6.1 shows the general model of mobile robot and trajectories which are divided into discrete segments.

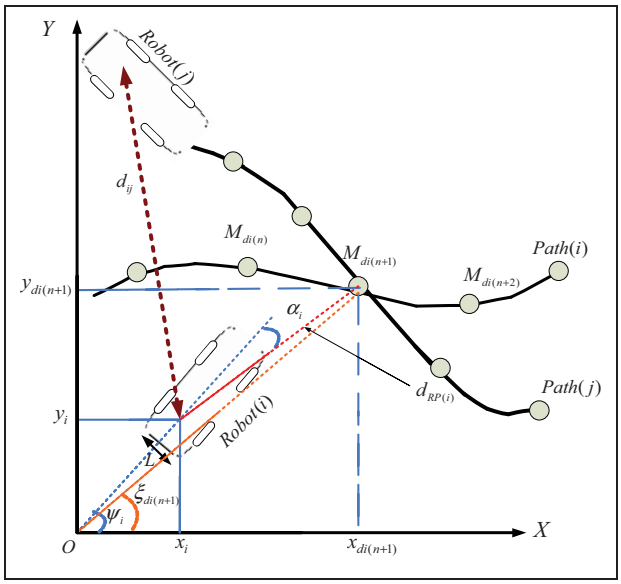


Figure 6.1 Mobile robot with two actuated wheels and intersected trajectories.

6.1.1 Fuzzy Coordination and Crash Avoidance Algorithm

Figure 6.2 illustrates the block diagram used for the control, coordination and crash avoidance for a group of mobile robots. The controller has three hierarchical levels consisting of a low PID level and two higher level fuzzy controllers. The first of the higher level controller is designed to instruct the robots to follow their generated trajectory coordinates, while the second higher level controller is designed to enable the robots to avoid collisions (crash avoidance). Posture sensors are used to localize the robots in the group and direct them

to their trajectories in a desired formation. Figure 6.3 shows the fuzzy system structure and notates the desired inputs and outputs.

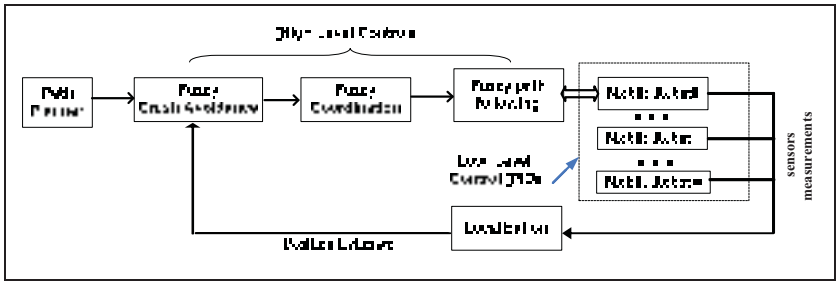


Figure 6.2 Fuzzy control structure.

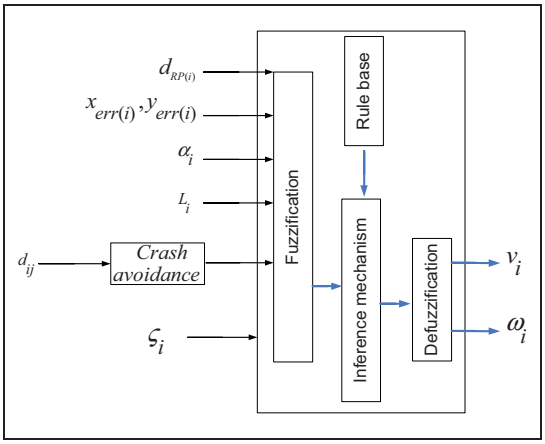


Figure 6.3 Infrastructure of multi robot control and coordination.

To achieve our goal of robot group coordination, we propose that each robot in the group has a local controller for trajectory following and coordination. The prerequisites required to achieve coordination across the whole robot group is that they are required to move along their trajectories while maintaining a desired inter-robot formation pattern, and they must also reach their final goals at the same time regardless of their trajectory lengths, number of crash points or their initial positions. To avoid a collision between any of the robots when they come into close proximity to each other, a control law for the linear and angular velocities must be assigned to the robots that are in danger of a collision. To solve this crash

avoidance problem, we propose a control law for the correction of the linear and angular velocities as:

$$v_{crash}(t) = v_{di}(t) + \tilde{v}_{di}(t) \tag{6.2}$$
$$\omega_{crash}(t) = \omega_{di}(t) + \tilde{\omega}_{di}(t)$$

where $\tilde{v}_{di}$ and $\tilde{\omega}_{di}$ are the correction terms.

This requires that for any trajectory, the robots must reach their final goals in the same time duration and if they lose coordination for any period of time, they are able to resume it once the crash problem is resolved such that for all different trajectories $\left\| P_i - M_{di(n)} \right\| \to 0$.

6.1.2 Fuzzy Coordination and the Crash Avoidance

The form of the control law equation for trajectory tracking and coordination is as follows:

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} f_1(d_{RP(i)}, d_{ij}, \alpha_i, x_{err(i)}, y_{err(i)}, L_i, \varsigma_i) \\ f_2(d_{RP(i)}, d_{ij}, \alpha_i, x_{err(i)}, y_{err(i)}, L_i, \varsigma_i) \end{bmatrix} \tag{6.3}$$

The functions f_1 and f_2 are the control laws of a Sugeno type fuzzy controller which can take both fuzzy inputs and outputs.

To solve the problem of a crash between robots, a fuzzy control law for linear and angular velocities of robots is considered with respect to the priority status of the robots in the group, with the crash avoidance behavior designed on a priority basis. The robots are given a priority number rating with the lowest number robot always having precedence over the higher numbers. If any of the robot's sensors indicate that a collision between them is imminent $\boldsymbol{d_{ij}(t) \leq d_{ij(min)}}$, the robots with the higher numbers must 'stop' until the priority robot has moved away from the collision area. The fuzzy control algorithm then recalibrates their trajectories so they reach their target point within specific time duration.

Therefore, a control law is enacted by the fuzzy controller, and η_i will be added to α_i.

$$if \;\; d_{ij}(t) \leq d_{ij(min)} \;\; \Rightarrow \alpha_i(new) = \lambda_i + \alpha_i + \eta_i \tag{6.4}$$

If the robots are at a significant distance from each other then this angle will be zero, but as they move closer together this angle increases. As the robots travel along their predefined trajectories, if they move inside the minimum danger zone (represented by the circles on Figure 6.6), then the controller considers that a crash is imminent and the lesser priority robot will stop. This allows the higher priority robot to either continue (if there is now no danger of a crash) or create a direction change (if a crash scenario is considered unavoidable). This trajectory change is calculated with respect to this new angle of η_i being added, and after some distance the robot resumes the original predefined trajectory. (See trajectory change direction on Figure 6.6). Membership functions of this input are shown in the Figure 6.4. Figure 6.5 shows the changes of η_i related to changes of distance between robots (d_{ij}),

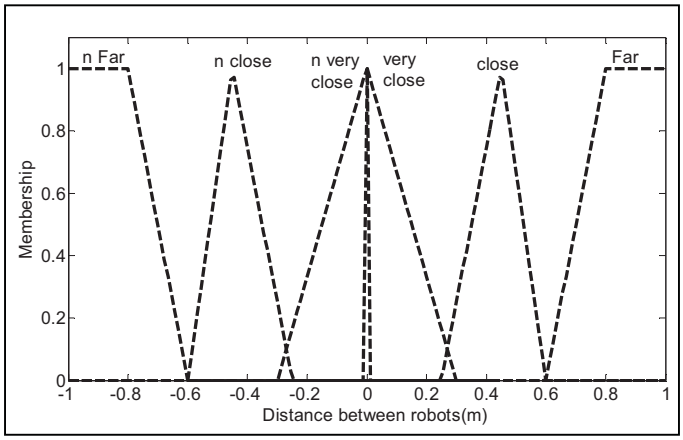


Figure 6.4 Membership function of $\mathbf{d_{ij}}$.

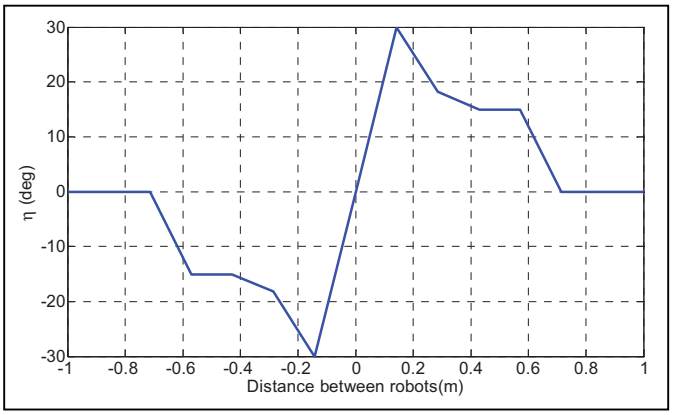


Figure 6.5 η_i Obtained by fuzzy controller and $\mathbf{d_{ij}}$.

Figure 6.6 shows the scenario in which robots i and j move toward each other. The circles around the robots show the minimum distance that is allowed for the robots to get close to each other before they are considered to be in danger of a crash. As can be seen in this figure, robot j $(j > i)$ will stop and robot i will change its desired trajectory to avoid a crash with the other robot.

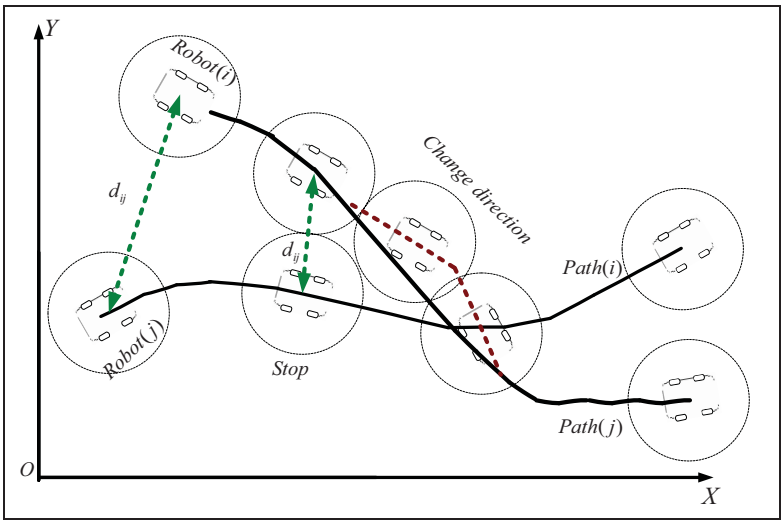


Figure 6.6 Crash avoidance between robots.

Equation (6.5) represents new linear and angular velocities of robots i and j .

$$if\ \ d_{ij}(t) \le d_{ij(min)}\,, x_j < x_i \quad \Rightarrow \begin{cases} v_i^*(n+1) = v_i(n+1) \\ \omega_i^*(n+1) = \omega_i(n+1) \pm \widetilde{\omega}_{di} \\ v_j^*(n+1) = 0 \\ \omega_j^*(n+1) = \omega_j(n) \end{cases} \tag{6.5}$$

Figure 6.7 represents the trajectory tracking, coordination and crash avoidance algorithm.

6.1.3 Experimental Results

Figure 6.8 shows the design of the experimental set up for the control, trajectory planning, coordination and crash avoidance for a team of mobile robots. The robot trajectories are

generated by the trajectory generator and the robot positions are measured by sensors, while trajectory following, coordination and crash avoidance are performed by the fuzzy controller. To evaluate the integrity of the proposed algorithm, experimental tests were performed using three mobile robots employing different trajectories.

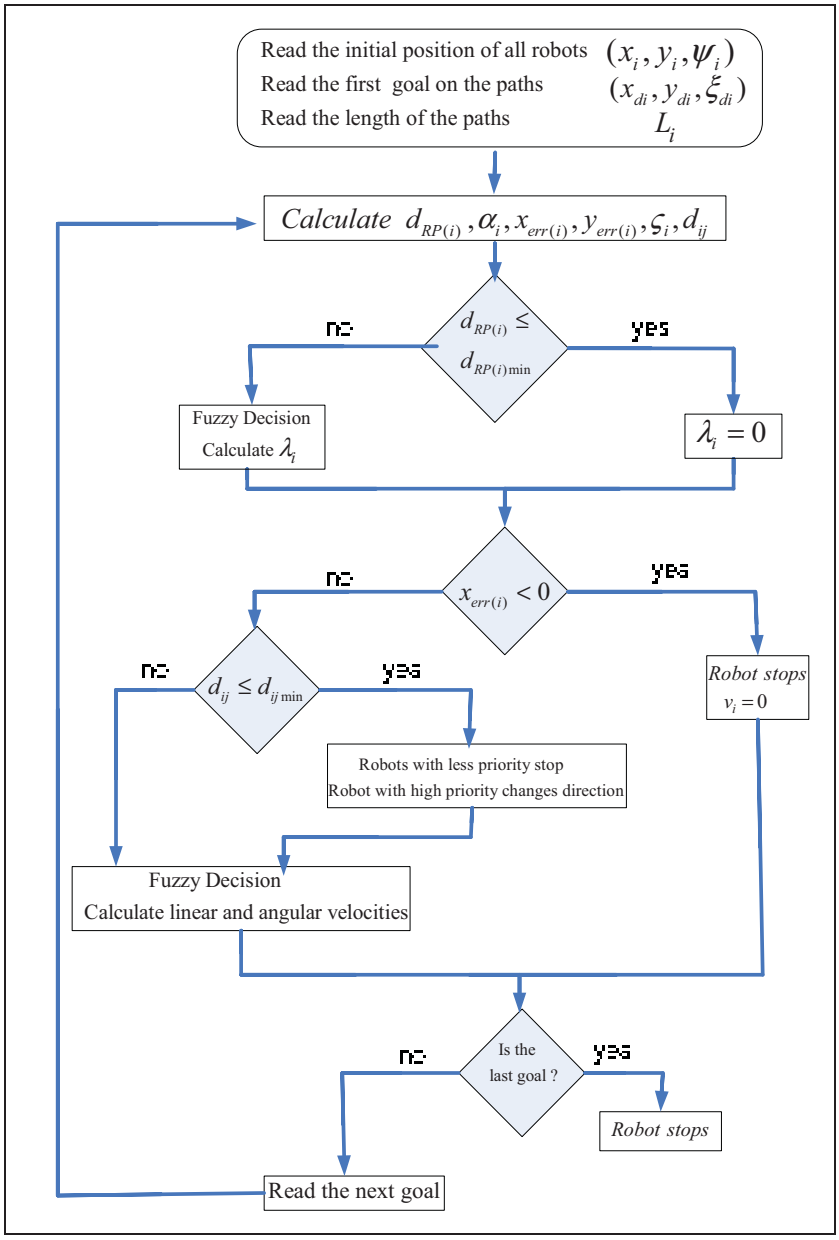


Figure 6.7 Flow chart of trajectory tracking, coordination and crash avoidance.

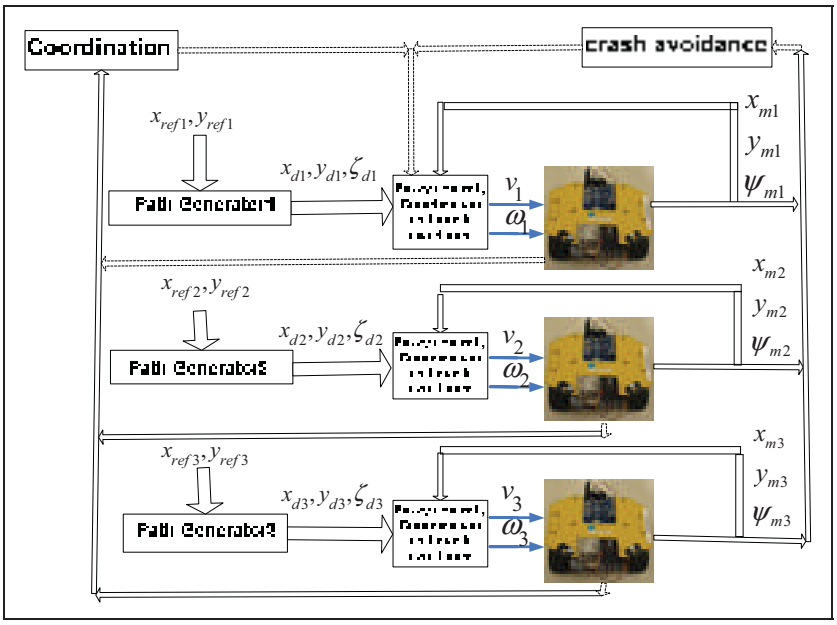


Figure 6.8 The general view of experimental setup.

6.1.3.1 Experimental Tests

In this section, we discuss the results of using two different robots formations. These are:

1) trajectories in which robots 1 and 2, then robots 2 and 3 arrive at the same points;
2) trajectories in which robots 1 and 2, then robots 1 and 3 and finally 2 and 3 arrive at the same points.

In the first formation experimental test, the trajectories being considered are robots 1 and 2, and then robots 2 and 3 arriving at the same sampling points at the same time. In this scenario, the length of trajectories and the initial positions of robots are:

$[x_1(t_0), y_1(t_0)\psi_1(t_0)]^T = [0,1,0]^T, [x_2(t_0), y_2(t_0)\psi_2(t_0)]^T = [0,0,0]^T, [x_3(t_0), y_3(t_0)\psi_3(t_0)]^T = [0,-1,0]^T$
$L_1 = 8.14\ m\ , L_2 = 8.8m, L_3 = 9.41\ m$

Figure 6.9 depicts the reference and the actual robots trajectories in this scenario. Figure 6.10 shows the detail of the avoidance crash points using robots 1 & 2 and 2 & 3. Figure 6.10(a)

shows that when robots 1 and 2 move close to each other, robot 2 stops and robot 1 changes its trajectory. Figure 6.10(b) shows that when robot 3 and 1 move toward each other, robot 3 stops and robot 2 changes its trajectory to avoid a crash. Figure 6.11 shows distance between all the robots in the formation when they move to follow their individual trajectories.

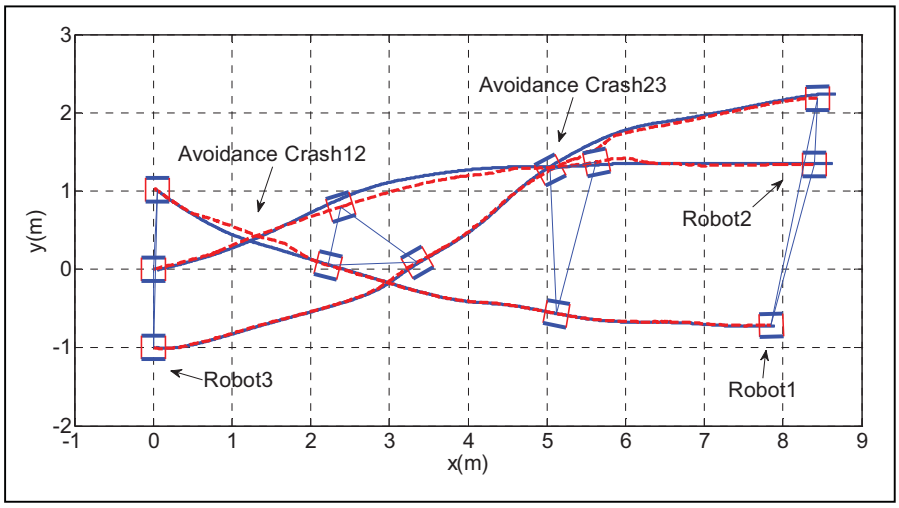


Figure 6.9 Reference and real robots' trajectories.

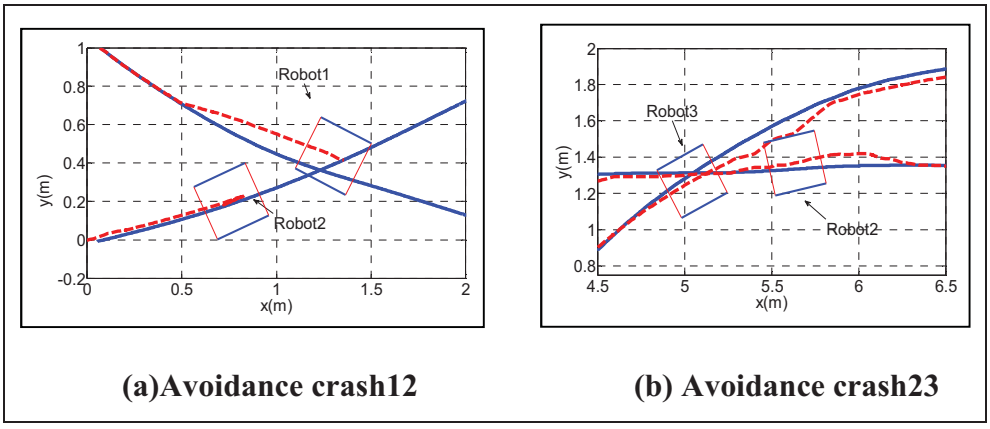


Figure 6.10 Avoidance crash points between robots.

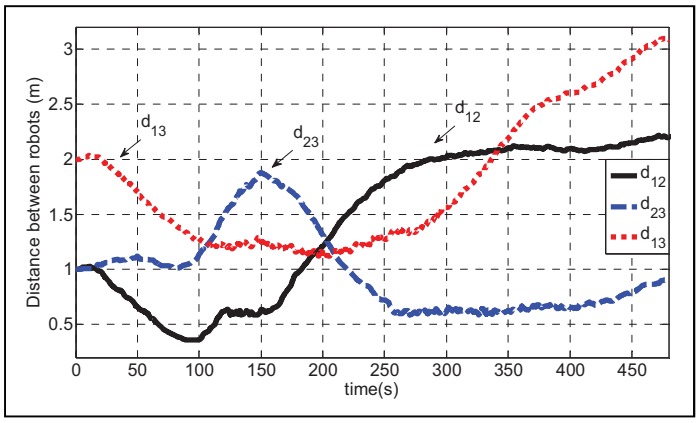
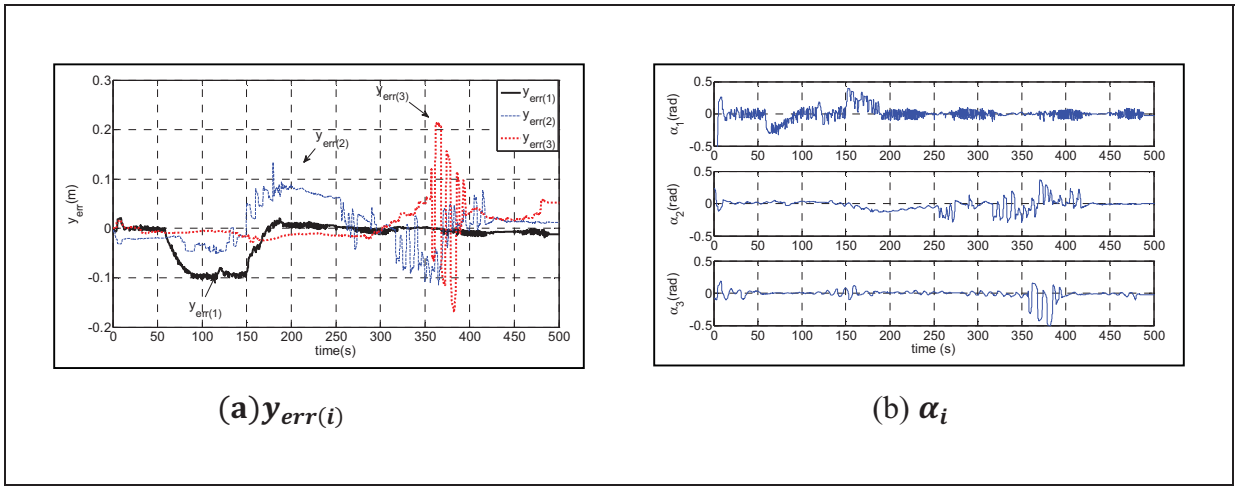


Figure 6.11 Distance between the robots d_{ij}.

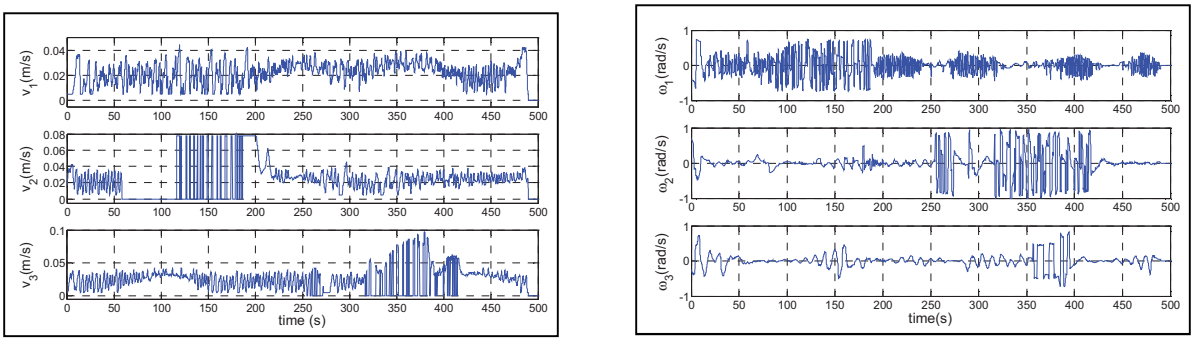
The trajectory tracking errors y_{err} and α are shown in Figure 6.12. As can be seen in these figures, when the robots are close to each other the errors increase, and when robots are far from each other these errors decrease to zero.



(a) $y_{err(i)}$ (b) α_i

Figure 6.12 Trajectory tracking errors.

The linear and angular velocities are plotted in Figure 6.13. These figures show that the velocities of robots 2 and 3 decrease to zero at some points to avoid a crash.



(a) Linear (b) Angular

Figure 6.13 Velocity of robots.

In the second formation experimental test, the trajectories being considered are when all robots will arrive at the same point at the same time. In this scenario, there are possibilities of three crashes. The trajectory lengths and initial positions of robots are defined as:

$[x_1(t_0), y_1(t_0)\psi_1(t_0)]^T = [0,2,0]^T, [x_2(t_0), y_2(t_0)\psi_2(t_0)]^T = [-1,0,0]^T, [x_3(t_0), y_3(t_0)\psi_3(t_0)]^T = [0,-2,0]^T$
$L_1 = 9.77\ m\ , L_2 = 8.46\ m, L_3 = 9.77\ m$

Figure 6.14 depicts the reference and actual robots trajectories in this scenario. This figure shows that to keep formation and coordination between the robots and avoid crashes, the robots must change their predefined trajectories at some points. Figure 6.15 shows the details of the avoidance crash points using robots 1 & 2 and 3. This figure shows that when the robots move close to each other, robot 2 and 3 stop and robot 1 changes its trajectory to avoid a crash.

As robot 1 moves away from the area of crash, robot 2 begins to move and changes its desired trajectory to avoid a crash with robot 3. When robot 2 moves away from the area of the crash, robot 3 which has been prevented from moving begins to travel on its desired trajectory. As can be seen in these figures, the fuzzy algorithm allows for a temporary loss of coordination and instructs the robots to recalculate their next move whilst avoiding a collision with any another robot. Once a robot has stopped or changed its desired trajectory, and there is now no danger of a collision, it is allowed to resume its trajectory and continue

with the group formation. Figure 6.16 shows the distance between all the robots in the formation when they move to follow their individual trajectories.

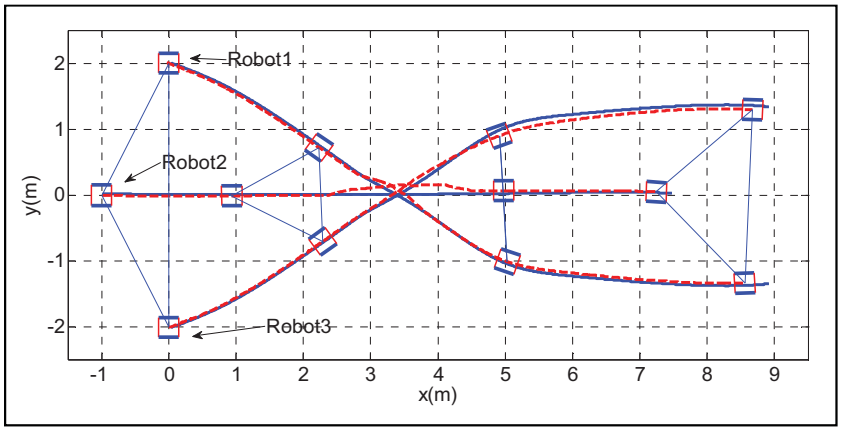


Figure 6.14 Reference and real robots' trajectories.

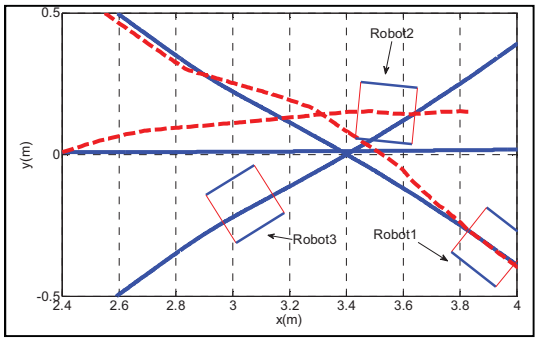


Figure 6.15 Crash avoidance between robots with a change of trajectories.

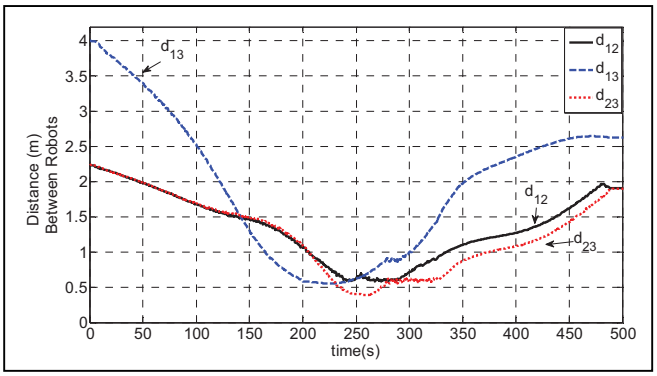
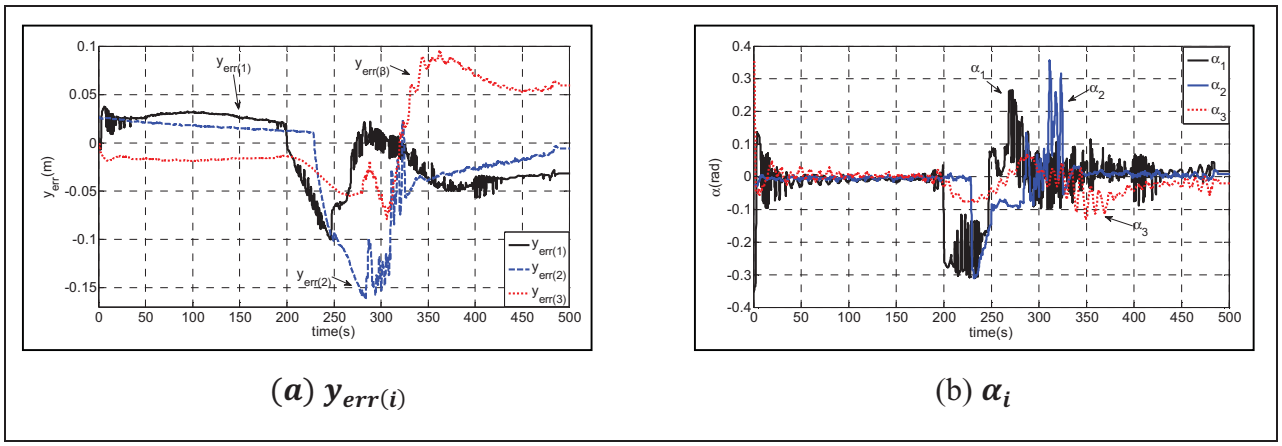


Figure 6.16 Distance between robots d_{ij}.

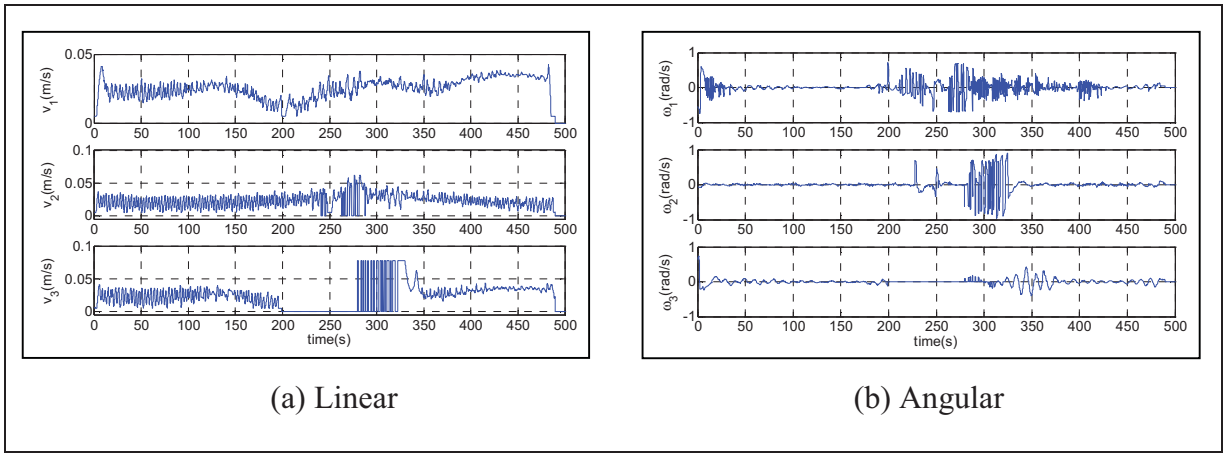
The trajectory tracking errors y_{err} and α are shown in Figure 6.17. As can be seen in these figures, when the robots are close to each other these errors increase and when robots are far from each other these errors decrease to zero.



(a) $y_{err(i)}$ (b) α_i

Figure 6.17 Trajectory tracking errors.

The linear and angular velocities are plotted in Figure 6.18. These figures show that the velocities of robots 2 and 3 decrease to zero at some points to avoid a crash.



(a) Linear (b) Angular

Figure 6.18 Velocity of robots.

6.2 Fuzzy Obstacle Avoidance and the Coordination Algorithm

In this section, the design and implementation of an intelligent coordination algorithm is introduced for a team of MMR's that are confronted with obstacles in an unknown environment (Mehrjerdi, Saad et Ghommam, 2010e). The robots in the group have inbuilt sensors to observe their surrounding environments. The developed algorithm bases its characteristics on the efficiency of fuzzy logic which enables it to coordinate the MMR group and follow their trajectories. Simultaneously, it processes the environmental data from the robot sensors to monitor the environment in order to locate and avoid any obstacles - including other robots in the group. When an obstacle is detected by any of the robot's inbuilt sensors, they direct that robot to move around the obstacle by either changing its velocity and / or direction. As well as obstacle avoidance, the controllers work to make the MMR group arrive concurrently at their target points by adjusting each of the robot's velocities as they move along their desired trajectories. This means that the group will arrive within the same time duration, regardless of the length of each individual trajectory or number of obstacles that confront each robot.

The experimental results of three mobile robots traveling on different trajectories in unknown environments are presented to show the accuracy of obtaining control, coordination and obstacle avoidance by using the designed fuzzy algorithm.

The challenge for this section is to modify the algorithmic model so that it still creates an intelligent coordinated behavior in situations where MMR's are placed in environments containing obstacles on their trajectories. To achieve the coordination and trajectory processing for the MMR group, it is necessary to have position and sonar sensor feedbacks which measures the distance between the robots and any obstacle that may appear on their trajectories. The designed intelligent algorithm must synchronize the robots within the group to execute common or diverse tasks by performing several simultaneous functions. It must have the ability to instruct all robots in the group to discern their environment, avoid a collision, track their trajectories, alter their directions and velocities, and transmit their

locations and environmental data to other robots in the group formation. The choice of using a fuzzy coordination algorithm for robot control is that it gives both flexibility and adaptability to the individual robots when maintaining group coordination. They must be able to dynamically change their velocity or desired trajectory when confronting obstacles, or when they lose coordination with the rest of the group. The algorithmic model must precisely guarantee that all the robots within the group reach their target points both individually and in group formation.

The robots are equipped with sonar sensors which can measure the relative position of any object (other robots or environmental obstacles) if and only if, the object is within a given distance. The model of the mobile robot with inbuilt lateral and longitudinal sonar sensors can be seen in Figure 6.19. In this section, we are particularly interested in monitoring obstacles that are of unknown shape and size and possibly moving dynamically in time (robots can be considered as a dynamic obstacle for each other).

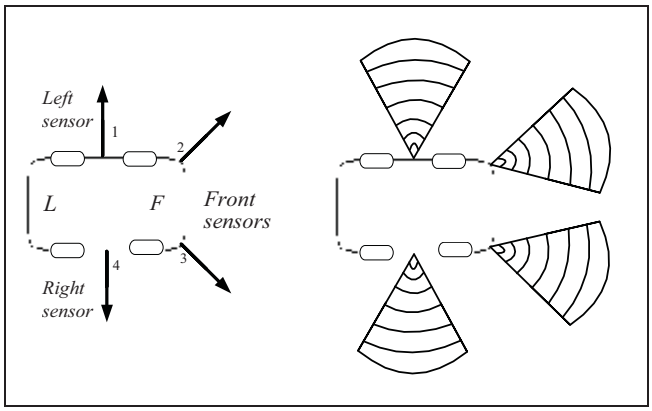


Figure 6.19 Model of mobile robot with inbuilt sonar sensors.

Figure 6.20 shows two mobile robots in an unknown environment in the presence of an obstacle and displaying discretized trajectories. In this figure, $\mathrm{ro}_{z(i)}$ shows the distance between the robots and obstacles where $z = 1, \ldots, 4$ represents the number of sonar sensors, and in order to avoid a collision then: $\mathrm{ro}_{z(i)} \leq \mathrm{ro}_{z(\min)}$.

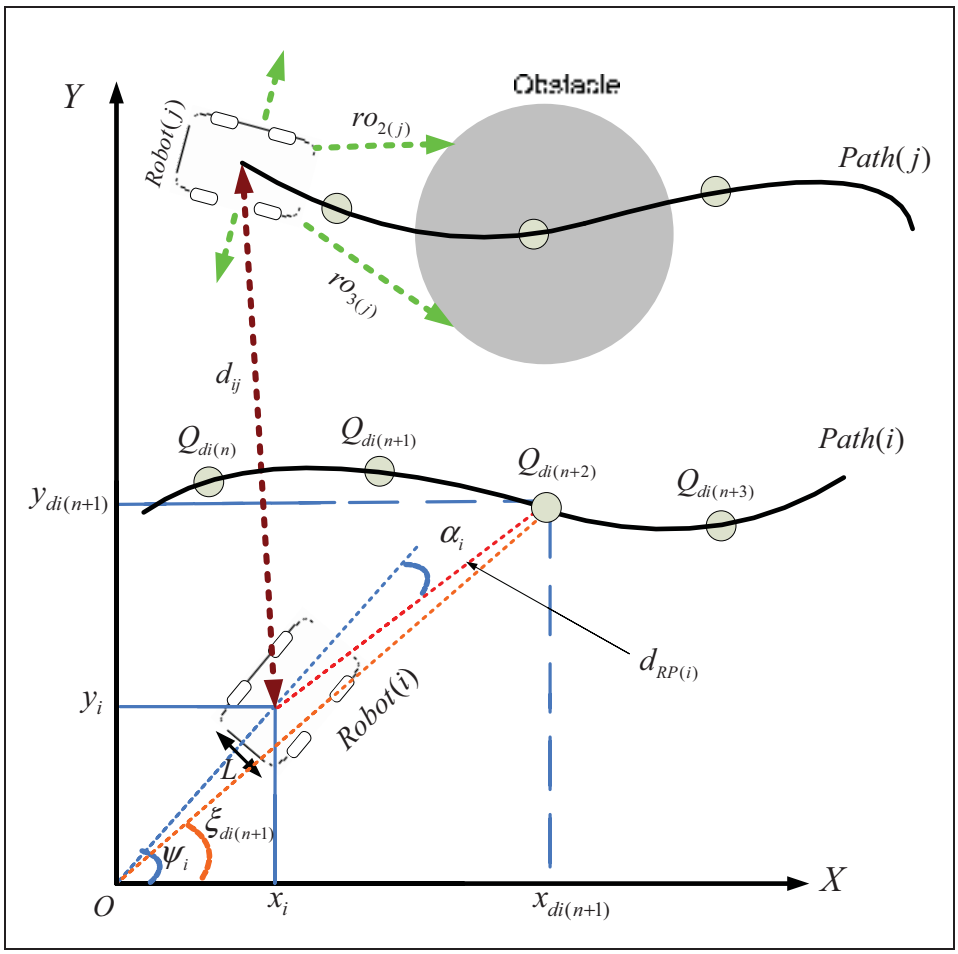


Figure 6.20 Model of mobile robots in an unknown environment.

Figure 6.21 illustrates the block diagram used to control and synchronize a group of mobile robots traveling in unknown environments. In this section, fuzzy logic is the prime controller and therefore performs the tasks of trajectory tracking, obstacle avoidance and group coordination. The secondary controller is a PID which ensures that the motor velocity to the robot's left and right front wheels is always accurate. The direction angle of each robot is determined by the desired trajectories, the formations of the MMR group and the presence of obstacles in an unknown environment.

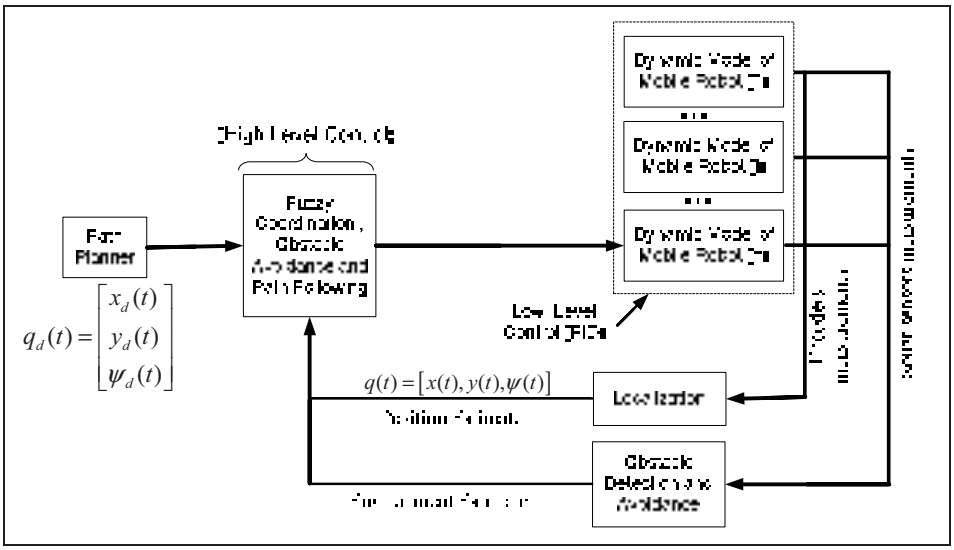


Figure 6.21 Infrastructure of multi robot control and cooperation.

Figure 6.22 shows the fuzzy inference system structure with desired inputs and outputs. We add $ro_{z(i)}$ as a new input to the fuzzy algorithm, and the output of the fuzzy controller determines the linear and angular velocities of the individual robots.

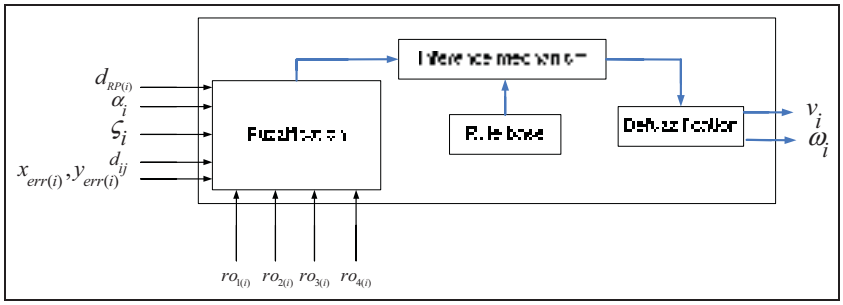


Figure 6.19 Block diagram of the fuzzy inference system.

One set of inputs for the fuzzy system is data coming from sonar sensors mounted on the robots and includes both the distance and the angle to an obstacle that appears on the trajectory of any of the robots. The measurements of the robot's position and velocity are the second input to the algorithm and this information is combined to calculate the most appropriate trajectory that the robot should pursue to avoid the obstacle as well as stay within the group formation. The fuzzy rules applied to the robots are defined by the kinematic

limitations which are bounded by both the linear and angular velocities and the length and curvature of the individual trajectories. The designed fuzzy model employs three behaviors being trajectory tracking, group coordination and obstacle avoidance.
The form of the control law equation for trajectory tracking and cooperation is as follows:

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} f_1(d_{RP(i)}, d_{ij}, \alpha_i, x_{err(i)}, y_{err(i)}, ro_{z(i)}, \varsigma_i) \\ f_2(d_{RP(i)}, d_{ij}, \alpha_i, x_{err(i)}, y_{err(i)}, ro_{z(i)}, \varsigma_i) \end{bmatrix} \tag{6.6}$$

The functions f_1 and f_2 are the control laws of a Sugeno type fuzzy controller which can take both fuzzy inputs and outputs. We propose to solve the problem of propelling a robot along a continuous desired trajectory in an unknown environment by observing obstacles on the trajectories and considering the robot moving between discontinuous sampling points while avoiding obstacles and other robots in the environment. The trajectories are modeled by a fifth order polynomial and divided into segments for analysis with the same sampling point numbers regardless of the shape, curvature or trajectory length.

The robots are fitted with four sonar sensors to detect and avoid obstacles in the environment. These consist of two primary sensors located on the front of the robot giving a wide angle view ahead and two secondary sensors, one on each side giving lateral views. The newly designed algorithm has two modes of operation that it can quickly switch between depending on whether obstacles are present or not in the environment. These two modes of operation are termed ‘fuzzy trajectory tracking and coordination’, and ‘fuzzy obstacle detection and avoidance’.

The fuzzy trajectory tracking and coordination mode is only required when there are no obstacles observed in the local environment by any of the robots’ sonar sensors. This first mode of operation is only required to perform velocity and direction changes for the robots due to their differing trajectory lengths, enabling the group to arrive at their individual destinations within the same time.

The second mode, fuzzy obstacle detection and avoidance, is enacted once any of the robots’ sensors detect either a static or dynamic obstacle (another robot is also considered an

obstacle). In this situation, the velocity and direction changes are adjusted by a newer version of the algorithmic processing that uses the sonar information to move around the obstacles. Once the robot has cleared the offending obstacle, the algorithm reverts to the previous version of algorithmic processing to maintain group coordination and guide them along their designated trajectories.

To solve the problem of obstacle avoidance, a fuzzy control law for the angular velocity of robots is considered. If any of the robot's sensors indicate that a collision with an obstacle is imminent $\mathrm{ro}_{\mathrm{z(i)}} \leq \mathrm{ro}_{\mathrm{z(min)}}$, the fuzzy control algorithm recalibrates a new trajectory to move around the obstacle. Therefore, a control law is enacted by the fuzzy controller, and Θ_i will be added to α_i.

$$if\ ro_{z(i)} \leq ro_{z(min)} \Rightarrow \alpha_i^{**} = \alpha_i + \Theta_i \tag{6.7}$$

If the robot is at significant distance from the obstacle then Θ_i will be zero, but as they move closer then this angle increases. This trajectory change is calculated with respect to this new angle of Θ_i being added, and after moving around the obstacle the robot resumes the original predefined trajectory. Membership functions of the front sonar sensor inputs are shown in the Figure 6.23. Figure 6.24 shows the changes of Θ_i related to changes of the distance between the robot and obstacles observed by the front sonar sensors ($ro_{z(i)}$),

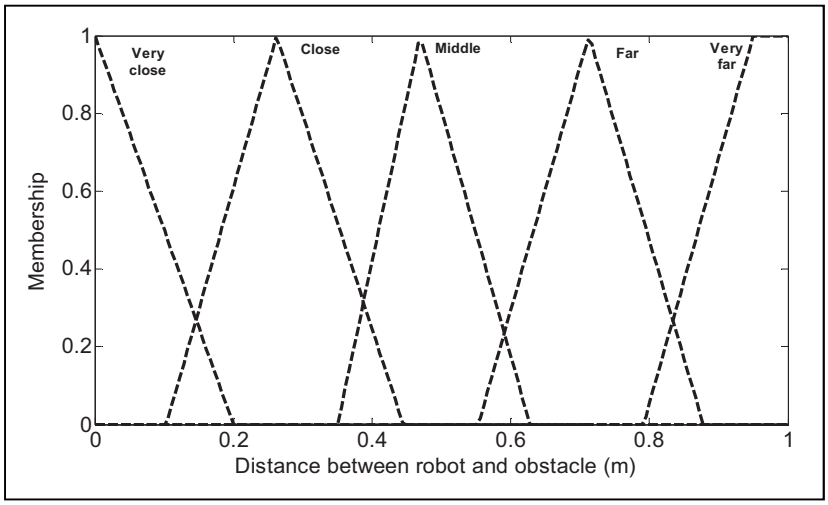


Figure 6.20 Membership function of $\mathrm{ro}_{\mathrm{z(i)}}$.

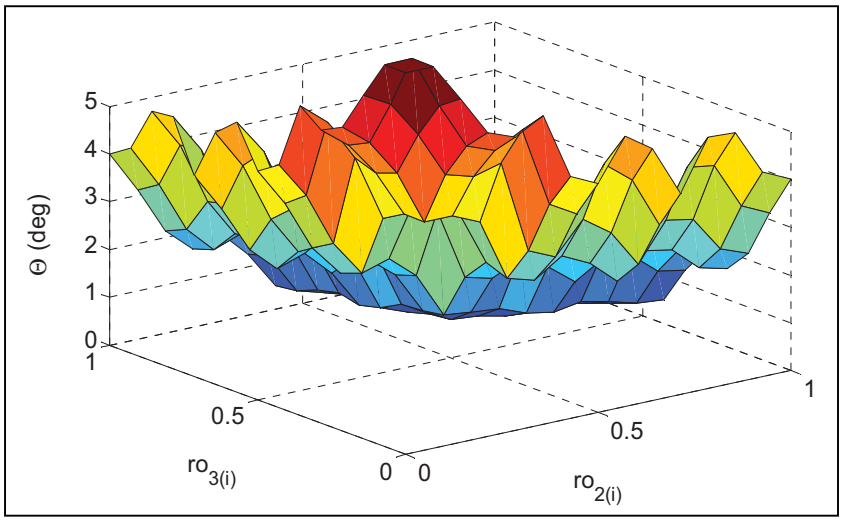


Figure 6.21 Θ_i Obtained by fuzzy controller and $ro_{z(i)}$.

In this section we consider four general cases in which robots may confront obstacles in unknown environments and where fuzzy obstacle and coordination rules should be considered.

A) Case study 1:

If any of the robots in the group see an obstacle on its desired trajectory then it must recalculate another trajectory to avoid a collision as can be seen in Figure 6.25. The robot does this by detecting the obstacle with its sonar sensors and uses the fuzzy logic algorithm to determine the best way to move around it. Once the robot has successfully passed the obstacle and sees that its trajectory is clear, then it resumes its trajectory to arrive at the target point as part of the group formation. Meanwhile, the other robots will continue until this robot passes the obstacle and returns to its desired trajectory.

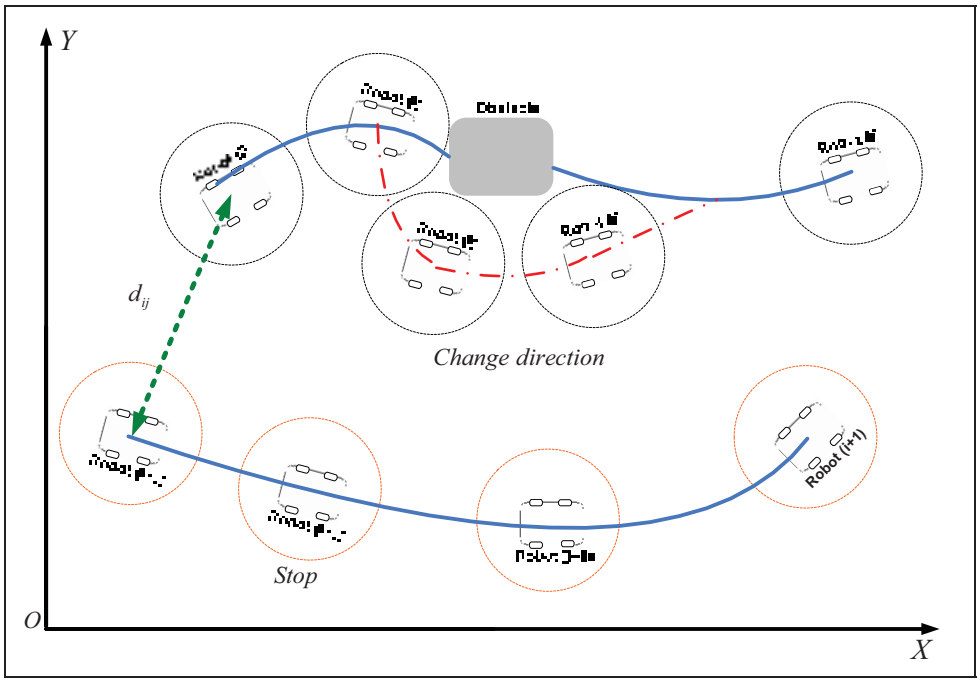


Figure 6.22 Case study 1: Mobile robot observes an obstacle.

B) Case study 2:

In this situation, a robot confronts an obstacle observed by its sonar sensors. However, when the robot tries to change its desired trajectory and direction to avoid this immediate collision, there appears another robot on its new trajectory and so there is also a chance of a crash with this robot. This situation can be seen in the Figure 6.26. In this scenario to keep the group coordination, the robot with less priority will stop to allow the robot confronted with the obstacle to pass around that obstacle. If there is no danger of a crash, the robots keep moving on their desired trajectories.

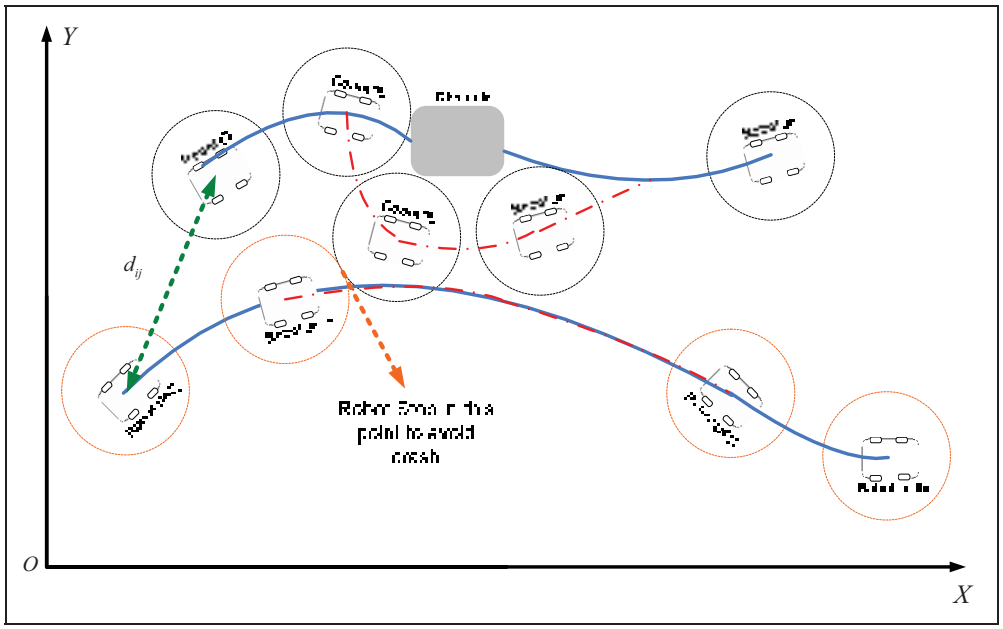


Figure 6.23 Case study 2: Mobile robot observes an obstacle and there is a chance for a crash with other robots.

C) Case study 3:

In this scenario, more than one robot in the group observes obstacles on their desired trajectories with their sonar sensors. However, when they try to change their desired trajectories and direction to avoid these obstacles, there is also a chance of collision between these robots on their new trajectories. This situation can be seen in the Figure 6.27, and in this scenario in order to keep the group coordination, the robot with less priority will stop until the robot with a higher priority (in this case robot i) can successfully pass around the obstacle. If there is no danger of a crash between the robots, then they will continue on their desired trajectories.

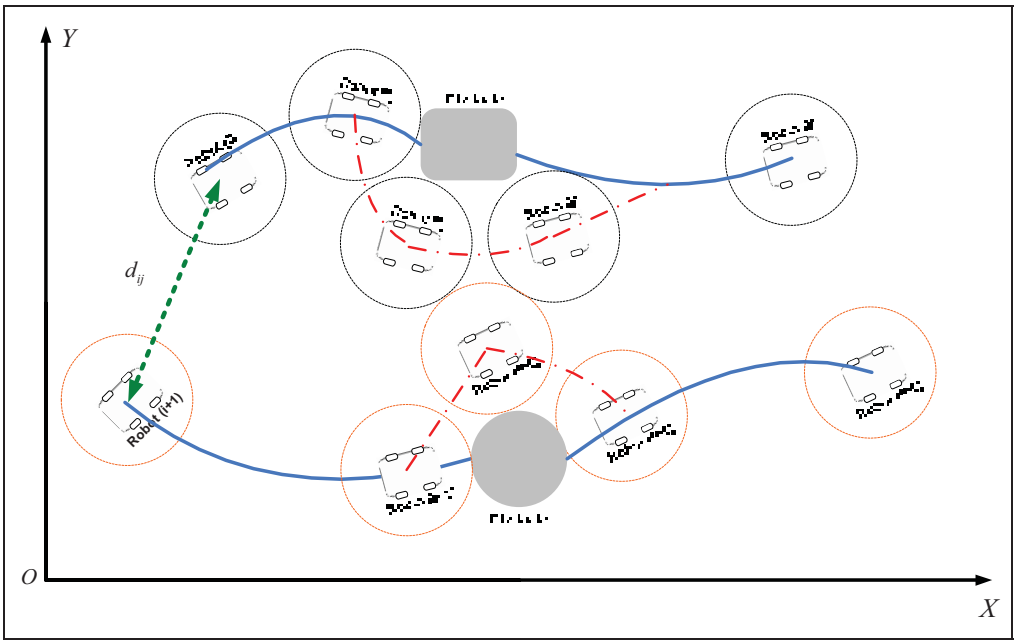


Figure 6.24 Case study 3: Mobile robots observe obstacles and there is a chance for a crash with other robots.

D) Case study 4:

In this scenario, a robot observes an obstacle on its desired trajectories by its sonar sensors. However, due to the shape or the position of the obstacle this robot becomes stuck in local minimum and cannot pass around the obstacle in the desired time allocated for it to do so. A short time-delay is given for a robot which is losing its trajectory coordination with the other robots in the group. If it cannot pass an obstacle and is completely stuck in local minimum, then once this time duration has elapsed, its coordination to the other robots will be lost then they will continue on their trajectories without it. This situation is shown in the Figure 6.28, and the stuck robot is now out of the group coordination.

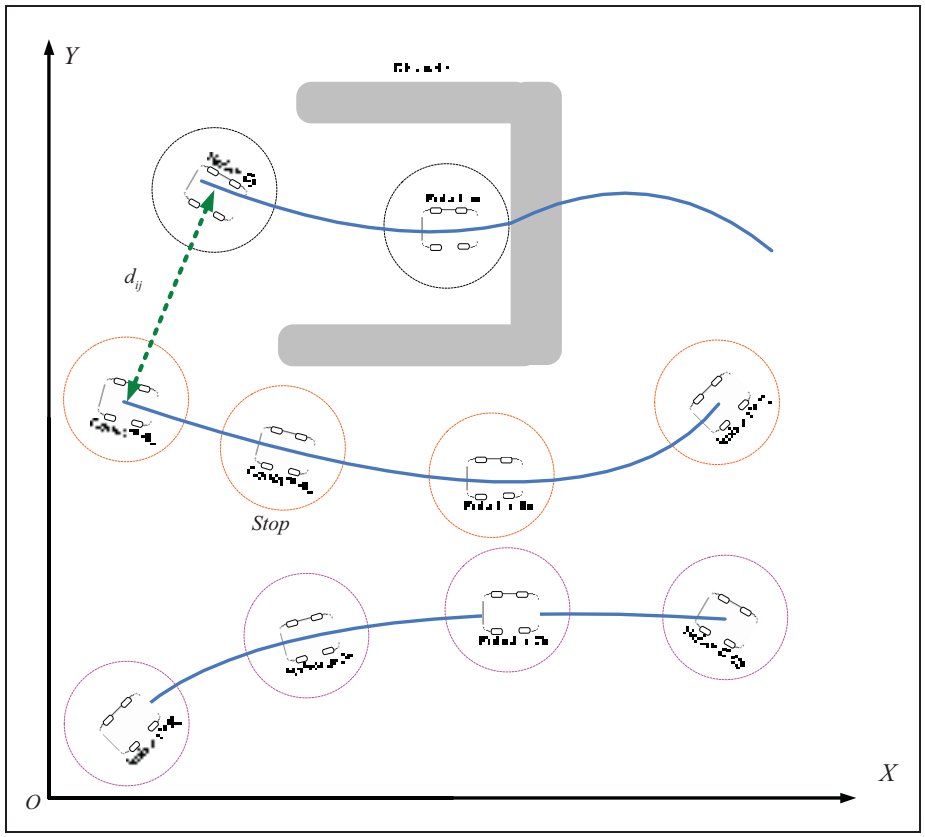


Figure 6.25 Case study 4: Mobile robot observes an obstacle and stuck in local minimum.

Flow chart in Figure 6.29 represents the algorithm steps to detect obstacles and other robots, avoid collisions with obstacles and other robots, group formation, trajectory tracking and coordination.

6.3 Simulation and Experimental Results

To evaluate the performance of the proposed cooperation and control scheme, experimental tests were conducted with three mobile robots traveling on different trajectories. Fig. 6.30 shows the EtsRo mobile robot used for the experimental setup. EtsRo is a mobile robot with four sonar sensors installed on the forward and lateral sides of the robot.

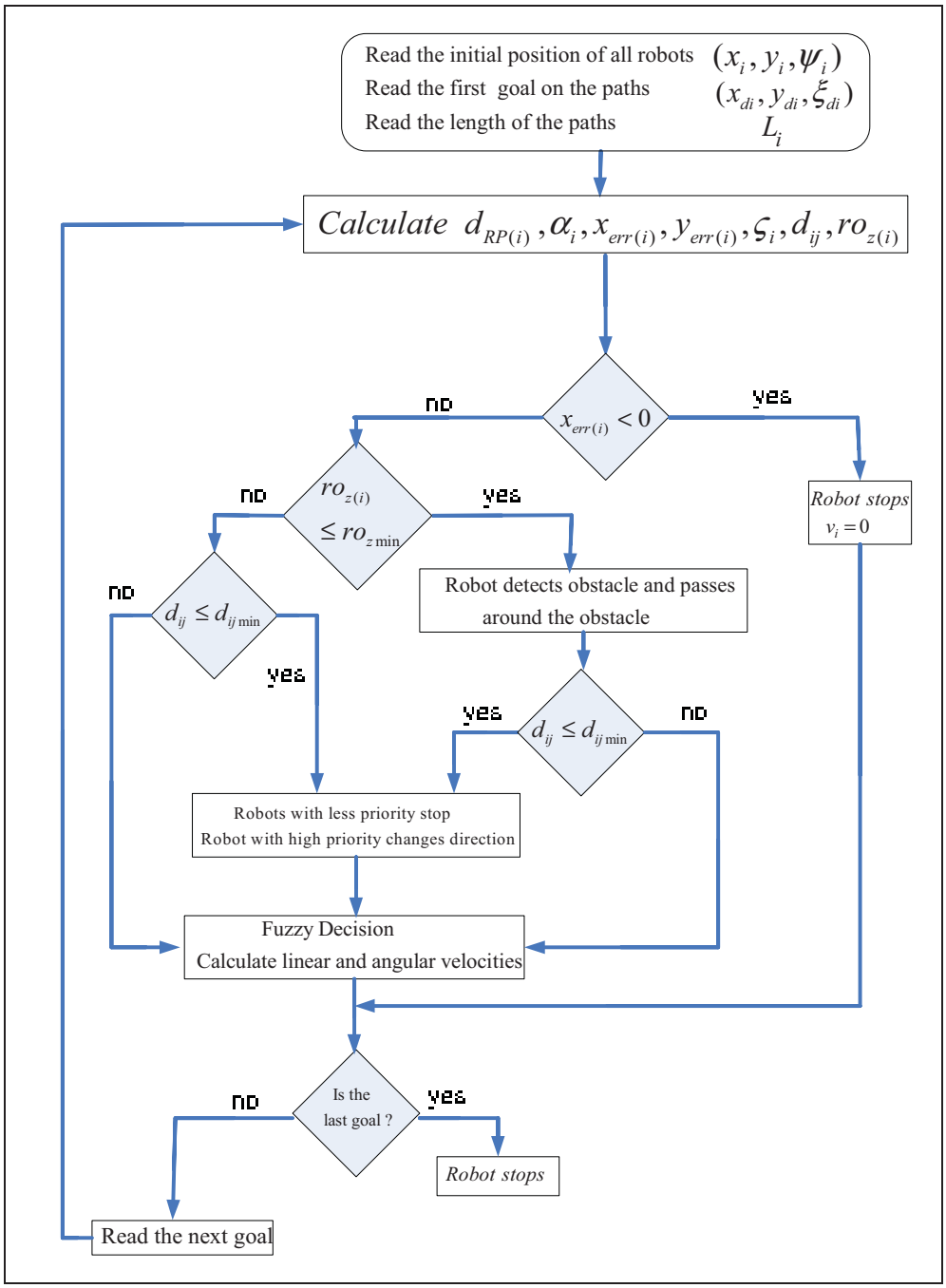


Figure 6.26 Intelligent flow chart of obstacle detection, obstacle avoidance, trajectory tracking and coordination.

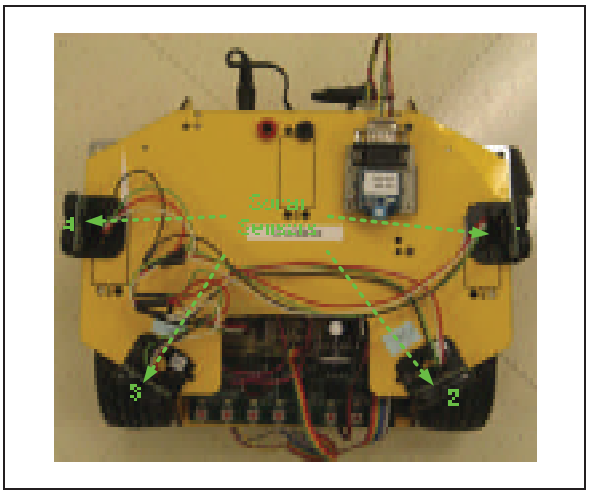


Figure 6.27 The mobile robot (EtsRo).

6.3.1 Simulation Results

In this section, we discuss the results of three different formations of the robots. These formations are:

1) where two robots observe obstacles on their desired trajectories;
2) where one robot confronts an obstacle and changes its desired trajectory, but then observes another robot on this new trajectory;
3) where one of the robots is stuck by an obstacle on the path and loses coordination.

In the first test, robots 1 and 2 confront obstacles on their desired trajectories. Figure 6.31 shows the reference and the actual robots trajectories in this scenario. The trajectory tracking error y_{err} is shown in Figure 6.32. As can be seen in these figures, the robots detect obstacles and recalculate other trajectories to avoid a collision. When the trajectories are clear, the robots resume their desired trajectories and keep coordination.

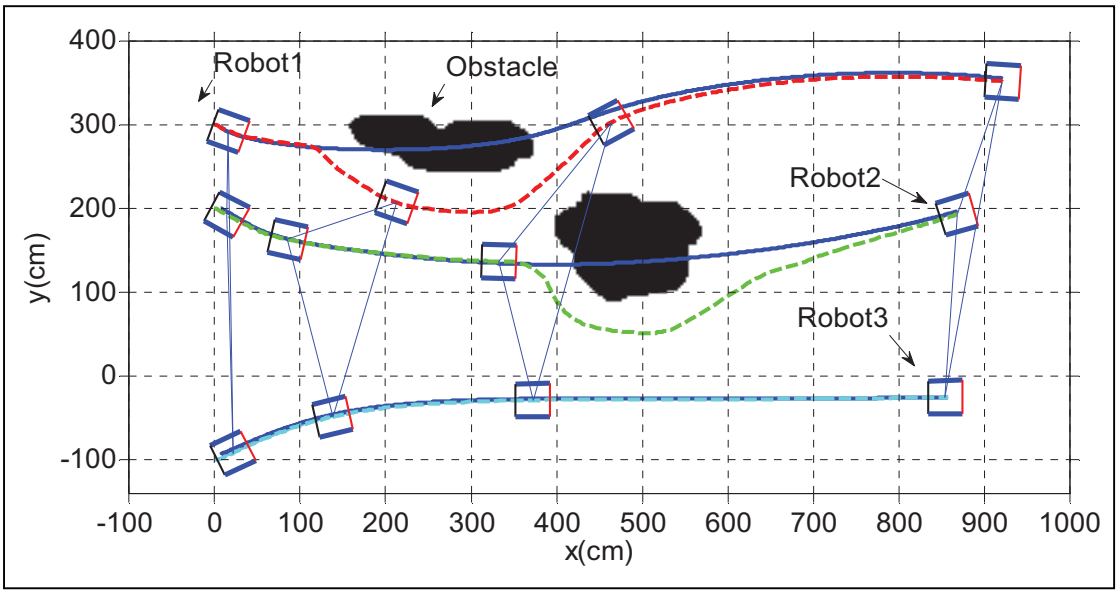


Figure 6.28 Reference and real robots' trajectories.

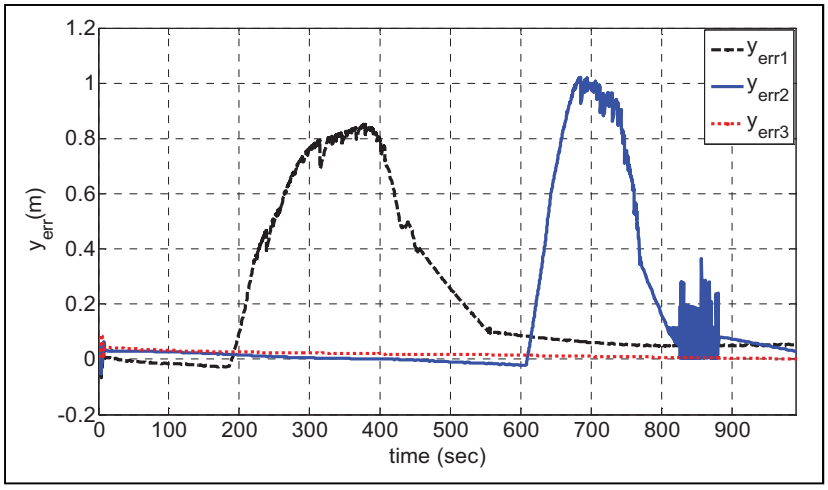


Figure 6.29 Trajectory tracking errors $y_{err(i)}$.

The linear velocity and distance between the first robot and the obstacle observed by the sonar sensors are plotted in Figure 6.33 and Figure 6.34 respectively. These figures show that the robots travel with different velocities relative to the length of the trajectories on which they travel, and the appearance and shape of the obstacles on their trajectories.

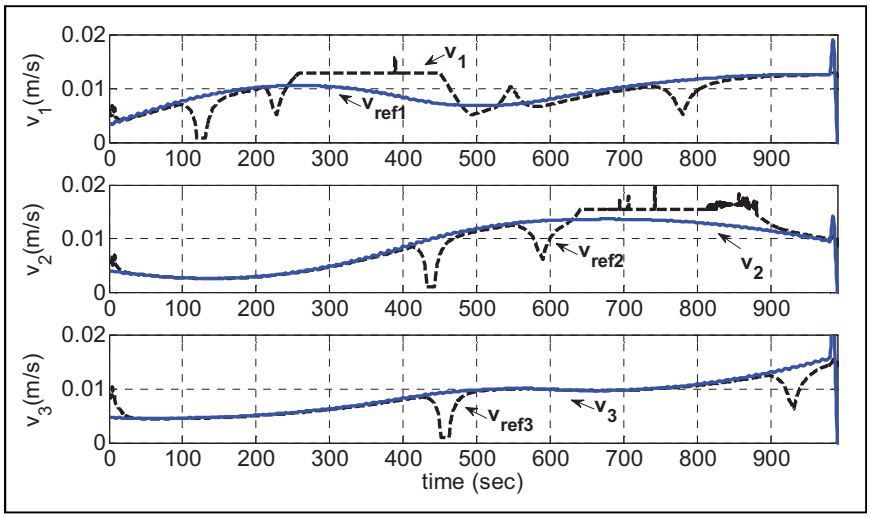


Figure 6.30 Linear velocity of robots.

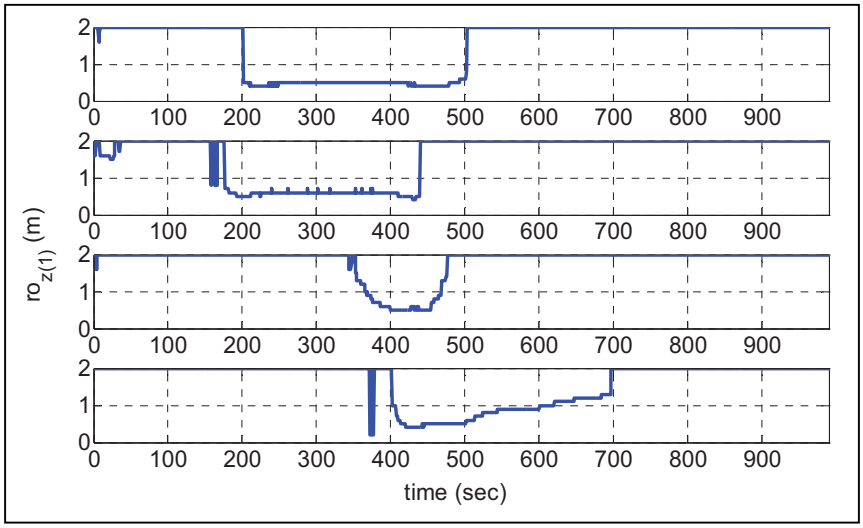


Figure 6.31 Distance between robot 1 and an obstacle observed by its sonar sensors.

In the second test, when robot 1 recalculates another trajectory to avoid the obstacle on its desired trajectory, robot 2 then also appears on its new trajectory and there is a chance of imminent collision. In this situation to avoid a crash between robots, robot 2 will stop until robot 1 moves around the obstacle on its trajectory. When there is no danger of a crash, robot 2 will then continue on its desired trajectory. Figure 6.35 shows both the reference and the

actual robots' trajectories where the robots move around obstacles and maintain the group formation. The trajectory tracking error y_{err} is shown in Figure 6.36.

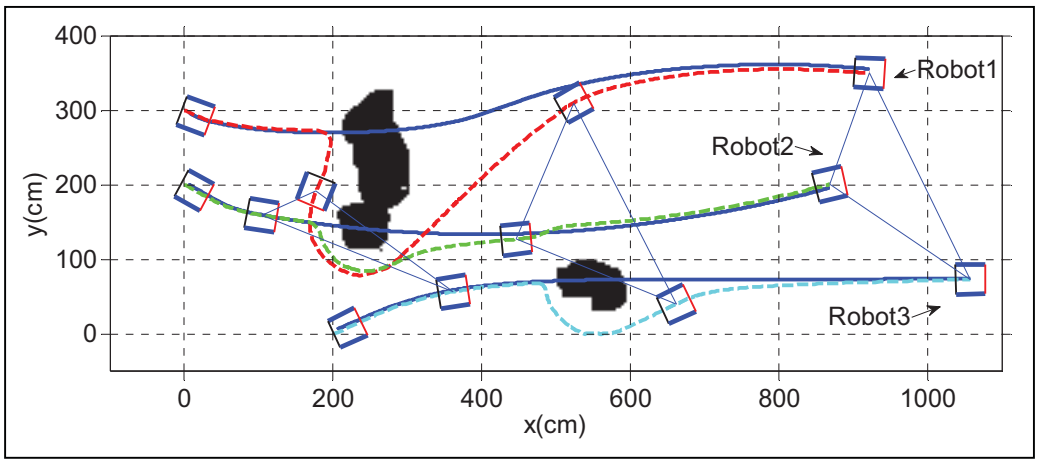


Figure 6.32 Reference and real robots' trajectories.

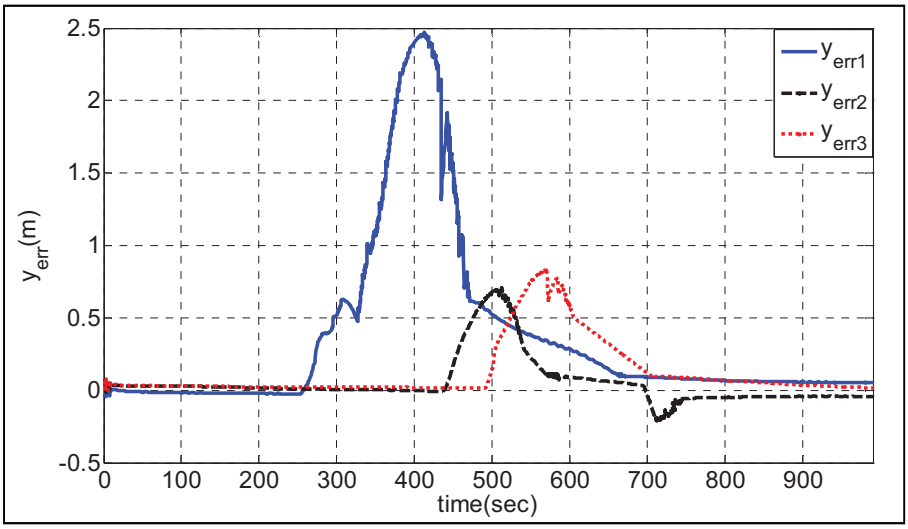


Figure 6.33 Trajectory tracking errors $y_{err(i)}$.

The linear velocity and distance between the first robot and the obstacle observed by the sonar sensors are plotted in Figure 6.37 and Figure 6.38 respectively. As can be seen in Figure 6.38, the linear velocity of robot 2 decreases to zero to avoid a crash with robot 1.

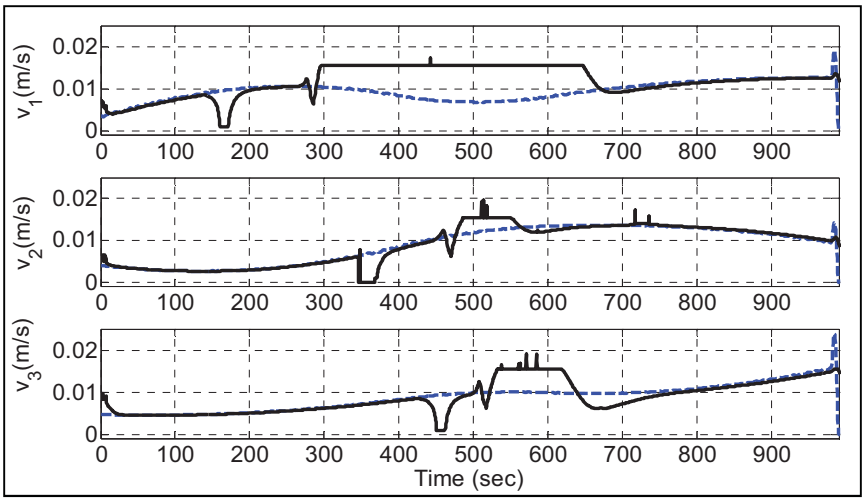


Figure 6.34 Linear velocity of robots.

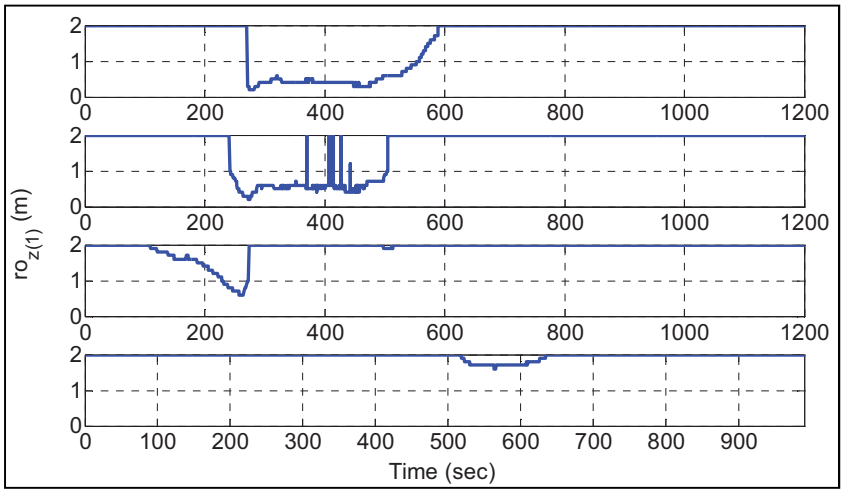


Figure 6.35 Distance between robot 1 and an obstacle observed by its sonar sensors.

In the last scenario, when robot 1 travels on its desired trajectory, it is stuck in local minimum by an obstacle. In this situation the other robots continue on their desired trajectories and keep coordination with each other but they lose coordination with this stuck robot. Figure 6.39 shows both the reference and the actual robots' trajectories in this scenario where robot 1 stuck in local minimum. The trajectory tracking error y_{err} is shown in Figure 6.40.

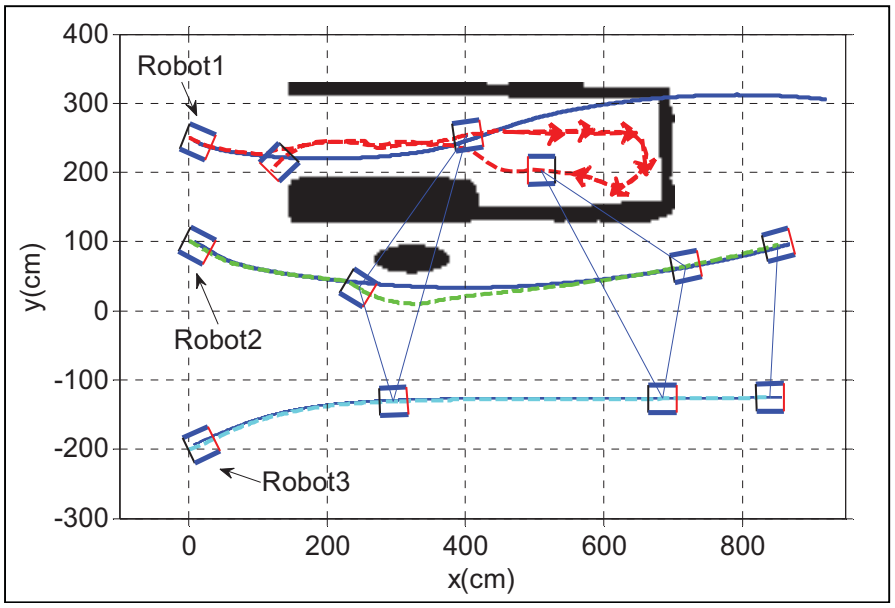


Figure 6.36 Reference and real robots' trajectories.

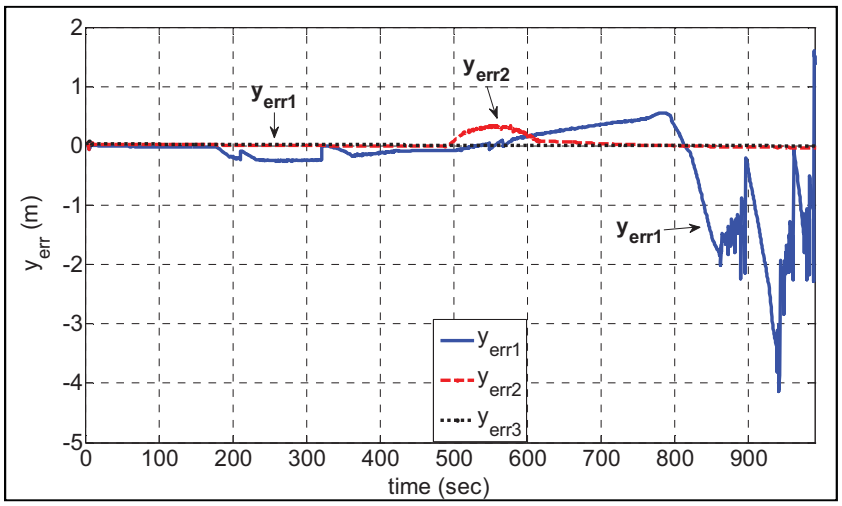


Figure 6.37 Trajectory tracking errors $y_{err(i)}$.

The linear velocity and distance between the first robot and an obstacle that is observed by its sonar sensors are plotted in Figure 6.41 and Figure 6.42 respectively.

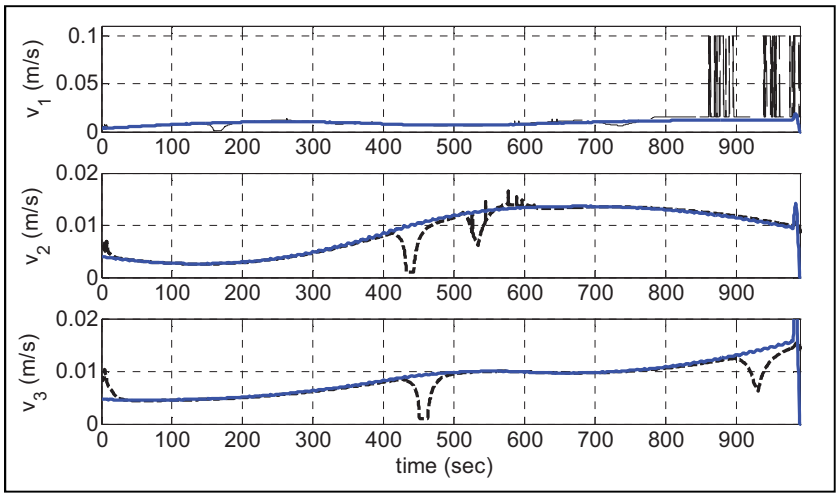


Figure 6.38 Linear velocity of robots.

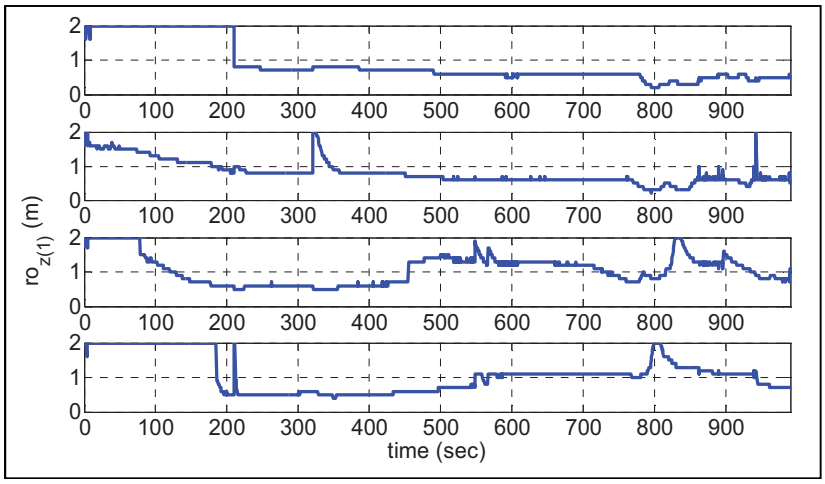


Figure 6.39 Distance between robot 1 and an obstacle observed by its sonar sensors.

6.3.2 Experimental Results

In this scenario, robots 1 and 3 confront obstacles on their desired trajectories. Figure 6.43 shows the reference and the actual robots' trajectories in this scenario. The trajectory tracking error y_{err} is shown in Figure 6.44. As can be seen in these figures, the robots detect obstacles

and recalculate other trajectories to avoid a collision. When their trajectories are clear, the robots resume their desired trajectories and keep group coordination.

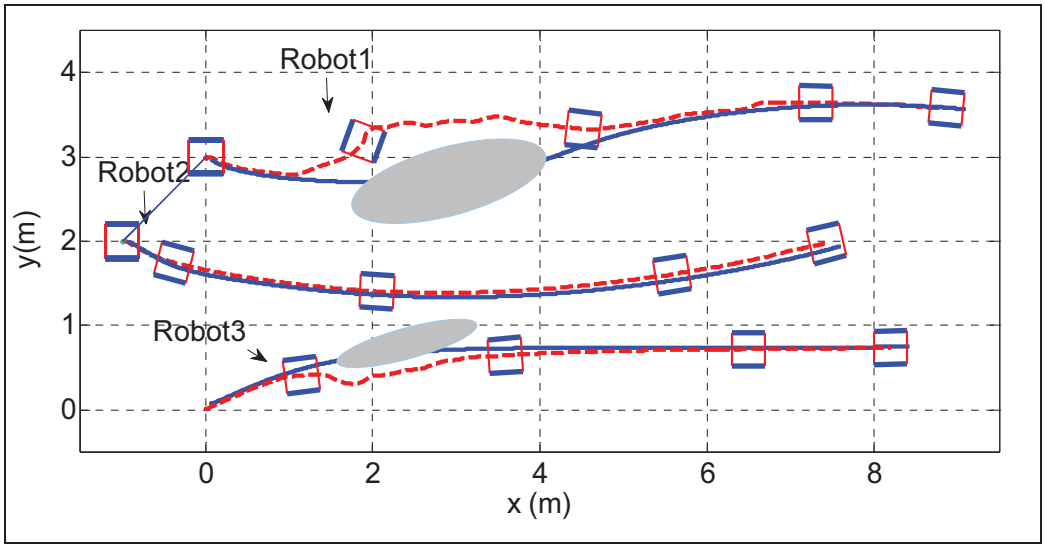


Figure 6.40 Reference and the actual robots' trajectories.

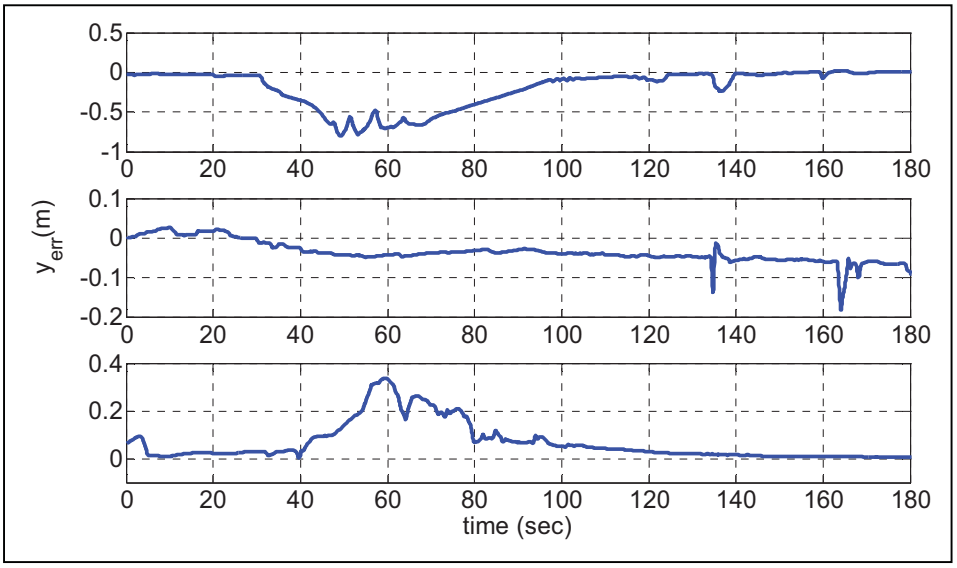


Figure 6.41 Trajectory tracking errors $y_{err(i)}$.

The linear and angular velocities of robots are plotted in Figure 6.45 and Figure 6.46 respectively. These figures show that the robots travel with different velocities relative to the

length of the trajectories on which they travel as well as the appearance and shape of obstacles they encounter on their trajectories.

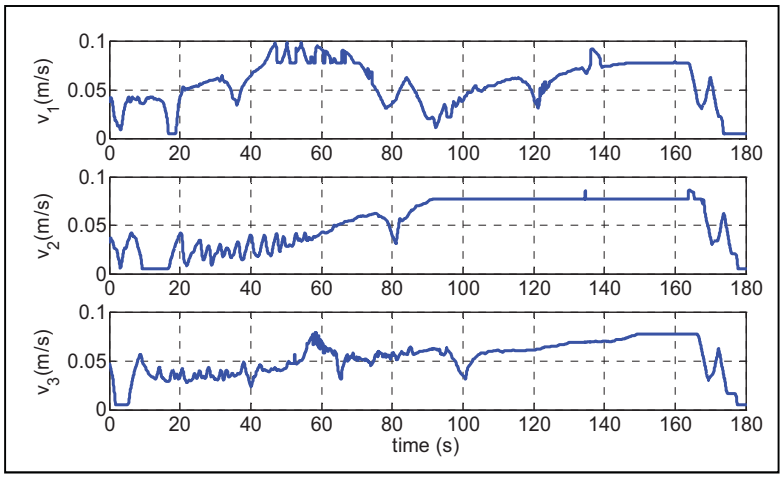


Figure 6.42 Linear velocity of the robots.

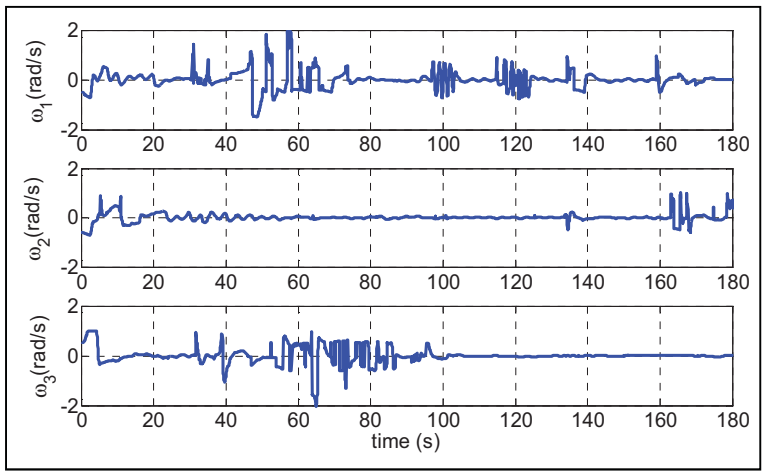


Figure 6.43 Angular velocity of the robots.

6.4 Conclusion

In this chapter, the design and implementation of a dual layered intelligent coordination algorithm is introduced for a team of MMR's that are confronted with obstacles in unknown environments. The developed algorithm has two modes of operation termed 'fuzzy trajectory tracking and coordination', and 'fuzzy crash, obstacle detection and avoidance'. The first mode of operation is only required to perform the velocity and direction changes for the robots due to their differing trajectory lengths, therefore enabling the group to arrive at their individual destinations within the same time duration. The second mode, fuzzy crash, obstacle detection and avoidance, is enacted once any of the robots' sensors detect either a static or dynamic obstacles, including other robots. Once the encountered obstacle is no longer a threat to a robot then the control is reverted back to the first mode of operation. The experimental results in three different unknown formation scenarios demonstrate the accuracy of obtaining control, coordination and obstacle avoidance by using the designed fuzzy algorithm.

CONCLUSION

In this thesis we present techniques for the control and coordination of a group of mobile robots moving in formation in an unknown environment that includes obstacles. The robots in the group are required to simultaneously reach their desired target points while maintaining an overall group formation. This research focuses on the question of how to facilitate a smooth and efficient trajectory tracking, cooperative group behavior and obstacle avoidance for a MMR group. This thesis proposes a solution by dividing the research into three areas, the dynamic tracking control to deliver smooth robot movement, the control algorithm for efficient robot coordination, and an intelligent coordination algorithm that instructs the robots to avoid any obstacles along their paths.

Firstly, an efficient dynamic tracking control is developed for a nonholonomic mobile robot that is based on three modes of operation. These are exponential sliding mode, Lyapunov technique and fuzzy control. A two level controller is designed incorporating a low level PID controller for the right and left motors, and a high level controller to control the speed and movement of the robot. The high level controller uses a feedback controller utilizing the nonlinear function. The developed exponential sliding mode control reduces chattering on the control input compared to conventional sliding modes, and delivers a high dynamic tracking performance in a steady state mode. The experimental results obtained using an EtsRo mobile robot show the effectiveness of the theoretical outcomes.

Secondly, control algorithms are developed for the efficient coordination of a group of mobile robots. These algorithms enable the robots to work both individually and in meaningful robot formations. The first approach is designed to incorporate a low level PID controller for the right and left motors, and a high level controller to coordinate the speed and movement of the robot group. The high level controller is designed using a feedback controller utilizing the exponential sliding mode function.

In the second approach, a control algorithm and efficient coordination architecture is developed for a group of mobile robots using a combination of Lyapunov technique with graph theory embedded within a virtual structure. The high level controller is designed using a feedback controller utilizing the Lyapunov function. The experimental results on a multi-robot platform show the effectiveness of the theoretical result and the performance of the system against short term communication loss or failure, as well as position measurement errors.

Finally in the third approach, the design and implementation of a dual layered intelligent coordination algorithm is introduced for a team of MMR's that are confronted with obstacles in unknown environments. The developed algorithm has two modes of operation termed 'fuzzy trajectory following and coordination', and 'fuzzy obstacle detection and avoidance'. The first mode of operation is only required to perform the velocity and direction changes for the robots due to their differing trajectory lengths, therefore enabling the group arrive at their individual destinations within the same time duration. The second mode, fuzzy crash, obstacle detection and avoidance, is enacted once any of the robots' sensors detect either a static or dynamic obstacle on their trajectories, including other robots. Once the obstacle is no longer a threat to a robot then control is reverted back to the first mode of operation. The experimental results in three different unknown formation scenarios demonstrate the accuracy of obtaining control, coordination and obstacle avoidance by using the designed fuzzy algorithm.

However, in further work these concepts could be readily expanded and improved by including the following additions.

1) To improve the individual robot's performance and task abilities by adding sophisticated sensors such as cameras for a better analysis of unknown environments.
2) To add onboard micro computers to the robots making them more autonomous and powerful in their ability to process decisions.
3) To use low level programming such as ROS instead of Matlab.
4) To increase the number of robots in the team and develop new coordination algorithms.

BIBLIOGRAPHIE

Antonelli, G., F. Arrichiello et S. Chiaverin. 2009. « Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control ». *IEEE Transactions on Control Systems Technology,* vol. 17, n° 5, p. 1173–1182.

Antonelli, G., S. Stefano et G. Fusco. 2007. « A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking ». *IEEE Transactions on Fuzzy Systems,* vol. 15, n° 2 (Apr.), p. 211–221.

B. Siciliano, et O. Khatib. 2008. *Handbook of Robotics*. Springer.

Balch, T., et R. C. Arkin. 1998. « Behavior-based formation control for multi robot teams ». *IEEE Transactions on Robotics and Automation,* vol. 14, n° 6, p. 926–939.

Barnes, L. E., M. A. Fields et K. P. Valavanis. 2009. « Swarm Formation Control Utilizing Elliptical Surfaces and Limiting Functions ». *IEEE Transactions on Systems, Man, Cybernetics-Part B: Cybernetics,* vol. 39, n° 6 (Feb.), p. 1434–1445.

Bartolini, G., A. Ferrara, A. Pisano et E. Usai. 2001. « On the convergence systems ». *International Journal of Control,* vol. 74, p. 718–731.

Bartolini, G., A. Ferrara et E. Usai. 1998. « Chattering avoidance by second-order sliding mode control ». *IEEE Transactions on Automatic Control,* vol. 43, n° 2, p. 241–246.

Bartolini, G., A. Ferrara, E. Usai et V. I. Utkin. 2000. « On multi-input chattering- free second order sliding mode control ». *IEEE Transactions on Automatic Control,* vol. 45, n° 9 (Sep.), p. 1711–1717.

Bartolini, G., A. Pisano, E. Punta et E. Usai. 2003. « A survey of applications of second order sliding mode control to mechanical systems ». *International Journal of Control,* vol. 76, n° 9/10, p. 875–892.

Bartolini, G., A. Pisano et E. Usai. 2001. « Digital second order sliding mode control for uncertain nonlinear systems ». *Automatica,* vol. 37, n° 9, p. 1371–1377.

Beard, R. W., H. Lawton et F. Y. Hadaegh. 2001. « A coordination architecture for spacecraft formation control ». *IEEE Transactions on Control Systems Technology,,* vol. 9, n° 6 (Nov.), p. 777–790.

Boiko, I. 2005. « Analysis of sliding modes in the frequency domain ». *International Journal of Control,* vol. 78, n° 13, p. 969–981.

Boiko, I., L. Fridman, A. Pisano et E. Usai. 2007a. « Analysis of chattering in systems with second-order sliding modes ». *IEEE Transactions on Automatic Control,* vol. 52, n^{o} 11 (Nov.), p. 2085–2102.

Boiko, I., L. Fridman, A. Pisano et E. Usai. 2007b. « Performance analysis of second-order sliding-mode control systems with fast actuators ». *IEEE Transactions on Automatic Control,* vol. 52, n^{o} 6, p. 1053–1059.

Cao, Y. U., A. S. Fukunaga et A. B. Kahng. 1997. « Cooperative mobile robotics: antecedents and directions ». *Autonomous Robots,* vol. 4, n^{o} 1, p. 1-23.

Chang, C. Y., J. P. Sheu, Y. C. Chen et S. W. Chang. 2009. « An Obstacle-Free and Power-Efficient Deployment Algorithm for Wireless Sensor Networks ». *IEEE Transactions on Systems, Man, Cybernetics-Part A: Systems and Humans,* vol. 39, n^{o} 4, p. 795 – 906.

Chang, Y. C., et B. S. Chen. 2000. « Robust tracking designs for both holonomic and nonholonomic constrained mechanical systems: adaptive fuzzy approach ». *IEEE Transactions on Fuzzy Systems,* vol. 8, n^{o} 1 (Feb.), p. 46–66.

Chen, C. Y., T. H. S. Li, Y. C. Yeh et C. C. Chang. 2009a. « Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots ». *Mechatronics,* vol. 19, p. 156–166.

Chen, J., D. Sun, J. Yang et H. Chen. 2009b. « A Leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme ». *The International Journal of Robotics Research,* vol. 29, n^{o} 6.

Choi, S., et M. Kim. 1997. « New discrete-time, fuzzy-sliding-mode control with application to smart structures ». *Journal of guidance, control, and dynamics,* vol. 20, n^{o} 5 (Sep./Oct.), p. 857–864.

Chwa, D. K. 2004. « Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates ». *IEEE Transactions on Control Systems Technology,* vol. 12, n^{o} 4 (Jul.), p. 637–644.

Coelho, P., et U. Nunes. 2005. « Path-following control of mobile robots in presence of uncertainties ». *IEEE Transactions on Robotics and Automation,* vol. 21, n^{o} 2 (Apr.), p. 252–261.

Consolini, L., F. Morbidi, D. Prattichizzo et M. Tosques. 2008. « Leader–follower formation control of nonholonomic mobile robots with input constraints ». *Automatica,* vol. 44, p. 1343–1349.

Corradin, M. L., et G. Orland. 2002. « Control of mobile robots with uncertainties in the dynamical model: a discrete time sliding mode approach with experimental results ». *Control Engineering Practice,* vol. 10, p. 23–34.

Cruz, C. D. L., et R. Carelli. 2008. « Dynamic model based formation control and obstacle avoidance of multi-robot systems ». *Robotica,* vol. 26, n° 3, p. 345-356.

Cruz, C. De La, et R. Carelli. 2006. « Dynamic modeling and centralized formation control of mobile robots ». In *Proc. IEEE Conf. Ind. Electron* (Nov.). p. 3880–3885.

Das, T., et I. N. Kar. 2006. « Design and implementation of an adaptive fuzzy logic based controller of wheeled mobile robots ». *IEEE Transactions on Control Systems Technology,* vol. 14, n° 3 (May), p. 501–510.

Defoort, M., A. KokosY, T. Floquet, W. Perruquetti et J. Palos. 2009. « Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach ». *Robotics and Autonomous Systems,* vol. 57, p. 1094–1106.

Dierks, T., et S. Jagannathan. 2010. « Neural network output feedback control of robot Formations. ». *IEEE Transactions on Systems, Man, Cybernetics-Part B: Cybernetics,* vol. 40, n° 2 (Apr.), p. 383–399.

Do, K. D. 2009. « Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges ». *Robotics and Autonomous Systems,* vol. 57, n° 1 (Jan.), p. 34–47.

Dudek, G., M. R. M. Jenkin et D. Wilkes. 1996. « A taxonomy for multi-agent robotics ». *Autonomous Robots,* vol. 3, n° 4, p. 375–397.

Egerstedt, M., et X. Hu. 2001. « Formation constrained multi-agent control ». *IEEE Transactions on Automatic Control,* vol. 17, n° 6 (Dec.), p. 947–951.

Encarna, P., et A. Pascoal. 2001. « Combined trajectory tracking and path following for marine craft ». In *Proceedings of 9th Mediterranean Conference on Control and Automation*.

Fallah, C. 2007. « Étude de la commmande par mode de glissement sur les systèmes mono et multi variables ». Master's thesis, Montreal, CA, Quebec University (ETS).

Farinelli, A., L. Iocchi et D. Nardi. 2004. « Multirobot Systems: A Classification Focused on Coordination ». *IEEE Transactions on Systems, Man, Cybernetics-Part B: Cybernetics,* vol. 34, n° 5 (Oct.), p. 2015–2028.

Fax, J. A., et R. M. Murray. 2004. « Information flow and cooperative control of vehicle formations ». *IEEE Transactions on Automatic Control,* vol. 49, n° 9 (Sep.), p. 1465–1476.

Ferrara, A., et M. Rubagotti. 2008. « Second-order sliding-mode control of a mobile robot based on a harmonic potential field ». *Control Theory & Applications, IET,* vol. 2, n° 9, p. 807–818.

Fierro, R., et F. L. Lewis. 1995. « Control of a nonholonomic mobile robot: backstepping kinematics into dynamics ». In *Proc.34th IEEE Conf. Decision Control*. p. 3805–3810

Fridman, L. 2003. « Chattering analysis in sliding mode systems with inertial sensors ». *International Journal of Control,* vol. 76, n° 9/10, p. 906–912.

Fukao, T., H. Nakagawa et N. Adachi. 2000. « Adaptive tracking control of a nonholonomic mobile robot ». *IEEE Transactions on Robotics and Automation,* vol. 16, n° 5, p. 609–615.

Ge, S. S., et C. H. Fua. 2005. « Queues and artificial potential trenches for multirobot formations ». *IEEE Transactions on Robotics and Automation,* vol. 21, n° 4 (Aug.), p. 646–656.

Geracioti, T. D., M. A. Kling, J. R. Josephvanderpool, S. Kanayama, N. E. Rosenthal, P. W. Gold et R. A. Liddle. 1989. « Meal-Related Cholecystokinin Secretion in Eating and Affective-Disorders ». *Psychopharmacology Bulletin,* vol. 25, n° 3, p. 444-449.

Gerkey, B. P., et M. J. Mataric. 2004. « A formal framework for the study of task allocation in multi-robot systems ». *International Journal of Robotics Research,* vol. 23, n° 9, p. 939–954.

Ghommam, J., H. Mehrjerdi, M. Saad et F. Mnif. 2010. « Formation path following control of unicycle-type mobile robots ». *Robotics and Autonomous Systems,* vol. 58, n° 5, p. 727–736.

Ghommam, J., et F. Mnif. 2009. « Coordinated Path-Following Control for a Group of Underactuated Surface Vessels ». *IEEE Transactions on Industrial Electronics,* vol. 56, n° 10, p. 3951–3963.

Godsil, C., et G. Royle. 2001. *Algebraic Graph Theory*. Coll. « Graduated Texts in Mathematics ». New York: Springer.

Gu, D., et H. Hu. 2008. « Using Fuzzy Logic to Design Separation Function in Flocking Algorithms ». *IEEE Transactions on Fuzzy Systems,* vol. 16, n° 4, p. 826–838.

Gu, D., et Z. Wang. 2009. « Leader–Follower Flocking: Algorithms and Experiments ». *IEEE Transactions on Control System Technology,* vol. 17, n° 5, p. 1211–1219.

Guzzoni, D., A. Cheyer, L. Julia et K. Konolige. 1997. « Many robots make short work ». *AI Mag,* vol. 18, n° 1, p. 55–64.

Hagras, H., et M. Colley. 2005. « Collaborating multi robotic agents for operations in inaccessible environments ». *The IEE Forum on Autonomous Systems*, (Nov.).

Harmati, I., et A. Saffiotti. 2009. « Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework ». *Robotics and Autonomous Systems,* vol. 57, n° 1, p. 75–86.

Hollinger, G., J. Djugash et S. Singh. 2007. « Coordinated Search in Cluttered Environments Using Range from Multiple Robots ». In *6th International Conference on Field and Service Robotics*.

Hong, Y., J. Hu et L. Ao. 2006. « Tracking control for multi-agent consensus with an active leader and variable topology ». *Automatica,* vol. 47, n° 2, p. 1177–1182.

Howard, A., M. J. Mataric et G. S. Sukhame. 2002. « Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem ». In *Proc. Int. Symp. Distributed Autonomous Robotic Systems*. p. 299–308.

Hu, Y., W. Zhao et L. Wang. 2009. « Vision-Based Target Tracking and Collision Avoidance for Two Autonomous Robotic Fish ». *IEEE Transactions on Industrial Electronics,* vol. 56, n° 5 (May), p. 1401–1410.

Huang, J., S. M. Farritor, A. Qadi et S. Goddard. 2006. « Localization and follow-the-leader control of a heterogeneous group of mobile robots ». *IEEE/ASME Transactions on Mechatronics,* vol. 11, n° 2 (Apr.), p. 205–215.

Hung, J. Y., W. Gao et J. C. Hung. 1993. « Variable structure control: A survey ». *IEEE Trans. Ind. Electron,* vol. 40, n° 1, p. 2-22.

Huntsberger, T., P. Pirjanian, A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. J. Ganino, M. Garrett, S. S. Joshi et P. S. Schenker. 2003. « CAMPOUT: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration ». *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 33, n° 5, p. 550–559.

Hwang, C. H., et N. W. Chang. 2008. « Fuzzy Decentralized Sliding-Mode Control of a Car-Like Mobile Robot in Distributed Sensor-Network Spaces ». *Control Engineering Practice,* vol. 16, n° 1 (Feb.), p. 97–109.

Hwang, C. L. 2004. « A novel Takagi-Sugeno-based robust adaptive fuzzy sliding-mode controller ». *IEEE Transactions on Fuzzy Systems,* vol. 12, n° 5 (Oct.), p. 676–687.

Hwang, C. L., et L. J. Chang. 2007. « Trajectory Tracking and Obstacle Avoidance of Car-Like Mobile Robots in an Intelligent Space Using Mixed H2/H∞ Decentralized Control ». *IEEE/ASME Transactions on Mechatronics,* vol. 12, n° 3 (Jul.), p. 345–352.

Iguria, A., et A. M. Howard. 2009. « An Integrated Approach for Achieving Multirobot Task Formations ». *IEEE/ASME Transactions on Mechatronics,* vol. 14, n° 2, p. 176–186.

Inalhan, G., D. Stipanovic et C. Tomlin. 2002. « Decentralized optimization, with application to multiple aircraft coordination ». In *Proc. 41st IEEE Conf. Decision and Control*. p. 1147–1155.

Jolly, K. G., R. S. Kumar et R. Vijayakumar. 2010. « Intelligent task planning and action selection of a mobile robot in a multi-agent system through a fuzzy neural network approach ». *Engineering Applications of Artificial Intelligence,* vol. 23, n° 6, p. 923–933.

Keviczky, T., F. Borelli et G. Balas. 2006. « Decentralized receding horizon control for large scale dynamically decoupled systems ». *Automatica,* vol. 42, p. 2105–2115.

Keviczky, T., F. Borrelli, K. Fregene, D. Godbole et G. J. Balas. 2008. « Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle ». *IEEE Transactions on Control System Technology,* vol. 16, n° 1, p. 19 –32.

Kim, M. S., J. H. Shin, S. G. Hong et J. J. Lee. 2003. « Designing a robust adaptive dynamic controller for nonholonomic mobile robots under modeling uncertainties and disturbances ». *Mechatronics,* vol. 13, n° 5, p. 507–519.

Klavins, E. 2003. « Communication complexity of multi-robot systems ». *Algorithmic Foundations of Robotics,* vol. 7, p. 275–292.

Ko, N. Y., D. J. Seo et R. G. Simmons. 2008. « Collision-free motion coordination of heterogeneous robots ». *Journal of mechanical science and technology,* vol. 22, n° 11, p. 2090–2098.

Lee, G., et N. Y. Chong. 2009. « Decentralized formation control for small-scale robot teams with anonymity ». *Mechatronics,* vol. 19, n° 1 (Feb.), p. 85–105.

Lee, J. H., C. Lin, H. Lim et J. M. Lee. 2009. « Sliding Mode Control for Trajectory Tracking of Mobile Robot in the RFID Sensor Space ». *International Journal of Control, Automation, and Systems,* vol. 7, n° 3, p. 429–435.

Levant, A. 1993. « Sliding order and sliding accuracy in sliding mode control ». *Int. J. Control,* vol. 58, p. 1247–1263.

Levant, A. 2003. « Higher order sliding modes, differentiation and output feedback control ». *International Journal of Control,* vol. 76, n° 9/10, p. 924–941.

Levant, A. 2005. « Quasi-continuous High-order Transactions on Sliding-mode Controllers ». *IEEE Transactions on Automatic Control,* vol. 50, n° 11, p. 1812–1816.

Li, H., S. X. Yang et M. L. Seto. 2009. « Neural-Network-Based Path planning for a Multirobot System with Moving Obstacles ». *IEEE Transactions on Systems, Man, Cybernetics-Part C: Applications and Reviews,* vol. 39, n° 4, p. 410–419.

Li, T. H. S., S. J. Chang et W. Tong. 2004. « Fuzzy target tracking control of autonomous mobile robots by using infrared sensors ». *IEEE Transactions on Fuzzy Systems,* vol. 12, n° 4 (Aug.), p. 491-501.

Maalouf, E., M. Saad et H. Saliah. 2006. « A higher level path tracking controller for a four-wheel differentially steeres mobile robot ». *Robotics and Automation Systems,* vol. 54, p. 23–33.

Mariottini, G. L., G. Pappas, D. Prattichizzo et K. Daniilidis. 2005. « Vision based localization of leader–follower formations ». In *Proc. 44th IEEE Conf. Decis. Control*. p. 635–640.

Mbede, J. B., X. Huang et M. Wang. 2003. « Robust neuro-fuzzy sensor-based motion control among dynamic obstacles for robot manipulators ». *IEEE Transactions on Fuzzy Systems,* vol. 11, n° 2, p. 249–261.

Mehrjerdi, H., et M. Saad. 2010. « Chattering Reduction on the Dynamic Tracking Control of a Nonholonomic Mobile Robot using Exponential Sliding Mode ». *Journal of Systems and Control Engineering (Accepted)*.

Mehrjerdi, H., M. Saad et J. Ghommam. 2010a. « Formation and path following for multiple mobile robots ». In *IEEE International Symposium on Industrial Electronics* (4-7 July). Bari, Italy.

Mehrjerdi, H., M. Saad et J. Ghommam. 2010b. « Fuzzy Crash Avoidance and Coordination between Mobile Robots ». In *18th Mediterranean Conference on Control and Automation*. p. 592-597. Marrakech, Morocco.

Mehrjerdi, H., M. Saad et J. Ghommam. 2010c. « Hierarchical Fuzzy Cooperative Control and Path Following for a Team of Mobile Robots ». *IEEE/ASME Transactions on Mechatronics,* vol. (In press).

Mehrjerdi, H., M. Saad et J. Ghommam. 2010d. « Intelligent cooperative behavior and crash avoidance between multi mobile robots ». *Fuzzy Set and Systems (Submitted)*.

Mehrjerdi, H., M. Saad et J. Ghommam. 2010e. « Perceptive formation and obstacle avoidance for a squad of mobile robots in unknown environment ». *IEEE/ASME Transactions on Industrial Electronics (Submitted)*.

Mehrjerdi, H., M. Saad, J. Ghommam et A. Zerigui. 2010a. « Cooperation Control for a Team of Mobile Robots Based on Fuzzy Logic ». In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (6-9 July). Montreal, Canada.

Mehrjerdi, H., M. Saad, J. Ghommam et A. Zerigui. 2010b. « Optimized Neuro-fuzzy coordination for multiple four wheeled mobile robots ». *Information Technology Journal,* vol. 9, n° 8, p. 557-570.

Moreau, L. 2005. « Stability of multiagent systems with time-dependent communication links ». *IEEE Transactions on Automatic Control,* vol. 50, n° 2 (Feb.), p. 169–182.

Nouyan, S., R. Gross, M. Bonani, F. Mondada et M. Dorigo. 2009. « Teamwork in Self-Organized Robot Colonies ». *IEEE Transactions on Evolutionary Computation,* vol. 13, n° 4, p. 695–711.

Ogren, P., E. Fiorelli et N. E. Leonard. 2004. « Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment ». *IEEE Transactions on Automatic Control,* vol. 40, n° 8 (Aug.), p. 1292–1302.

Olfati-Saber, R., et R. M. Murray. 2002. « Distributed cooperative control of multiple vehicle formations using structural potential functions ». In *Proc. 15th IFAC World Congr.* p. 1–7.

Park, B., S. S. Yoo, J. B. Park et Y. H. Chou. 2009. « Adaptive Neural Sliding Mode Control of Nonholonomic Wheeled Mobile Robots With Model Uncertainty ». *IEEE Transactions on Control Systems Technology,* vol. 17, n° 1, p. 207–214.

Park, M. S., D. Chwa et S. K. Hong. 2006. « Decoupling Control of A Class of Underactuated Mechanical Systems Based on Sliding Mode Control ». *International Joint Conference SICE-ICASE*, p. 806–810.

Peasgood, M., C. M. Clark et J. McPhee. 2008. « A complete and scalable strategy for coordinating multiple robots within roadmaps ». *IEEE Transactions on Robotics,* vol. 24, n° 2 (Apr.), p. 283–292.

Pereira, G. A. S., V. Kumar et M. F. M. Campos. 2008. « Closed loop motion planning of cooperating mobile robots using graph connectivity ». *Robotic and Autonomous Systems,* vol. 56, n° 4, p. 373–384.

Purvis, K. B., K. J. Astrom et M. Mhammash. 2008. « Estimation and optimal configurations for localization using cooperative UAVs ». *IEEE Transactions on Control Systems Technology,* vol. 16, n° 5 (Sep.), p. 947-958.

Ray, A. K., P. Benavidez, L. Behera et M. Jamshidi. 2009. « Decentralized Motion Coordination for a Formation of Rovers ». *IEEE Systems Journal,* vol. 3, n° 3, p. 369–381.

Ren, W. 2007. « Multi-vehicle consensus with a time-varying reference state ». *Systems & Control Letters,* vol. 56, n° 7-8 (Jul.), p. 474–483.

Ren, W., et R. W. Beard. 2005. « Consensus seeking in multi-agent systems under dynamically changing interaction topologies ». *IEEE Transactions on Automatic Control,* vol. 50, n° 5 (May), p. 655–661.

Ren, W., et N. Sorensen. 2008. « Distributed coordination architecture for multi-robot formation control ». *Robotics and Autonomous Systems,* vol. 56, n° 4 (Apr.), p. 324–333.

Riachy, S., Y. Orlov, T. Floquet, R. Santiesteban et J. P. Richard. 2008. « Second-order sliding mode control of underactuated mechanical systems 1: Local stabilization with application to an inverted pendulum ». *International Journal of Robust and Nonlinear Control,* vol. 18, (Mar.), p. 529–543.

Sankaranarayanan, V., et A. D. Mahindrakar. 2009. « Control of a Class of Underactuated Mechanical Systems Using Sliding Modes ». *IEEE Transactions on Robotics,* vol. 25, n° 2, p. 459–467.

Schneider-Fontan, M., et M. Mataric. 1998. « Territorial multi-robot task division ». *IEEE Transactions on Automatic Control,* vol. 14, n° 5, p. 815–822.

Sepulchre, R., D. Paley et N. E. Leonard. 2007. « Stabilization of planar collective motion: All-to-all communication ». *IEEE Transactions on Automatic Control,* vol. 52, n° 5 (May), p. 811–824.

Shima, T., M. Idan et O. M. Golan. 2006. « Sliding-mode control for integrated missile autopilot guidance ». *Journal of guidance, control, and dynamics,* vol. 29, n° 2 (Mar./Apr.), p. 250–260.

Shtessel, Y. B., I. A. Shkolnikov et M. D. J. Brown. 2003. « An asymptotic second-order smooth sliding mode control ». *Asian J. Control,* vol. 5, n° 4, p. 498–504.

Skrjanc, I., et G. Klancar. 2010. « Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves ». *Robotics and Autonomous Systems,* vol. 58, n° 1 (Jan.), p. 1–9.

Slotine, J. J., et W. Li. 1991. *Applied Nonlinear Control*. Coll. « Englewood Cliffs ». NJ: Englewood Cliffs, Prentice-Hall.

Takahashi, J., T. Yamaguchi, K. Sekiyama et T. Fukuda. 2009. « Communication Timing Control and Topology Reconfiguration of a Sink-Free Meshed Sensor Network With Mobile Robots ». *IEEE/ASME Transactions on Mechatronics,* vol. 14, n° 2, p. 187–197.

Utkin, V., J. Guldner et J. Shi. 1999. « Sliding Modes in Electromechanical Systems ». In. London, U.K.: Taylor & Francis.

Utkin, V. I. 1977. « Variable structure systems with sliding modes: A survey ». *IEEE Trans. Autom. Control,* vol. AC-22, n° 2, p. 212–222.

Viguria, A., et A. M. Howard. 2009. « An Integrated Approach for Achieving Multirobot Task Formations ». *IEEE/ASME Transactions on Mechatronics,* vol. 14, n° 2, p. 176–186.

Wang, H. O., K. Tanaka et M. F. Griffin. 1996. « An approach to fuzzy control of nonlinear systems: Stability and design issues ». *IEEE Transactions on Fuzzy Systems,* vol. 4, n° 1 (Feb), p. 14-23.

Wang, Z., M. Zhou et N. Ansari. 2003. « Ad-hoc robot wireless communication ». In *Proceedings IEEE Conference on Systems, Man and Cybernetics* (Oct.). p. 4045–4050. Washington, DC.

Xu, Y. 2008. « Chattering Free Robust Control for Nonlinear Systems ». *IEEE Transactions on Control Systems Technology,* vol. 16, n° 6 (Nov.), p. 1352–1359.

Yang, J. M., et J. H. Kim. 1999. « Sliding mode control of for trajectory tracking of nonholonomic wheeled mobile robots ». *IEEE Transactions on Automatic Control,* vol. 15, n° 3 (Jun.), p. 578–587.

Yang, T. T., Z. Y. Liu, H. Chen et R. Pei. 2008. « Formation Control and Obstacle Avoidance for Multiple Mobile Robots ». *Acta Automatica Sinica,* vol. 34, n° 5 (May), p. 588–593.

Zhu, A., et S. X. Yang. 2007. « Neurofuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments ». *IEEE Transactions on Systems, Man, Cybernetics-Part C: Application and Reviews,* vol. 37, n° 4 (Jul.), p. 610–621.