UNIVERSITAT ROVIRA I VIRGILI

SWARM ROBOTIC SYSTEMS: YPOD FORMATION WITH ANALYSIS ON SCALABILITY AND STABILITY

Purushotham Muniganti

ISBN: / DL:

Swarm Robotic Systems: Y-Pod Formation with analysis on scalability and stability

Ph. D. Thesis

Purushotham Muniganti

UNIVERSITAT ROVIRA I VIRGILI

Department of Electronic, Electrical and Automatic Control Engineering

ISBN: / DL:

ISBN: / DL:

ISBN: / DL:

ISBN: / DL:

Purushotham Muniganti

Swarm Robotic Systems: Y-Pod Formation with analysis on scalability and stability

Ph. D. Thesis

Supervised by

Dr. Albert Oller Pujol

Department of Electronic, Electrical and Automatic Control Engineering

Intelligent Robotics and Computer Vision Research Group

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2016

ISBN: / DL:

ISBN: / DL:

UNIVERSITAT ROVIRA I VIRGILI
Departament d'Enginyeria Electrònica, Elèctrica i Automàtica
Escola Tècnica Superior D'Enginyeria
Campus Sescelades
Avinguda dels Països Catalans, 26
43007 Tarragona
Espanya
Tel.: + 34 977 559 610 / 559 728
Fax: + 34 977 559 605

Jo, Albert Oller Pujol, professor titular del Departament d'Enginyeria Electrònica, Elèctrica i Automática de la Universitat Rovira i Virgili,

FAIG CONSTAR que aquest treball, titulat "Swarm Robotic Systems: Y-pod Formation with Analisys on Scalability and Stability", que presenta Purushotham Muniganti per a l'obtenció del títol de Doctor, ha estat realitzat sota la meva direcció al Departament d'Enginyeria Electrònica, Elèctrica i Automàtica d'aquesta universitat.

Tarragona, 2 de Juliol de 2015

El director de la tesi doctoral

Albert Oller Pujol

Acknowledgements

I am particularly grateful to my supervisor Dr. Albert Oller Pujol, who inspired me with his passion for interdisciplinary research, encouraged me during particularly hard and crisis time, challenged me intellectually and ultimately made this opportunity of professional and personal growth possible. Dr. Albert Oller Pujol guided me to pursue important problems that will have practical impact and was always available to guide me whenever I approached him. His extensive knowledge, experience, and exceptional ability to find new approaches for difficult problems were pivotal in this work and my development as a researcher. This work would not have been completed without his encouragement and patience. I feel privileged to have had the opportunity to study under his guidance and for learning many valuable lessons under him during the past four years.

The interdisciplinary nature of my project required me to gather knowledge in many areas. I greatly benefited from the workshops, conferences and discussions with many senior researchers, and also gained knowledge from robotics exhibition, which has been conducted every year in Rovira i Virgili University under my supervisor. I am also grateful for the discussions and fun time with researchers in my research group and particularly to Jaume Rodirguz Melendez for his contribution to my work and to Julian Cristiano for sharing his analyses. I also give thanks to Dr. Domenec Puig Valls for his moral support and encouragement in all aspects and I extended my thanks and blessings from Dra. Maria Cristina Andreu for her enormous help and guidance in education as well as supporting my family over here in Tarragona. And to my colleagues at the Intelligent Robotics and Computer Vision research lab for fostering a positive, supportive and fun environment.

I would like to thank my friends at Tarragona especially Subba Rao Mandava, Marco A.Piccolino-Boniforti, Roberto Caspani, Balakrishna Bugga, Prasad Ganji, Suva Paria, Satyanarayana Rao Rachakonda and my favourite volleyball team friends in Tarragona for their kind friendship for the past several years which made my stay at Tarragona memorable.

ISBN: / DL:

Finally, I would have never managed to accomplish this daunting task without the loving support of my family, I cordially thank my father, Mr. Muniganti Murali and my mother Mrs. Muniganti Tarabai and my wife, Mrs. Muniganti Madhavilatha and my two children Muniganti Koushik, Muniganti Kranti for their patience, confidence, and trust, and the so many wonderful friendships that I was blessed with during my stay in Tarragona and back home.

ISBN: / DL:

To my mother and my father

To my wife and two sons

ISBN: / DL:

Abstract

The context of this work is an active area of research community which is “swarm formation”. In general, swarm system has most striking examples from nature: social insect colonies are able to build sophisticated structures and regulate the activities of millions of individuals by endowing each individual with simple rules. When applying rules extracted from natural systems to artificial problems, essentially requires different control parameters in order to fulfil the system performance in terms of scalability, flexibility and robustness.

This thesis contributes to the investigation of the swarm formation shape and controller, which is important in swarm robotics too since coordinated behaviour of a group of robots to form a pattern when viewed globally. In this regard, global shape formation is one of the ongoing problems in artificial swarm intelligence. In nature, it is performed for various purposes, such as natural disaster and flock of large birds flying together while forming a shape in order to reduce the air resistance. There exist various shape formations in the literature, but in this thesis, approached new strategy, i.e. Y-Pod, which has vast applications compared to other formation techniques. The Y-Pod is a node which connected with three segments and it will appears different for 2D and 3D environments with respect to angles and shapes.

The main objective of the proposed approach is to form a Y-Pod shape using with linear controller that significantly define the resulting behavior. We have proposed system settling time and pole based approach with respect to equilibrium strategy, to control the swarm system. The proposed linear controller guarantee that the system stability and scalability based on steering analysis and pattern index matching techniques. In addition, with the help of pattern index matching technique, we justify the absolute minima and system synchronization problems in order to overcome the redundancy issues in communication networks. In this process, parameters are chosen based on desired formation as well as user defined constraints. This approach compared to others, is simple, computationally efficient, scales well to different swarm sizes, to both centralized and decentralized swarm models.

ISBN: / DL:

ISBN: / DL:

Contents

ISBN: / DL:

List of Tables

List of Figures

ISBN: / DL:

ISBN: / DL:

Chapter 1

1. Introduction

This chapter provides an introduction to the work presented in this thesis. Particularly, the overview, objectives are briefly described. Then, the arguments exposed in this thesis will be briefly presented, along with the outline of this work.

1.1 Overview

The context of this work is the innovative young field of swarm systems. In swarm systems perspective, present and future active area of research community is swarm robotics. This field gives an exciting basic platform for young researchers to get involved and share new ideas to scrutinize their minds in analytical and heuristic approaches. Starting in the early 1980's the attention of researchers was attracted by the idea of creating groups of mobile robots able to collaborate in order to accomplish one or more predefined tasks. The basic principles behind this new approach to the robotics cooperation, coordination and other interactions among themselves was directly inspired by the observation of natural systems. In general, robotics is already an interdisciplinary field uniting mechanical engineering, electrical engineering, artificial intelligence and cognitive science. However, swarm robotics is even more extensive, including fields of biology, chemistry, statistical physics and even philosophy. Swarm robotics was first initiated in the early 80's by Beni and Wang (Beni and Wang 1989) but the name of the 'swarm' was first mentioned by Alex Meystel in the discussion of cellular robotics systems. For problems in biology, the evolutionary approach involves studying the fascinating array of observed behaviour in natural collectives (flocks, schools, herds, etc.) from the perspective of evolution by natural selection. This approach

ISBN: / DL:

provides important insights into the mechanisms that drive group behavior in natural collectives. In other words, the development in swarms of mathematical models is to explain evolutionary puzzles such as cooperation and altruism (Huberman 1993, Jeanne 1986, Camazine 1998), optimization problems such as travelling salesman problem (Dorigo and Gambardella 1997, Bonabeau et al. 2000), even in business (Bonabeau and Meyer 2001). Although in flocks, and schools of fishes (see Reynolds 1987, Shaw 1962, Parrish 1997). Perhaps the most striking examples are from nature: social insect colonies are able to build sophisticated structures and regulate the activities of millions of individuals by endowing each individual with simple rules. According to the environmental activities, a colony can adjust its behaviour through assigning different numbers of insects to different tasks or adjusting the behaviour of individual insects. Scalability, flexibility and robustness are three main advantages for such systems (Bonabeau et al. 1999).

This thesis particularly focuses on swarm formation, which is important in swarm robotics too, since coordinated behaviour of a group of robots to form a pattern when viewed globally. In this regard, global shape formation is one of the ongoing problems in artificial swarm intelligence. In nature, it is performed for various purposes. For instance, a flock of large birds fly together while forming a V shape in order to reduce the air resistance (Weimerskirch et al. 2001), and search and rescue swarms (Reynolds 1987) could be used in disaster areas such as earthquake zones, searching through darkened, stricken vessels or burning buildings. Shape formation in artificial intelligence systems usually requires particular task-oriented performances, which include forming sensing grids (Spears et al. 2004), unknown environment exploring and mapping in space, underwater, or hazardous (Nawaz et al. 2010), and forming a barricade for surveillance or protecting an area/person (Cheng et al. 2005).

Besides, aforementioned explanations of swarm formation literature, there exist various shapes. Some of these shape formations are unable to overcome some features:

- Scalability: To cover the complete environment in all size using callable solutions.
- Stability. To maintain steering angular velocity without oscillation.
- Flexibility. The robots are in particular sizes to cover the entire environment.

ISBN: / DL:

- Fault tolerant. Some robots fail in formation due to hardware capabilities.
- Redundancy. Problems exist due to its complexity in network topology.

In this sense, it is very important to decide the best shape and control mechanisms in order to overcome all the above mentioned problems. As said before, this thesis addresses the Y-Pod shape formation. The Y-Pod was first initiated by Alvin Swimmer (Alvin 2001) for 3D morphing architecture, which has vast applications compared to other formation techniques. The Y-Pod has wide range of applications in various fields such as: 3D morphing architecture, cell biology, molecular dynamics, geodesic spheres, nano, micro and macro connector technology architecture for new prototypes and Y-Pod communicator models that, move, collapse, walk, illuminate, reflect, rotate and fly swim and also redundancy problems in network topologies.

1.2 Objectives of the research

This thesis contributes to the investigation of development of swarm formation and formation controller in non-holonomic robots. The main objective of the proposed approach is to obtain a Y-pod swarm robot formation, and control the shape in order to maintain the Y-Pod. In addition, both stability and scalability of Y-Pod shape must be guaranteed meanwhile the system evolves. In order to accomplish these objectives, this thesis deals with the following specific tasks:

1. To obtain a Y-Pod formation, we propose several parameters, which significantly define the resulting behaviour. Those parameters are related to system equilibrium such as settling time and frequency poles, which are used directly into the control law expressions.

2. The proposed linear controller and simulation tuned parameters are combined to control Y-Pod swarm formation in terms of orientation and swarm movement as a whole. Parameters are chosen based on initial conditions as well as user defined constraints. This approach, compared to others, is simple, computationally efficient, scales well to different swarm sizes and to both centralized and decentralized swarm models.

ISBN: / DL:

3. System scalability and stability analysis are performed using pattern index matching technique to evaluate the performance shape of Y-Pods. In such a way, networked topologies or redundancy problems might overcome.

4. An extension to obstacle avoidance problem of Y-Pod formation is proposed based on Jacobian potential fields approach. Furthermore, we show that this work can be extended to 3D environments under some assumptions for future research work.

This research work was studied and developed at IRCV (Intelligent Robotics and Computer Vision Research Group) at the Department of Electronics Electrical and Automation Engineering in the Rovira I Virgili University.

1.3 Thesis Outline

This thesis is devised in three main topics: the linear controller for non-holonomic robots in 2D environments, the analysis of results under different input states, and both the extension to obstacle avoidance problem and in 3D environments. The structure of this document is as follows.

In Chapter 2, our attention is focused on the investigation about the background information of swarm robotics. The basic concepts that lie behind the idea of using swarm robots are illustrated. Additionally, delivered ideal thoughts of researcher involved, acquired and reached by swarm robots are shown.

Chapter 3 particularly elevates the swarm formation related works with respect to various shape formations, formation control methods, behaviour based systems, graph theoretic approaches, and potential field techniques are augmented.

Chapter 4, exposes Y-Pod approach and its applications and discusses our proposed linear controller in order to form a Y-pod swarm formation in 2D environments. Several cases are discussed, such as static, linear, quadratic and combined. Moreover, to cope with switching

ISBN: / DL:

problems a fusion controller approach is proposed using sigmoid functions. All these topics are illustrated by means of simulation.

Chapter **5** exposes the extension to obstacle avoidance problem. Jacobian potential field approach is used by adding an extra term into the control law. The extension to 3D is exposed briefly but only for “floating robots”. Some results are described briefly.

Chapter **6** collects the final conclusion about the obtained results and the possible guidelines for future work.

ISBN: / DL:

Chapter 2

2. Background Information

This chapter briefly introduces the basic concepts that are behind the idea of using swarm systems to accomplish predefined tasks. Due to interdisciplinary approach of swarm robotics, we augmented the roots of swarm in step by step process such as: agents, mobile robots, swarm robots and inter-relationships among them.

2.1 Agents and Multi-agent Systems

Agents are one of the most prominent and attractive technologies in the engineering field. There are several universal consensus on the definition of an agent (Padgham and Winikoff 2004). However, the Wooldridge and Jennings definition is increasingly adopted and suitable for all areas. "An agent is a computer system that is situated in some environment, and that is capable of autonomous actions in this environment in order to meet its delegated objectives" (Wooldridge 2001). The definition could be applied to robotics as follows:

> *"an agent is an autonomous entity which can perceive through sensors and act through actuators"* (Russell and Norvig 1995).

They group agents into various classes based on their degree of perceived intelligence and capability. According to (Weiss 1999, Wooldridge 2009) an intelligent agent must meet the following three requirements:

- Reactivity: intelligent agents are able to perceive their environment, and respond in timely fashion to changes that occur in it in order to satisfy their design objectives.

ISBN: / DL:

- Pro-activeness: intelligent agents are able to exhibit goal –directed behavior by taking the initiative in order to satisfy their design objectives.
- Social ability: intelligent agents are capable of interacting with other agents in order to satisfy their design objectives.

An agent–to-agent interaction is basically the encounter of two robots with mutual influence. In a simple example, such an interaction occurs when robots A and B recognize each other as obstacles and they rotate away from each other to avoid a collision. This is called collision avoidance behavior. If only robot A perceives B this encounter could also be an interaction but with influence to robot A only. This is an inadequacy of the literal sense of "interaction" as B's contribution is only passive by entering the sensor range of robot A. For simplicity, the mere encounter of two robots is called collision.

Multi-agent systems are at least the historical basis for swarm robotics, just as they are for swarm intelligence. Multi-agent system is more than simply a group of agents: is a set of agents interacting with other agents in order to reach their design objectives. This grouping of agents constitutes a multi-agent system. In this regard, agent to agent interaction, agent to other agents and agent to environment interaction occurs on the task to perform the system tasks. In a microscopic description of a multi-agent system the individual are trajectories of agents and distances between agents. The macroscopic level of a multi-agent system is the group level: individual agents are not addressed, instead group fractions or densities. In fact, trajectories of individual agents cannot be determined.

A multi-agent environment in which agents operate can be defined in different ways. It is helpful definition to view the following as referring to the way the environment appears from the point of view of the agent itself (Ramchurn et al. 2004). Furthermore, multi-agent systems serve as models for robotic systems as, for example, in the robot soccer domain (Vetulani 2002). In multi-agent systems, agents perform the tasks in deterministic vs stochastic, static vs dynamic, discrete vs continuous, homogeneous vs heterogeneous and single vs multiple agents but not limited to applications, the possibility of accomplishing different tasks at the same time is of interest in robotic missions.

ISBN: / DL:

2.2 Mobile Robots and Multi-robot Systems

Mobile robotics can arguably be dated back to 1515, when Leonardo da Vinci presented a mechanical lion at the arrival of King Francis in Lyon. This lion possibly was nothing more than an automation based on a clockwork but able to move. At least in the 1940s a variety of projects began developing wheeled robots. Today we have a huge variety of autonomous mobile robots that are legged, wheeled, or have belts, and even some are also unmanned aerial vehicles and autonomous underwater vehicles.

A robot, in the loosest sense, may be defined as a physical agent conveying itself by perception and motor actuation by the way of some level of autonomy. An autonomous robot is one that can perform a specific set of tasks without human supervision as opposed to a tele-operated robot which requires constant human supervision. A robot that has full autonomous capabilities will embody some level of artificial intelligence so the robot can 'choose' the right actions and in some cases adapt and/or learn from its changing environment or its own choices. Nowadays, robots are used for service and industry applications (also military ones), especially those that are dull, dirty or dangerous. They can come in many shapes and sizes and can be classified into 3 groups: unmanned ground vehicles (UGVs), unmanned air vehicles (UAVs), and unmanned underwater vehicles (UUVs). Single mobile robot systems have been used for security (Everett 2003), medicine (Miyawaki et al. 2005), and domestic tasks (Jones 2006, Sahin and Guvenec 2007).

Localization challenge relates to all the tasks and behaviors associated with knowing the location of each robot in the swarm. This field also includes the areas of mapping and navigation. Localization has been studied extensively (Leonard and Durrant 1991, Rothermich et al. 2005). Research on localization concentrates on two major tasks. First, there is the task of localizing an individual robot using an a priori map of the environment. Second, there is the task of localizing a robot while building a map of the environment at the same time. Some research has been done on localization using multiple robots, which is called cooperative localization. Statistical and probabilistic techniques are the most common tools used in all of the methods researched. The use of landmarks and of methods based on the cooperation between various robots has been investigated.

ISBN: / DL:

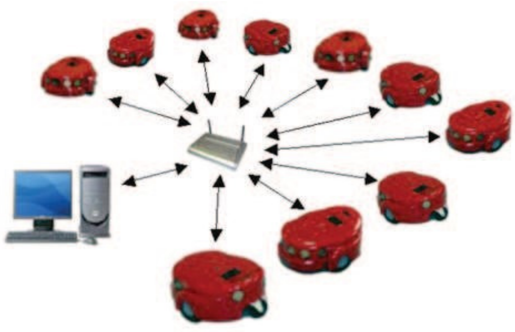
In some cases a single robot is not sufficient. For example, in the case of pushing objects (Mataric et al. 1995) the use of a single robot would be both unrealistic and inefficient. For these tasks, several mobile robots can be utilized to accomplish a task that would otherwise be very difficult or impossible for a single robot. A good overview of the work in multi-robotic systems can be found in (Jones et al. 2004, Arkin 1999, Goldberg and Mataric 2000). Concerning with this thesis, the major result of this research is that there are practical and effective strategies for knowing the location of all the robots in the system. If this were not so, then the use of many algorithms would be impractical because there would be no way to accurately measure the distances between the robots of the system. Some more research has been done in other application: distributed surveillance by using unmanned aerial vehicles (UAVs) in order to gather intelligence information (Parker 2008), distributed manipulation involving using multiple mobile robots in order to manipulate an object (Nouyan 2009).

Finally, other areas of interest include design and learning, sensor and hardware issues, planning and task allocation, large-scale robot teams, and communication constraint and networks (McLurkin 2008, Nouyan et al. 2009).

2.2.1 Multi-robot Systems

Multi-robot systems are of mounting importance as robotics research progresses. Many new systems and experimental platforms have been developed. Recently, several large scale multi-robot systems have appeared in the literature. Multi-robot systems are of interest for tasks that are inherently too complex for a single robot or simply because using several simple robots can be better than having a single powerful robot for each task (Cao 1997). Groups of mobile robots are used for studying issues such as group behavior, resource cooperative robot teams and (ii) swarms of robots (Parker 1998). Conflict, and distributed learning. These systems have been used for a variety of tasks including but not limited to “Robocup Soccer” (Shmilovici et al. 2004); search and rescue (Sugiyama et al. 2006); terrain coverage (Zheng et al. 2005); foraging (Sugawara and Watanabe 2002); and cooperative manipulation (Mataric et al. 1995). These robots may be identical or they may contain a multitude of varying systems ranging from slightly different sensors to entirely distinct hardware and/or software platforms. Multi-robot systems can further be classified into two groups. In the first group, cooperative robot teams, each robot generally has different capabilities and control algorithms which when combined can be used to complete a task. In the second group, each robot generally has identical function and capabilities with the goal being the overall group behavior.

Architectures: The architecture of a multi-robot system provides the framework upon which missions are implemented, and determines the system functionality and boundaries. The most basic decision that is made when designing system architectures is defining the type of control the system will utilize. System control techniques of autonomous robots include: centralized control in which individuals receive commands from a central controller, and decentralized control where local control laws operates in individual robots producing a desired global and/or emergent behaviour. In centralized control methods (Cao et al. 2003, Zelinski et al. 2002, Egerstedt and Xiaoming 2001), a single computational unit oversees the whole group and plans the motion control of the group accordingly.



(a) Centralized architecture of the "Decabot"

(b) Example of a decentralized architecture "Project".

Fig.2.1: Applications inspired from nature to swarm robots

Fig. 2.1(a) shows an example of a centralized architecture with the block in the middle being the main control block, all robots communicating through this block. In centralized control methods, the entire multi-robot system is dependent on one controller, so these methods are not very tolerant to failure. Decentralized control methods (Balch and Hybinette 2000, Lenord and Fiorelli 2001, Sugihara and Suzuki 1996) lack a central control unit and follow two forms: distributed control when all robots are equal with act to control and hierarchical control when control is locally centralized (see Fig.2.1(b)). In decentralized control methods,

ISBN: / DL:

the desired behaviour is produced using only local control laws operating on individual robot members. These local control laws depend on the specific model and methodology used. A key feature of centralized control is that each of the robot members communicates directly with each other. Decentralized control methods are advantageous over centralized ones in that they are more fault-tolerant, scalable, and reliable.

Interaction among robots: Multi-robot systems may either be made up of homogeneous or heterogeneous units. A group of robots is said to be homogeneous if the capabilities of the individual robots are identical and heterogeneous otherwise. Any difference in software or hardware can make a robot different from another. Higher levels of heterogeneity introduce more complexity in task allocation since robots have different capabilities. This requires robots to have a greater knowledge about each other. In a heterogeneous system, it is necessary to prioritize a robot's tasks based on its capabilities, whereas in a completely homogeneous system all robots have equal capabilities and priorities. The team size can either hinder or help depending on the task and team composition. Within a multi-robot system, robots may communicate following several information structures (Dudek et al. 1993, Cao et al. 1997) including: (i) interaction via the environment, (ii) interaction via sensing, and (iii) interaction via communications Interaction by means of the environment involves using the surrounding features as the communication medium. An example of such interaction would be some type of landmark navigation (Fukuda et al. 1995). Interaction via sensing involves using sensory data such as range measurements to sense other robots (Mataric 1992, Balch and Arkin 1998, Lietmann 1981), and interaction via explicit communication through either directed or broadcast messages (Stefano and Antonelli 2003, Stefano et al. 2005), where attention is also focused on the possibility of using a behavioral approach with assigned priority-levels. More recent works point out the idea of controlling a homogeneous group of mobile robot based on *graph control theory*. This is a fully decentralized approach applied to achieve goals like herding, leader-based optimal control and formation keeping or flocking.

Geometry: Coordination and Formations includes all the strategies used by the swarm in order to make the individual robots work together in order to perform a certain task (Parrish 2002). The coordination strategy must maintain the cohesion of the swarm and prevent collisions between the swarm members. It must also allow the robots to act in such a way that they can perform the actions that they need to perform in order to accomplish their assigned task. The two strategies for coordination are the centralized approach and the distributed approach (Kazadi 2009).

In the centralized approach, a central planner plans and controls the actions of all the robots. In the distributed approach, each agent is autonomous and controls its own actions. Some research has been done on economy-based architectures that use negotiation to coordinate a multi-robot system.

Micro-robot systems: The word “micro” in micro-robotics has two meanings here. First, it indicates the small size of the robots, which is today, however, far above micrometres and rather in the millimetre’s or centimetre’s range. Second, it indicates the high path accuracy that can actually go down to nanometre’s scale. The connection to swarm robotics is drawn because a small overall size of swarm robots is at least beneficial if not necessary. Otherwise, it would barely be feasible to deploy many dozens or even hundreds of robots under laboratory conditions. Micro-robotics served as the stepping stone to swarm robotics (Fatikow and Rembold 1993, Fatikow et al. 2000, Worn et al. 2000).

As an example, we want to highlight the MINIMAN, the MiCRoN, and the I-SWARM project (see Fig. 2.5). These projects show the transition from just building small robots to small groups of small robots, and finally to large groups or swarms of small robots. According

to the official statements (Fahlbusch et al. 1999) the main idea of the MINIMAN project was the development of a smart micro-robot with five degrees of freedom and a size of a few cubic centimetres, capable of moving and manipulating by the use of tube-shaped and multi-layered actuators. Equipped with micro-machined grippers, the robot takes over high-precise grasping, transport, manipulation and positioning of mechanical or biological micro-objects, under a light microscope or within the vacuum chamber of a scanning electron microscope.

The MiCRoN Project was to develop a system that is based on a cluster (five to ten) of small (in the size of a few cm3) mobile autonomous robots. These wireless micro-robots, each equipped with on-board electronics for control and communication, cooperate to accomplish a range of tasks associated with assembly and processing from the Nano- to the micro-range.

In the I-SWARM project the goal was to produce a “true” (robotic) self-organizing swarm concerning the size of the swarm and the size of the individual robots (Seyfried et al. 2005). The idea was to establish a mass production for autonomous robots making, in the production of numbers as high as 103 feasible. The robot itself has a size of just 3mm × 3mm × 3mm, is equipped with four infrared emitters and sensors, three vibrating legs, and a solar cell as energy supply as well as additional sensor (see Fig. 2.5).

2.3 Swarm Robots

In the last decades researchers have discovered the variety of the interesting insect or animal behaviours in nature (see Fig. 2.2): a flock of birds’ sweeps across the sky, a group of ants’ forages for food, a school of fish swims, turns, flees together, etc. (Berder 1954, Bonabeau et al. 1999). We call this kind of aggregate motion “Swarm behaviour”. Recently biologist and computer scientists in the field of “Artificial life” have studied how to model biological swarms to understand how such “Social animals” interact, achieve goals, and evolve. A social insect colony usually consists of thousands of individuals which are able to do many sophisticated jobs without a centralized control mechanism.

In fact, social insects work autonomously and their teamwork is essentially self-organized. Coordination between individuals arises from different interactions between insects or between insects and environment (the stigmergic mechanisms). Although these interactions might be primitive, as a whole they result in efficient solutions to difficult problems such as finding the shortest route to a food source among myriad possible paths.

(a) Flock of birds with V-shape formation

(b) Ant collaboration and cooperation each other's

(c) Butterfly fishes schooling.

(d) Four lions collaborate during hunting.

Fig.2.2: Examples from nature

As well as the animal behavior has evolved in collaborative direction in order to increase the survival probability of the species, researchers start thinking that maybe a multi-robot system can be modelled as a swarm system, where all the agents collaborate such that the chances to accomplish a predefined task is increasing.

Swarm robots can be defined as follows:

> *"Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed in such a way that a desired collective behaviour emerges from the local interaction among agents and between the agents and the environment"* (Sahin 2005).

ISBN: / DL:

Starting from these considerations derived from natural systems, researchers have identified two main collaborative paradigms: the intentionally cooperative systems and collective cooperative systems. The idea behind the first paradigm is that all the robots in group have knowledge about the presence of other team members, and are able to coordinate each other exploiting global information like the state and the capabilities of teammates. And the idea relying to collective ones is that the robots have local knowledge but interaction mechanisms drive to emergent behaviour. Swarm robots systems is the intersection among three main topics (Muniganti and Oller 2010): swarm intelligence, bio-robotics, and self-organized systems (see Fig. 2.3).

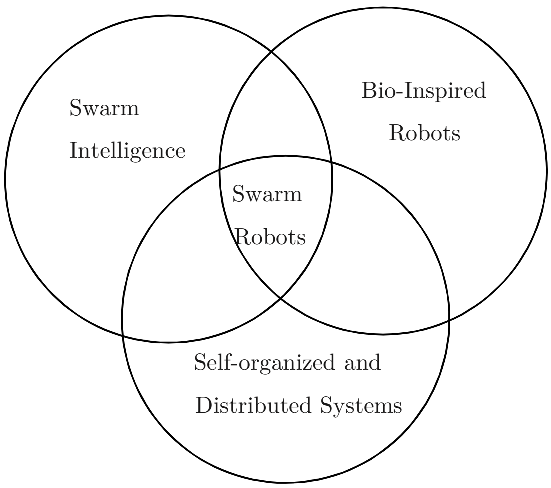


Fig.2.3: Swarm robots are intersected with various fields

There is research on scalable decentralized systems. In fact, drawing the borderline between multi-agent systems and software simulations of swarm robotic systems is challenging because the transition from large decentralized multi-agent systems to swarms is blurred. This is true, especially, for scalable swarm robotics. Thus, the software and the modelling subfield of swarm robotics can be regarded as multi-agent systems with emphasis on scalability and biological inspiration (see Fig. 2.4).

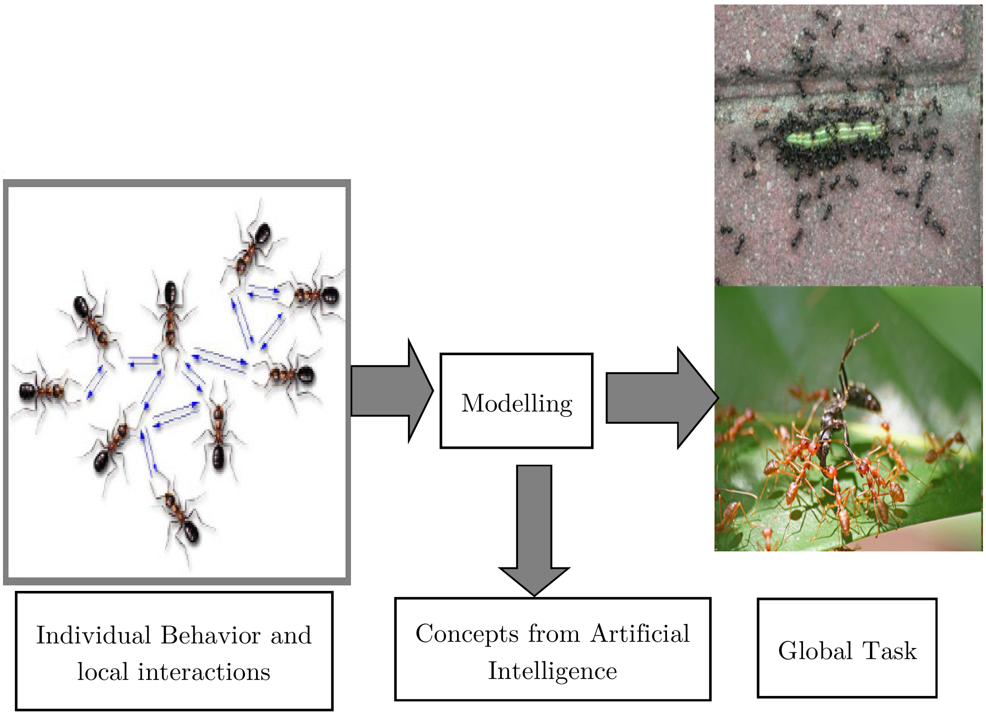


Fig.2.4: Examples of nature formation in swarms and its analogy into engineering.

A swarm is a large system of agents/robots that are all working on a single task (Farinelli et al. 2004). These systems have a degree of autonomy in the same way that the individual agent/robot does. These swarms of robots or autonomous vehicles have significant advantages over conventional kinds of teams. In conventional teams, there is usually a very complex central controller that uses deliberative control strategies to control the behavior of the team. Thus, the controller is complex and the system is not very flexible or scalable. If one robot fails, often the system will no longer be able to function properly. On the contrary, swarms are based on groups of simple and reactive agents. The agents can perform a small set of simple behaviors and can sense and react to their environment and to each other. The interaction of the individual robots allows more complex behaviors to emerge. Thereby, the swarm can perform complex tasks while using very simple control strategies for each agent and the group as a whole. In fact, the macroscopic modelling and control architectures of swarm robotic systems is a major area of research in robotics (Gonzales et al. 2004).

The “Nerd Herd experiment” was one of the first experiments on swarm robotic systems (Mataric 1995) dealing with a decentralized control architecture, where 20 robots use very little explicit communication rather than *stigmergy* (i.e. communication through environment modification). Nowadays, experiments perform foraging, dispersion, surrounding, and herding (see Fig. 2.5): each single robot is supposed to have very minimal capabilities in terms of sensing and acting on the environment, but they are able to collaborate in order to show complex behaviors. Systems like that, where simple agents can perform complex goals, are called *superadditive*, meaning that the whole can do more than the single. Recent experiment “SwarmBots” (McLurkin 2004) performed with 100 robots (see Fig. 2.5) has pointed out the possibility of merging different simple tasks in order to achieve complex goals.

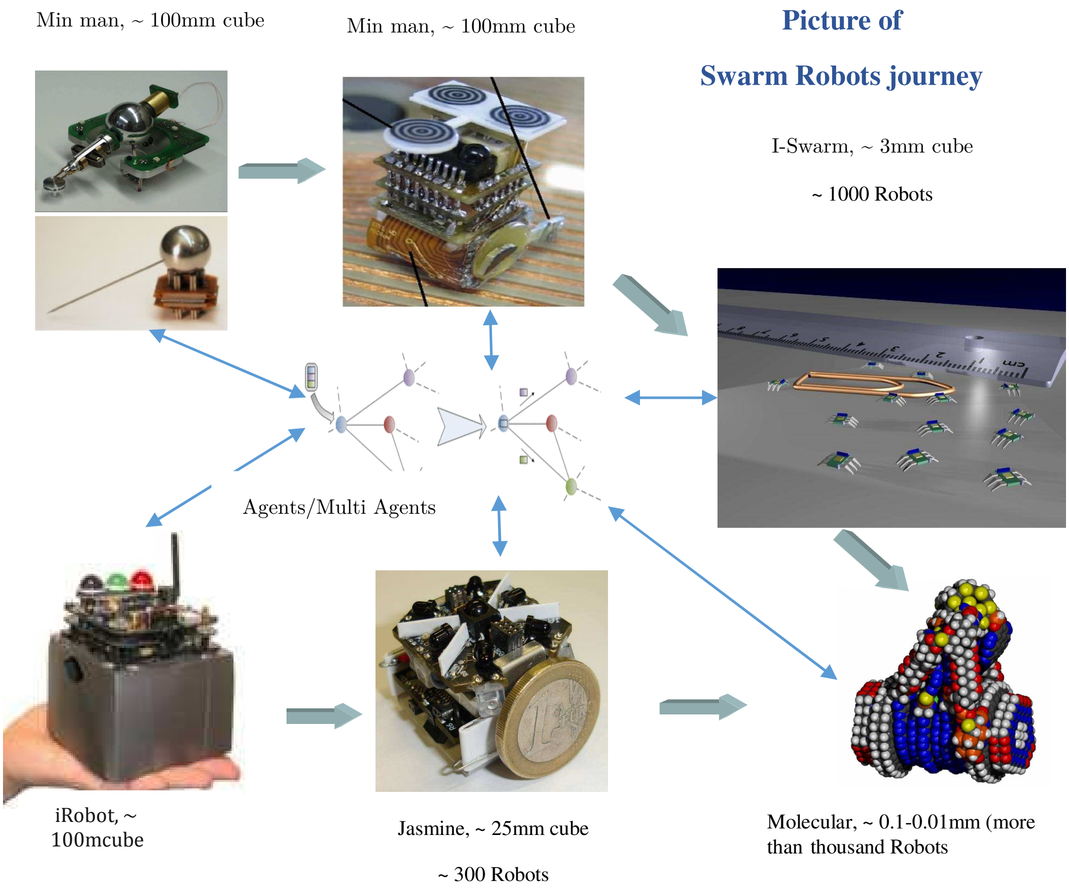


Fig.2.5: Picture of swarm robots from agents, mobile, micro and swarm robots

ISBN: / DL:

There is much recent work related to real applications and technical issues. However, there are still different approaches to swarm system architectures. Here follows a brief discussion.

Centralized: a global supervisor receives information from swarm members and the swarm behavior is coordinated from a single control point. This architecture found in (Milatinovic and Lima 2006, Bekey and Khoshnevis 1998) has the advantage that as all the information is collected by a single unit, this *node* of the system can be powerful enough to calculate the control law for each robot, considering also the opportunity of having complex tasks.

Decentralized: it is the most used architecture to control swarm robotic systems, and can be considered as the opposite of the centralized approach. In a decentralized architecture, each robot acts based only on knowledge of local teammates' state and of environment. This subsumption approach is robust to failure but, on the other hand, presents limits to power computation, which means that it can be non-trivial to implement complex tasks in distributed fashion. An example of such architecture is Swarmbot experiment (see Fig.2.6), where 18 robots organize themselves in order to drag a body in a hypothetic disaster scenario (Machael et al. 2007).

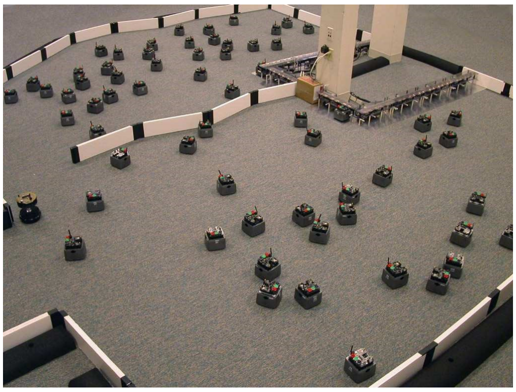


Fig.2.6 A picture of the Swarmbot experiment

Hierarchical: this architecture, directly inspired by military command protocol, is suitable for some civil applications. It is based on the idea that some robots can command as supervisors of a local and relatively small group of team members (see Fig. 2.7). Once again, as in the centralized approach, weakness of this architecture can be found in recovering from failures of robots high in the command tree.

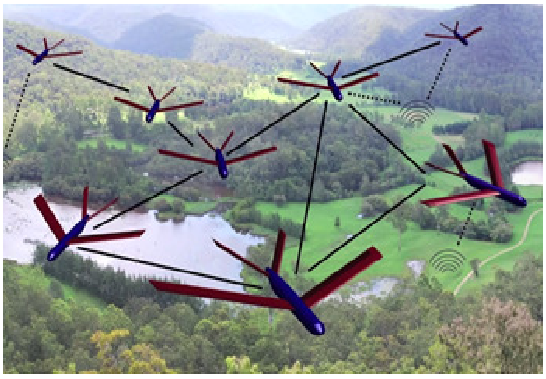


Fig.2.7: Example of hierarchical robotic swarms: airplanes in formation, or networked sensors.

Hybrid: this approach tries to be a trade-off between the centralized and decentralized architecture. In particular, it is based on the idea that one or more high level supervisors allocate tasks and resources, and single low level robots exploit local information to accomplish a predefined task. The hybrid architecture has been used in many applications based on multi-robot control (Hugli et al. 1998, Jacobsen 1998).

Since researchers are working on groups of mobile robots with homogeneous and heterogeneous groups (see Fig. 2.8), their attention focuses on the possibility of obtaining a large spectrum of emergent behaviors such as manipulation, traffic control, foraging, coverage, flocking and formation keeping. In particular, the last two have attracted researchers' attention for many reasons. Among them, the fact that they can be used to model the behavior of social insects like ants (*foraging*) or social animals like birds (*flocking*).

(a) A flying quad-rotor UAV (b) Khepera III robot by K-Team (c) E-Puck robot by Cyber-bot
Fig.2.8.Examples of robots used in real swarm robotics experiments

ISBN: / DL:

In a foraging domain, where a very large number of mobile robots are involved, objects like pucks are distributed in the environment and the global goal is to find and collect all them. The main issue in this application is how to coordinate robots in order to explore as fast as possible all the terrain without interfering with each other. Thus, *foraging* is strictly related to the *coverage problem*, whose solution can be applied to real world problems such as mapping, surveillance, industrial surface cleaning, demining, search and rescue, or toxics waste removal (Hollis et al. 2000, Bruckstein et al. 2000, Wyman et al. 1997). In these cases we can talk of *weakly cooperating systems*: the solution of the foraging and coverage problem usually requires minimum communication between teammates. Furthermore, flocking and formation keeping are global goals that can be considered, such as different realizations of the same basic problem that is to find a way to coordinate the movements, in a similar way as in foraging domain.

The most recent architectures point out by researchers in order to control a homogeneous group of mobile robot is based on *graph control theory*. This fully decentralized approach, initially used on groups of massless-point agents, is based on concepts borrowed from the graph theory, and it exploits the *Laplacian solution* to the *consensus problem* (or *rendezvous problem*) in order to achieve goals like herding, leader-based optimal control and formation keeping or flocking.

ISBN: / DL:

Chapter 3

3.Related Work of Robot Formation

This chapter presents an overview of the related work in swarm formation. Particularly, swarm formation related works with respect to various shapes, formation control methods, behaviour based and potential field based swarm formations are presented.

A robot formation can be defined as a "group of robots that moves as a collective maintaining pre-determined positions and orientations among its members". Having a pre-defined global shape is not always a requirement, and sometimes only relative positions between the members are defined. We can say that robots in formation are required to keep a fixed distance and angle relative to other robots in the group.

A comparative study on existing multi-robot and swarm formation methodologies is presented, but the formation is discussed and compared including both specific geometric formations and flocking formations. Only multi-robot systems focusing on formation control are included in this comparison. The formation control strategies are analysed from their control methods and shape representations.

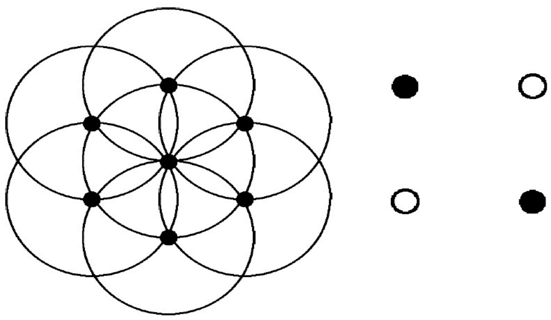
3.1 Formation control methods

The most fundamental and key aspect of formation control is the method of control used in the multi-robot or swarm system. The control method follows three different types, including (i) centralized, (ii) completely decentralized, and (iii) hybrid. The majority of the approaches are hybrid. The robots need to maintain a specific formation shape while maintaining the correct formation position among other robots. In this section different control and shape

ISBN: / DL:

representation methods are surveyed. In formation control for a group of robots, different control topologies can be adopted depending on the specific scenarios and/or missions. There may be one or more leaders in the group, while the other robots follow one or more leaders in a specified way. Each robot has on-board sensing and computation abilities.

In some applications, robots only have limited communication ability. In general, global knowledge about the system is not available to each robot. A centralized controller is not utilized, and in this case the design of each robot controller has to be based on local information. If there is no assigned leader, then each robot must coordinate with the others by relying on some global consensus to achieve the common goal. Various types of shapes have been employed in formation control. The specific shape might be scenario or mission dependent. The more common formation shapes are columns, lines, wedges, triangles, and circles. Some works use lattices (Chou and Feng 2008, Jeong and Lee 2014), but are made of those basic shapes mentioned above (see Fig. 3.1):



(a) (b)

Fig. 3.1: Lattice formations (a) Hexagonal, made of triangles (b) Square, made of lines and columns

The concept of formation control has been studied extensively in the literature with applications to the coordination of multiple robots (Ando et al. 1995, Barfoot and Clark 2004), unmanned air vehicles (UAVs) (Chaimowicz and Kumar 2004, Koo and Shahruz 2001,Shao et al. 2006), autonomous underwater vehicles (AUVs) (Kalantar and Zimmer 2004, Shao et al. 2006), and spacecraft (Beard et al. 2000, Lawton and Beard 2000, Saaj et al. 2006). Four main control frameworks have emerged to address the swarm formation

ISBN: / DL:

problem including the behavior-based and potential fields, leader-following, graph-theoretic and virtual structure approaches.

The following sections summarize relevant work in formation control techniques such as leader-follower, behaviour-based, graph-theory, virtual control, and some other techniques are included.

3.1.1 Leader-follower

The leader-follower (Mariottini et al. 2007, Shao et al. 2007) approach is decentralized,where each robot is assigned a unique identifier (ID). The robots typically try to maintain some desired distance and desired relative angle to some of their neighbours and/or virtual points. Some robots are designated as leaders while other robots are designated as followers. There can be as few as one leader, or there can be several leaders with a hierarchical structure. The leaders are generally tracking a predefined trajectory and the followers are tracking their leader(s) in some manner dependent on the algorithm. Such algorithm for maintaining formations utilizes a leader-follower or graph-based method.

In (Fredslund and Mataric 2002), local sensing and minimal communication between agents is used to maintain a predetermined formation. Each robot in the group keeps a single ‘friend’ robot which it perceives via a special sensor, and maintains a specific angle at all times in relation to this ‘friend’. Broadcast communication is utilized but is minimal, only sending robot IDs, directional changes, and formation messages. Each robot has access to the number of robots in the system as well as the type of formation. With each formation, each robot has a specified angle to keep between its friend and the frontal direction. This algorithm is limited to only formations which can adhere to the chain of friendship which limits it to no more than two loose ends or frontally concave formations.

In (Egerstedt and Xiaomig 2001), a coordination strategy for maintaining formation over a given trajectory is presented. The formation control is achieved through the tracking of virtual reference points. The leader of the path acts as a reference point for the robots to follow. The robots move in a triangular formation avoiding obstacles, and if the tracking errors are bounded then the formation error is stabilized.

ISBN: / DL:

In (Leonard and Fiorelli 2001), artificial potentials define the interaction control forces between adjacent vehicles and define the desired inter-vehicle spacing. The approach is inspired by biology considering attraction and repulsion to neighbours as well as velocity matching. Virtual leaders or beacons (not an actual vehicle) are used to manipulate group geometry and motion direction. Constant prescribed formations of schooling and flocking are demonstrated, but the approach is only applicable to homogeneous formations. Closed-loop stability is proven with Lyapunov function using kinetic and potential energy of the robot system.

In (Kostelnik et al. 2002), an approach is provided for multi-robot formations including obstacle avoidance using local communication and sensing. The approach is behavior-based but integrates social roles representing positions within the formation using local communication to improve performance. New agents are allowed to join the formation by role changes when necessary. The local communication is fixed, and the locally information travels to the leader which knows the entire shape of the current formation and decides on necessary role changes. This information is then passed back to the necessary followers, updating the information. The roles or positions for the robots are decided dynamically and changed as the formation grows.

In (Desai et al. 1998), leader-follower patterns are used for formation control. In this approach, it is assumed that only local sensor based information is available for each robot. In (Sisto and Gu 2006), a fuzzy logic leader-follower approach is presented for formation control. Maintaining correct formation position while avoid collisions is investigated here. Separate fuzzy-logic controllers are developed for formation position control and internal collision avoidance. Circle, wedge, line, and column formations are presented.

In (Olfati-Saber and Murray 2003), the authors show a graph theory called graph rigidity which is very helpful in representation and control of formations of multiple vehicles. These rigid graphs identify the shape of the formation and the interconnections lead to automatic generation of potential functions. The basis is that performing graph operations allows the creation of larger rigid graphs to be formed by combining smaller rigid sub-graphs. This

ISBN: / DL:

work has specific applicability in the area of dynamic reformation as well as splitting and merging of vehicles in a distributed manner.

In (Desai 2001), a graph theoretic framework for formation control of moving robots in an area containing obstacles is presented. Control graphs are used to determine the behavior and transitions that are possible between different formations or control graphs. Each team's model consists of a lead robot, shape variables containing relative positions of robots, and a control graph that describes the behaviors of the robots in the formation. This method is scalable to large groups despite the computational complexity of growing control graphs due to its decentralized design.

In (Fierro and Das 2002), another graph-based approach consisting of a four-layer hierarchical modular architecture for formation control is proposed. The group control is at the highest level layer generating a desired trajectory for the whole group to move. The next layer manages formation control implementing a physical network, a communication network, and a computational network (control graph). The formation is maintained by using only local communication and relative position. Next, there are two layers, one to control robotic kinematics and one to handle robot dynamics. This system is very scalable to heterogeneous systems because of the layered approach with the adaptable kinematics and dynamic layers. This method also allows for various formations and both centralized and decentralized methods of control graph (Fierro et al. 2001) assignment are described.

In (Jadbabaie 2003), nearest neighbour rules are used to control the motion of the robots updating each robot's heading based on the average of its heading plus its neighbors' headings. Undirected graphs are used to represent robot interactions. This method claims that, despite the absence of a centralized coordination mechanism and the dynamic neighbour changes, there will be an overall emergent coordinated motion. No particular formation is exhibited but overall robot motion is in the same direction. While the leader-follower and graph-theoretic approaches are logical and easily implemented, there are limitations. Each leader is a single point of failure for the formation so this makes these systems weaker than completely decentralized systems. Reassigning leadership and information flow in the event of a failure can be difficult and computationally expensive. In

ISBN: / DL:

addition, if there is no explicit response from the followers to the leader, and if the follower has a difficulty, the formation cannot be maintained.

3.1.2 Behavior-based systems and Artificial Potential Fields

Behavior-based systems integrate several goal oriented behaviors concurrently in order to reach a goal. In the behavioral approach to formation control (Cao et al. 2002, Dudenhoeffer and Jones 2000, Balch and Arkin 1998, Belta and Kumar 2004, Liu et al. 2006, Reif and Wang 1995), each agent has several desired behaviors, and the control action for each swarm member is defined by a weighting of the relative importance of each behavior. In addition there are many formation control strategies which utilize potential fields (Balch and Hybinette 2000, Elkaim and Kelnley 2006, Yao et al. 2006). Behavior-based methods and potential fields are often combined in formation control of mobile robot systems.

In (Cao et al. 2002), a behavior based formation control method is used in which a leader is referenced to determine formation position. Each of the robots is equipped with some primitive motor behaviors. The behaviors have control parameters which are tuned using a genetic algorithm. During the motion, the leader decides its next position based on its knowledge about the goal and environment and then broadcasts its anticipated position to the followers. The use of genetic algorithms for optimizing the formation control is interesting, but the drawback is that the system requires almost global knowledge about the environment and is dependent on receiving this via broadcast communication. In (Balch and Arkin 1998), the behavioral approach is applied to formation-keeping for mobile robots, where control strategies are consequent of several simultaneous behaviors. In this approach, line, column, diamond and wedge formations are presented. For each formation, each robot has a specified position based on an identification number.

In (Lawton et al. 2003), complex formation manoeuvres are broken down into a sequence of behaviors to achieve formation patterns. A bidirectional ring topology is used to maintain the formation of the system. The advantage of this approach is that it can be implemented when only neighbour position information is available. Because of the way formation patterns are defined, this approach is limited in directing rotational manoeuvres for the group.

In (Lawton and Beard 2000), a behavioral-based approach is used to obtain formation control laws to maintain attitude alignment among a group of spacecrafts. The approach utilizes velocity feedback and the other passivity-based damping. Behavior-based methods and potential fields are often combined in formation control of mobile robot systems (Balch and Arkin 1998, Cao et al. 2003). In these approaches, each robot has basic motor schemas which generate a vector representing the desired behavioral response to sensory input. These motor schemas include behaviors such as obstacle avoidance and formation keeping.

In (Balch and Hybinette 2000), a strategy to arrange a large scale, homogeneous team in a geometric formation utilizing potential functions is presented. The method is inspired by and is similar to the process of molecular covalent bonding. Various robot formations result from the usage of different attachment sites. Attachment sites are constructed relative to the other agents in the team. Formation is maintained in the presence of obstacles. Local sensing is sufficient to generate and maintain formation. Robots are not assigned specific locations, but attracted to the closest teammates or attachment sites. Behaviors such as "move to a goal" and "avoid an obstacle" are utilized for robotic control. In (Dudenhoeffer and Jones 2000), formation control is achieved via a group formation behavior based on social potential fields. The robot's behavior is based on a subsumption architecture where individual behaviors are prioritized with respect to others. This work extends the work in (Reif and Wang 1995) using the social potential fields method by integrating agent failure and imperfect sensory input. This method uses only local information and is scalable to very large groups of robots.

In (Monteiro and Bicho 2002), the behavior-based formation control is modelled by a non-linear dynamic systems for trajectory generation and obstacle avoidance in unknown environments. The desired formation pattern is given through a matrix which includes parameters to define the leader, desired distance, and relative orientation to the leader. These parameters are then used to shape the vector field of the dynamical system that generates the control variables.

In (Ge and Fua, 2005, Ge et al. 2004), the desired formation pattern is represented in terms of queues and formation vertices. The desired pattern and trajectory for the group of robots

ISBN: / DL:

is represented by artificial potential trenches. Each robot is attracted to and moves along the bottom of the potential trench, automatically distributing with respect to each other. Although the behavior-based approach has the advantage of formation feedback through neighbour-based communication and it is highly decentralized; it is extremely difficult to analyse mathematically and has limited ability in precise geometric formation keeping. If a very precise formation is required, another method, such as a virtual structure method, should be used.

3.1.3 Graph theory

In more recent years, graph theory has been studied as a new way to solve many different problems, such as traffic routing problems (Ogunsanya 1986, Diestel 1997), payload transport, task assignment, air traffic control and many other applications, included robotics. There are two main paradigms to achieve formation for multi-vehicle systems: the first approach is based on the idea of using a rigid structure to represent the desired formation and to control the robots behavior basing on inter-vehicle potential fields; on the contrary, the second one is based on the idea of representing the group of vehicles as a graph (where communication links are represented by edges) performing. Graph-based works with changing shapes problems are explained in (Desai 2001, Olfati-Saber and Murray 2002, Desai 2002, Olfati-Saber et al. 2003).

3.1.4 Virtual frame control

Virtual frame formation control plays a vital role due to its virtual structure .In the virtual structure approach (Fujibayashi et al. 2002, Lewis and Tan 1997, Tan and Lewis 1996), the entire formation is treated as a single rigid body. The concept of the virtual structure was first introduced in (Tan and Lewis 1996). The virtual structure approach is typically used in spacecraft or small satellite formation flying control (Beard et al. 2000). The virtual structure can adapt its shape expanding in a specified direction while maintaining a rigid geometric relationship among multiple agents. These approaches were proposed to acquire high precision formations control for mobile robots.

ISBN: / DL:

In (Fujibayashi et al. 2002), a virtual structure method is proposed for self-organizing formation control in which it is assumed that elements are not connected to each other and can move in a continuous space. The goal is to arrange the elements into the spatial pattern of a crystal using virtual springs to keep neighbouring elements within close proximity. Each pair of elements within a certain range is connected with a virtual spring. The elements form triangular lattices with random outlines.

In order to determine the desired outline, virtual springs are broken with a certain probability. The candidate springs for breaking are chosen based on the connections of its neighbors. Elements interact locally and have no global information, but the tuning of parameters for different formations and number of robots is computationally expensive. The main advantage of the virtual structure and graph theoretic approaches to formation control is that it is simple to prescribe the behavior for the entire group. The formation structure is generally very tight and precise in these methods during tasks. The main disadvantage is in the computation complexity of some of these methodologies, as well as the centralized nature which make these systems less robust to failure.

3.1.5 Other formation control strategies

There exist several formation controls in various applications which are closely related to swarm robots. In this scenario we propose a glance on researchers' works. There are also many other formation control strategies which do not easily fit into the categories previously discussed. In (Lindhe et al. 2005), a distributed coordination algorithm is presented for multi-robot systems in which a particular method of navigation function with Voronoi partitions is used. The robots navigate, maintaining a flocking formation, while avoiding obstacles. Inter vehicle communication is achieved by using a global list of positions where every single vehicle can only get a list of its neighbours within a specified radius.

Although For some special tasking, if robots can form a specific geometric pattern or formation, they can interact with others and/or perform the task more efficiently (Sugihara and Suzuki 1996 Leonard and Fiorelli 2001). Hence, the formation control becomes an interesting research topic in swarm robotic systems. In order to form a specific formation

ISBN: / DL:

pattern, the interaction between robots should be well regulated. When characterising the system dynamics of a swarm robot team, related properties of a formation such as stability and connectivity are also important, and should be analysed, and robots must maintain the formation shape while performing some tasks.

Formation keeping can be viewed as the maintenance of relations among robots. For instance, more complex formation like a regular hexagon can be combined by the three basic formations. Detailed mathematic preliminary about non-linear attractor dynamics approach was proposed by Godenstein in (Godenstein et al.1999). Some researchers also worked in different strategies i.e., robots identified their neighbour robots and environmental characteristics via the result of processed images (Das et al. 2002a, Fierro et al. 2001 and Tanner et al. 2004).

In (Zelinski et al. 2003), the formation problem is solved for a group of autonomous vehicles by providing inter and intra vehicle constraints as well as a time limit for reconfiguration. The nominal input trajectory for each vehicle is determined so that the group begins in the initial position and ends in the final position in the specified amount of time. The information is represented as a particular form of input signals so the formation problem can be reformulated as an optimization problem and solved more efficiently especially for large groups of vehicles. However, this method suffers from a single point failure, since it utilizes a central controller.

In (Dunbar and Murry 2002), the stabilization and manoeuvring of vehicles is achieved through model predictive control. Each individual vehicle may be governed by nonlinear and constrained dynamics. The vehicles are stabilized to acceptable equilibriums rather than precise locations for each individual. The individual trajectories of autonomous vehicles moving in formation were generated by solving an optimal control problem at each time step. This is computationally demanding and hence not possible to perform in real-time. In (Yamaguchi 2001), a distributed control scheme for multi-robot systems is presented. Each robotic vehicle has its own coordinate system, and it senses its relative position and orientation in reference to others in order to create a group formation. Despite the presence

ISBN: / DL:

of a supervisor, the robot vehicles are stabilized. The stability of the vehicles is proven only for symmetrical formations.

In (Sughiara and Suzuki 1996), approximate pattern formation is achieved by sharing position information to other robots. In this method, it is assumed that robots have global position information. An algorithm is developed for each pattern formation which includes circles, polygons, lines, filled circles, and filled polygons. Robots can also be split into an arbitrary number of equal or near equal group size. Although this method is decentralized, the information sharing of the global position of each group member to the whole group is a significant drawback.

In (Kowalczyk 2002), a target assignment strategy for formation building of multiple robots scattered in the environment is presented. The algorithm first begins with assigning each robot a target point in the desired formation. Trajectories, including collision avoidance, are generated by a central planner. Priorities of areas around the robots are integrated so robots will avoid each other when in a certain threshold. Sensing is global and the method is dependent on a central controller. In (Spry and Hedrick 2004), a formation control methodology based on generalized coordinate system is presented. The generalized coordinates characterize the vehicle's location, orientation and the shape of its formation. This allows the group to be controlled as a single entity. Force-based and velocity-based controls are developed. Similar ideas utilizing coordinated systems for shape representations are presented in (Yamakita and Saito 2004, Zhang et al. 2003).

In (Koo and Shahruz 2001), a hierarchical, centralized planning method to achieve a desired formation for a group of UAVs is presented. The desired flight trajectories for each UAV are determined by a leader which is more capable than the other team members. To achieve flight formation according to a given scenario, each UAV independently takes off towards its corresponding trajectory and locks onto it in finite time. Only the leader is equipped with sensors, and it communicates to the other team members what trajectories to track via a communication channel. This method is very prone to failure, is very risky in dynamic environments, and scales very poorly with growing team sizes. In (Chaimowicz and Kumar

ISBN: / DL:

2004, Chaiomowicz et al. 2005) swarms of unmanned ground vehicles (UGVs) are coordinated with the use of unmanned aerial vehicles (UAVs).

In (Chaiomowicz and Kumar 2004), a hierarchy is formed between the UAV and the UGVs. The UAV is in charge of determining the grouping and merging of swarms as well as the swarm distribution and motion of the group. The UGVs are at the lowest level of the hierarchy, and at the highest level there is a centralized planner for the whole system, a single UAV. This system is centralized with each robot communicating to its central planner. The shape of the formation is determined by the central planner in the form of a directed graph. Due to the dependence on a central planner, this method is prone to failure, but the UGV swarm-UAV coordination proves interesting. There are also many other methods in the application of formation control.

In (Kobayashi et al. 2003), genetic algorithm and reinforcement learning are used for robot formation control and obstacle avoidance. In (Hirota et al. 1995), neural networks and radial basis functions are used to achieve formation control. Vision is used for formation control in (Das et al. 2002, Marottini et al. 2007, Michaud et al. 2002; Vidal et al. 2003, Moshtag et al. 2006).

3.1.6 Comments

There are many ways to describe an object or behaviour, and the way we are doing this is through mathematical models. This helps us to understand the object whether it is a bird or a robot. When working with robots we can use these mathematical models to both predict and control its actions based on a set of data given by a controller and a set of sensors.

The direct approach in programming a robotic swarm is insufficient due to the huge amount of robots that appear in such systems. In order understand the system behaviour and characteristics and to identify the specific parameters, we need to have a specific mathematical model to understand the system viabilities. This thesis proposes to overcome these challenges by supporting the design process with method. When applying rules extracted from natural systems to artificial problems we need specific mathematical models to understand the internal mechanism of the systems with respect to real and simulation environments (Muniganti and Oller 2010). Natural systems have flexibility, scalability and reliability but artificial systems are not. This is the reason why the mathematical model can play a vital role in modelling swarm systems. Concerning with swarm robot systems, individual robot behaviour is simple but they exhibit complex behaviours for the desired tasks. We need specific parameters to understand the whole behaviour of the system. Real experiments are very expensive and time-consuming, and also have some problems in size, noise and other environmental issues for the overall system performance (Barraquand and Latombe 1989). Using the mathematical analysis we can rapidly and efficiently study the swarm systems in order to understand the behaviour of the system with simulated environments.

ISBN: / DL:

Chapter 4

4. Y-Pod Swarm Formation in 2D Environment

In this chapter, Y-Pod and its applications are explained. Also proposed linear controller to form a Y-Pod swarm formation in 2D environment is discussed. In this sense, the system reacts to static, linear, quadratic and combined cases, while performing these tasks there exist switching problems in the simulations. In order to overcome switching problems, we applied, sigmoid function and fusion controller. Thereafter, Y-Pod shape performance is evaluated by using the pattern index matching technique. On the other hand, the stability analysis is justified based on steering analysis. Finally, results and discussions are presented.

In this chapter Y-pod swarm formation based on linear control is presented. This method applies to swarm systems, although the roots of these methods are related to dynamical systems and control theory. In particularly, the method applies to non-holonomic systems. The designer has some expected requirements while devising swarm behaviour: the solution proposed here is to support the design phase by a model delivering predictions that correspond to reality. Although we design the controller for real robot performance also, but the work is only delivered in a simulation environment, due to large amount of robots involved in swarm systems.

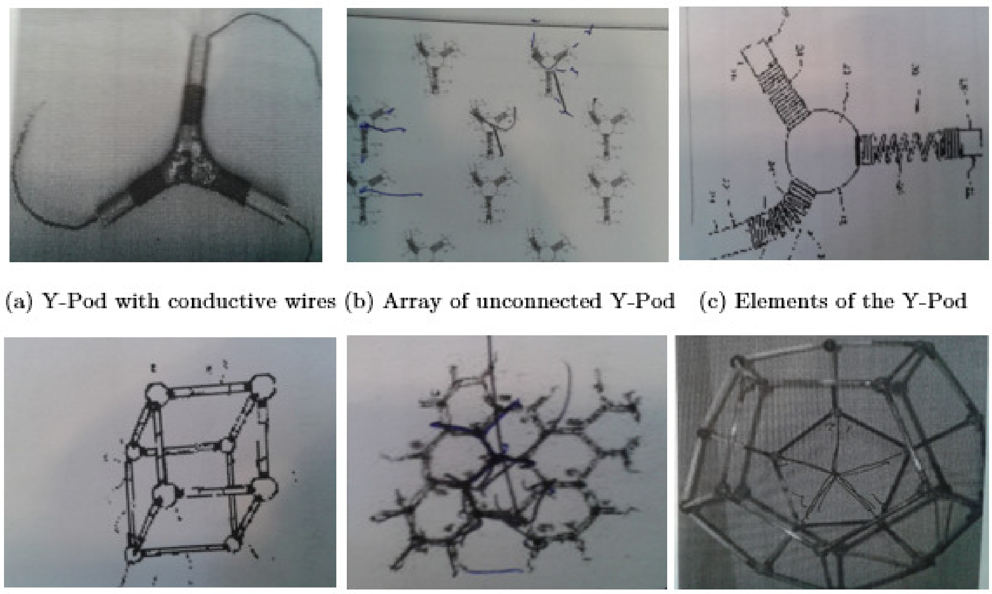
4.1 Y-Pod and its Applications

In this section, Y-Pod and its applications are shown. In swarm robotics, there exist various shape formations in the literature, but this thesis introduces Y-Pod shape, which has vast applications compare to other formation techniques and also have some important advantages were found, they are the following:

ISBN: / DL:

1. The Y-Pod can be utilized for formation strategy on all scales
2. Global shape formations
3. Changes shapes
4. Easy to expand
5. To overcome the redundancy problems
6. Self-organized and self-repair problems

In 2001, Dr.Alvin Swimmer (Alvin 2001) had been introduced to this Y-pod shape for 3D morphing architecture for various purposes. He argued that the world of nature is not totally flat, not a square, rigid and laminar; the world of nature is cellular, vascular and of variable rigidity. There are no straight lines or right angles, nor are there perfect spheres in nature. All these concepts are observed from various fields (see Fig. 4.1). In order to cover and maintain perfect shapes, we need emergent shape formations and one of them is Y-Pod.



(a) Y-Pod with conductive wires (b) Array of unconnected Y-Pod (c) Elements of the Y-Pod

(d) Cube made with Y-pod connector (e) Y-Pod flat sheet (f) Y-Pod Geodesic sphere

Fig.4.1: Snapshot of Y-pod use in many different structures

We propose to apply these Y-Pods as a formation strategy in swarm robots. It's assumed as an equilateral triangle (see Fig. 4.3). In general, Y-Pod is not exactly an equilateral triangle

ISBN: / DL:

and it contains the node with three segments connected with edges. When many Y-Pods are connected, they will appear in different planes with different shapes with respect to angles and orientations (Muniganti and Oller 2015). Although Y-Pods shape will change from one to plane to another plane based on torsion and dihedral angles (see Fig. 4.2).

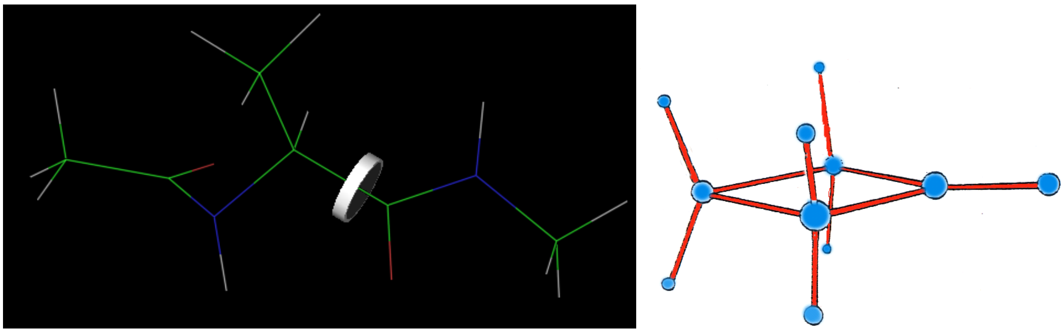


Fig.4.2: Y-Pod shape contains node with three corresponding segments. Edges will change in 2D and 3D planes due to torsion and dihedral angles.

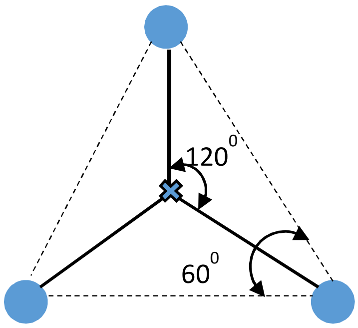


Fig.4.3: Y-Pod shape assumed as an equilateral traingle with segments and angles .

This shape has the advantage that if we choose any formation, there should be involved Y-Pod. Then, can be useful to overcome the network problems. Besides, it has wide range of applications (see Fig. 4.1). This new approach is able to design systems that function on the nano, micro and macro connector for communication (Steve 2010) and network technologies.

ISBN: / DL:

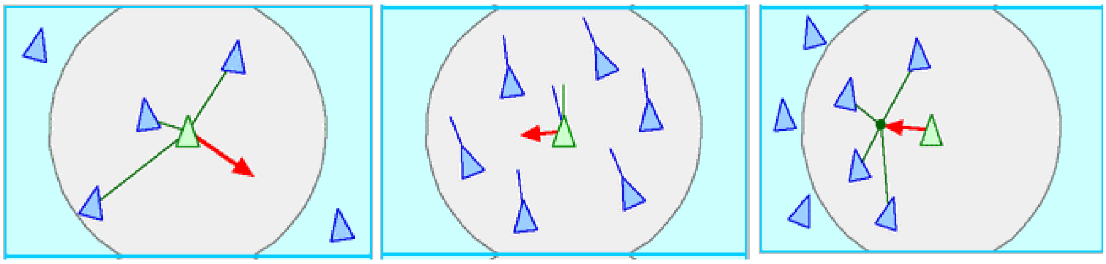
4.2 Proposed model of Y-Pod swarm system

Our approach is inspired and borrowed from molecular dynamics theory in organic and biological systems with AMBER force field but it will modified in to harmonic function form and derived to a parametrized control law (Barraquand and Latombe 1989, Oller and Garcia 2002). In this setting, our method will be explained in step-by-step process which contains AMBER force fields, harmonic function form, and non-holonomic linear controller with key modifications. The discussion of holonomic and non-holonomic constraints is very usual in robotics systems. Holonomic robots are those that can move in all directions freely regardless of pose, which is unrealistic due to physical limitations (think of parallel parking). However, in the simulations, holonomic robots prove to be extremely useful to demonstrate the effects of different behaviors without worrying about robot morphology and kinematics (Indiveri 1999). Some holonomic systems do exist, and certain mathematical logics can be employed to make non-holonomic systems appear holonomic. On the other hand, non-holonomic robots are characterized by constraint equations involving the time derivatives of the system configuration variables. These equations are non-integrable and typically arise when the system has less control signals than state variables. Typically, a car-like robot has two control signals (linear and angular velocities) while it moves in a 3-dimensional configuration space (x, y, θ). As a consequence, any path in the configuration space does not necessarily correspond to a feasible path for the system (Chiaverini et al.2005). This is basically why the purely geometric techniques developed in motion planning for holonomic systems do not apply directly to non-holonomic ones.

Before explaining the model, we have summarized relationship between molecular dynamic theory and our approach with respect to amber force fields as follows: swarm formation needs to maintain coherence in the environment. As discussed in chapter 3, one of the most preferred control techniques is the artificial potential field approach, where Reynold's rules are very used, with attraction, alignment and repulsion compounds (AAR) (see Fig. 4.4), or social potentials (Li et al. 2005). Instead of such potential fields, the simulation of biomolecules is done by force fields: the so-called AMBER technique (Assisted Model Building with Energy Refinement). Such force fields are the cornerstone of molecular

ISBN: / DL:

mechanics (Junmei et al. 2004), working well both for the biological and organic molecules, especially in proteins and nucleic acids.



(a) Repulsion (b) Alignment (c) Attraction

Fig. 4.4: Reynolds' rules as an example of AAR compounds.

In molecular dynamics, researchers are amused by the functions which atoms play in a completely local way, which is similar to autonomous robots to perform tasks. With this evidence, biologists and chemists even call proteins nature's robots in a recent book (Tanford and Reynolds 2004). In robotics, robot team forming is for multiple mobile robots to establish robotic formation which is optimal for performing a given task (Parker et al. 2005). By merging these two approaches, each atom is considered as an autonomous robot, and Y-Pod is a process of autonomous robot team formation (Chuang 2007). This analogy is reasonable because a Y-Pod is a set of connected atoms, and formation needs the atom to form a symmetrical cohesion. After understanding both approaches, it indicates that, molecular dynamics and swarm robotics has same features that deal with large amount of atoms in molecular dynamics and large amount of robots in swarm robotics.

Consideration of the aforementioned description and analogy of both theories, we adopted techniques in both ways and considered our system as follows: path planning is a procedure which specifies motion trajectories of multiple autonomous mobile robots to form a robotic team with a required formation. Since molecules consist of large amount of atoms which come together to form a Y-Pod structure, it is considered that each atom act like a mobile robot and takes adequate path to form a robotic team. Based on the amber force field, each atom-robot searches for its position and maintain Y-Pod structure by controlling distance and angle. On the other hand, multiple atom-robots may form sub-teams, so each one maintains distance and angles with respect to their destination point to maintain the Y-pod.

ISBN: / DL:

Thereafter, virtual leader carries the Y-Pod structure to reach the desired destination point of the team. Of course, the equation and parameters are not the exactly same as organic or biological approaches because of physical and kinematic control laws are different in magnitudes.

Classical amber force field in organic or biological molecules can be expressed as follows,

$$E = \sum_{bonds} k_r \left(r - r_{eq}\right)^2 + \sum_{angle} k_\theta \left(\theta - \theta_{eq}\right)^2 + \sum_{dihedrals} \frac{v_n}{2}(1 + \cos(n\phi - \gamma) + \sum_{i<j} \frac{A_{ij}}{R^m{}_{ij}} - \frac{B_{ij}}{R^n{}_{ij}} + \sum_{i<j} \frac{q_i\, q_j}{\varepsilon R_{ij}} \quad (4.1)$$

Where r, θ, ϕ are variable atom poses and E is the total amount of energy. The formula has some equilibrium structural parameters in terms of physical and chemical constants such as, γ is phase angle for torsion angle. Eq. (4.1) contains bonded and non-bonded terms. In this work bonded terms in 2D environment are prevented, so the last three terms (torsion, dihedral angles. Specific atomic phenomena) are not used because they are active only in 3D analysis.

The final bonded terms equation is as follows:

$$E = \sum_{bonds} k_r \left(r - r_{eq}\right)^2 + \sum_{angle} k_\theta \left(\theta - \theta_{eq}\right)^2 \quad (4.2)$$

Now Eq. (4.2) becomes a harmonic function form, bond between two atoms nothing but distance between two robots, and the angle between to atoms is angle between two robots. Then r_{eq} and θ_{eq} are equilibrium values with respect to set-points, and k_r, k_θ are constants concerning with sensitivity.

In the AMBER technique, organic molecules have displacement and vibration characteristics: the more energy the more displacements and vibrations will have. Then, when energies tend to zero the system tend to be more stable (Muniganti and Oller 2015). According to this, mobile robot energies under formation are the amount of steering w.r.t steering of the set-

ISBN: / DL:

point (equilibrium point). When the amount of steering tends to the steering of the set-point this means our system is stable.

So, Eq. (4.2) can be rewrite as follows:

$$\sum_{robot} \omega = \sum_{robot} k_r \left(r - r_{eq}\right)^2 + \sum_{robot} k_\theta \left(\theta - \theta_{eq}\right)^2 \qquad (4.3)$$

Eq. (4.3) in terms of robotics terminology is as follows,

$$\sum \omega(t) = \sum_{i=1,2,3.}^{robots} k_r(t)\,(r_i - r_{eq})^2 + \sum_{i=1,2,3.}^{robots} k_\theta(t)\,(\theta_i - \theta_{eq})^2 \qquad (4.4)$$

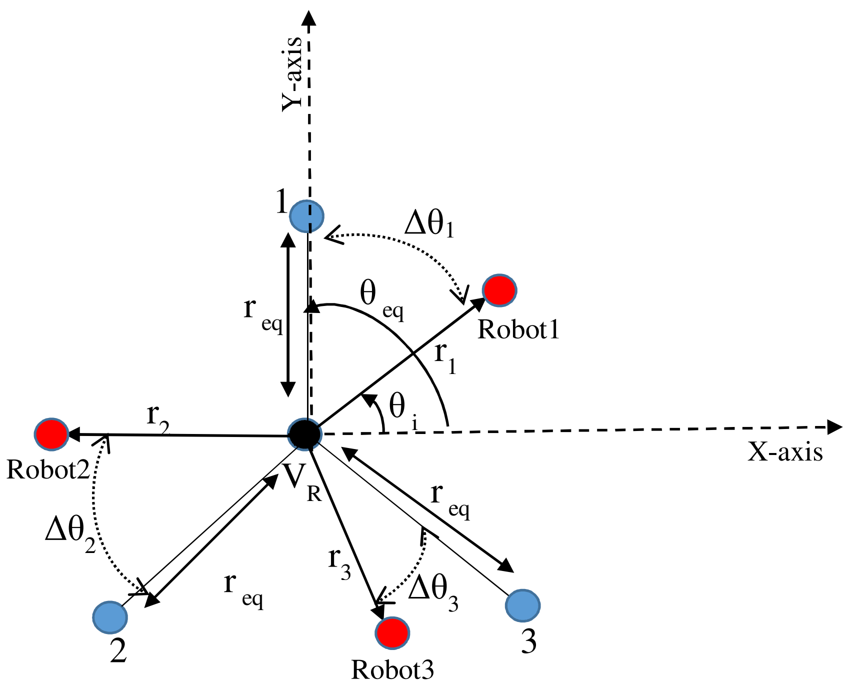


Fig.4.5: Y-Pod analogy w.r.t.to amber forces field with our control approach.

Fig. 4.5 is the geometrical description of the Eq. (4.4) which can be modified into a kinematic control law for each single robot, so which can be written as follows:

$$\omega(t) = k_r . F(r,t) + k_\theta . F(\theta,t) \qquad (4.5)$$

ISBN: / DL:

, where k_r and k_θ are parameters.

In order to obtain constant parameters from Eq. (4.5), we will derive them in terms of settle time parameter based on pole placement approach, so then we need to derive the control law based on the kinematic model of the robot (Scheuer and Fraichard 1997).

A kinematic model of the robot is a mathematical description of the capabilities and dependencies of the robot. Although the inertia will be neglected in our model, we will get a good response in the simulation of the system. Hence a kinematic model of the robots is created describing the mathematical relations between the inputs and outputs of the system. The robot is able to turn on its own axis without any movement in xy-plane by applying velocities to wheels, and both the robot and the moving goal move in the horizontal plane. The goal manoeuvres are not a priori known to the robot so the aim is to design a closed-loop control law for robots (Shiller and Lu 1990), which insures reaching the moving goal.

We assume that the following conditions are satisfied.

- The robot is faster than the moving goal, and the goal moves in a smooth path.
- The minimum turning radius of the robot is smaller than the minimum turning radius of the moving goal.
- The robots provide the control system with the necessary information about the target and the environment.
- The target's speed, orientation, and position are exactly known. Such data could be measured without problems in real robots.

The equations of the mobile robot kinematics moving with linear velocity and angular velocities are:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{v}\,cos\theta \\ \dot{v}\,sin\theta \\ \omega \end{pmatrix} \tag{4.6}$$

where x and y represents the robot co-ordinates in terms of a fixed co-ordinates system, ω represents angular velocity and θ is the angle determined by the robot orientation w.r.t. x-

ISBN: / DL:

axis (see Fig. 4.6). Let us suppose that the robot's movement must be controlled in a way such that the robot must follow the horizontal axis $(x, 0)$ with a constant linear velocity v.

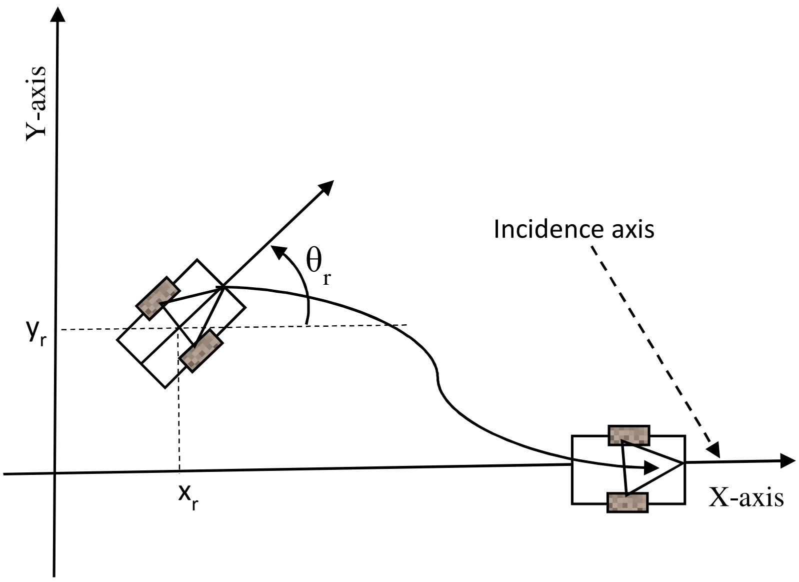


Fig.4.6: Axis reference of a mobile robot.

In order to obtain a control algorithm the values of angular velocity will be obtained in terms of robot pose: the robot will tend to follow the horizontal line $(x_r, 0)$. Therefore, the evaluation of a control law $\omega(y_r, \theta_r)$ will make tends to evolve (y_r, θ_r) to $(0,0)$ values.

The kinematics Eq. (4.6) can be rewritten as:

$$\begin{aligned} &\dot{x} = F(x, \omega), \\ &\text{where} x = (y_r, \theta_r)^T, \quad F(x, \omega) = (v \sin\theta_r, \omega)^T \end{aligned} \tag{4.7}$$

The system equilibrium points (X_{eq}, ω_{eq}) are those points x and ω where the dynamics cancels out:

$$X_{eq} = \begin{pmatrix} Y_{eq} \\ \Theta_{eq} \end{pmatrix} = \begin{pmatrix} Y_{eq} \\ \{0, \pm\pi\} \end{pmatrix}, \qquad \omega_{eq} = 0 \tag{4.8}$$

ISBN: / DL:

The system dynamics (Barraquand and Latombe 1989) is given by the next expression when the tangent function is linearized:

$$\begin{pmatrix} \dot{y}_r \\ \dot{\theta}_r \end{pmatrix} \approx \begin{pmatrix} 0 & v \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y_r \\ \theta_r \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \omega \tag{4.9}$$

In linear control systems theory, these equations are a state space representation of the mobile robot movement. In fact, it can be demonstrated (Barraquand and Latombe 1989) that the system is controllable and that a control law can be obtained in order to guarantee the evolution to the equilibrium and also its stability.

From the Eq. (4.9) we obtain a second order linear equation for angle θr which can be written in terms of parameters α_1 and α_2:

$$\ddot{\theta}_r + (\alpha_1 + \alpha_2)\dot{\theta}_r + (\alpha_1 \alpha_2)\theta_r = 0 \tag{4.10}$$

This linear second order differential equation draws the evolution of $\theta r(t)$ function in terms of α_i parameters: these two parameters are the poles of the system. Using Laplace transformation properties, the corresponding characteristic equation can be expressed in terms of ω_n and ζ parameters by this way:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = (s+\alpha_1) \cdot (s+\alpha_2) = 0 \tag{4.11}$$

, where‘s’ is the complex Laplace independent variable. So then, the settle time t_s of function $\theta_r(t)$ can be expressed in terms of ζ and ω_n parameters as follow: $t_s=4/(\zeta\omega_n)$.

It can be concluded that the movement control in the above conditions can be done, with constant linear velocity v and angular velocity ω given by the next control law:

$$\omega(t) = -\left(\frac{\alpha_1\alpha_2}{v}\right) \cdot y_r(t) - (\alpha_1+\alpha_2) \cdot \theta_r(t) \tag{4.12}$$

When $\zeta=1$ the poles are identical and the Eq. (4.12) can be rewritten with $\alpha_1=\alpha_2$. Fig. 4.7 and Fig. 4.8 show different plots of function $\theta_r(t)$ w.r.t different values of parameter ζ.

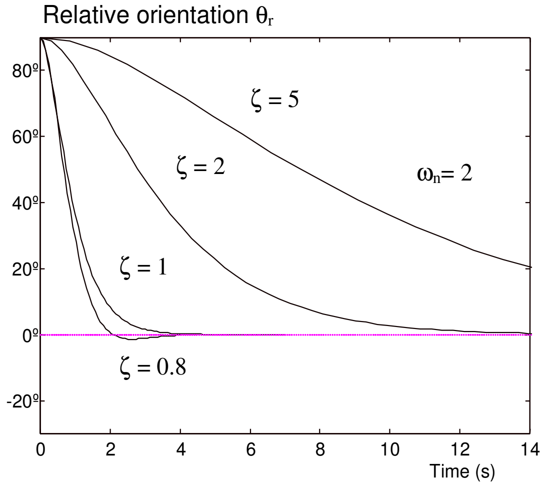


Fig.4.7: Plot of function θr(t) when ζ>1

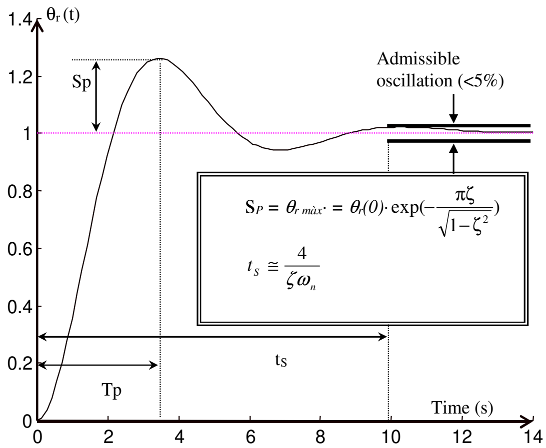


Fig.4.8: Plot of main parameters of function θ_r (t) when $\zeta<1$.

From the above figures we can choose ζ=1 (or 0.8) as acceptable but we consider 1 to entirely avoid oscillations: then, $\alpha=\alpha_1=\alpha_2$. Also we have $y_r(t)=r(t).\theta_r(t)$ so that Eq. (4.12) derives to:

$$\omega(t) = \frac{\alpha\alpha\,\theta(t)}{v}\, r(t) + 2\alpha\theta(t) \tag{4.13}$$

If we recall Eq. (4.5), we rewrite the Eq. (4.13) as follows:

$$\omega(t) = k_r.F(r,t) + k_\theta.F(\theta,t) \tag{4.14}$$

ISBN: / DL:

, where $k_r = \frac{\alpha\alpha\,\theta(t)}{v}$ and $k_\theta = 2\alpha$

Rewrite the Eq. (4.14) as follows

$$\omega_i(t) = \alpha_1(t)y_r(t) + \alpha_2(t)\theta_r(t) \quad (4.15)$$

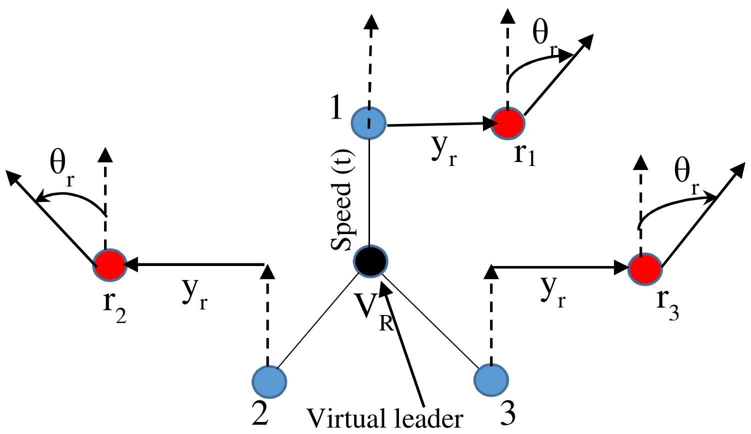


Fig.4.9: Schematically illustrated forces and moments acting on robots based on Eq. (4.15)

4.2.1 Follower and Virtual leader with control law approach

The path planning is a procedure which specifies motion trajectories of multiple autonomous mobile robots to form a robotic team with a required formation. In this framework, the virtual robot leader (V_R) is in charge of carrying Y-Pod structure. In this scenario, virtual leader is a virtual reference point which influences the Y-Pod structure and its corresponding robots. In other words, virtual leader robots acts as moving goal position. Since the virtual leader position must be the result of some path planning procedure, typically the position should be the centre of mass for the desired Y-pod. From the point of view of corresponding robots, distances and angles are computed in reference to Y-Pod nodes. In addition, virtual leader avoids the obstacles in the environment: in this concern, virtual leader and corresponding follower robots will avoid future collisions in order to reach the desired destination point (Farinelli et al. 2004, Hsu and Liu 2005).

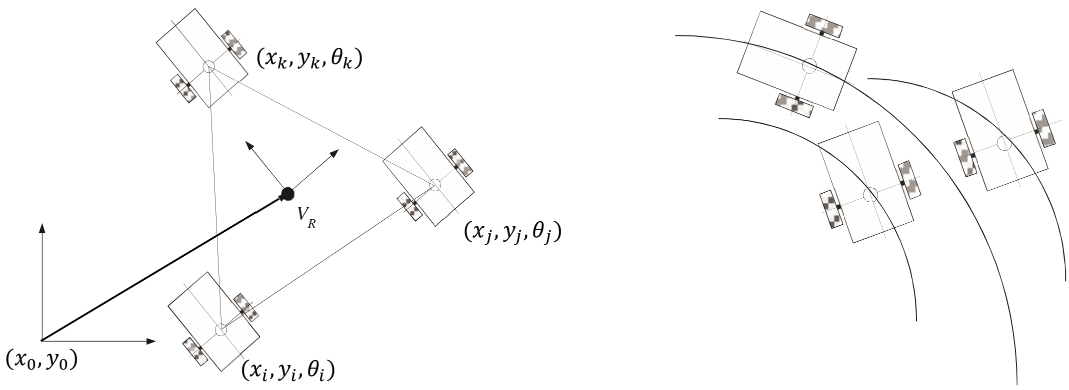


Fig.4.10: (left) Reference points to corresponding robots w.r.t. Y-Pod, (right) Path of moving Y-Pod with three robots.

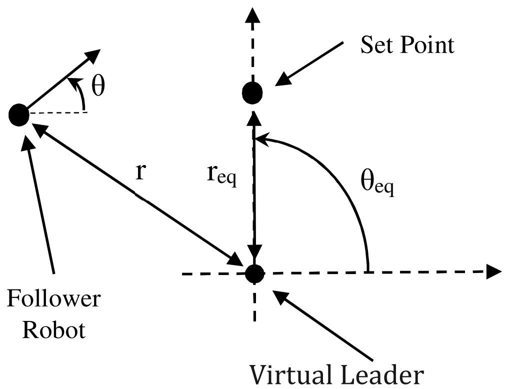


Fig. 4.11: Geometric representation of variables based on follower robot and virtual leader.

In Fig. 4.10 and Fig. 4.11 show the geometrical representation of follower robots and virtual leader with notations. Here we consider virtual leader is a reference point positioned at the centre of the Y-Pod moving with linear velocity (v_r), angular velocity (ω_r) and orientation angle (θr).

ISBN: / DL:

4.2.2 Pattern matching index of Y-Pod using network topology

In this section briefly discusses patter matching index technique, and to explain some components that should be useful for further elaborations (for more information, section 3.1.5). A collection of simple robots can replace a complex robots via cooperation. In the process of cooperation, forming a specific formation pattern will enhance the efficiency of the formation and data transmission between robots. In order to form a specific formation pattern, the interaction between robots should be well regulated. When characterising the system dynamics of a swarm robot team, related properties of a formation such as stability and connectivity are also important, and should be analysed, although robots must maintain the formation shape while performing some tasks.

Aforementioned explanations, we refer geometric pattern matching technique to produce the better measurements for the Y-Pod shape evolution. In this scenario, focus on matching process which aims to find the commonalities between geometric features. In other word, pattern or shape matching techniques may be based on geometrical, topological, or semantical information, or a combination of these three. Here, we focus on geometrical approach via Euclidian geometry.

The process of identifying patterns located on different environment by means of geometrical, topological and semantical information is called pattern or point matching. A pattern is represented by a set of points in the Euclidean plane that form a geometric figure such as a circle, a line , triangle or some other arbitrary shapes . Given a particular pattern as input, the robots must position them-selves with respect to each other such that the location of the robots correspond to points in the pattern. The arbitrary pattern formation problem, that of forming any pattern given in input, has also been studied (Lorenzo et al. 2011, Wang et al. 2010, Hackeloeer et al. 2013, and Rosen and saalfeld 1985).

ISBN: / DL:

Geometrical point matching techniques

Geometrical point or pattern matching techniques only consider geometrical information (i.e., coordinates) for evaluating a point matching. Even though the distance between point coordinates may be calculated in any p-norm, the only metric of practical relevance is the Euclidean distance metric.

On the other hand, A Pattern is represented by a set of distinct points $(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$ n>=1, in the two dimensional Euclidean plane. A pattern P_i is said to be isomorphic to a pattern P_i ,if P_i can be obtained by a combination of translation, rotation and uniform scaling of pattern P_i. The size of pattern P_i is its cardinality and will be denoted by n_i. In this sense define some special cases such as *point*, *two-point and polygon(n).*

***Point*:** The pattern consisting of a single point.

Two-point: The only possible pattern consisting of exactly two points.

Polygon(n): For any n>=3, this is the pattern consisting P1, P_2,.., P_n that are vertices of a regular convex polygon of n sides.

With the evidence of aforementioned analogy. In Euclidean geometry, Brahmagupta's formula finds the area of any cyclic quadrilateral (one that can be inscribed in a circle) given the lengths of the sides. Brahmagupta's formula gives the area *k* of a cyclic quadrilateral whose sides have lengths a, b, c, d as:

$$k = \sqrt{(s-a)(s-b)(s-c)(s-d)} \qquad (4.16)$$

Where s, the semi-perimeter, is defined to be as follows:

$$s = \frac{a+b+c+d}{2} \qquad (4.17)$$

A triangle may be regarded as a quadrilateral with one side of length zero. From this perspective, as *d* approaches zero, a cyclic quadrilateral converges into a triangle, and

ISBN: / DL:

Brahmagupta's formula simplifies to next one, also called Heron's formula. This formula states that the area of a triangle whose sides have lengths *a*, *b*, and *c* is

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{4.18}$$

, where *s* is the semi-perimeter of the triangle, that is,

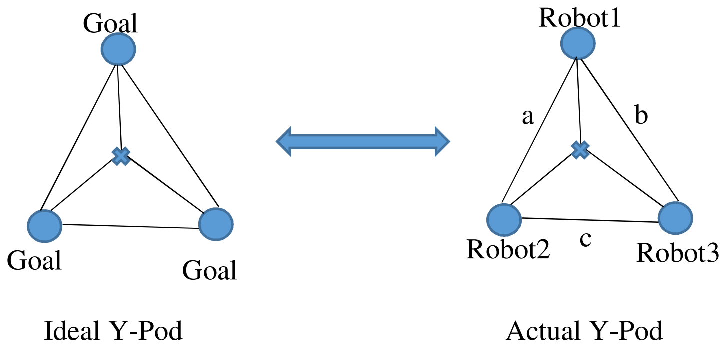


Fig.4.12: ideal Y-Pod and actual Y-pod strategy for pattern matching index,

$$s = \frac{P}{2} = \frac{a+b+c}{2} \tag{4.19}$$

When a triangle is equilateral ($\underline{a}=a=b=c$) then the area equals to $\sqrt{3}(\underline{a}/2)^2$ and next equation holds:

$$A=\sqrt{3}(\underline{a}/2)^2 \tag{4.20}$$

In order to define an index *J* we need a dimensionless quantity so we propose the next expression:

$$J^{area} = \frac{A}{P^2} \tag{4.21}$$

For an equilateral triangle, Eq. (4.21) equals to $\underline{J}^{area}=\sqrt{3}/36$ but when used for general triangles ($a\neq b\neq c$) the previous index holds the next expression:

$$J^{area} = \sqrt{\left(1-\frac{a}{p}\right)\left(1-\frac{b}{p}\right)\left(1-\frac{c}{p}\right)} \tag{4.22}$$

, where is the perimeter $p=(a+b+c)$. Eq. (4.22) can be normalized as follows:

$$J_{area} = \frac{J^{area}}{\underline{J}^{area}} = \frac{\sqrt{\left(1-\frac{a}{p}\right)\left(1-\frac{b}{p}\right)\left(1-\frac{c}{p}\right)}}{\sqrt{3}/36} \tag{4.23}$$

In the other hand, we want to build another index to evaluate the deformation of the triangle. For an equilateral triangle, Eq. (4.21) equals to $\underline{J}=\sqrt{3}/36$ but when used for general triangles ($a\neq b\neq c$) the previous index holds the next expression:

$$J_{elongation} = \frac{1}{3}\sqrt{\sum_{3sides} abs(\underline{a}-a_i)^2} \tag{4.24}$$

Relation between ideal Y-Pod and actual Y-Pod considered in three cases those are:

1. Index to compute if actual triangle is big or smaller than ideal Y-Pod

$$J_{area} = \begin{cases} ideal & if\ J_{area} = 1 \\ bigger & if\ J_{area} > 1 \\ smaller & if\ J_{area} < 1 \end{cases}$$

2. Index to compute actual deformation related to ideal Y-Pod

$$J_{elongation} = \begin{cases} ideal & if\ \ J_{elongation} = 0 \\ deformed & if\ \ J_{elongation} > 0 \end{cases}$$

3. New index:

$$J_{new\ index} = \frac{W_{area} \cdot J_{area} + W_{elongation} \cdot J_{elongation}}{W_{area} + W_{elongation}}$$

Where W_{area} and $W_{elongation}$ are weighted parameters.

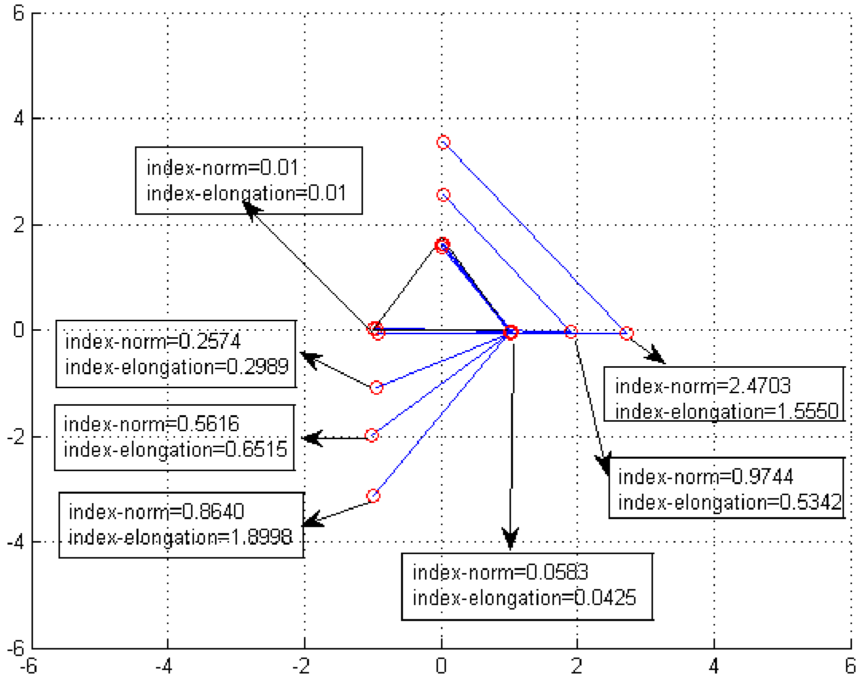


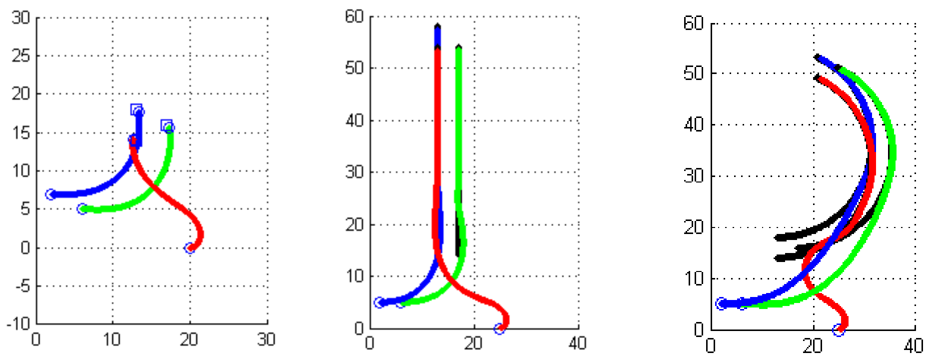
Fig.4.13: Matlab simulation test result of index matching formula,

In the Fig. 4.13 show the index performance evolution of the Y-Pod shape, with the assumptions of Y-Pod is an equilateral triangle. It can see, that the index is zero exactly at all corner positions of the Y-Pod, which indicate red circle. Although red circles appears far away from the corner position of the equilateral triangle, it can treated as index elongation. Index elongation is nothing but pull the Y-Pod shape to various locations. In this process, we receive different values, compare to index norm. Index norm is 0.01 at corner positions, it can says that shape evolution is exactly matches. Consequently, index elongation is more than the index norm, in this case, we can say that, shape evolution is not exactly same, which means index is mismatched.

ISBN: / DL:

4.3 Switching problems

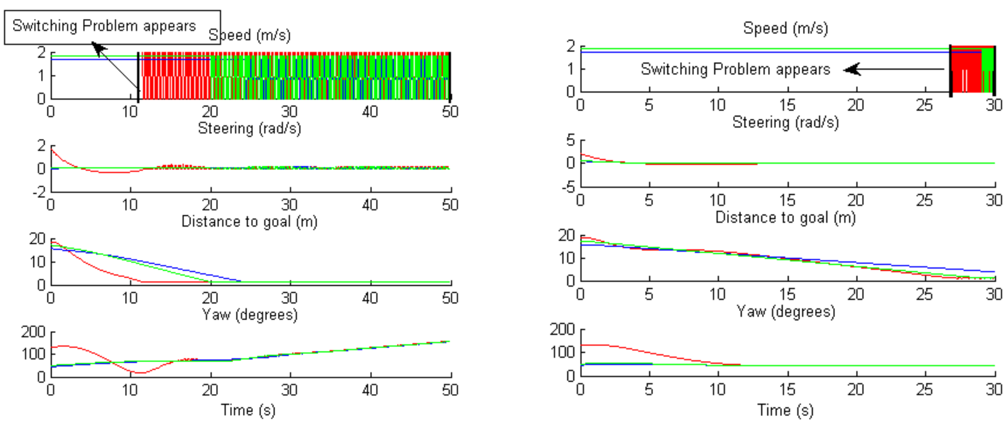
After using the proposed controller, we have done simulation experiments with three robots in various positions with static, linear and quadratic cases tested in the matlab simulation environment. We got some interesting results but appeared some switching problems as follows.



(a) 3-robots in static path (b) 3-robots in linear path (c) 3-robots in quadratic path

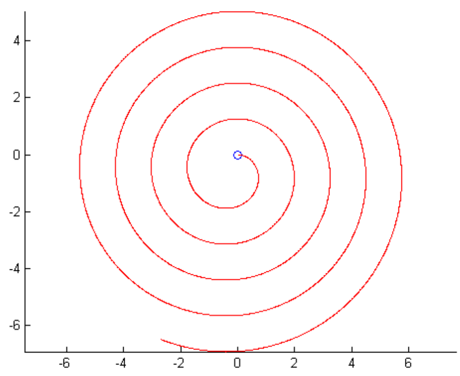
Fig.4.14: shows the 3 robots simulation run in different case

In the Fig. 4.14 the controller was tested with 3- robots run in the simulation and observed that the simulator worked well enough but with respect to speed we observed switching problems associated with the above results shown in Fig. 4.14 as follows

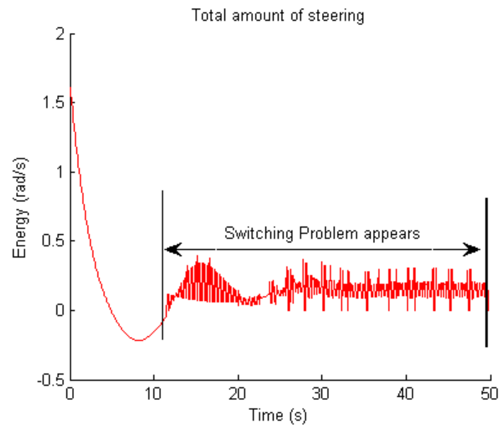


(a): Switching occurs w.r.t.to speed at 11 sec/time (b): Switching w.r.t.to speed at 20sec/time

ISBN: / DL:



(c): Switching occurs around the goal pose



d): Switching at12 sec/time in amount of steering

Fig.4.15: Switching problem appears in diferent levels

The term switching describes the phenomenon of finite-frequency, finite-amplitude oscillations appearing in many sliding mode implementations. These oscillations are caused by the high-frequency switching of a controller exciting unmodelled dynamics in the closed loop. Fortunately, preventing switching usually does not require a detailed model of all system components. Rather, a controller may first be designed under idealized assumptions of no unmodelled dynamics. The solution of the switching is of great importance when exploiting the benefits of a controller in a real-life system (Dorigo 2005). Without proper treatment in the control design, switching may be a major obstacle to the implementation of sliding mode and velocity control in a wide range of applications. Our proposed controller overcomes all the requirements except the velocity control, which is the problem between robots and desired destination point (goal position).

Under realistic conditions, a switching prevention scheme should be selected to meet the system specifications and to ensure a good system performance. Switching is a mismatch between two cases but it depends on the situation, this type of problems exist in work. Various issues for example were observed in our work velocity mismatch in the results that we treated as a velocity control between robots and destination goal point.

ISBN: / DL:

4.3.1 Fusion control with sigmoid function

Mostly, fusion control is useful in sensor fusion(Fredrik 2001) and sliding mode controller in robotic studies, but in this issue, fusion control is used based on sigmoid functions, and to control the velocity of speed for the near and far goal fixed positions of the robot destination point. Fusion control plays an important role in order to control the speed around the goal position in various directions, i.e target velocity control, which measure the initial robot velocity and target velocity. In this concern, need to explain briefly as follows.

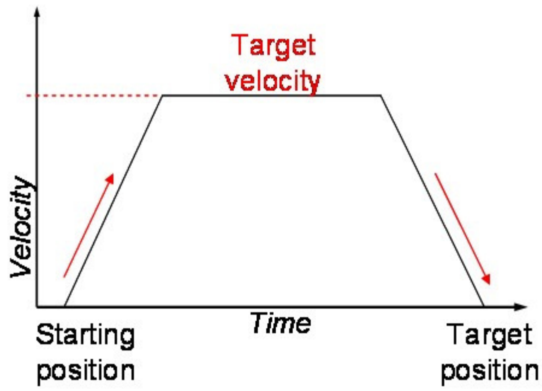


Fig.4.16: A Typical velocity profile.

The motion controller (Siemens 2010) uses the desired target position, maximum target velocity, and acceleration values to determine how much time it spends in the three primary move segments (which include acceleration, constant velocity, and deceleration).

For the acceleration segment of a typical trapezoidal profile, motion begins from a stopped position or previous move and follows a prescribed acceleration ramp until the speed reaches the target velocity for the move. Motion continues at the target velocity for a prescribed period until the controller determines that it is time to begin the deceleration segment and slows the motion to a stop exactly at the desired target position. If a move is short enough that the deceleration beginning point occurs before the acceleration has completed, then the actual velocity attained may fall short of the desired target velocity. Sigmoid function is a bounded differential real function that is defined for all real input values and has a positive derivative at each point.

One obvious solution to make the control function continuous or smooth is to approximate function $v(\sigma) = -\rho sign(\sigma)$ by some continued or smooth function. Here we used this function in order to overcome the switching problems. For instance, it could be replaced by a "sigmoid function" in order to control the velocity matching between robot velocity and goal velocity.

The technique presented below allows to switch from one state to another state. The new control law will control the movement of the robots switching from one state to another state when certain transition conditions arises. Then new controller continuously switch between two states instead of constant discrete values. The technique is based on the behaviour of the hyperbolic tangent function that continuously switches from -1 to +1. By doing various manipulations and changes of scale and appropriate displacements, the below Eq. (4.25) function evolves from the values a_1 and a_2 whose value is maximum transition rate for $x=x_0$.

$$f(x) = \frac{(a_2 - a_1)}{2} \cdot \tanh(C \cdot (x - x_0)) + \frac{(a_2 + a_1)}{2} \qquad (4.25)$$

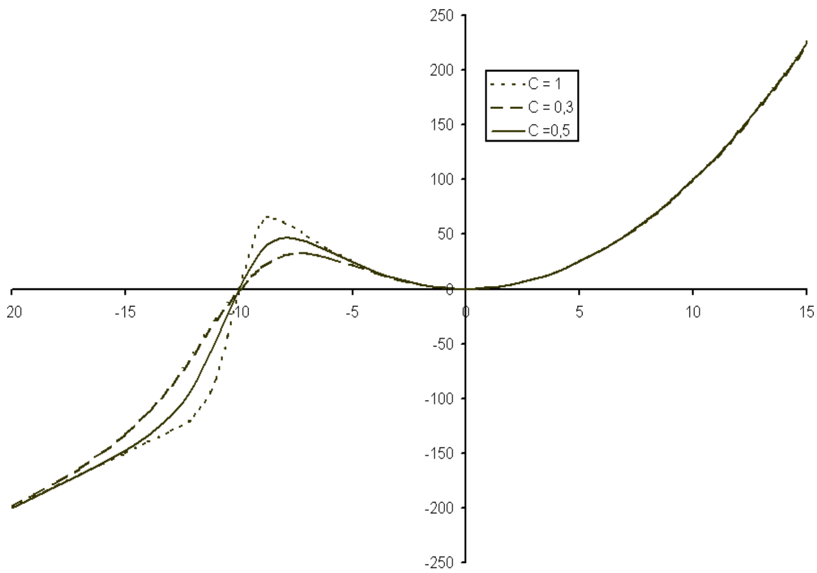


Fig.4.17: Example of continuous switching between a_1=x and a_2=x^2 functions at x_i=-10.

ISBN: / DL:

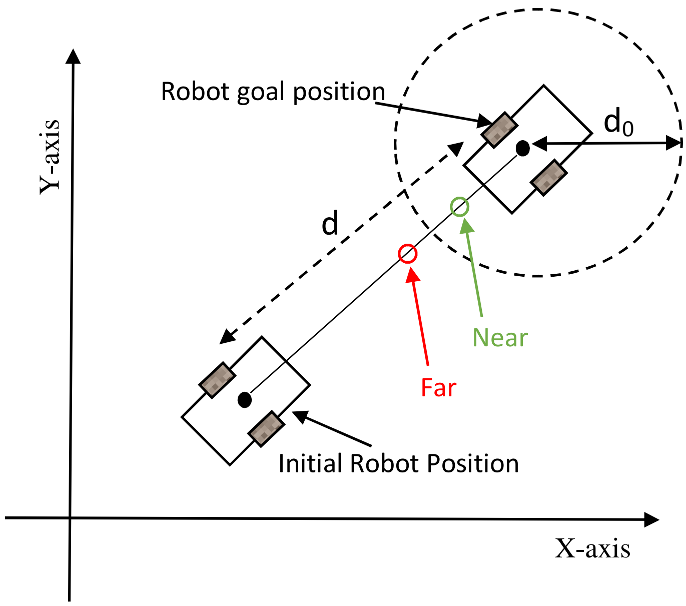


Figure 4.18: Geometric representation of far and near goal positions

$$v(t) = \frac{(v_{far}(t) - 0))}{2} \tanh(\mathrm{C}.(\mathrm{d} - \mathrm{d}_0) + \frac{v_{far}((t) + 0))}{2} \qquad (4.26)$$

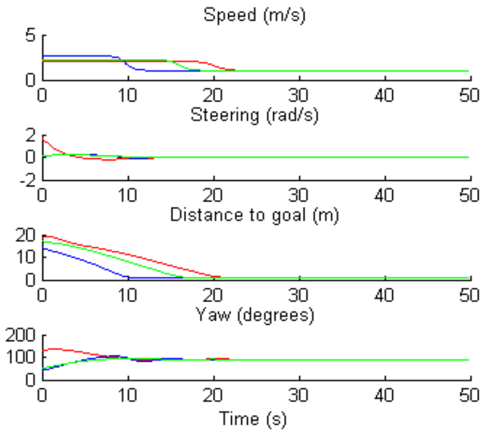
As for the linear velocity control we choose the option that the robot move at a constant linear velocity at some distance begins to slow to zero. In this case, the combined control law proposed in Eq. (4.25) should take the values of a1 and a2 depends on the circumstances of the system control and setup of parameters.

Syntax:

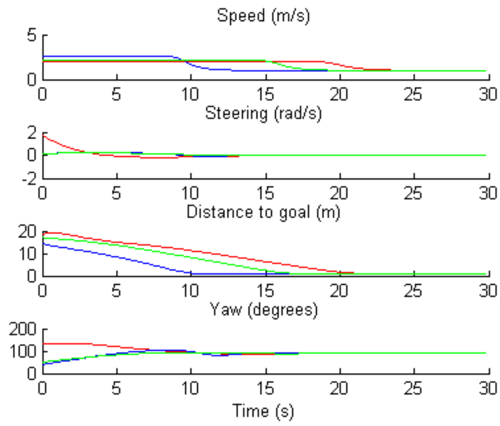
```
#define fusion_controller

#if def fusion_controller
        C=1; % C is the sigmoid function rate parameter
do
dist0=DIST_TH % where DIST_TH is the threshold depends on environment.

        speed_max=v;

        a1=[0 0 0];     % NEAR

        a2=speed_max;   % FAR

        v= (a2-a1)/2.*tanh(C*(dist-dist0)) + (a2+a1)/2; % transition rate

end
```

ISBN: / DL:

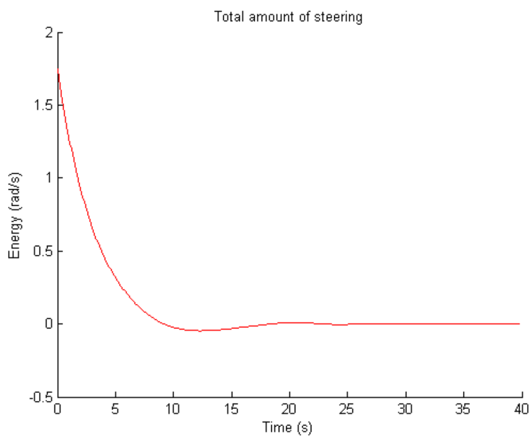
After using the fusion control and sigmoid function, we overcome the switching problems in the entire simulations for all cases. The following results show without switching problems, now settle time Ts is computed for each Y-Pod.



(a)



(b)



(c)

Fig.4.19: Simulation results without switching problems

ISBN: / DL:

4.4 Simulation experiments with proposed controller

The proposed control law has been simulated in Matlab and its performance in terms of scalability, stability, and additional characteristics are examined. The simulations were run to form a Y-Pod shape and formation under different scenarios. The control law was applied in static, linear, quadratic and combined motions to a large swarm of robots. As an initial study, three robots were scattered in the environment and their behaviour was analysed. The three robots were assigned the positions $[x, y, \theta]^T$, with x and y being the initial positions of the robots and θ their orientation with respect to the virtual leader V_R (reference point). The virtual leader motion's is described in three different terms: without movement, constant speed and no steering (ω=0) and constant speed with constant steering $\omega \neq 0$ that is static mode, linear and curved paths. In all simulations, the path of each robot and Y-Pod is computed. The results regarding the speed, steering, and distance to goal and yaw angle θ (relative orientation of robots with respect to the leader) are expounded in the present section, as well as the energy value (relative to the total amount of steering) and index matching range.

The Y-Pod shape and formations appear in each case to rely on various variables such as: the pole frequency, the settle time, the robots' initial positions, speed of robot (maximum, minimum), speed of virtual leader, the maximum simulation time, the number of robots and sample time. The results shown below were mostly limited to systems of 3-robots with single Y-Pod shape and 9-robots with three Y-Pods. Although entire simulations, the sample time was set to 1/10 meter per seconds to each iteration. Additional simulations were carried out with 6 robots, 12 robots, 15 robots and 18 robots corresponding to two Y-Pods, four Y-Pods, five Y-Pods and six Y-Pods respectively. These simulations were carried out to understand the swarm's behaviour, and the results are shown in the appendix.

4.4.1 Static

The study analyses the Y-Pod formation in the static case. That is, the robots move from the initial position to form a static Y-Pod (at goal position), which maintain a static position at the goal location. In this scenario the robots are initially placed in different positions in the simulation arena and our control law is used to form a Y-Pod shape at a fixed settle time Ts=14 sec. The robots are placed at (2, 5), (6, 5) and (25, 0) coordinates and are indicated with blue circles in Fig. 4.20. The corresponding goal positions are set to (12, 21), (17, 18) and (15, 24) indicated with stars. The paths and trajectories are shown red, blue and green.

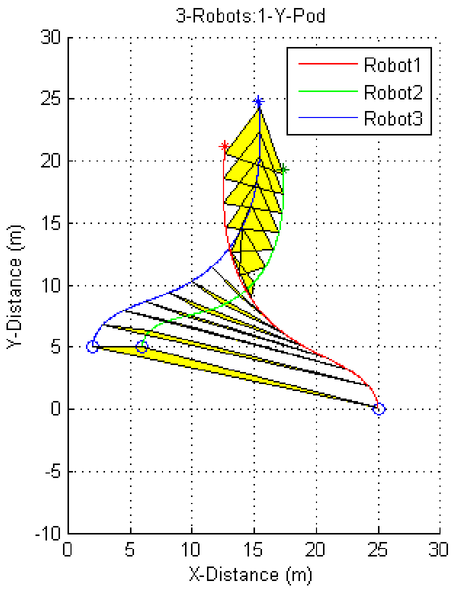
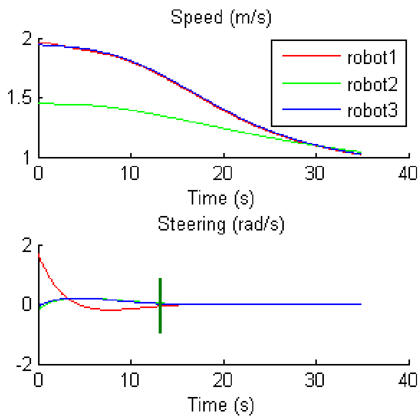
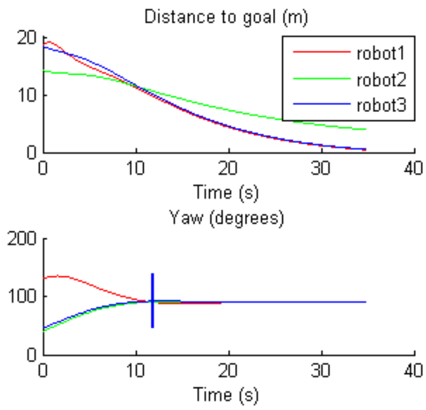


Fig.4.20: Y-Pod form with yellow face colour



(a): Speed and steering



(b): Distance to goal and Yaw angle

Fig 4.21: Variables of the 3 robots forming a Y-Pod without movement. The evolution of the speed, steering, distance to goal and yaw angles of individual robots are shown.

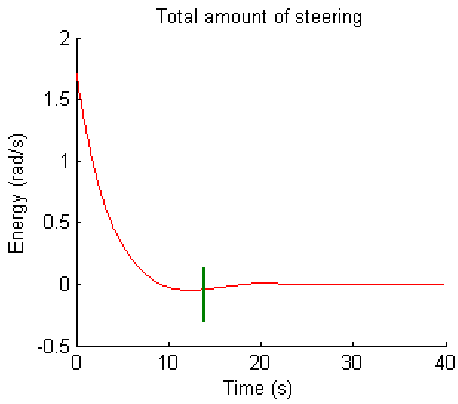


Fig. 4.22: Energy values of simulation of the 3 robots,the total amount of steering tends to zero at 13.2 seconds.

Fig. 4.21 shows the speed and steering, distance to goal and yaw angle of the three robots demonstrating that robots tends to the set goals within the allowed time frame. For instance, Figure 1b shows that the three robots are moving at constant speed and tends to zero after 13.2 seconds. Rather, Fig. 4.21 show the relative orientation of all three robots tending to 90° degrees. We observe that robot2 and robot3 have a similar characters compared to robot1, because robot 1 has a more remote initial position compared to the other two robots. Consequently, robot1 is seen to steer towards negative angles after 3 sec. In this process, steer to positive angles after 13.2 sec.

To further complete our study, we tested the same static case with increasing number of robots forming more Y-Pods. The simulation was run in the same conditions as in the above case. The results of static mode with 9 robots (3 Y-Pods) are shown in Fig. 4.23. The initial positions are different for all robots forming a Y-Pod of a different color (green, blue and red). The settle time of the 3 Y-Pods was set to Ts=14 seconds. Each Y-Pod possesses a same pole and speeds to reach different destination points, as seen in Fig. 4.24. Speed, distance to goal, yaw angle and steering were simulated up to, reach the desired destination points.

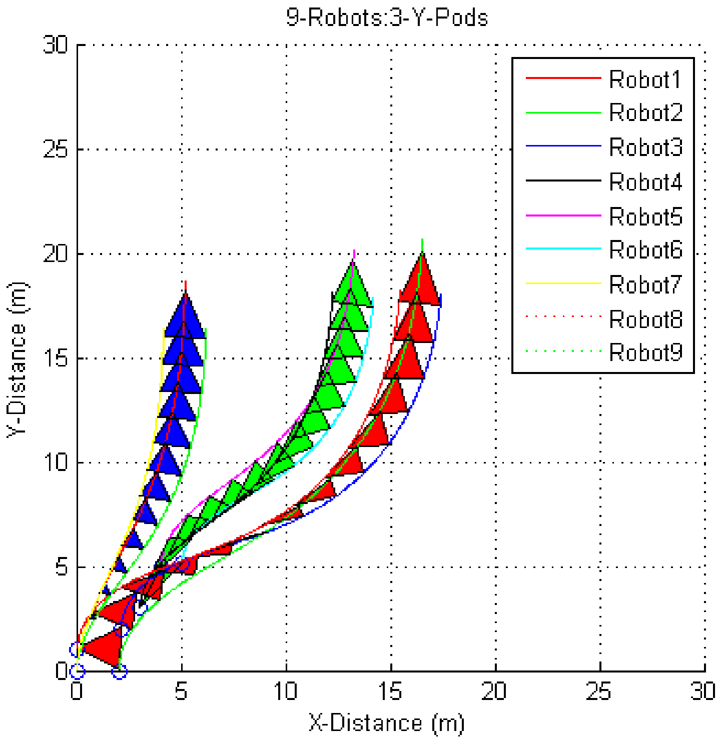


Fig. 4.23: 9-Robots (3 Y-Pods) with the face colors. Red, Green and Blue.

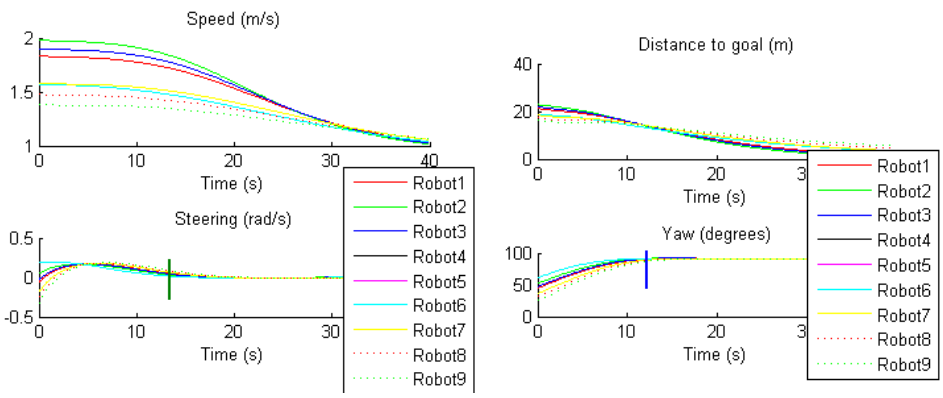


Fig. 4.24: Performance of Y-Pod formation in terms of speed, steering, distance to goal and yaw angle θ with respect to time in the static mode.

ISBN: / DL:

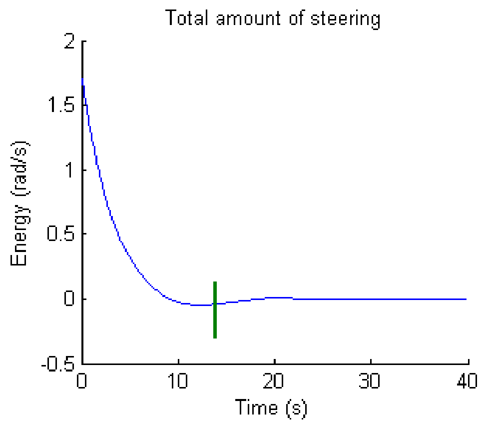


Fig. 4.25: Energy values according to simulation of the 9 robots 3 Y-Pods, i.e. total amount of steering tends to zero after 13.2 seconds

4.4.1.1 Analysis of results

In this section the results of 3-robots with single Y-Pod and 9 robots with 3 Y-Pods are analysed. Table. 4.1 shows, the effect of increasing the number of robots and their corresponding Y-Pods (see appendix for results including more than 9 robots).

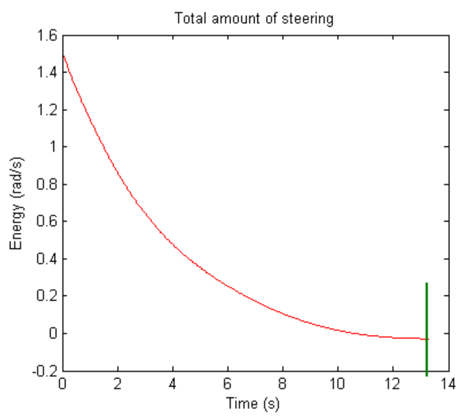


Fig.4.26: Energy values according to the simulation of the robots and Y-Pods. The total amount of steering angular velocity tends to zero at 13.2 seconds.

Fig. 4.26 shows, when energies tend to zero the system is in steady state. According to this, the energy of the mobile robots under formation is proportional to the amount of steering with respect to the set-point (equilibrium point). When the amount of steering tends to zero the system reaches steady state. Energy plot show the controller perform accurately in terms

ISBN: / DL:

of time. Particularly in the static case, the settle time was set to 13.2 seconds and plot show that the steady state is reached in the range of 13 to 14 seconds. The values of steering angles show that our systems is stable, and the amount of steering plot show similar features.

Static-Case Robots&Y-Pods	Theoretical Settle-time Ts [s]	Simulated result measure in ω(t) Actual-time Ta [s]	Total amount of Steering time interval[0s to 14s]
3Robots:1-Y-Pod	13.2	13.2	**13.2**
6Robots:2-Y-Pods	13.2	13.2	13.2
9Robots:3-Y-Pods	13.3	13.5	13.2
12Robots:4-Y-Pods	13.5	13.6	13.2
15Robots:5-Y-Pods	13.7	13.8	13.2
18Robots:6-Y-Pods	13.8	14.0	13.2

Table. 4.1: Settle time, Actual time and steering performance in static-case

The data in Table. 4.1 demonstrates that the controller has the ability to perform the same task when the number of robots is gradually increased. The corresponding settle time and actual time increases slightly, however the energy over the time interval is constant in all cases.

ISBN: / DL:

4.4.2 Linear

Having demonstrated that Y-Pods can be formed in a static case, we further implemented our controller with the linear case, i.e. the virtual leader carries the Y-Pod, and moving linearly in the arena. The desired goal is placed far away from the initial positions of three robots. The steering angle is set to zero (ω=0) and the robots speed is set constant.

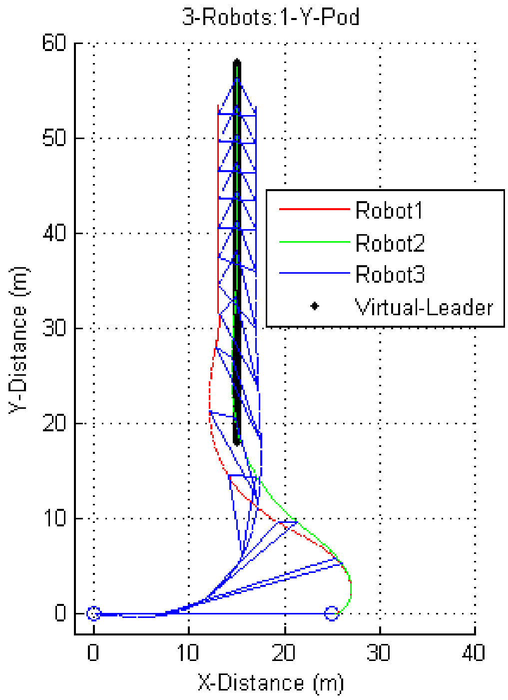
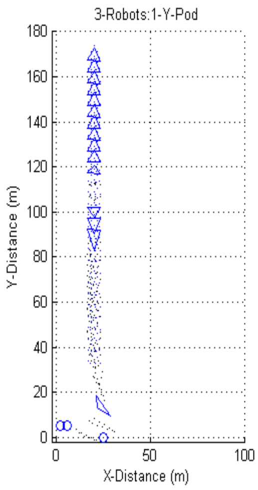


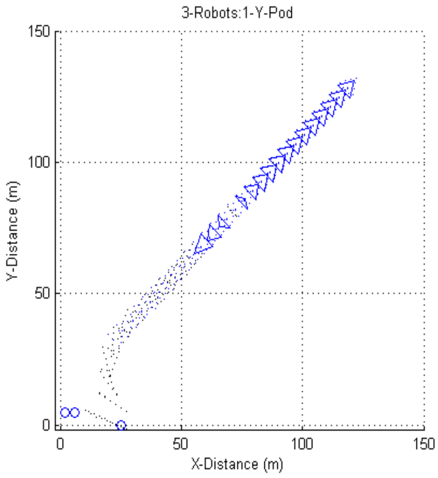
Fig.4.27: X and Y coordinates of Y-Pod (blue triangle) in the linear case with constant speed and steering ω=0. The three robots are moving to maintain the Y-Pod, the virtual leader is indicated with the black dot at the centre of the Y-Pod.

In addition, new figures show in the next pages, which contains Y-Pod shapes and index performances. In this sense, Y-Pod is drawn with dotted and solid lines, when index values (>1) gives an bad values and treated as dotted lines, consequently, index values (<1) gives an better values and its treated as solid lines. On the other words, in this sections , extra results are shown based the patterm matching index technique for individual and comination of Y-Pods. Moreover, delevered steering analysis for individual and combination of Y-Pods are augumented in the follwing figures.

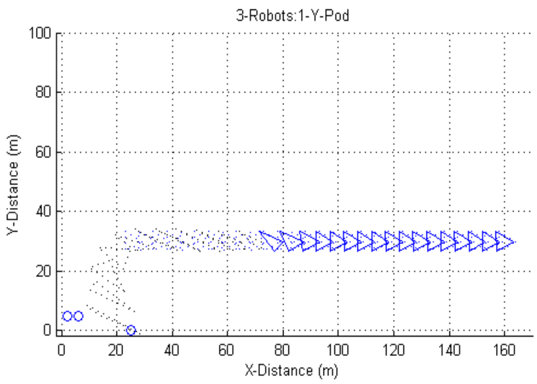
ISBN: / DL:



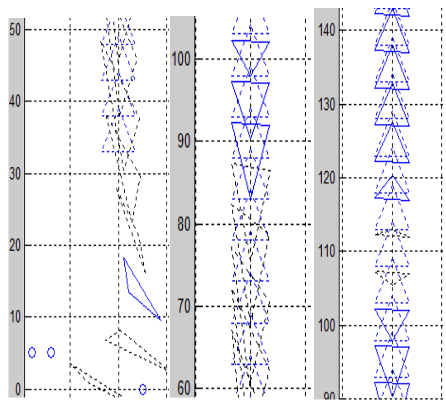
(A):Orientation-Yaw: θ=90⁰



(B):Orientation-Yaw:θ=45⁰



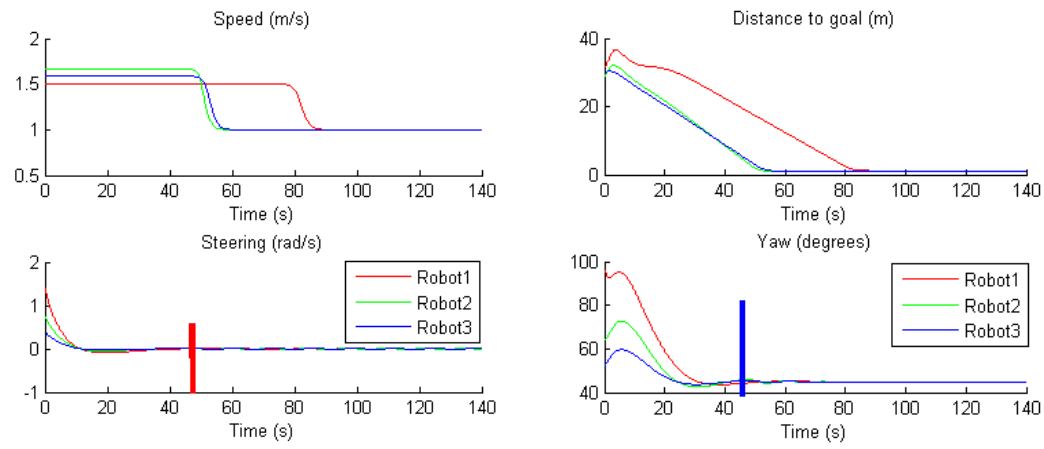
(C): Orientation-Yaw: θ=0⁰ and 180⁰



(D): Snap shot of movement while Y-Pod in linear way

Fig.4.28: Different directions of the Y-Pod movement (A, B, C), and D is the Snap shots of Y-Pod evolution in a linear case, the black dotted line represent a bad index , consequently, blue dotted line represents the ideal Y-Pod and it is initially located at (20 30), and blue solid line represents the actual Y-Pod respectively.

The simulation arena was set with the axis limits (-2 150 -2 150). The theoretical settle time from our controller is set to 34.1 seconds for single Y-Pod and 36.7 seconds for three Y-Pods. While operating the Y-Pod in a linear case, the speed, distance to goal, yaw angle, total amount of steering (i.e. total energy) and index performance is monitored. In this process, individual and combination of Y-Pods are discussed, the results are particularly focused on linear way with the orientation θ=45°.



(a): Speed and steering

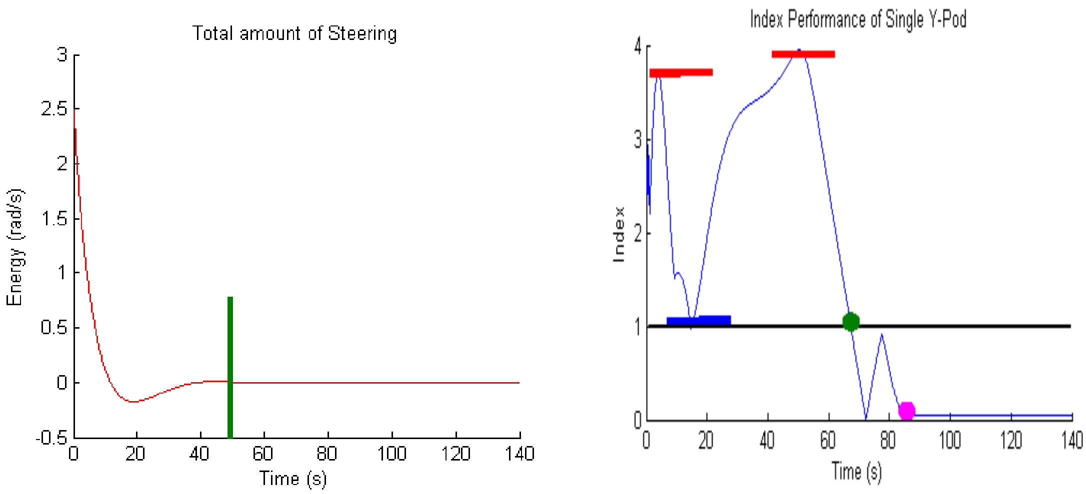
(b): Distance to goal and Yaw angle

Fig.4.29: Speed, steering, distance to goal and yaw angle θ with respect to time of the virtual leader controller carrying single a Y-Pod.

As seen in Fig. 4.29 delivered the results of three robots performance. It consist, (left-a) speed and steering, (right-b) distance to goal and yaw angle. In this sense, (left-a) show, three robots have different initial positions, this is the reason why robot2 and robot3 speeds become constant after 45 seconds and robot 1 is constant after 90 seconds.

In other words, the steering of each robot has some oscillations up to 45 seconds. Thereafter no oscillations are appeared. So, at 45 seconds time all the robots perform constant steering. Finally conclude that the system is stable approximately after 45 seconds, which indicate with the Red line in the Fig. 4.29.

On the hand, in Fig. 4.29 (right-b), consist the distance and yaw angle of each robot. In this issue, The distance from the initial position of robots to goal positions are performed according to the speed such as, robot1, robot2 and robot 3 distances tends to zero concerning with the speed at each iteration sample time by 1/10s. Although yields, distance behaviour same as in speed case over the time. Rather, each robot is pointing towards 45° yaw angle. It show , up to 45 seconds time all robots has some oscillation in order to reach the exact 45° orientation, afterword yaw angle is constant, which is represents the blue line in fig. 4.29. Moreover, yaw angle has constant movement, due to steering is i.e. ω=0 in this particular case. Finally, conclude that, the three robots has same characteristic with respect to speed, distance, steering and yaw angle.



(A): Total amount of steering (energy)   (B): Index performance between 0 to 1

Fig.4.30: (left) Evolution of total amount of steering and (right) Index performance for three robots and single Y-Pod in linear case.

In addition, Fig. 4.30 shows total amount of steering (left-A) and shape evolution using index performance (right-B). For instance, (left-A) gives the total of amount steering of all three robots. Three robots has constant steering at 45 seconds, which represents green line in the above in the Fig. 4.30. It can be seen that the steering tends to zero at 45 seconds. Afterword's, it perform the constant steering. We can confirm, that all robots has synchronization features at 45 seconds in order to form a perfect shapes. Therefore, it appears that our system is synchronized and appears stability features.

Now, we would like to provide for a convenient way to monitor the shape evolution while the Y-Pod is moving toward the target. Y-Pod shape evolution is studied in section 4.2.2,

ISBN: / DL:

where the motion pattern is continuously measured and compared to an ideal Y-Pod to actual Y-Pod. Difference between ideal and actual Y-Pods are fixed with the range 0 to 1.

Before explains the performance of index for shape evolution, we arose some questions , those are, the robot positions at each point is properly maintains or not, is the shape formation at every sample time, robots corresponding Y-Pods are properly connected or not. Those are the main questions raised. In order to overcome these questions, we used pattern matching index technique (see section 4.2.2) to justify the shape evolution.

In the Fig. 4.30 (right-B) depicts the shape evolution graph in which a threshold is symbolised by a black line corresponding to an index value of 1. In this issue, represents red bar is bad shape, blue bar is trying to reach good shape, green circle is good shape and pink circle is absolute minima respectively.

The upper part of the threshold corresponds to the bad index values and lower part to the good index values. In other words, upper part of the threshold the red bars are placed at the highest index value peaks corresponding to an extremely mismatched shape, while the blue bar is trying to make the profile is converting to a good shape.

In the lower part of the threshold green circle correspond to a perfect shape matching. Furthermore, pink circle correspond to absolute minima. In this sense, Fig. 4.30 (right-B) it, appears that up to 65 seconds the shape does not match , but a good shape evolution received approximately over the time at 65 seconds, which indicate the green circle. Thereafter the system adopts a good shape continuously. Meanwhile 85 seconds, it appears perfect shape, it can says that absolute minima, and moving constantly without changing the shape while the system perform the task. Therefore, overall system maintains good shape upon reaching at the 65 seconds. Furthermore, theoretical settle time of our controller is 34.1 seconds, while the index performance, the settle time occurs at 65 seconds. It gives us the Y-Pod shape. In this regard, we can form a Y-Pod formation in any scales. It can see that the system has scalability in all sizes.

Based on the total amount of steering and index performance analogy from the Fig. 4.30, the keen observations as follows that, total amount of steering will produces the system is synchronized at 45 seconds settle time. Other hand shape evolution is appeared at the 65 seconds settle time. The difference between theoretical settle time of our controller and total amount of steering is approximately 20 %, even more shape evolution is approximately 20%. In other words, vary, settle time from the simulations of total amount of steering and index shape evolution is 20 %.

Analysis from the above results of Fig. 4.29 and 4.30, gives us, the synchronization of all robots based on speed and steering plots are achieved. Moreover, without oscillations with constant orientation of the distance and yaw angle of all three robots tends to zero (constant steering-stable) are obtained. In other words, the total amount of steering and shape evolution of Index performance plots are gives us the system settle times. In all cases experiment settle time obtained 45 seconds expect index (i.e. 65 second).With this evidence, we can form a Y-Pod shape formations with stability and scalability for more number of robots with corresponding Y-Pods.

Furthermore, we tested the same phenomena with increasing the number of robots and Y-Pods. In this scenario, the simulation was run with 9 robots and 3 Y-Pods.

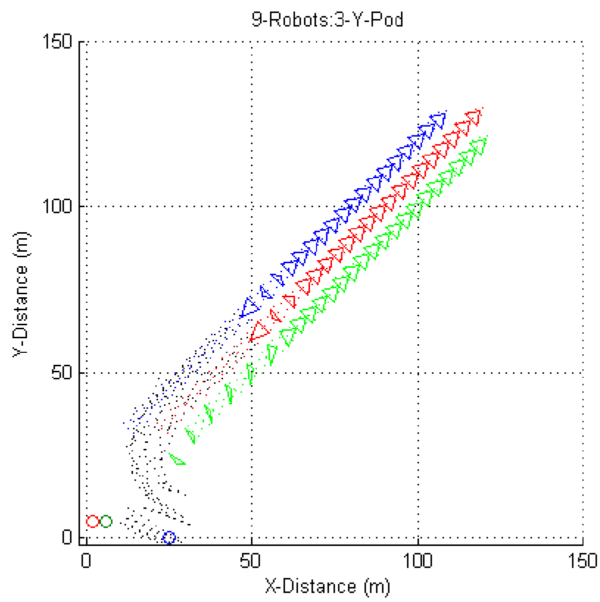
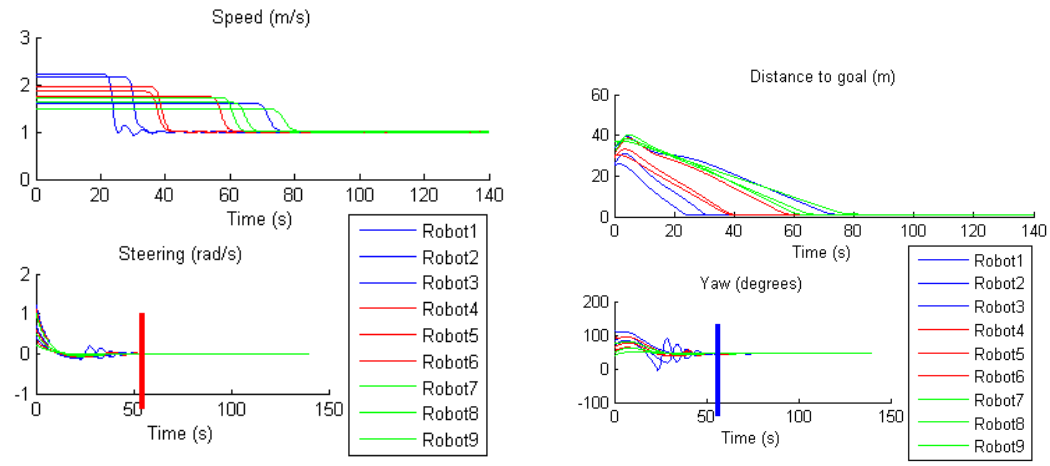


Fig.4.31: Y-Pods in a linear case at constant speed and steering angle ω=0.Y-Pods shown in blue, red and green.

ISBN: / DL:

The theoretical settle time from our controller is set to 36.7 seconds for three Y-Pods. In this context, results and simulation set-ups has same feature like 3 robots case. So we restricted our detailed explanation in this study, due to its similar features of 3 robots and single Y-Pod case. The difference here is the increasing number of robots corresponding its Y-Pods. In addition added, extra results of individual Y-Pod with respect to steering and index evolution respectively.



(a): Speed and steering

(b): Distance to goal and Yaw angle

Fig.4.32: Data relative to speed, steering, distance to goal and yaw angle θ of 9 robots in linear case.

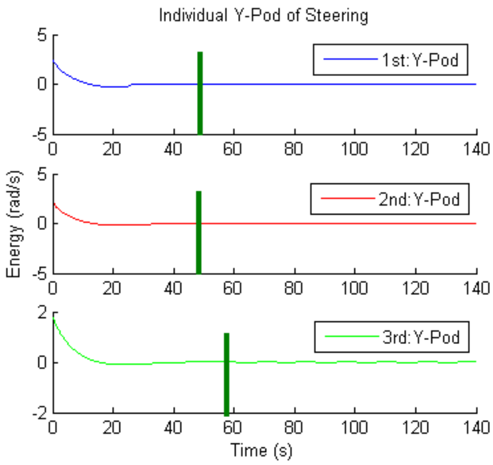
In the Fig. 4.32 show the results of nine robots performance. It consist, (left-a) speed and steering, (right-b) distance to goal and yaw angle. In this sense, (left-a) show, nine robots have different initial positions. Speeds become constant after 55 second and some of them are at 90 seconds depends on the initial positions of each robots.

In other words, the steering of each robot has some oscillations up to 55 seconds. Thereafter no oscillations are appeared, all the robots perform constant steering. Therefore, the system is stable approximately after 55 seconds, which indicate with the Red line in the Fig. 4.32.

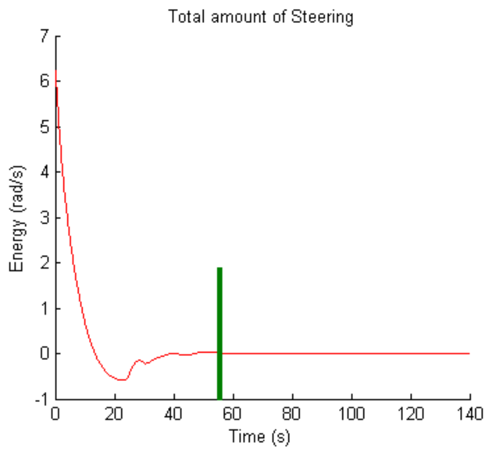
On the other hand, in Fig. 4.32 (right-b), consist the distance and yaw angle of each robot. It yields, distance behaviour same as in speed case over the time. Moreover, yaw angle has

ISBN: / DL:

constant movement, at 55 seconds settle time, due to its steering ω=0 in this particular case. Finally, conclude that, the nine robots has same characteristic with respect to speed, distance, steering and yaw angles.



(A) individual Y-Pod steering

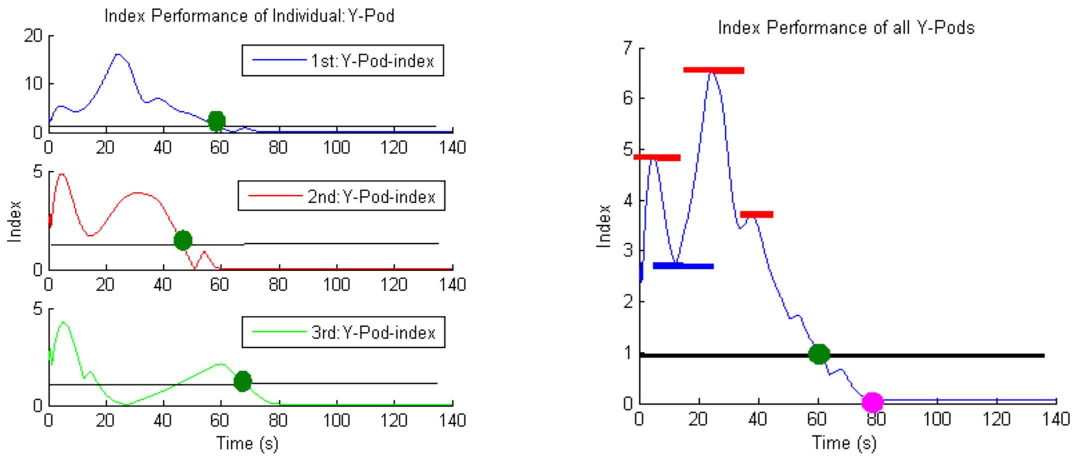


(B) Total amount of steering

Fig.4.33: Evolution of total amount of steering of 3 Y-Pods in linear case.

In addition, Fig. 4.33 shows individual Y-Pod steering (left-A) and total amount of steering (right-B). For instance, (left -A) gives the individual of Y-Pod steering. In this process 1st 2nd and 3rd Y-Pods steering is constant at 54, 50 and 58 seconds respectively, which indicate with green line. (right -B) gives the total of amount steering of three Y-Pods. Three Y-Pods has constant steering at 55 seconds, which represents green line in the above Fig. 4.33. It can be seen that the steering tends to zero at 55 seconds. In other words, the steering is tends to zero approximately 55 seconds for total amount of steering and individual steering of Y-pods. Therefore, the system gives us the settle time is 55 seconds.

ISBN: / DL:



(A) Individual Y-Pod Index (B) Average Y-Pod Index

Fig.4.34: Evolution of index performance of 3 Y-Pods in linear case

In the Fig. 4.34 depicts the shape evolution graphs, individual Y-Pods (left-A) and Average Y-Pods (left-B). For instance, individual Y-Pod index (left-A) case, a good shape evolution received approximately over the time at 65, 50, 70 seconds corresponding 1st, 2nd and 3rd Y-Pods respectively. On the other hand, average Y-Pod index (left-B) case a good shape evolution received approximately over the time at 65 seconds. Therefore shape evolution of both the cases approximately appeared good shape at 65 seconds.

4.4.2.1 Analysis of results

The minimum and maximum theoretical settle time of our controller is set to 34.1 and 40.2 seconds for the combination of robots and Y-Pods. In this context, results of 3,6,9,12,15 and 18-robots corresponding 1, 2, 3, 4, 5 and 6 Y-Pods are analysed. Additional results of more robots (Y-Pods) show in Tables. 4.2 and 4.3 refer appendix. In this concern, Table. 4.2 show results based on total amount of steering and index performance data. Which involves, row contains, theoretical settle time, total amount of steering and index performance respectively. Consequently, column contains combination of robots and Y-Pods. In addition, another Table 4.3 is presented with data concerning with single Y-Pod to understand the system synchronization and scalability. We examine simulated results to understand the behaviour of whole system in linear case. The keen interest is to understand the performance in terms of steering and index. In this issue, combinations of robots and Y-Pods are presented with settle time measures based on steering and index performance.

ISBN: / DL:

Linear-Case Robots&Y-Pods	Settle-Time Ts [s]	Steering performance time interval [s]				Index performance time interval [s]			
		0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
3Robots:1Y-Pod	34.1	45	Stable	stable	Stable	65	absolut min	-do-	-do-
6Robots:2Y-Pod	35.0	45	Stable	-do-	-do-	65	absolute min	-do-	-do-
9Robots:3Y-Pod	36.7	55	Stable	-do-	-do-	65	absolute min	-do-	-do-
12Robots:4Y-Pod	38.0	55	stable	-do-	-do-	65	absolute min	-do-	-do-
15Robots:5Y-Pod	39.1	58	Stable	-do-	-do-	68	absolute min	-do-	-do-
18Robots:6-YPod	40.2	58	stable	-do-	-do-	70	absolute min	-do-	-do-

Table. 4.2: Settle-time using total amount of steering and index performances data.

Results from the Table. 4.2, the difference between theoretical settle time of our controller and the simulation results settle time of steering and index performance is approximately similar, but little variation occurs, that approximately considered as 20%. So with this evidence, we justify that, achieved good shape and its evolution is stable in the steady state with respect to steering. In addition, with respect to absolute minima based on settle-time achieved perfect shapes. We can justify that the shape has ability to apply in communication and network problems.

Table. 4.3 show data concerning with individual Y-Pod in a linear case. In this process used the 18 robots with 6 Y-Pods (i.e. combinations 18 robots 6 Y-pods in linear case), which is split in to individual Y-Pod and performed one after another. With different initial position of robots and Y-Pods. In this regards, individual Y-Pods act and maintain the same feature like combinations of Y-Pods.

For instance theoretical settle time of our control law is set to minimum and maximum is 31.2 to 36.4 seconds, more or less similar to combination of Y-Pods. There exist little variations compare to combination of Y-Pods, due to its initial positions. Table. 4.3 show the data concerning with individual Y-Pods in a linear case to identify and compare the settle time in terms of steering and index performance evolutions.

ISBN: / DL:

Linear	Individual	Each Y-Pod Steering Time [s]				Each Y-Pod Index Time [s]			
Individual Y-Pod	Settle-TimeTs [s]	0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
1st -Y-Pod	31.2	40	stable	stable	stable	55	absolute min	-do-	-do-
2nd -Y-Pod	31.4	42	stable	-do-	-do-	55	absolute min	-do-	-do-
3rd -Y-Pod	33.4	42	stable	-do-	-do-	60	absolute min	-do-	-do-
4th –Y-Pod	34.2	43	stable	-do-	-do-	61	absolute min	-do-	-do-
5th –Y-Pod	35.0	45	stable	-do-	-do-	60	absolute min	-do-	-do-
6th –Y-Pod	36.4	45	stable	-do-	-do-	65	absolute min	-do-	-do-

Table 4.3: Individual Y-Pod settle-time measured based on steering and index Performance.

Results from the Table. 4.3, the difference between theoretical settle time from our controller and the simulation results of settle time of steering and index performance is approximately similar, but little variation occurs, that is considered as 10%. So with this evidence, the robot does not appear oscillations and steering tends to zero with proposed settle time. Therefore robots moving in constant steering without oscillations, and it exhibits the system is well synchronized. Apart from this, achieved good shape and its evolution is stable in the steady state with respect to synchronization. With this evidence, guarantee the system can have more number of robots and Y-Pods in order to form a large scale swarm formation without synchronization problems.

4.4.3 Quadratic

In this section the virtual leader carry the Y-Pod in a quadratic approach like curved path with the limiting axis (-2 150 -2 150), and maximum time 1500 seconds for particularly established arena. In this concern, the steering angle set-up is not equal to zero ($\omega \neq 0$) and it has constant speed. Even, orientation angle is not constant because it vary due to Y-Pod moves in the different ways. Moreover, robots are placed at random positions far enough from the goal location, so that all three robots are free but still under influence of the virtual leader. It's clear that the initial and goal position of the robots and other parameters are same as in the previous sections. So we are not going to explain those characteristics.

ISBN: / DL:

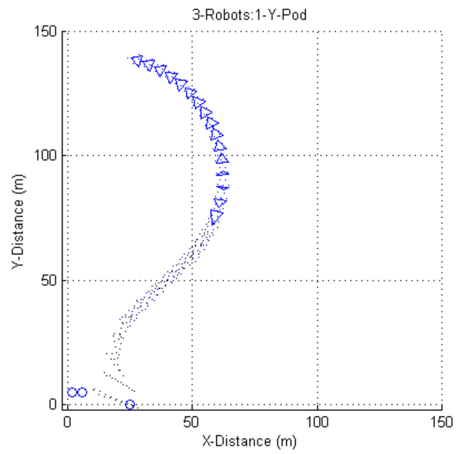
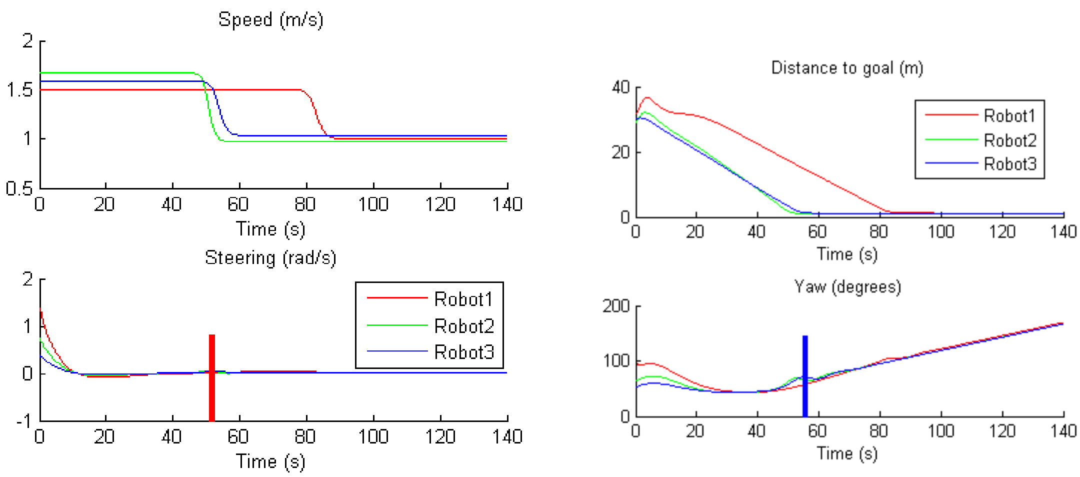


Fig.4.35: Y-pod with 3 robots moving in a quadratic way, constant speed and steering angle ω≠0



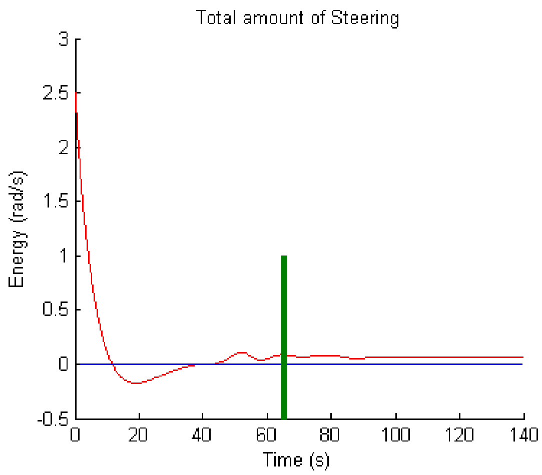
(a): Speed and steering

(b): Distance to goal and Yaw angle

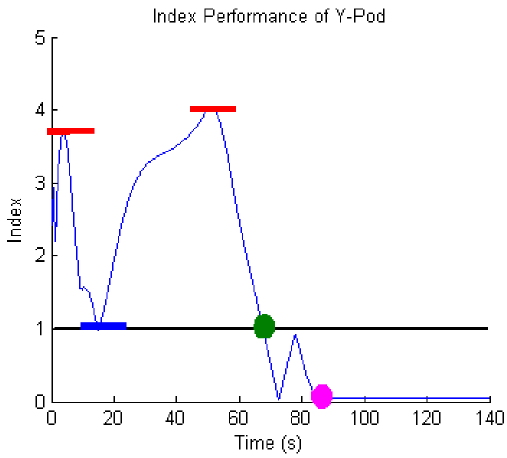
Fig.4.36: Plot of variables of 3 robots (1-Y-Pod) in quadratic case.

For instance, in the Fig. 4.36, three robots moving in quadratic way. It consist, (left-a) speed and steering, (right-b) distance to goal and yaw angle. In this issue, (left-a) the steering of each robot has some oscillations up to 50 seconds, which can says that, settle time from the simulation data is achieved 55 seconds. Thereafter no oscillations are appeared. So, at 50 seconds time all the robots perform constant steering. Finally conclude that the system is stable approximately after 50 seconds, which represents with the Red line in the Fig. 4.36.

In addition, Fig. 4.36 (right-b), consist the distance and yaw angle of each robot. In this issue, particular focus on yaw angle, which is differ from time due to its curved trajectories up to 58 seconds, it appears oscillations. Thereafter moves without oscillation, which is represents the blue line in fig. 4.36. It will move with different yaw angle, start with 45^0 angle, increases up to 150^0, it gives us, our Y-Pod moving in curved way. Moreover, yaw angle has constant movement, due to steering is $\omega \neq 0$ in this particular case. Finally, conclude that, the three robots moves in different yaw angle but we observe no oscillation at 58 seconds settle time onward.



(A): Total amount of steering (energy)



(B): Index performance between 0 to 1

Fig. 4.37: 3robots (1-YPod), (left) total amount of steering, (right) index performance in quadratic case

Fig. 4.37 gives us the total amount of steering (A) and index performance (B). In this regard, from left-A total amount of steering is constant at 65 seconds, which gives the system settle time, it indicates with green line in Fig 4.38. There after 65 seconds it exhibits constant steering but it will not tends to zero because of steering $\omega \neq 0$, its represents has a blue line in the Fig. 4.37. Subpart right-B indicates the shape evolution , it can see that after 67 seconds appears a good shape which indicate with green circle, continues the shape at 85 sec it will be perfect shape it is treated as absolute minima which indicate the pink circle.

Particular observations of the quadratic case such as, the total amount of steering and index performance from the Fig. 4.37, the key identification as follows that, total amount of steering will produces the system is synchronized at 65 seconds settle time. Other hand shape evolution is also appeared at the 67 seconds settle time, which vary in 2 seconds with respect to steering and index. Its gives us the robots are well regulated with perfect shape. Theoretical settle time of our controller and total amount of steering is approximately 10 %, even for shape evolution is approximately 10% variations appears. It has better results compare in linear way even due to quadratic way.Furthermore, we tested the same with increasing the number of robots and Y-Pods. In this scenario, the simulation was run with 9 robots and 3 Y-Pods.

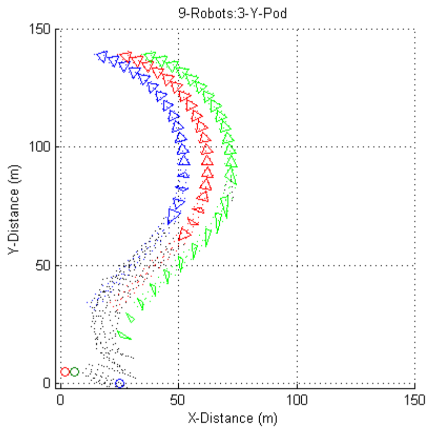
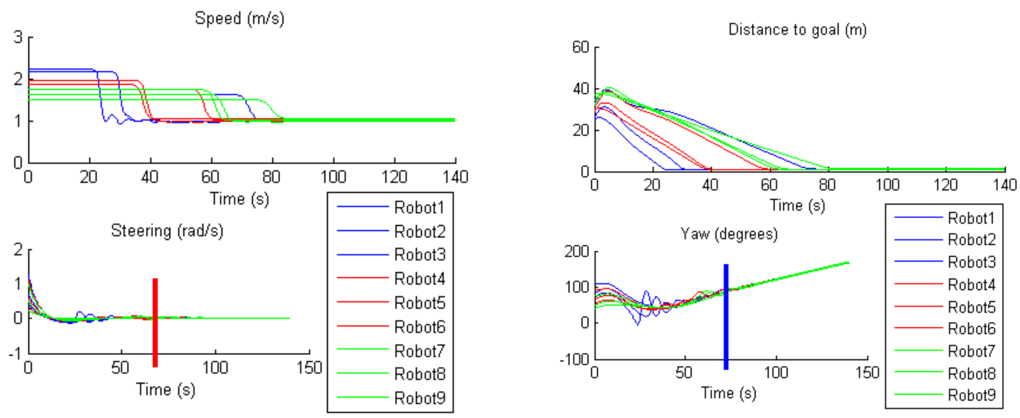


Fig. 4.38: 9-Robots 3-Y-pod is moving in a quadratic way .i.e. constant speed and steering angle ω≠0.



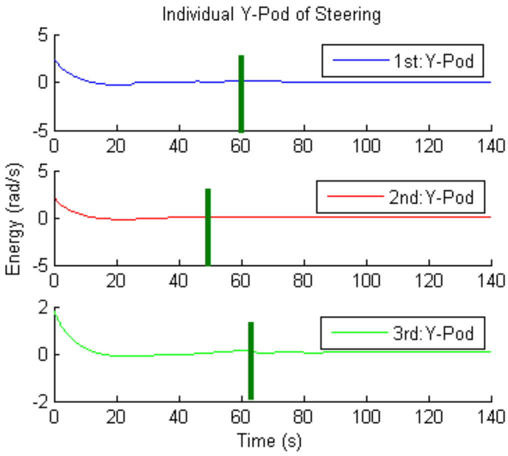
(a): Speed and steering

(b): Distance to goal and Yaw angle

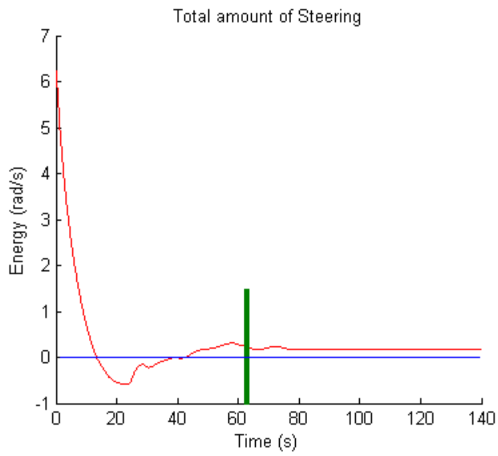
Fig. 4.39: 9-robots, 3-yopds in terms of speed, steering, distance to goal and yaw angle θ w.r.t.to time.

ISBN: / DL:

In the Fig. 4.39 focus on yaw angle, which is differ from time due to its curved trajectories up to 65 seconds and appears oscillations. Thereafter moves without oscillation and robot steering left side gives us 65 seconds of time has constant steering, these simulations results indicate the settle time of the system.



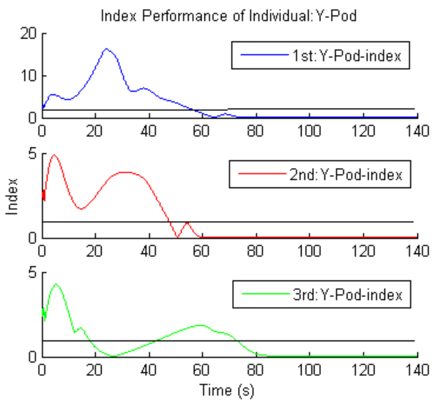
(A) individual Y-Pod steering



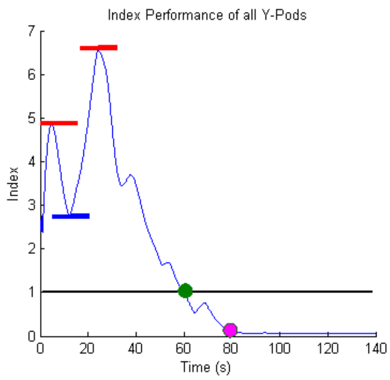
(B) Total amount of steering

Fig. 4.40: Individual and Total amount of steering 9-robots, 3-Y-Pods.

Above Fig. 4.40 show the results of individual steering (left) vs total amount of steering (right) steering, individual steering of each Y-Pod has constant steering at 60,58 and 63 seconds . Contrary, total amount of steering has 65 seconds. Compare to both left and right cases in the Fig. 4.40, it has similarities. Constant steering indicate that the simulation settle-time which has 65 seconds in this particular case.



(A) Individual Y-Pod Index



(B) Average Y-Pod Index

Fig. 4.41: individual and average of index 9-robots, 3-yopds in quadratic case.

The Fig. 4.41 show the results of shape evolution graphs, individual Y-Pods (left-A) and Average Y-Pods (left-B). For instance, individual Y-Pod index (left-A) case, a good shape evolution received approximately over the time at 60, 55, 75 seconds corresponding 1st, 2nd and 3rd Y-Pods respectively. On the other hand, average Y-Pod index (left-B) case a good shape evolution received approximately over the time at 63 seconds. Therefore shape evolution of both the cases approximately appeared good shape at 65 seconds.

4.4.3.1 Analysis of results

In this section illustrate the results of quadratic approach, the results has small variations compare to linear approach, due to its curvature path. It yields, interesting results and explained based on simulated data augmented in Tables. 4.4 and Table. 4.5. In addition tables contain the same components as in linear case, so we restrict our explanation about those components.

We examine simulated results to understand the behaviour of whole system in quadratic case. The keen interest is to understand the performance in terms of steering and index. Settle time measures based on steering and index performance.

Quadratic-Case Robots&Y-Pods	Settle-Time Ts [s]	Steering performance time interval [s]				Index performance time interval[s]			
		0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
3Robots:1Y-Pod	34.1	55	Stable	Stable	Stable	65	absolute min	-do-	-do-
6Robots:2Y-Pod	35.0	58	Stable	-do-	-do-	65	absolute min	-do-	-do-
9Robots:3Y-Pod	36.7	65	Stable	-do-	-do-	63	absolute min	-d0-	-do-
12Robots:4Y-Pod	38.0	68	Stable	-do-	-do-	69	absolute min	-do-	-do-
15Robots:5Y-Pod	39.1	68	Stable	-do-	-do-	69	absolute min	-do-	-do-
18Robots:6Y-Pod	40.2	68	stable	-do-	-do-	70	absolute min	-do-	-do-

Table. 4.4: Steering and index performance in quadratic case.

From the above data we can examine that has constant steering approximately at 65 seconds. For instance, in liner case the settle time of constant steering at 45 seconds and quadratic

ISBN: / DL:

case 65 seconds, so the difference is 20 seconds over here, because of its curvature path, and steering ω≠0. There is slight change in the path, these is the reason why after 65 seconds the system has constant steering. Index validation of Table. 4.4 And Table. 4.5 gives us overall data to make good shape evaluation at 60 seconds. Thereafter exists absolutely minima.

Results from the Table. 4.4, the difference between theoretical settle time from our controller and the simulation results of settle time of steering and index performance is approximately similar, but little variation exists that considered as 15%. So with this evidence, we justify that, achieved good shape and its evolution is stable in the steady state with respect to steering. In addition, absolute minima based on settle-time achieved perfect shapes. We can justify that the shape has abilities to apply in communication and network problems.

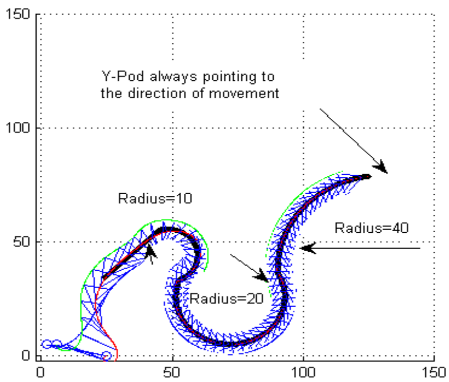
Quadratic	Individual	Each Y-Pod Steering Time [s]				Each Y-Pod Index Time [s]			
Individual Y-Pod	Settle-TimeTs [s]	0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
1st -Y-Pod	31.2	60	Stable	Stable	Stable	60	absolute min	do-	-do-
2nd -Y-Pod	31.4	60	Stable	-do-	-do-	63	absolute min	-do-	-do-
3rd -Y-Pod	33.4	65	Stable	-do-	-do-	65	absolute min	-do-	-do-
4th –Y-Pod	34.2	65	Stable	-do-	-do-	65	absolute min	-do-	-do-
5th –Y-Pod	35.0	68	Stable	-do-	-do-	68	absolute min	-do-	-do-
6th –Y-Pod	36.4	68	Stable	-do-	-do-	68	absolute min	-d0-	-do-

Table. 4.5: Individual Y-Pod Performance and analysis of steering and index based on quadratic case.

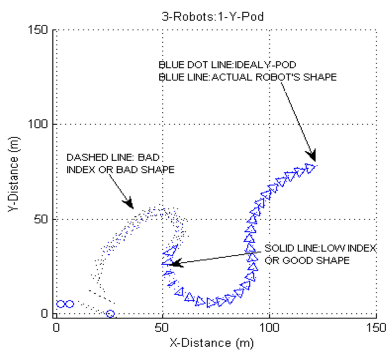
In addition, Table. 4.5, show the individual Y-pod performance, it gives us the individual Y-Pod settle time with steering and index data, some of the Y-Pods has different values, due to its distance from the initial position for each Y-pod. Steering of each Y-Pod can see in Table. 4.5, and it is constant at 60seconds, which gives us the settle time for the simulations. Other than this, individual Y-Pod index has good shape appears at 60 seconds. Therefore we conclude that the steering vs index has the same similarities, justify that it has good shape and also at the same settle time steering has steady state. With this evidence we can overcome the synchronization problems. Even apply for large scales of swarm systems in scalable approaches with stability features.

4.4.4 Combined case-A

Particularly, demonstrated the combine A and B cases. Main reasons, to perform these simulations to know that the Y-Pod can move in all direction to cover entire arena. Although the system performs in terms of linear and quadratic way. In this sense, we can see all possible states in one simulation to understand the steering and index performances. The Y-Pod will move in linear, quadratic and again in linear way. It consist of different steering and orientations , in this case steering $\omega \neq 0$ and has constant speed .Combined case with detailed explanation of Y-Pod with three robots are shown below , consider the black line is a virtual leader for the direction moment of corresponding robots.



(A)



(B)

ISBN: / DL:

Fig. 4.42: Single Y-Pod moving in the combined case with the constant speed and steering ω≠0. In this details sketch, three robots are moving to mainatins the Y-Pod, virtual leader indiacte the black line and it's centered in the Y-Pod to pointing toward the goal location, and Y-Pod is colored bule triangle shape to reach the desired desitaion positions. Particularly in Fig. 4.42(A) shown the y-pod moving with specific radius (B) .and shows indidex performnace regions expalind in Fig. 4.43.

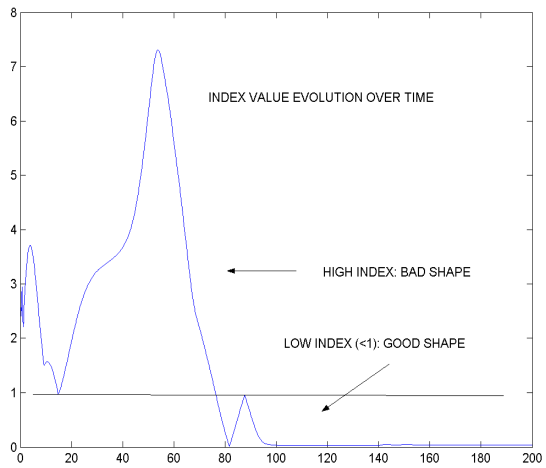
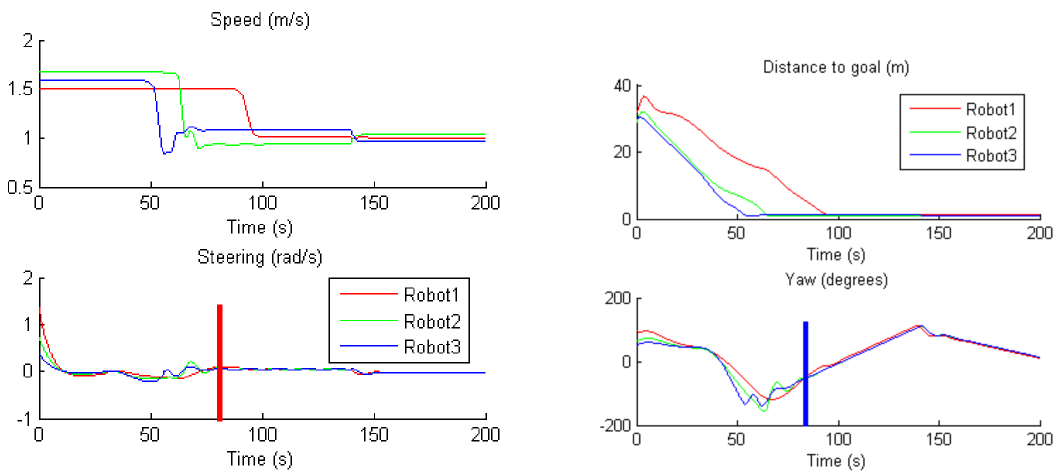


Fig. 4.43: Y-Pod index evaluation w.r.t to time and range 0 to 1



(a): Speed and steering

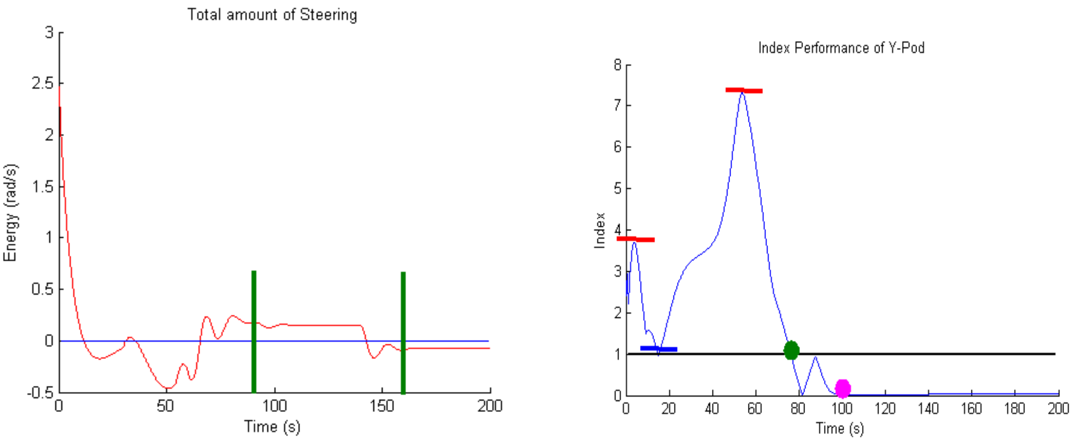
(b): Distance to goal and Yaw angle

Fig. 4.44: 3robots and single y-pod variables in terms of speed, steering, distance to goal and yaw angle θ.

The Fig. 4.44 show same characteristic as linear and quadratic approaches. So we restricted our discussion here. Only the difference is settle time, in the above cases has less settle time

ISBN: / DL:

rather than combined one. Because it involves both the approaches. In the Fig. 4.44 has constant steering at 85 seconds, which give the settle time from simulation results. In other word, yaw angle also obtains 85 seconds settle time, at this time all robots will overcome the oscillation and move in various directions in steady state constant yaw angle.



A. Total amount of steering B. Index performance between 0 to 1

Fig. 4.45: Evolution of total amount of steering and index performance

In addition Fig. 4.45 show the total amount of steering (A) and index performance (B). In this regard, from left-A total amount of steering is constant at 85 seconds, which gives the system settle time, it indicates with two green lines at 85 seconds and 165 seconds in Fig 4.45. There after exhibits constant steering but it will not tends to zero because of its combined path (i.e. linear and quadratic) and steering (ω≠0), its represents has a blue line in the Fig. 4.45.

Consequently, compare to other cases there exists two green line at 85 seconds and165 seconds in total amount of steering plot , it gives us the particular zone that the robots and Y-Pods turn in to another state (i.e. linear to quadratic state), with this behaviour we can justify that , our approach has some reactive based symptoms. In other words in this zone ,Y-Pod will move with constant steering but at 85 seconds settle time it has the steady state. Thereafter it maintains the steady state. On the other hand, in Fig. 4.45 right hand side -B indicates the shape evolution , it can see that after 80 seconds appears a good shape it indicate with green circle, continues the shape at 100 seconds it will be perfect shape with absolute minima which indicate the pink circle.Particular observations from the combined case such as, the total amount of steering and index performance from the Fig. 4.45, the key identification as follows that, total amount of steering will produces the system is synchronized at 85 seconds settle time. Other hand shape evolution is also appeared at the 80 seconds settle time, which vary in between 5% with respect to steering and index. Its gives us the robots are well regulated with perfect shape. Furthermore, we tested the same with increasing the number of robots and Y-Pods as follows:

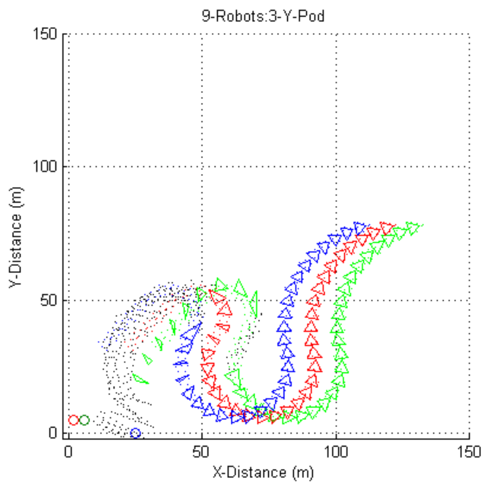
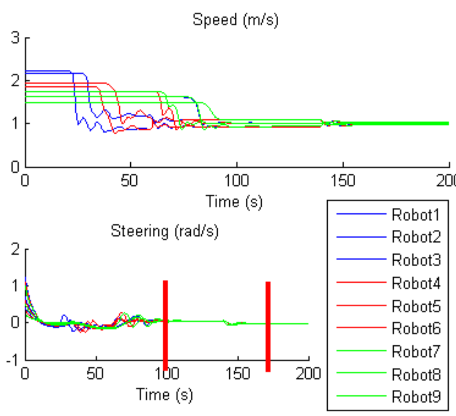
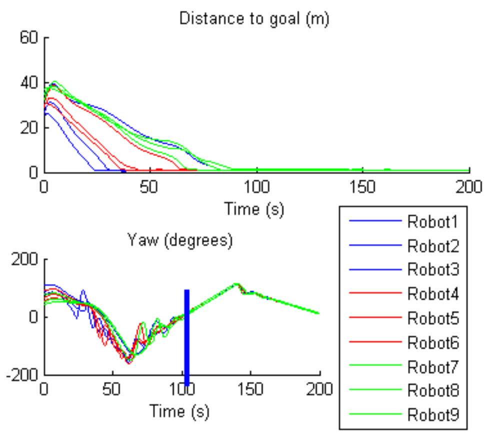


Fig.4.46: 3Y-pod is moving in a combined way, constant speed and steering angle ω≠0.



(a): Speed and steering

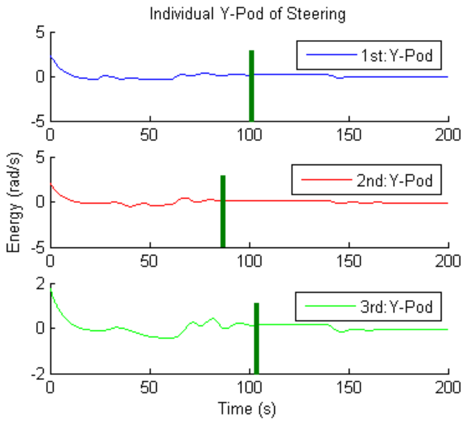


(b): Distance to goal and Yaw angle

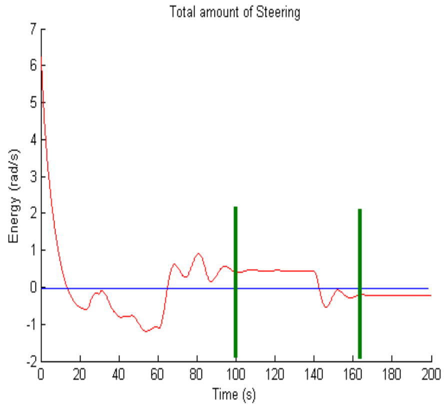
Fig.4.47: variables terms of speed, steering, distance to goal and yaw angle θ for combined case -A.

ISBN: / DL:

The Fig. 4.47 show results of nine robots in combined case A. In this issue received constant steering at 100 seconds, which give the settle time of simulation results. In other word, yaw angle also obtains 100 seconds settle time, at this time all robots will overcome the oscillation and move in various directions in steady state with constant yaw angle.



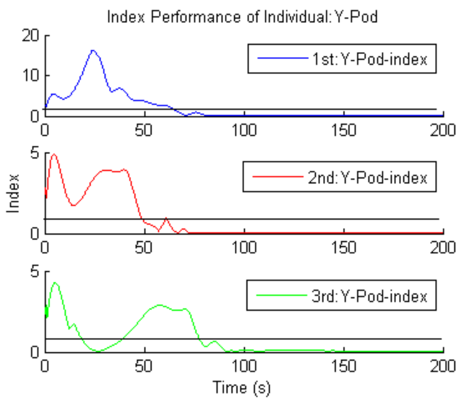
(A) individual Y-Pod steering



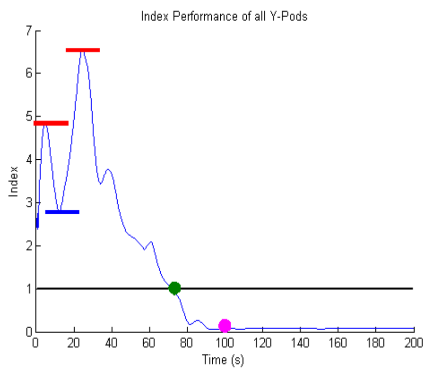
(B) Total amount of steering

Fig.4.48: Energy values according to the controller with linear state and the steering angles.

The Fig. 4.48 show the individual Y-Pod steering (A) and total amount of steering (B). In this regard, individual Y-Pod steering has obtain settle time 100 seconds from right-A total amount of steering is constant at 100 seconds, which gives the system settle time, it indicates with two green lines at 100 and 165 seconds. Compare to 3 robots case in combined-A, there exists little variation due to more no of Y-Pods which is 15%.



(A) Individual Y-Pod Index



(B) Average Y-Pod Index

ISBN: / DL:

Fig.4.49: 3Y-Pod and 9 robots index evolution w.r.t to time and range 0 to 1

From the Fig. 4.49 results of shape evolution graphs, individual Y-Pods (left-A) and Average Y-Pods (left-B). For instance, individual Y-Pod index (left-A) case, a good shape evolution received approximately over the time at 60, 50, 80 seconds corresponding 1st, 2nd and 3rd Y-Pods respectively. On the other hand, average Y-Pod index (left-B) case a good shape evolution received approximately over the time at 78 seconds. Therefore shape evolution of both the cases approximately appeared good shape at 78 seconds.

4.4.4.1 Analysis of results

In this section illustrate the results obtained from the simulated data in combined-A approach, the results has small variations compare to linear and quadratic case. It yields, interesting results and explained based on simulation data augmented in Tables. 4.6 and Table. 4.7.

Combined-CaseA Robots&Y-Pods	Settle-Time Ts [s]	Steering performance time interval [s]				Index performance time interval[s]			
		0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
3Robots:1Y-Pod	34.1	85	Stable	stable	stable	80	absolute min	86	-do-
6Robots:2Y-Pod	35.0	85	Stable	-do-	-do-	80	absolute min	18	-do-
9Robots:3Y-Pod	36.7	100	Stable	-d0-	-do-	78	absolute min	90	-do-
12Robots:4Y-Pod	38.0	100	Stable	-do-	-do-	80	absolute min	90	-do-
15Robots:5Y-Pod	39.1	110	Stable	-do-	-do-	80	absolute min	90	-do-
18Robots:6Y-Pod	40.2	110	Stable	-do-	-do-	80	absolute min	110	-do-

Table. 4.6: Steering and index performance with respect to robots Y-Pods in Combined case-A.

From the above data of Table 4.6 , can examine that, it has constant steering approximately at 85s for single Y-Pod and 100s from three Y-Pods. settle time archives from steering and index performance has 80 and 78 seconds approximately. Difference with respect to our theoretical controller and simulated settle time, which has 40% difference. Consequently compare to steering and index settle time, it is obtained 15 %.

ISBN: / DL:

GeneralA Individual Y-Pod	Individual Settle-TimeTs [s]	Each Y-Pod Steering Time [s]				Each Y-Pod Index Time [s]			
		0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
1st -Y-Pod	31.2	80	Stable	Stable	Stable	75	absolute min	-do-	-do-
2nd -Y-Pod	31.4	80	Stable	140	-do-	75	absolute min	-do-	-do-
3rd -Y-Pod	33.4	85	Stable	-do-	-do-	75	absolute min	-do-	-do-
4th –Y-Pod	34.2	100	Stable	-do-	-do-	80	absolute min	-do-	-do-
5th –Y-Pod	35.0	105	Stable	-do-	-do-	80	absolute min	-do-	-do-
6th –Y-Pod	36.4	110	Stable	-do-	-do-	85	absolute min	-do-	-do-

Table. 4.7: Individual Y-Pod Performance and analysis of steering and index based on Combined A.

In addition, Table. 4.7, show the individual Y-Pod performance, it gives us the individual Y-Pod settle time with steering and index data, some of the Y-Pods has different values ,due to its distance from the initial position vary from the poles for each Y-Pod. Steering of each Y-Pod can see in Table. 4.7. Overall difference in this case observed has 40% with the present simulated settle time with our theoretical settle time. In other words, steering and index receives 10 % variations. Therefore we conclude that the steering vs index has the same similarities, justify that it has good shape and also has steady state. With this evidence, we can overcome the synchronization problems. Even apply for large scales of swarm systems in scalable approaches with stability features.

4.4.5 Combined case-B

Combined case-B has same feature as combined –A, but the difference here is the movement of orientations. We would like to observe here itself, there exist any changes or not in the data in order to conclude the Y-Pods can move in any orientations with any circumstances. In this context, steering ω≠0 and constant speed. So we focus our discussion only on steering and index evolution, remaining components are neglected.

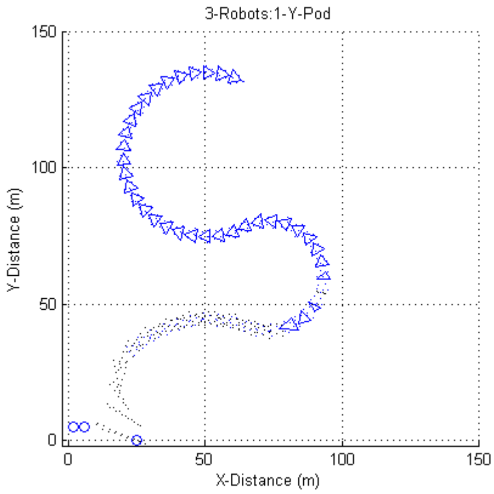
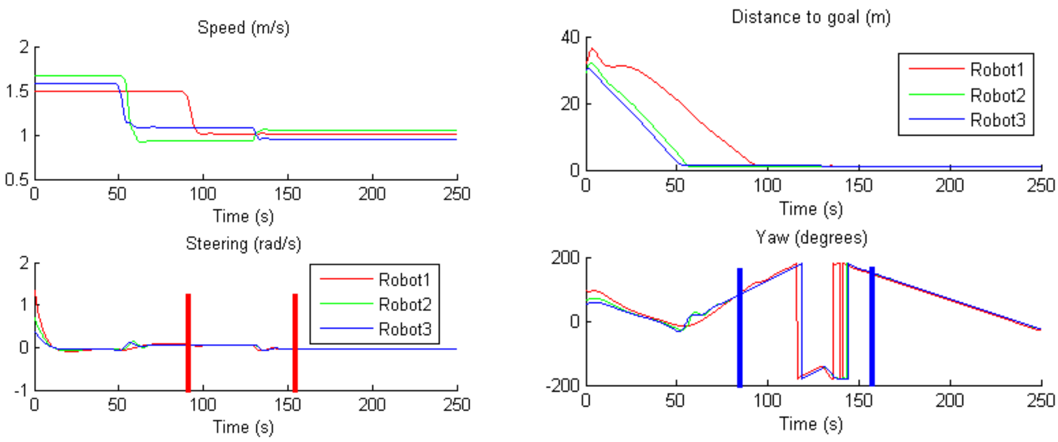


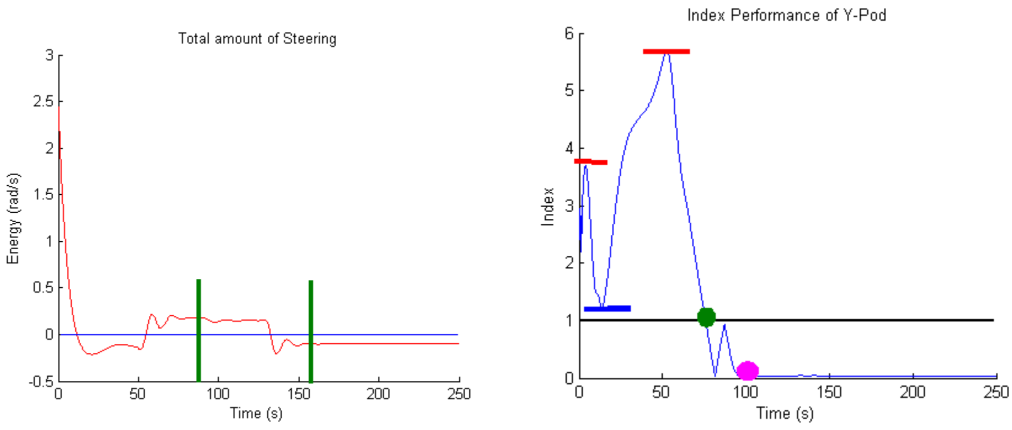
Fig. 4.50: three robots and single-Y-Pod is moving in a combined way.i.e. Constant speed and steering ω≠0.



(a): Speed and steering

(b): Distance to goal and Yaw angle

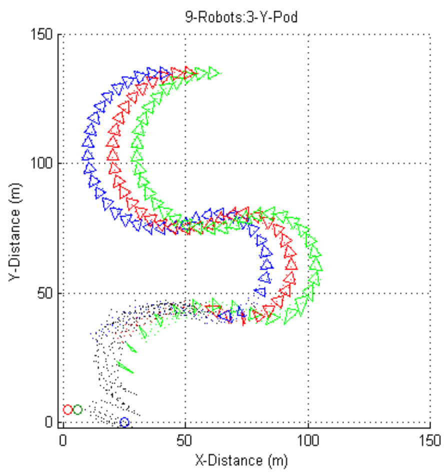
Fig. 4.51: 3 robots 1Y-Pod variables in terms of speed, steering, distance to goal and yaw angle θ.



A. Total amount of steering (energy) B. index performance between 0 to 1

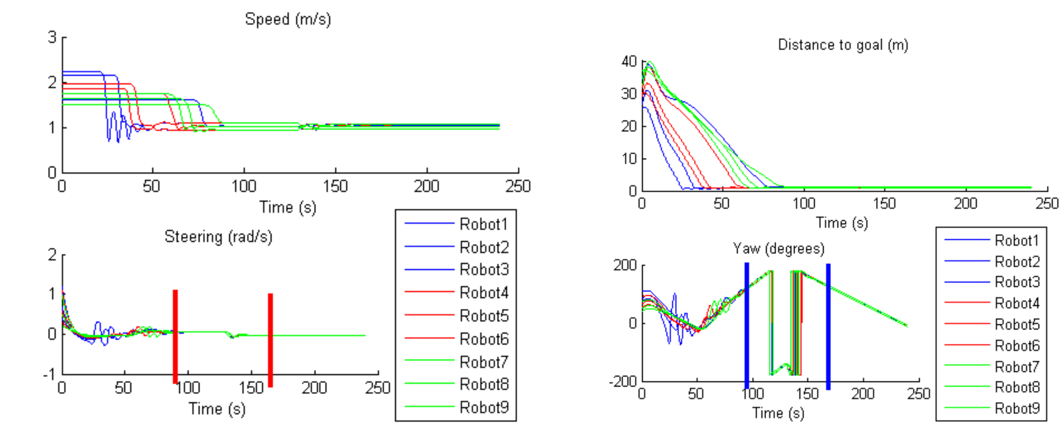
Fig.4.52: Total amount of steering and index performance in combined-B case.

In a Fig. 4.52 show the total amount of steering (A) and index performance (B). In this regard, from left-A total amount of steering is constant at 85 seconds, which gives the system settle time, it indicates with two green lines at 85 seconds and 165 seconds. On the other hand, in Fig. 4.52 right hand side -B indicates the shape evolution, it can see that after 75 seconds appears a good shape it indicate with green circle, continues the shape at 100 seconds it will be perfect shape with absolute minima which indicate the pink circle. Particular observation from the combined case-B is that, the steering analysis approximately same has combined case-A.



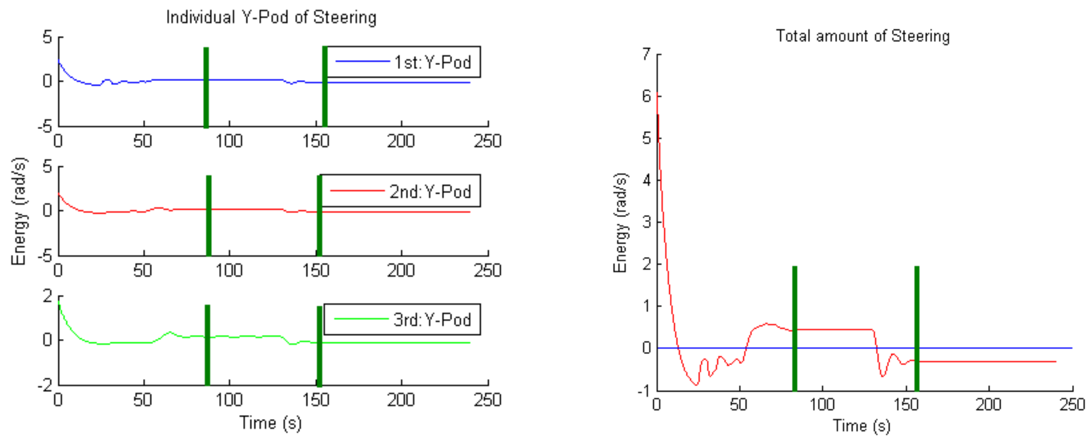
ISBN: / DL:

Fig.4.53: 3Y-Pod is moving in a combined way, constant speed and steering angle ω≠0.



(a): Speed and steering (b): Distance to goal and Yaw angle

Fig.4.54: 9 robots and 3-ypods and Virtual leader controller carrying y-pod formation in terms of speed, steering, distance to goal and yaw angle θ.

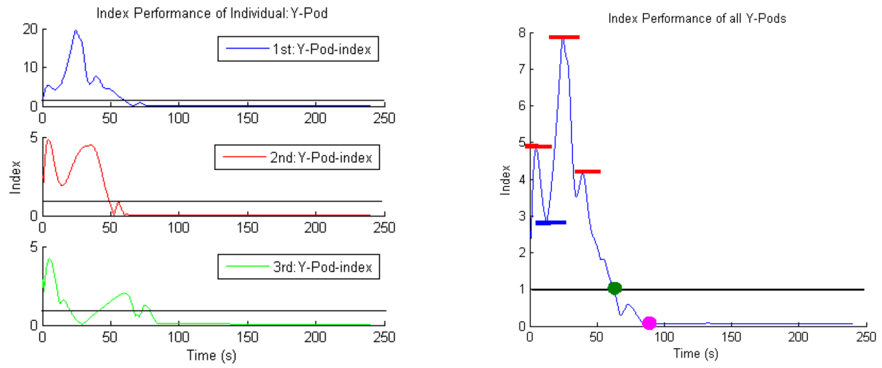


(A) Individual Y-Pod steering (B) Total amount of steering

Fig.4.55: Energy values according to the controller with combined state and the steering angles tends to zero with 9 robots and three Y-Pod. Its shows, total amount of steering control of robot in static, quadratic, linear cases and scalability with index.

The Fig. 4.55 show the individual Y-Pod steering (A) and total amount of steering (B). In this regard, individual Y-Pod steering has obtain settle time 80 seconds from right-A and total amount of steering is constant at 85 seconds left-B, which gives the system settle times.

Compare to combined case–A, the difference in settle time is more or less similar as combined case-B.



(A) Individual Y-Pod Index (B) Average Y-Pod Index

Fig.4.56:3Y-Pod and 9 robots with shape evolution graph in static, quadratic, linear cases.

From the Fig. 4.56 results of shape evolution graphs, individual Y-Pods (left-A) and Average Y-Pods (left-B). For instance, individual Y-Pod index (left-A) case, a good shape evolution received approximately over the time at 60, 50, 80 seconds corresponding 1st, 2nd and 3rd Y-Pods respectively. On the other hand, average Y-Pod index (left-B) case a good shape evolution received approximately over the time at 70 seconds. Particular observation in shape evolution is also same as combined case–A. With this evidence conclude that the system can move in all orientations, with well-regulated without synchronization problems.

4.4.5.1 Analysis of results

In this section illustrate the results obtained from the simulated data in combined-B approach, the results has similarities of combined-A case. The only difference is orientation direction in combined-B case. It yields, interesting results and explained based on simulated data augmented in Tables. 4.8, Table. 4.9. From the below Table 4.8 data examine that, constant steering is achieved at 85 seconds settle time for both single and 3 Y-Pods situation. The settle time achieved from steering and index performance has 85 and 75 seconds approximately. Difference with respect to our theoretical controller settle time it has some 40% difference. Consequently compare to steering and index settle time, is obtained 10 %.

ISBN: / DL:

Combined-CaseB Robots&Y-Pods	Settle-Time Ts [s]	Steering performance time interval [s]				Index performance time interval[s]			
		0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
3Robots:1Y-Pod	34.1	85	Stable	stable	stable	75	absolute min	73	-do-
6Robots:2Y-Pod	35.0	85	Stable	-do-	-do-	70	absolute min	73	-do-
9Robots:3Y-Pod	36.7	85	Stable	-do-	-do-	70	absolute min	74	-do-
12Robots:4Y-Pod	38.0	90	Stable	-do-	-do-	75	absolute min	74	-do-
15Robots:5Y-Pod	39.1	90	Stable	-do-	-do-	75	absolute min	78	-do-
18Robots:6Y-Pod	40.2	90	Stable	-do-	-do-	80	absolute min	78	-do-

Table.4.8: Steering and index performance with respect to no of robots and-Pods in Combined Case-B

CombineB Individual Y-Pod	Individual Settle-TimeTs [s]	Each Y-Pod Steering Time [s]				Each Y-Pod Index Time [s]			
		0-50	50-100	100-150	150-200	0-50	50-100	100-150	150-200
1st -Y-Pod	31.2	80	Stable	Stable	Stable	65	absolute min	-do-	-do-
2nd -Y-Pod	31.4	80	Stable	-do-	-do-	65	absolute min	-do-	-do-
3rd -Y-Pod	33.4	85	Stable	-do-	-do-	67	absolute min	-do-	-do-
4th –Y-Pod	34.2	85	Stable	-do-	-do-	80	absolute min	-do-	-do-
5th –Y-Pod	35.0	90	Stable	-do-	-do-	80	absolute min	-do-	-do-
6th –Y-Pod	36.4	90	stable	-do-	-do-	80	absolute min	-do-	-do-

Table.4.9: Individual Y-Pod settle-time measured based on steering and index Performance.

In Table. 4.9, the individual Y-Pod gives us the settle time, based on steering and index data. Overall difference in this case observed has 40% with theoretical settle time and consequently steering and index receives 10 % differences. Therefore we conclude that the steering vs index has the same similarities as combined case-A. In this regard, justify that system has good shape and also has steady state. With this evidence we can overcome synchronization problems. Even apply for large scales of swarm systems in scalable approaches with stability features. Furthermore we conclude that the combined case-A and combined case-B has same similarities in all aspects even changes the orientation direction.

ISBN: / DL:

Chapter 5

5. Y-Pod Formation with Obstacle Avoidance

In this chapter, our proposed controller is applied to Y-Pod swarm formation in order to avoid the obstacle in the simulation environment. Although extended the proposed techniques with Jacobian potential fields to avoid the obstacles for Y-Pod swarm formations. On the other hand, results and analysis are briefly explained.

5.1 Obstacle avoidance using proposed controller

In this chapter, we illustrate the obstacle avoidance problems based on the proposed controller from the section 4.2. In order to avoid obstacle, an extra term will be added to the proposed controller. An obstacle is modelled as a repulsive potential function based on the difference of vectors (Borenstein and Koren 1989) of obstacle and Y-Pod positions. Each Y-Pod will automatically try to avoid the obstacle due to controller steering command can change with different radius and it can generated by the added potential term. In this regard, the obstacle will basically be behaving like static with only a repulsive potential function (Erik and Kamesh 2011). An obstacle is modelled as follows:

$$\dot{d}^i_{obs} = -\sum_{j=1, j \# i}^{N} (d^i - d^j_{obs})[b_{obs} e^{-\frac{\left\| d^i - d^j_{obs} \right\|}{c_{obs}}}] \qquad (5.1)$$

Where the parameters b_{obs} and c_{obs} can be used to set the strength and region of influence of the obstacle. N is the no of robots to corresponding y-pods and i and j are robot and obstacle positions respectively. And $d^i - d_{obs}$ vector causing the Y-Pod to take evasive action, the obstacle function is very close to our propose controller and the difference is that the exponential term is now used as a repulsive part and there is no attraction.

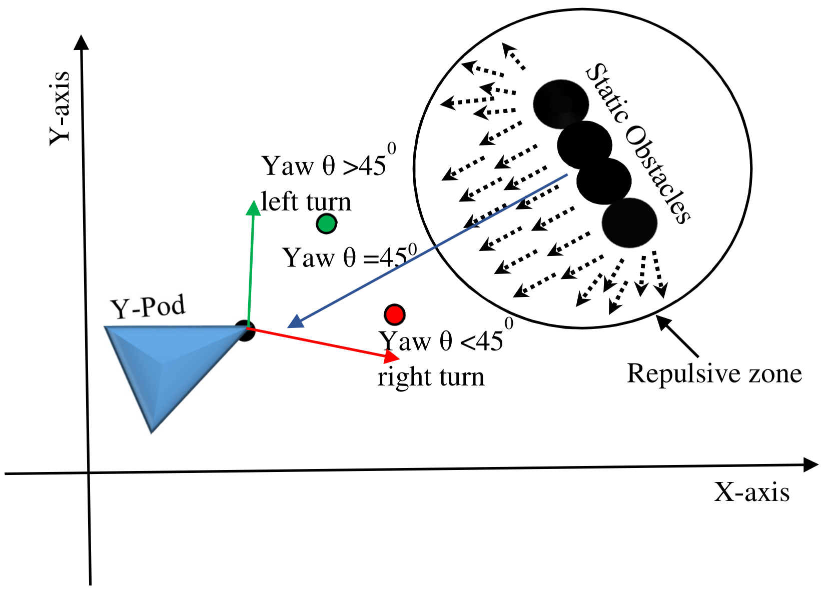


Fig.5.1: Schematically illustrated the static obstacles with repulsive forces.

However , when a Y-Pod encounters an obstacle on, or very close to its path, that obstacle will generate a large velocity in the opposite direction causing overshoot based on the approaches of equilibrium , pole frequencies and possible a local minima . For instance a simple situation with just two Y-Pods, placed so far apart, that they will attract each other. If there is an obstacle between the two Y-Pods, this obstacle will be exactly aligned with the two Y-Pod's paths. The obstacle avoidance (Khatib 1986) steering command will now generate a command with the size determined by the exponential part and the direction based on the difference vector of the Y-Pod and the obstacle. This will result in a vector trying to vary in radius to take turn with the help of steer of the Y-Pod. The net effect is that (in this case) over time both Y-Pods will come to a standstill when the attraction of each other matches the repulsion of the obstacle.

There is no steering command that generates lateral movement. If the obstacle is close to the path, but not on it, it will have a very small lateral component, insufficient to safely steer the Y-Pod around the obstacle with particular radius. To avoid this kind of behavior, an extra term $\dot{\boldsymbol{d}}^{i}{}_{\text{on}}$ is introduced. This term creates a steering command normal to vector $\boldsymbol{d}^{i} - \boldsymbol{d}_{obs}$, causing the Y-Pod to take evasive action. The strength of this command varies linearly with the distance to the obstacle. This ensures that the normal component starts acting before the Y-Pod gets dangerously close.

In the simulation a check is performed whether or not the Y-Pod is in close alignment with an obstacle. If not, there is no need to activate the extra term. This situation is explained in Fig. 5.2. Here, the green and red lines act like the attraction between the two robots, the blue, continuous lines indicate the repulsion force due to the obstacle (black dot) generated by the obstacle avoidance term. The dashed blue line indicates the steering command generated by the normal component. The black vectors are the sum vectors based on the three steering commands just discussed. In the simulation a check is performed whether or not the y-pod is in close alignment with an obstacle. If not, there is no need to activate the extra term. The net effect of the two commands will be a smoother avoidance of the obstacles. This extra avoidance term is implemented in order to overcome the obstacle and reach the desired destination point as follows:

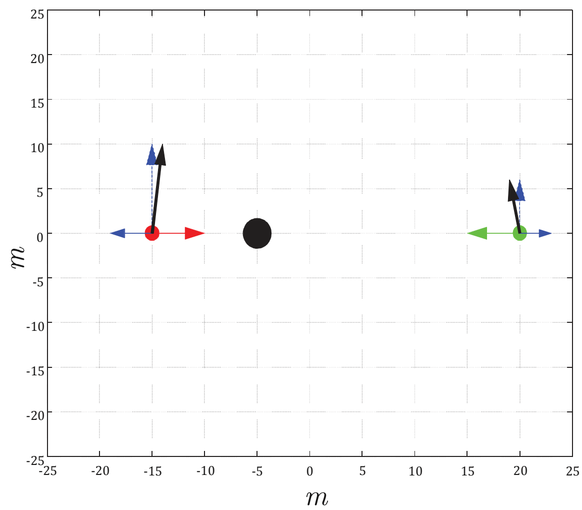


Fig.5.2: Effect of normal component on steering

ISBN: / DL:

- First, we check for all obstacles if they are within, or close to the path of the Y-Pod by calculating the inner-product ε:

$$\varepsilon = \frac{\vec{d}^i - \vec{d}^j_{obs}}{\|\vec{d}^i - \vec{d}^j_{obs}\|} \cdot \frac{\vec{d}^i - \vec{d}_r}{\|\vec{d}^i - \vec{d}_r\|}$$

- If, $\varepsilon < \varepsilon_d$ the obstacle is close enough that $\dot{d}^i_{on}$ needs to be activated. The user can set the sensitivity by varying the value of ε_d.

- If multiple obstacles are in close alignment with the Y-Pod path, the one closest to the Y-pod is selected.

On the other hand, we would like to test our controller with the Jacobean potential fields, because it has attraction and repulsion characteristics. In fact above controller serves as a repulsive potential function in order to avoid the static obstacles in the simulation arena. So we would like to compare both simulations with our proposed controller.

5.2 Simulation experiment with obstacle avoidance

The proposed techniques with two categories, repulsive potential functions and Jacobean potential fields, has been implemented in Matlab and performed the results to control the Y-Pod and to avoid the obstacles. The shape of the formation does not change during execution of the task. Thus the formation is kept only with interaction between robots with virtual leader (see Fig. 4.10). From the first category, robots has same characteristic which indicates in previous sections are posed in the simulation environment. For the second category, three robots are initially scattered in the environment and used controller to combines interaction between robots with attraction and repulsion of Jacobian forces are applied to execute the task. In both the categories, proposed controller act as a key role to navigate the robots in a linear way , when obstacles appears in the arena , potential functions are active in order to avoid the obstacle in above mentioned way in order to avoid the obstacle avoidance problems in the simulation arena. Using with the repulsive functions:

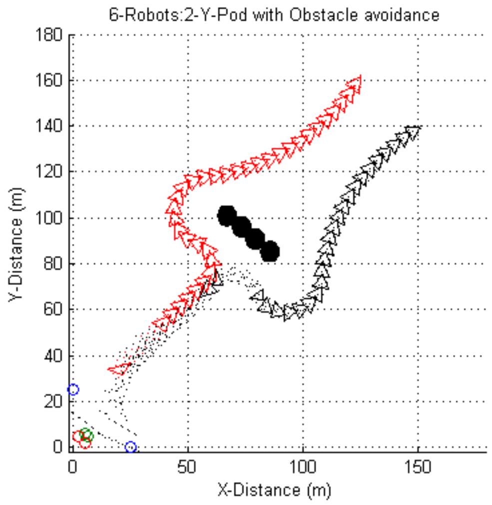
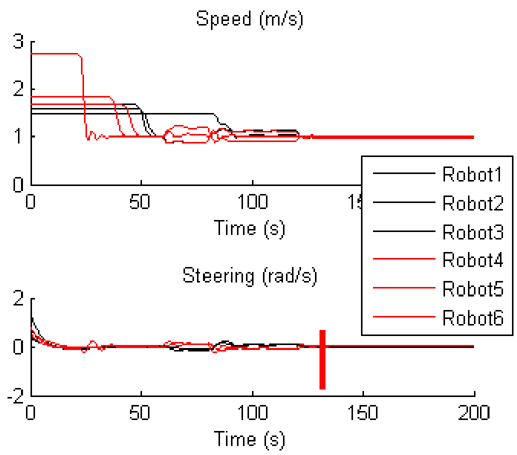
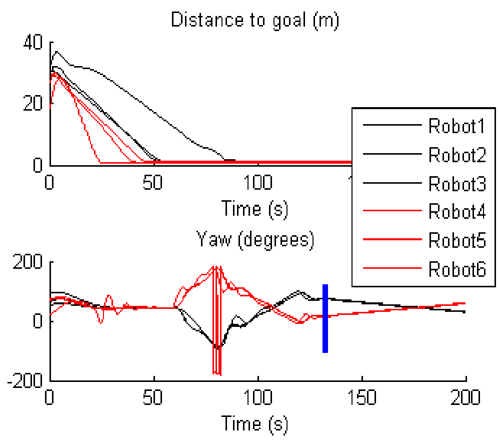


Fig.5.3: obstacle avoidance using repulsive potential functions and obstacles positioned as four circles in a rectangle and two y-pods indicate black and red are avoiding obstacles.



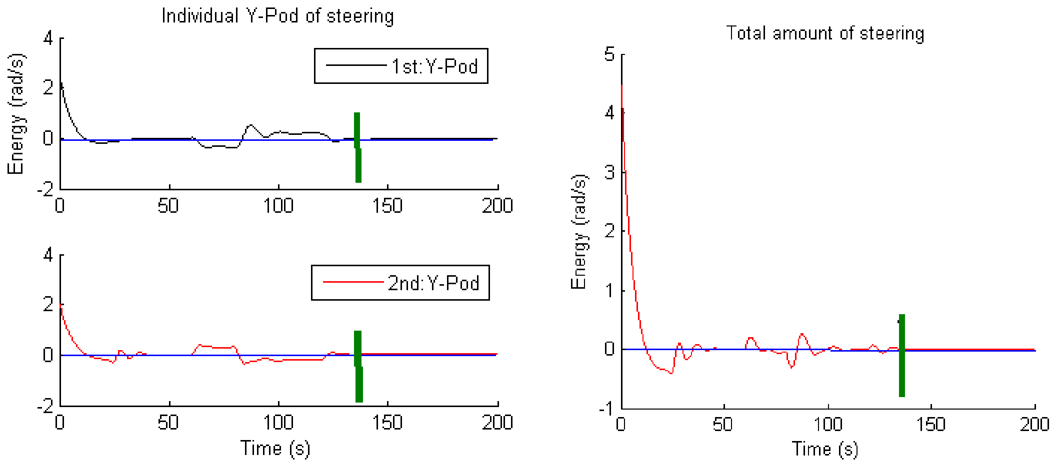
(a): Speed and steering



(b): Distance to goal and Yaw angle

Fig.5.4: Virtual leader controller carrying y-pod formation in terms of speed, steering, distance to goal and yaw angle θ and avoid the obstacles with repulsive potential functions

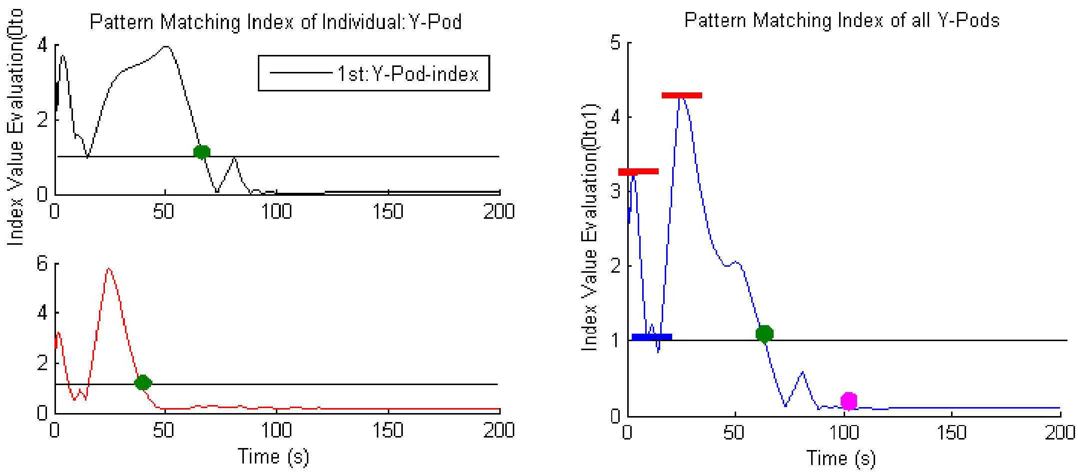
ISBN: / DL:



(A) individual Y-Pod steering

(B) Total amount of steering

Fig.5.5: Energy values according to the controller avoiding the obstacle using the repulsive potential functions and the steering angles tends to zero and shape with 6 robots and two Y-Pod. Its shows, total amount of steering control of robot in static, quadratic, linear cases and also involve obstacle avoidance criteria and scalability with index.



(A) Individual Y-Pod Index

(B) Average Y-Pod Index

Fig.5.6: shape evolution graph show with 6 robots and two Y-Pods. Its shows, total amount of steering control of robot in static, quadratic, linear cases and scalability with index. The controller avoiding the obstacle using the repulsive potential functions and the steering angles tends to zero and shape with 6 robots and two Y-Pod. Its shows, scalability with index.

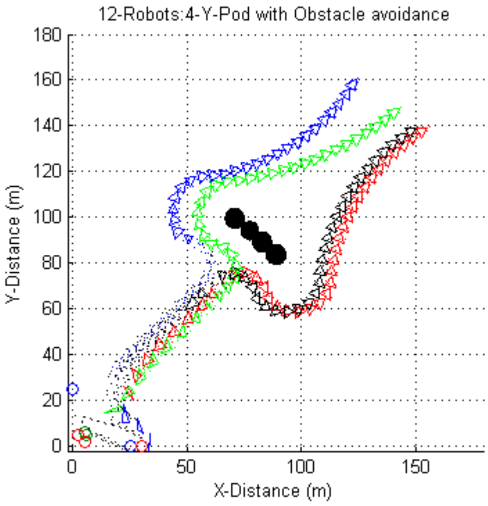
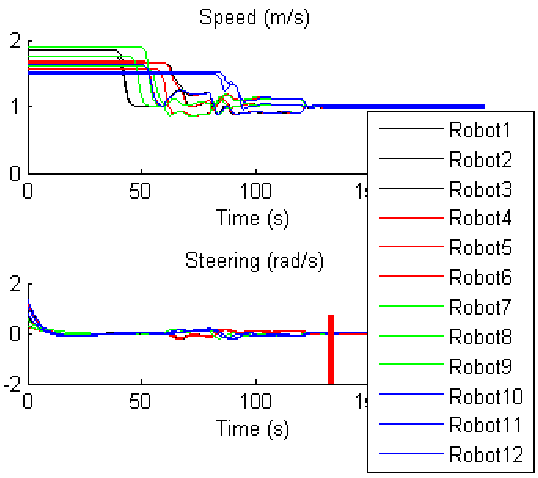
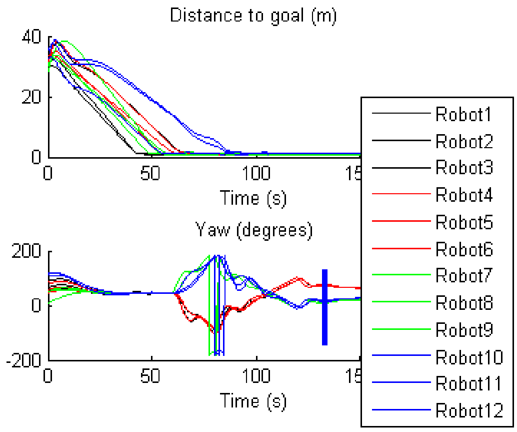


Fig.5.7:obstacle avoidance using repulsive potential functions and obstacles positioned as four circles in a rectangle and 12 robots 4 Y-Pods , it indicate black and red, green and blue are avoiding obstacles by using repulsive potential functions.

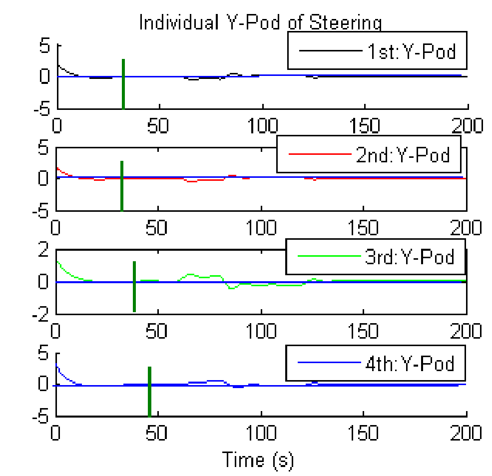


(a): Speed and steering

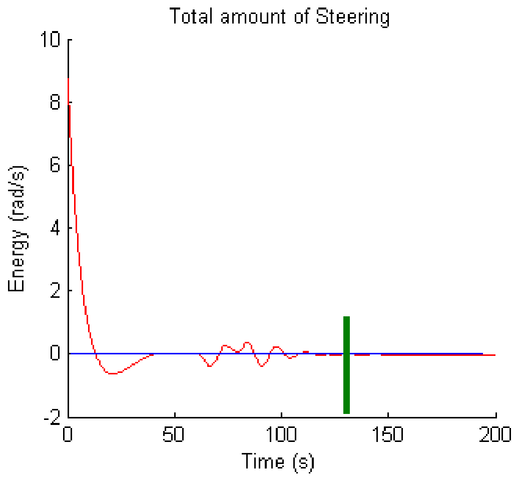


(b): Distance to goal and Yaw angle

Fig.5.8: 12 robots and 4 Y-Pods with Virtual leader controller carrying Y-Pod formation in terms of speed, steering, distance to goal and yaw angle θ and avoid the obstacles with repulsive potential functions.

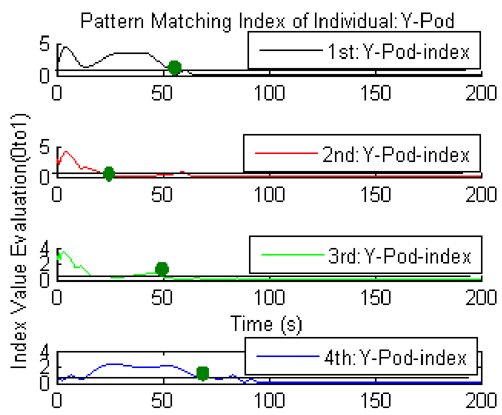


(A) individual Y-Pod steering

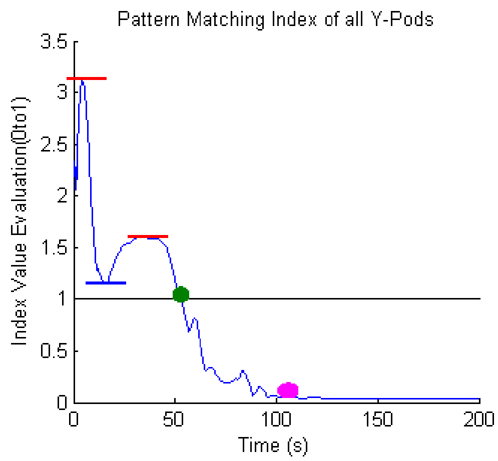


(B) Total amount of steering

Fig.5.9: Energy values according to the controller avoiding the obstacle using the repulsive potential functions and the steering angles tends to zero and shape with 12 robots and four Y-Pods. Its shows, total amount of steering control of robot in static, quadratic, linear cases and also involve obstacle avoidance criteria with stability.



(A) Individual Y-Pod Index



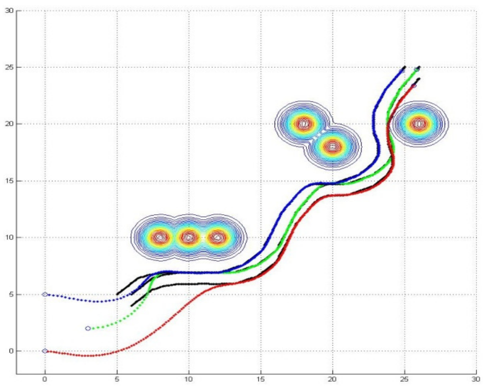
(B) Average Y-Pod Index

Fig.5.10: shape evolution graph show with 6 robots and two Y-Pods. Its shows, total amount of steering control of robot in static, quadratic, linear cases and scalability with index. The controller avoiding the obstacle using the repulsive potential functions and the steering angles tends to zero and shape with 12 robots and four Y-Pod. Its shows, scalability with index.

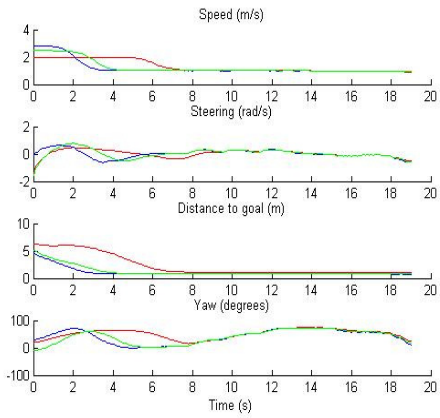
ISBN: / DL:

Using Jacobean potentials:

In this case. Three robots that drives between six obstacles of collision free path navigation to overcome the obstacles in the environment by using virtual controller and Jacobian force fields. Three robots will be assigned to positions [x, y, θ] ^{T}, x and y are initial positions of robots and θ is the orientation with respect to the virtual robot V_R (reference point) position. The virtual leader will move in constant speed with constant steering (ω≠0). This section presents that the robots are able to navigate and avoid the obstacles with free collision path in the environment to reach the desired destination point. In this scenario, six obstacles are placed in the environment at (7, 10), (10, 10), (12, 10), (17, 20), (20, 17) and (26, 20).



A. Obstacle avoidance trajectory



B.Speed,steering, distance, yaw angle

Fig.5.11:3 Robots and single Y-Pod paths and it consist of 6 obstacle with virtual robot following and avoiding obstacle using jocobian attraction and repulsive force fields with free-collision path.

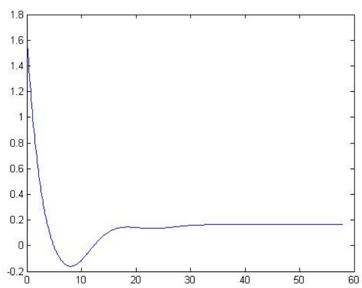


Fig.5.12: Total amount of steering according to the controller avoiding the obstacle with Jacobean potentials

ISBN: / DL:

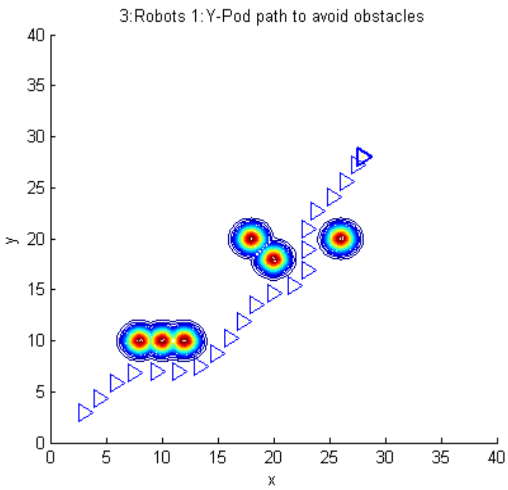


Fig.5.13: 3robots and single y-pod move with virtual leader in order to avoid the obstacle avoidance using the Jacobean potential fields with attraction and repulsive forces.

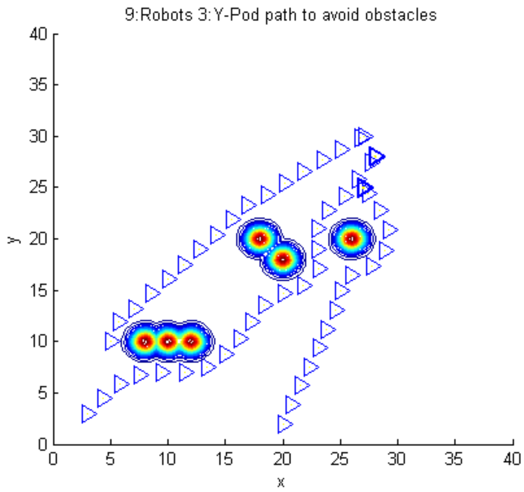


Fig.5.14: 9robots and three y-pod move with virtual leader in order to avoid the obstacle avoidance using the Jacobean potential fields with attraction and repulsive forces.

ISBN: / DL:

5.3 Analysis of results

In this section explains the analysis of results with respect to various parameters illustrated for each case. The results are performed with respect to all the cases with no of robots and no of Y-Pods and analysis show based and stability and scalability based on steering and pattern index matching.

Obstacle-Case Robots&Y-Pods	Settle-Time Ts [s]	Steering performance time interval[s]				Index performance time interval [s]			
		0-100	100-150	100-150	150-200	0-100	100-150	100-150	150-200
6Robots:2Y-Pod	35.0	140	Stable	Stable	Stable	65	Absolute	Absolute	Absolute
12Robots:4Y-Pod	38.1	140	Stable	Stable	Stable	60	Absolute	Absolute	Absolute
18Robots:6Y-Pod	40.6	140	Stable	Stable	Stable	65	Absolute	Absolute	Absolute

Table. 5.1: Steering and index performance in time interval [0 to 200] with respect to no of robots corresponding no of Y-Pods in obstacle avoidance problems

Obstacle Individual Y-Pod	Individual Settle-TimeTs [s]	Each Y-Pod Steering Time [s]				Each Y-Pod Index Time [s]			
		0-50	50-100	100-150	150-200	0-100	100-150	150-200	150-200
1st -Y-Pod	34.1	40	Stable	Stable	Stable	60	Absolute	Absolute	Absolute
2nd -Y-Pod	34.5	40	Stable	-do-	-do-	40	Absolute	-do-	-do-
3rd -Y-Pod	35.0	45	Stable	-do-	-do-	50	Absolute	-do-	-do-
4th –Y-Pod	35.4	48	Stable	-do-	-do-	65	Absolute	-do-	-do-
5th –Y-Pod	36.7	40	Stable	-do-	-do-	65	Absolute	-do-	-do-
6th –Y-Pod	38.4	45	Stable	-do-	-do-	70	Absolute	-do-	-do-

Table. 5.2: Individual Y-Pod Performance and analysis of steering and index based on obstacle avoidance approach. And the time is chosen at each interval [0 to 50].

ISBN: / DL:

Chapter 6

6. Conclusions and Future Works

In this chapter, the main conclusions arisen of the analysis and discussion of the results reported in this work are summarized. The chapter also reviews the dissertation's scientific contributions and then discusses directions for future research and application in certain topics in which the work of this thesis can continue. Finally, some concluding remarks are drawn.

6.1 Conclusions

In this thesis, formation control in swarm robotics has been analysed from the perspectives of scalability and stability by introducing a new implementation of a formation control methodology for a team of non-holonomic robots. In this sense, a formation is maintained by kinematic control based on linear control theory and dynamical system approaches. Each robot has local knowledge of the state of the formation, and moves with virtual leader technique: each virtual leader is in charge of carrying three goal positions matching a Y-pod shape. Hence, results of formation control with Y-Pod shapes are presented in different situations such as static, linear, quadratic and combined ones. Furthermore, some extended work is presented with static obstacles.

From the theoretical standpoint, the concepts introduced and discussed concern with formation control of swarm robots. There are a number of methods available for formation control. In the context of a swarm there are two main categories as earlier explained in the chapter 2: centralized and decentralized. Centralized control is a control method where each swarm member is controlled by a central controller. The members are fully dependent on the control inputs from the central controller. For a formation this means that each member

ISBN: / DL:

is guided along its own trajectory, provided by the central controller. In the other hand, in decentralized control the individual members have their own controller: this allows a member of a swarm to be autonomous. Instead of following a trajectory planned by a central controller, the members now react to each other's movements to stay out of harm's way and safely reach their goal.

The control strategy described in this thesis is a mix of centralized and decentralized control. This means that the control of the robot is spread between a central controller and a local controller. In this sense, single Y-Pod structure contains three goal positions with one virtual leader located at the centre of the Y-Pod. Each robot use a local controller based on the virtual leader movement which, in turn, has the responsibility to control all three robots: virtual leader can be treated as a centralized controller. But, in terms of many Y-Pods, two Y-Pods contain two virtual leaders which, in turn, have their own paths (trajectories) in autonomous way: this can be seen as a decentralized controller. Under this point of view, we argue that our approach is a mix of centralized and decentralized approaches: central controller will provide the virtual leader (Y-Pod) with a path and geometric requirements, while the distributed controllers would fulfil the formation by controlling safe distances and angles under settle time requirements.

The local controller we build use some previous work but we modified it in order to accomplish with the requirements of our system: the controller now is used to track the goal position while moving along some direction. The controller is based on two variables (angle and distance) relative to the line direction of the virtual leader. Since this controller is linear, corresponding parameters could be freely tuned but we can set them in order to match with settle time requirements. In such a way, we can design a controller that moves the robot to the goal position under time constraints. In addition, all the robots have same time requirements so that they move independently and synchronously to the Y-Pod goal positions.

Measuring the performance of the controller allows us to understand the behaviour of the system under time requirements. In such a way, we can use error variables (distance to goal, relative angle…) or control signals such us steering or speed. In this thesis we introduce also

ISBN: / DL:

the amount of steering as a way to evaluate such behaviour. Such signal comes from the amber force field techniques, where molecular chemists simulate organic molecules and evaluate the stability by the use of the amount of individual atoms energies. In our approach, such individual atoms energies are related to individual steering signals. That is the reason why we recall the amount of steering as energy.

Thinking on sensors network applications, we also need to fulfil geometric conditions. Then, we introduce a performance index to measure the pattern matching between the Y-pod and the actual positions of the robots. We don't measure the displacement but the difference between ideal and actual robot shapes. Such measure is centralized, and is a weighted index based on two different quantities: area and elongation. This index is a way to compute that the robots have good shape when less than one. A bigger value means bad shapes and, in network applications, results in bad connectivity situations.

As usual in control theory methodology, we tested the controller under three different canonical situations, that is, zero, first and second order for input variables. The meanings of these situations in our system are as follows: the goal position does not moves (zero order), the goal move through a line (first order) and the goal move through an arc (second order). In the simulation arena we simulated from 1 to 6 Y-Pods (3 to 18 robots) and made some assumptions. The most important relates to the speed of virtual leader and the initial conditions. The speed of the virtual leader is constant but no bigger that maximum speeds of individual robots, and the initial conditions parameters are used to set the control parameters: settle time is computed once the simulation starts taking into account the far the robot one, and nearest robots move slower.

In the first simulations some switching phenomena appeared and we solved by the way of command fusion technique. Since the switching appeared only on speed control signal, the controller works correctly but in real implementations this would be a big problem. The command fusion technique using the hyperbolic tangent works properly and solves the problem of switching. In fact, this command fusion applies only on speed control signal and steering control is not modified.

ISBN: / DL:

Simulation results show that the controller works properly so that all the robots move to the goal positions in a synchronous manner. Many simulations has been carried out so that we build many tables showing settle time measures based on energy and index performance plots. Tables show the results using 3 to 18 robots in static, linear, and quadratic situations. In addition more tables show the results in some combined situations such that the virtual leader changes from linear to quadratic and from quadratic to quadratic movements.

The results in the static and the linear cases show that the settle time measures in both steering and energy plots match with theoretical values. That is, the controller parameters are set to fulfil settle time requirements based on initial conditions, and the system evolves to the steady-state after some time. We measured this time at the energy plots and at the index of performance one: such measures show a constant shift of about 20%. We can argue that this shift is related to the fact that the controller in linear so that initial angles introduces some amount of time. In the same way, results in quadratic and combined cases show similar results with a time shift of about 45%. In a similar way, we can argue the same as in previous cases but here there is a new factor: the second order movement introduces some more difficulties so that the controller needs an extra time to reach the steady-state.

Chapter 4 shows a systematic procedure in order to evaluate the system performance in many different situations. We assume that some additional work could be done under more situations but the use of canonical signals allows us to conclude that further combinations will show similar results.

Chapter 5 shows some additional results with static obstacles where the virtual leader is in charge of avoiding the collision meanwhile carrying the Y-Pod. Such results are shown in order to simulate the system in a more realistic situation, but showing similar results as quadratic case, with a time shift of about 45%. Each virtual robot can sense the upcoming obstacles and generates a repulsive force so that finds a way to contour the obstacle. In this process, the Y-Pod turns and pulls all three goal positions so that the team-mates are avoiding the collision.

ISBN: / DL:

This set-up minimizes the information the central controller has to provide to the robots and it is a more safe and robust way of control: this feature improves the reliability of the system. All results under simulation show the effectiveness of the proposed methodology strategy. The desired formations are always created, regardless the original positions and angles of the robots, and the shape of the formation itself. Moreover, settle time requirements are also accomplished with some shift in time. Based on the results we conclude that the work presented in this thesis can be directly exported to a real robot, with just minor modifications. The main advantages of this new formation control methodology are the following:

- Stability is guaranteed based on a linear controller.
- The robots concerning with the same Y-Pod move synchronously.
- The steady-state is reached following initial time requirements. Results show an extra time shift that is introduced by the linear approximation.
- Unlike other artificial potential field strategies, there are no local minima because the controller fits the requirements of a harmonic function.
- The Y-Pod structure can provide a high degree of flexibility, since formations can be obtained with completely arbitrary shapes.

6.2 Future works

The concepts and the results presented in this dissertation pave the way for new applications and solutions to different formation and control problems. Some future research directions are summarized below. To extend the line of work presented in this thesis, there are still several ideas to be explored. The following section presents, the collision avoidance, interconnected dynamics for network communications, and 3D environments in order to obtain complex formation structures.

ISBN: / DL:

Collision avoidance problems:

In the future some work need to focus on collision avoidance because swarm formation can have huge amount of robots with a big degree of complexity. For instance, in our system the Y-pods can avoid collisions among then but the problem still remains among robots. In this regard, we need reactive solutions in order to avoid the collisions. In this scenario, we could consider the robots with some sensors but, then, both the linear controller and the parameters could not easily fulfil the time requirements. In fact, the state of the art does not still solve this because of the degree of complexity.

Interconnected dynamics for network communications:

Control problems such as millirobot control, distributed intelligence, swarm intelligence, congestion control in networks, collective motion in biology, oscillator synchronization in physics and game theory may be analysed under the theory of interconnected dynamic system (Antonelli 2013). By using interconnected dynamics, we can overcome the problems of damping and spring constant in lattices or other arbitrary formation. The swarm system can do various formation shapes: when connected to each other there exist a problem so, in order to overcome these type of problems, we can use the interconnected dynamics. This new approach will gives hand on research for our task to improve in thousands of robots to connect each other to perform perfect network communication. Then, we can overcome the redundancy issues, synchronization problems and also scalable issues.

Other formation frameworks

Additional theoretical studies on the formation framework can include, e.g., describing the formation with a dynamic graph, study its stability, managing dynamic link breaking/creating in accordance with some predefined objectives. We could include orientation potentials to regulate the relative orientation that robots should assume with respect to each other's, and the creation of other formation structures. Until now, every robot is connected to every other team-mate within a certain area of influence. Alternately, links can be programmed so that robots could be connected only to a certain subset of the

ISBN: / DL:

formation members for attraction and repulsion. Only repulsive forces would be considered for the other members, to avoid inter-robots collisions. There is an extensive area to explore here that depends on the application requirements.

3 Dimensional extension works

Future research activities will concern to 3D environments by extending the controller to 3 dimensional steering control variables. In such a way, we could extend the methodology to more complex swarm formations, thinking on underwater robots or flying ones. In fact, we start working in such direction and we found that only the floating-robot model works properly with such controller. Floating-robot kinematics model allows decomposing three steering control signals into two independent subsets: yaw and pitch steering in one hand, and roll steering in another. In addition, such model can be also used to quad-rotor flying robots. Next figures show some preliminary results where three robots (floating ones) reach the steady-state so that can follow the Y-pod. Following plot show three robots using a non-holonomic floating robot model (Aicardi et al. 2001) in a linear situation: Y-pod moves left-to-right and the floating-robot model.

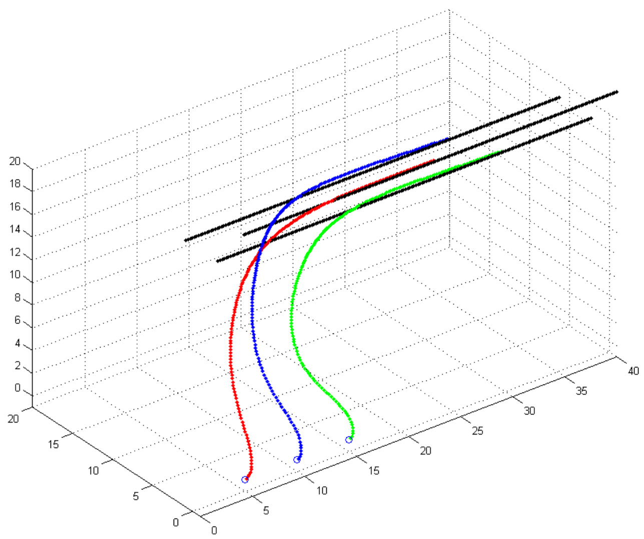


Fig.6.1: This figure shows the paths of the robots. Initially the robots are placed at the floor (y_plane=0) and are moving to the Y-pod position which is placed at some altitude and moves left-to-right

ISBN: / DL:

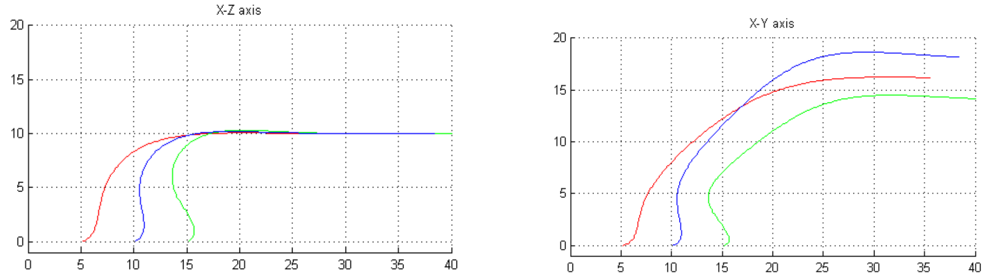


Fig.6.2: Next plots show XZ-plane and XY-plane paths: we can see the synchronization of the movements.

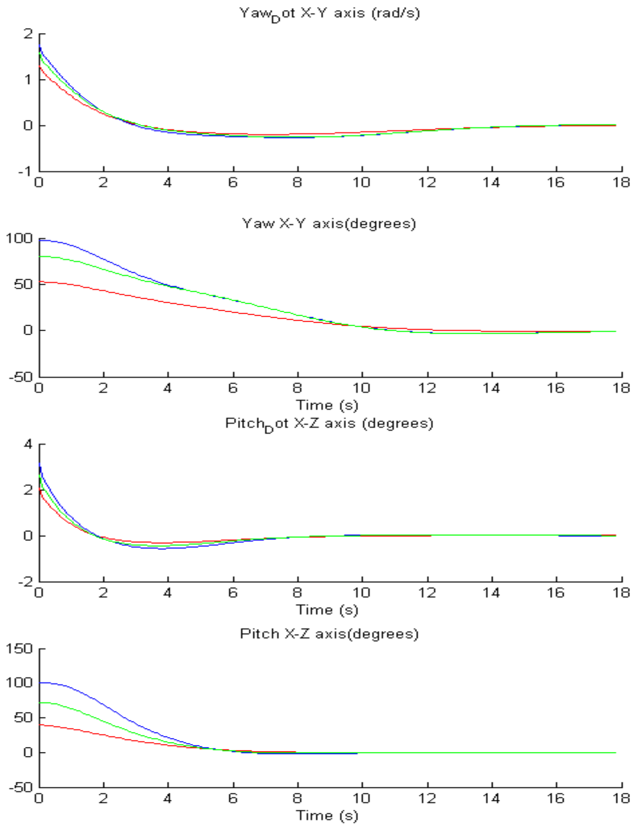


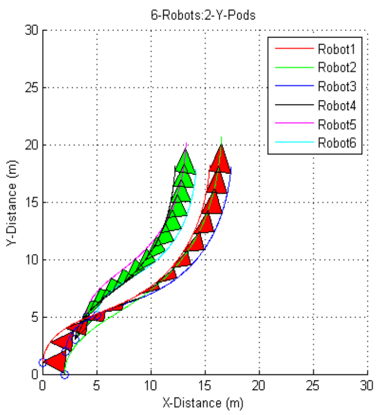
Fig.6.3: Shows angles (yaw and pitch), and the steering control signals (yaw_dot and pitch_dot variables) of the floating robots. In addition to (x, y, z) evolution in the previous plots, here we show how angles and steering control signals are also synchronize.

ISBN: / DL:

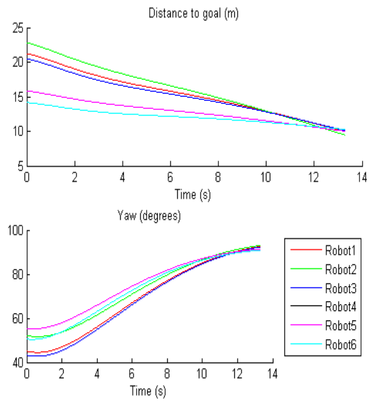
APPENDIX

This section show some additional results, those we mentioned in the tables such as table 1 to 12 increasing the no of robots 6, 12, 15 and 18 corresponding Y-Pods 2, 4, 5 and 6 of static, linear, quadratic, combined-A combined B and obstacle avoidance cases in chapter 4 in the results section. The main purpose of this section show the evidence of table values, swarm robotics system (i.e. working with huge amount of robots), and understand the analytical approach to predict the system behaviour in terms of speed, distance, yaw angle θ, energy (i.e. total amount of steering) and shape evolution of index. Here, we indicate robot is 'R' and corresponding Y-Pod is 'Y'.

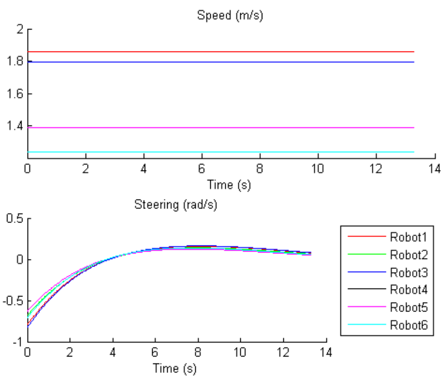
Static case: <u>R=6, Y=2:</u>



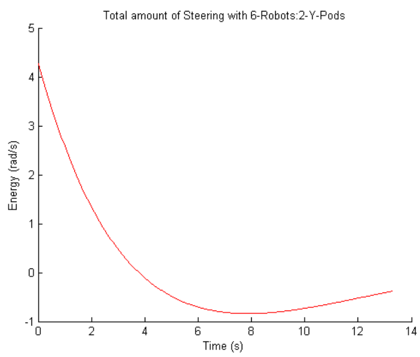
(i) 6-Robots and 2: Y-Pods



(ii) Distance to goal and Yaw angle θ

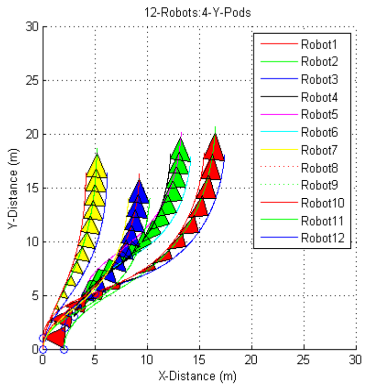


(iii) Speed and steering

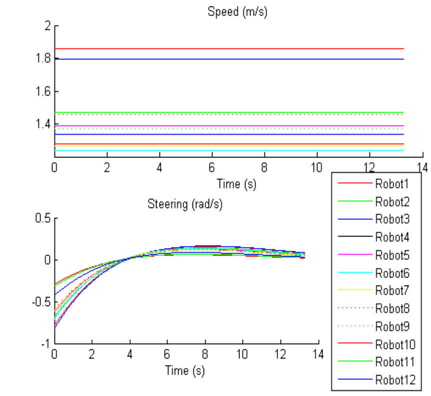


(iv) Amount of energy (Total steering)

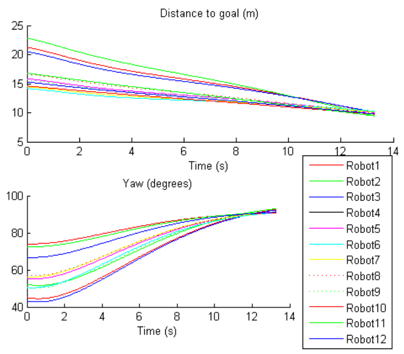
R=12, Y=4:



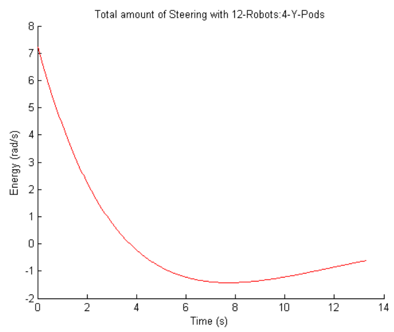
(v) 12-Robots and 4: Y-Pods



(vi) Speed and steering

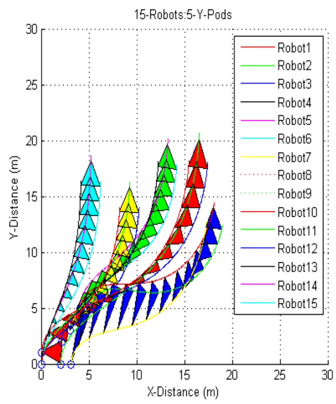


(vii) Distance to goal and Yaw angle θ

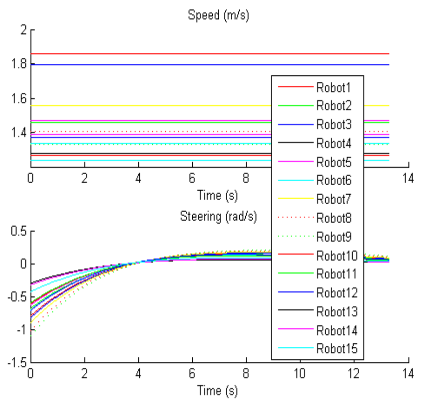


(viii) Amount of energy (Total steering)

R=15, Y=5:

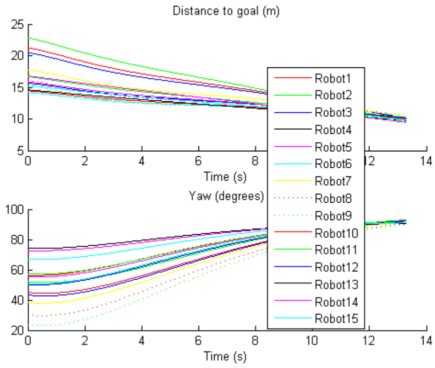


(ix) 15-Robots and 5: Y-Pods

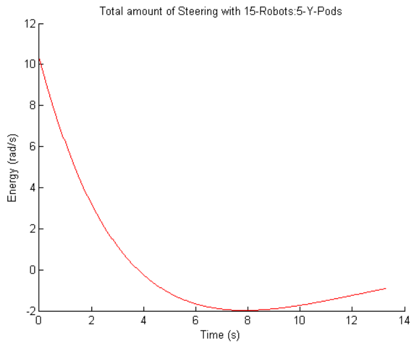


(x) Speed and steering

ISBN: / DL:

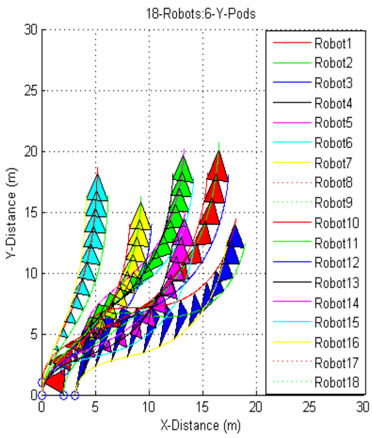


(xi) Distance to goal and Yaw angle θ

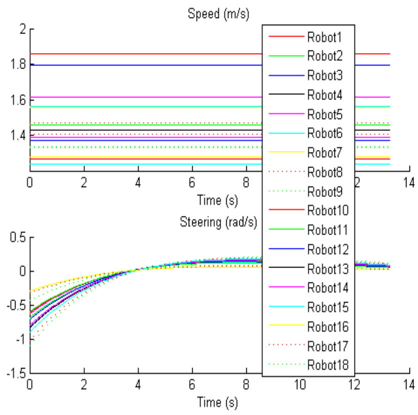


(xii) Amount of energy (Total steering)

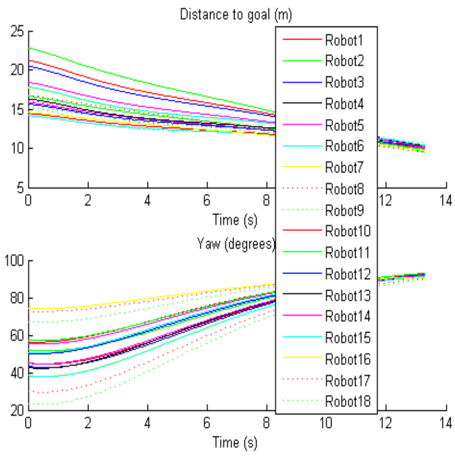
R=18, Y=6:



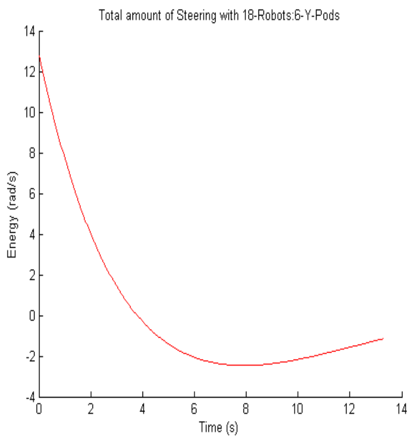
(xiii) 15-Robots and 5: Y-Pods



(xiii) Speed and steering



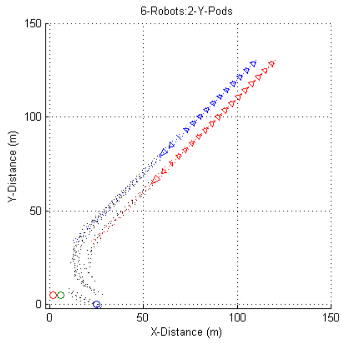
(xv) Distance to goal and Yaw angle θ



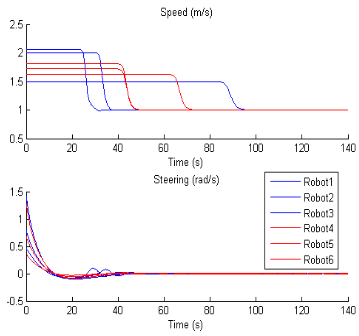
(xvi) amount of energy (Total steering)

UNIVERSITAT ROVIRA I VIRGILI
SWARM ROBOTIC SYSTEMS: Y-POD FORMATION WITH ANALYSIS ON SCALABILITY AND STABILITY
PURUSHOTHAM MUNIGANTI
ISBN: / DL:

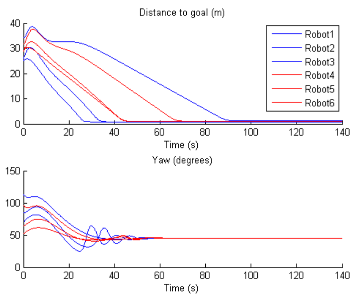
Linear case: R=6, Y=2:



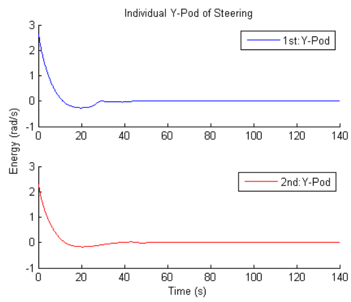
(i) 6-Robots and 2: Y-Pods



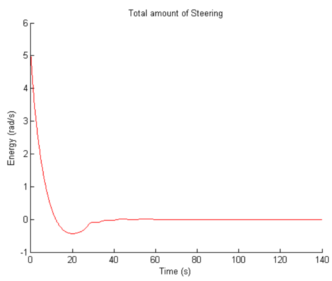
(ii) Distance to goal and Yaw angle θ



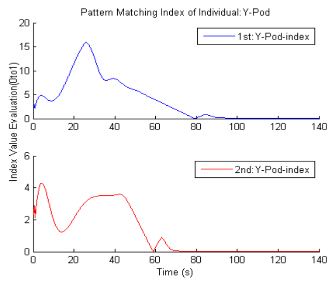
(iii) Speed and steering



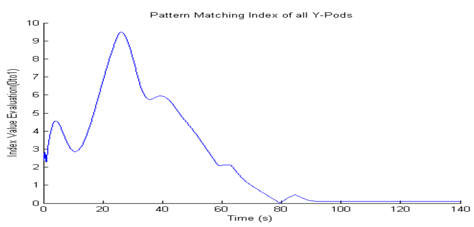
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

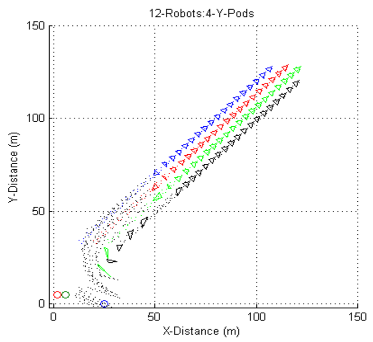


(vi) Index of individual Y-Pod

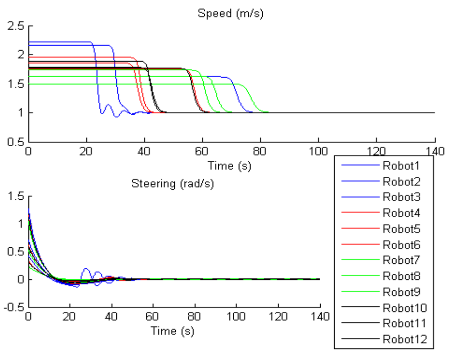


(vii) Total index of all Y-Pods

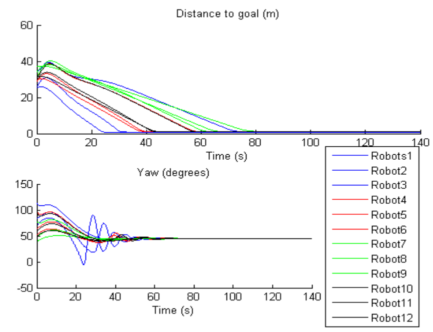
R=12, Y=4:



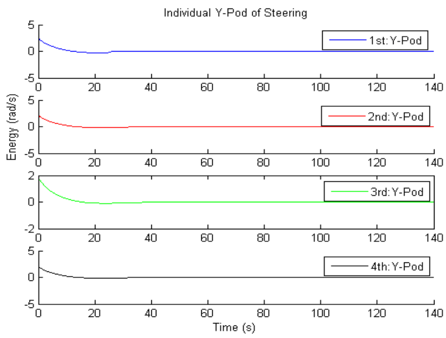
(i) 12-Robots and 4: Y-Pods



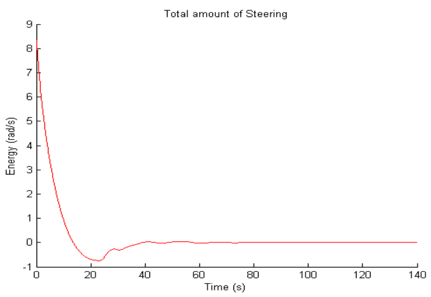
(ii) Speed and steering



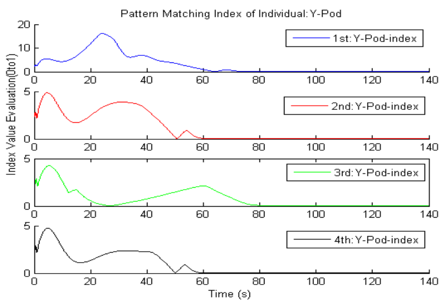
(iii) Distance to goal and Yaw angle θ



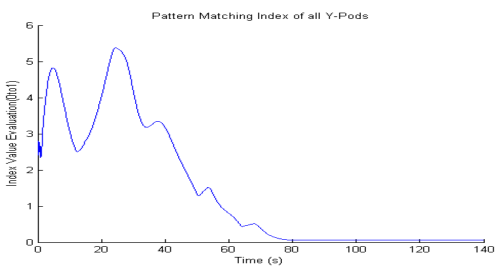
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

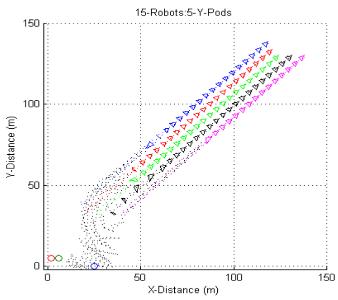


(vi) Index of individual Y-Pod

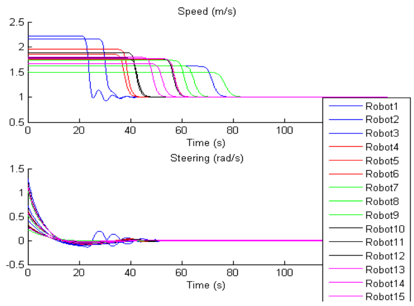


(vii) Total index of all Y-Pods

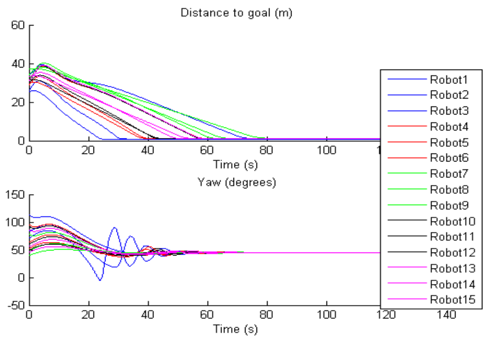
R=15, Y=5:



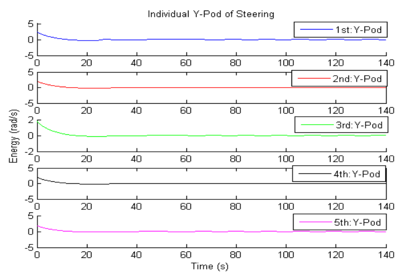
(i) 12-Robots and 4: Y-Pods



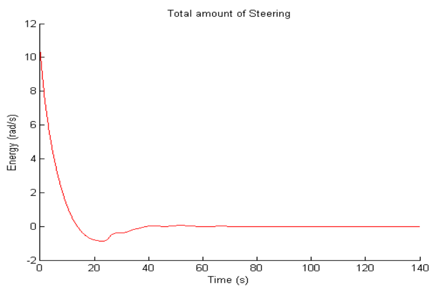
(ii) Speed and steering



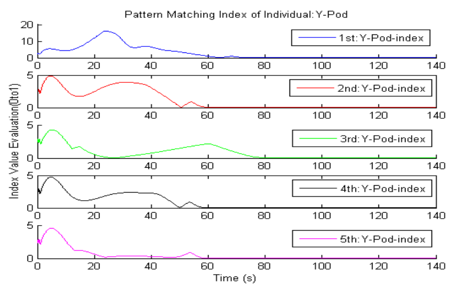
(iii) Distance to goal and Yaw angle θ



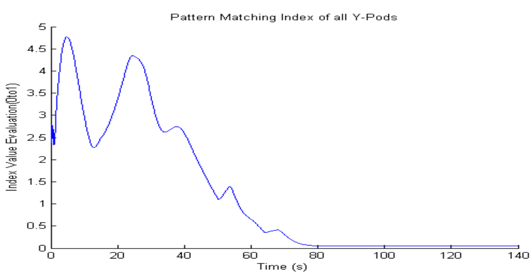
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

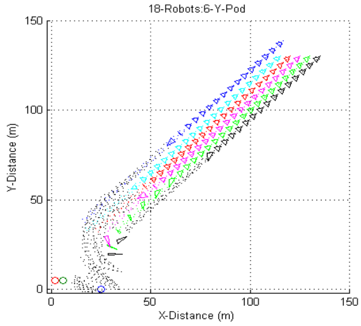


(vi) Index of individual Y-Pod

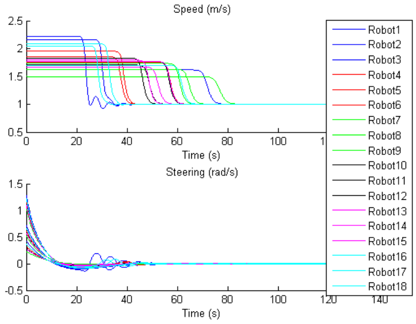


(vii) Total index of all Y-Pods

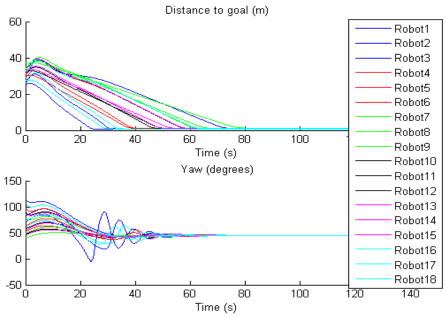
<u>R=18, Y=6:</u>



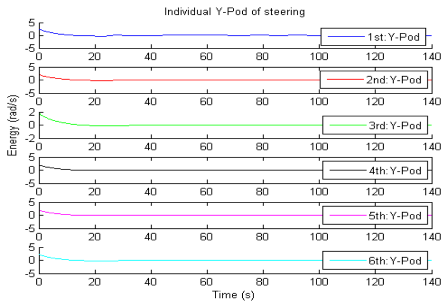
(i) 18-Robots and 6: Y-Pods



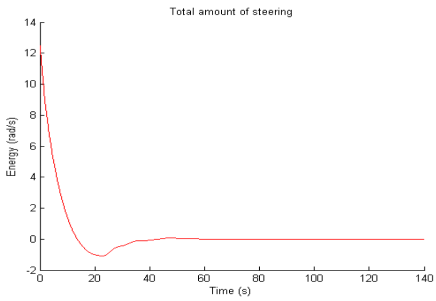
(ii) Speed and steering



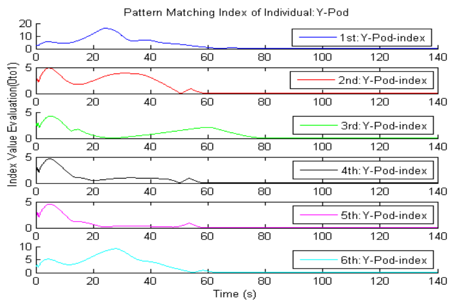
(iii) Distance to goal and Yaw angle θ



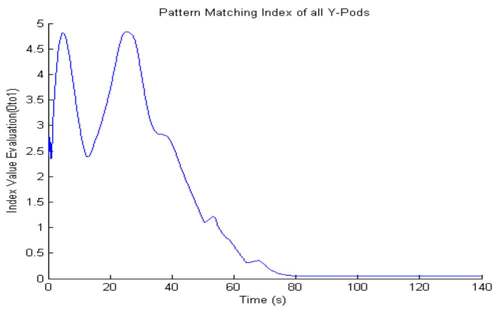
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

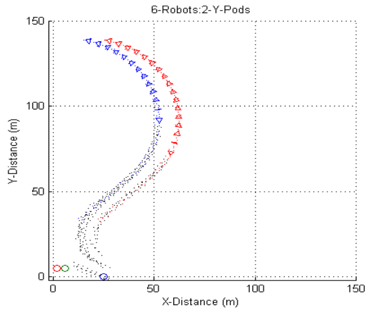


(vi) Index of individual Y-Pod

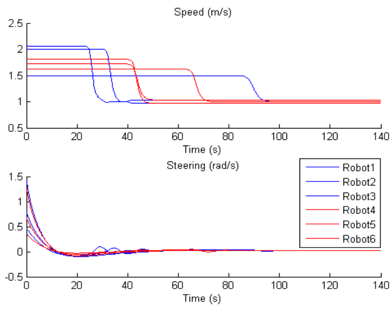


(vii) Total index of all Y-Pods

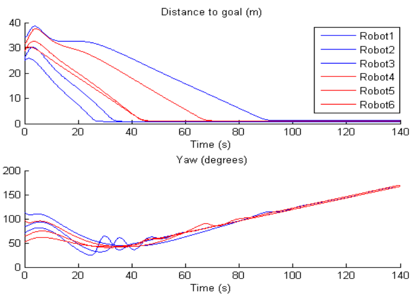
Quadratic case: R=6, Y=2:



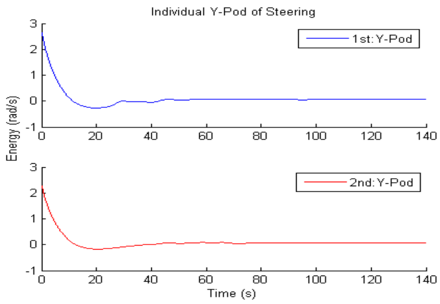
(i) 6-Robots and 2: Y-Pods



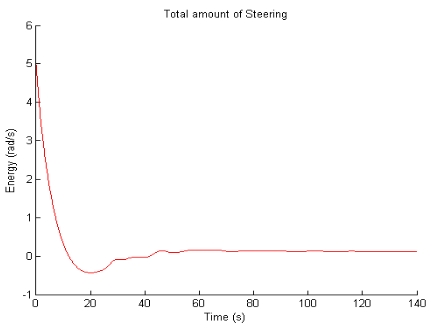
(ii) Speed and steering



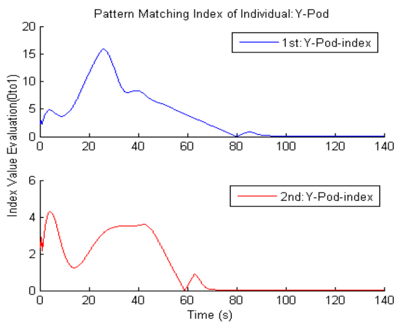
(iii) Distance to goal and Yaw angle θ



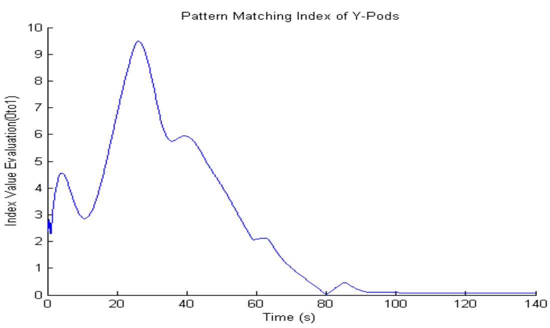
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

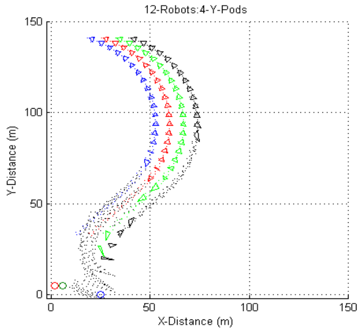


(vi) Index of individual Y-Pod

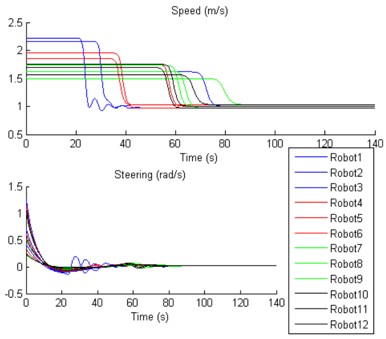


(vii) Total index of all Y-Pods

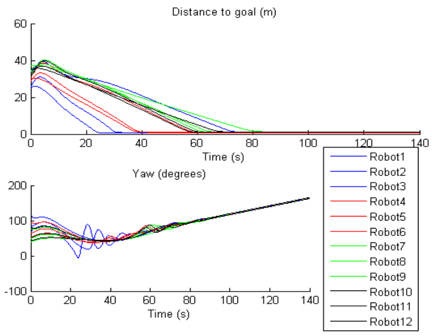
R=12, Y=4:



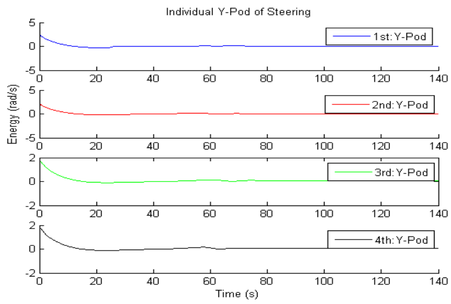
(i) 12-Robots and 4: Y-Pods



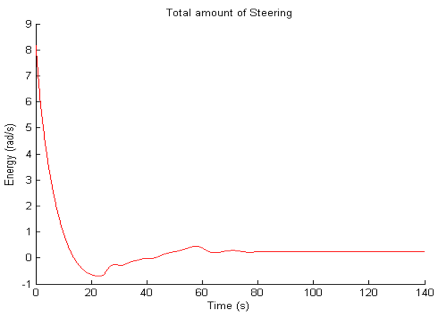
(ii) Speed and steering



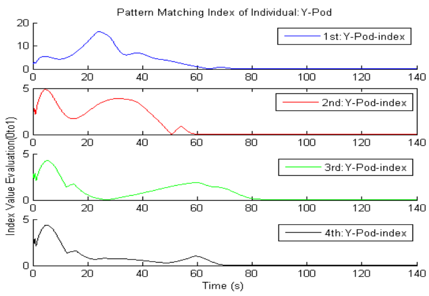
(iii) Distance to goal and Yaw angle θ



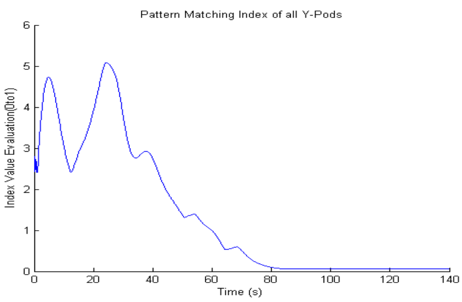
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)



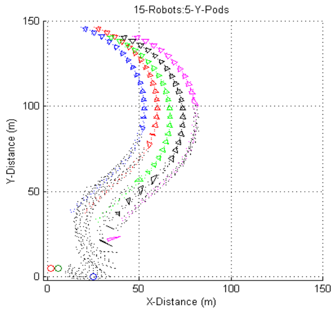
(vi) Index of individual Y-Pod



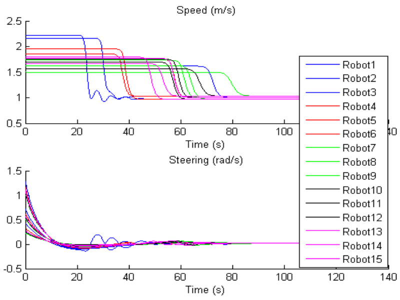
(vii) Total index of all Y-Pods

ISBN: / DL:

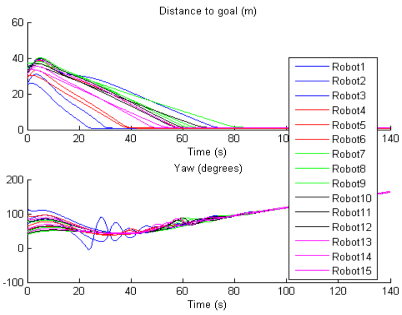
<u>R=15, Y=4:</u>



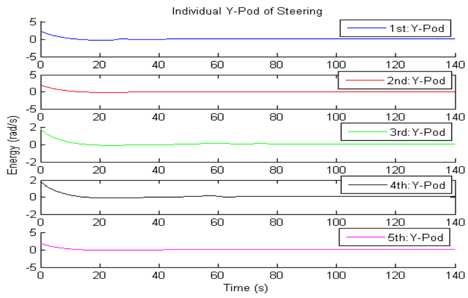
(i) 15-Robots and 5: Y-Pods



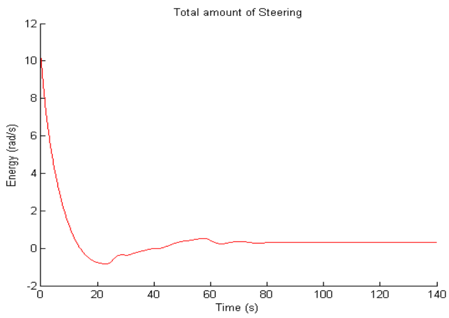
(ii) Speed and steering



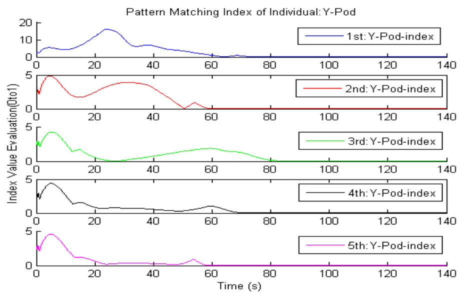
(iii) Distance to goal and Yaw angle θ



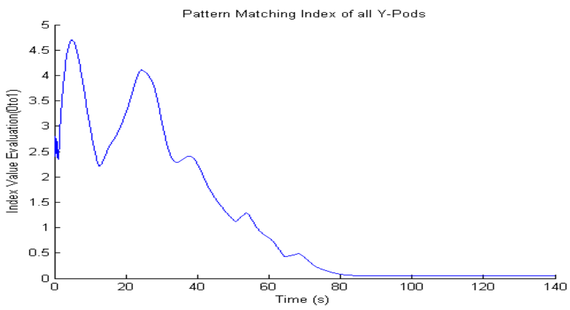
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

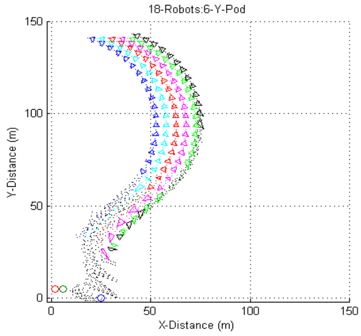


(vi) Index of individual Y-Pod

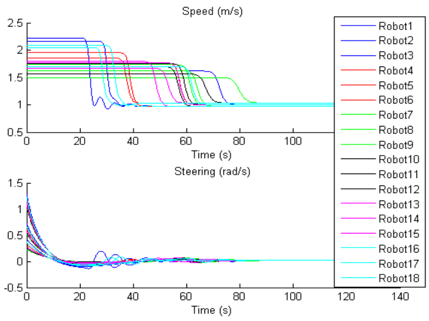


(vii) Total index of all Y-Pods

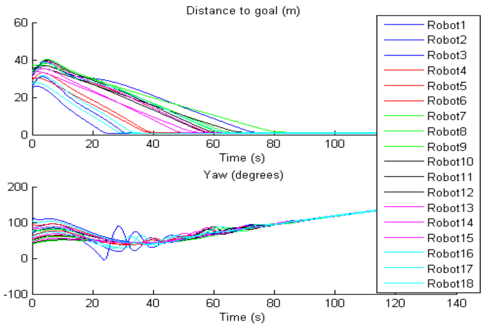
R=18, Y=6:



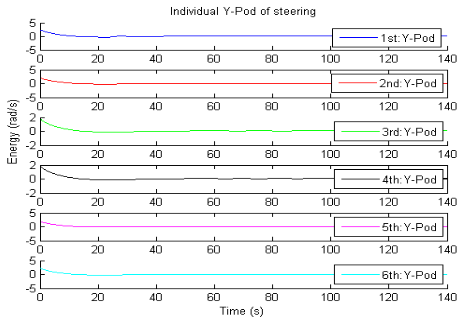
(i) 18-Robots and 6: Y-Pods



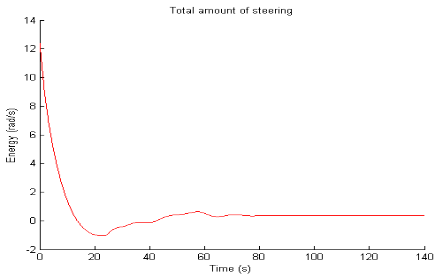
(ii) Speed and steering



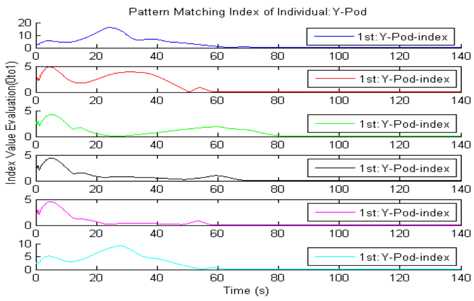
(iii) Distance to goal and Yaw angle θ



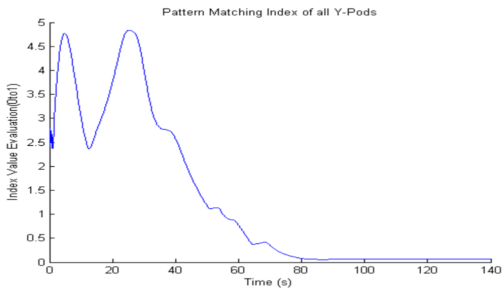
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)



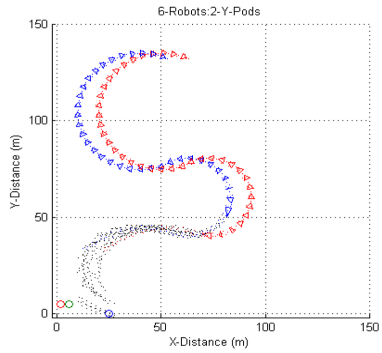
(vi) Index of individual Y-Pod



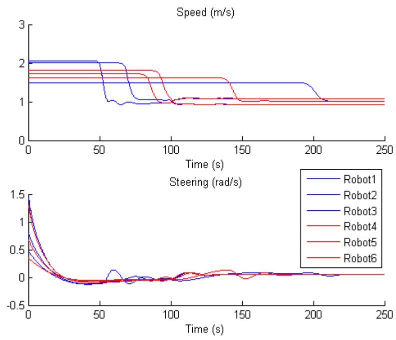
(vii) Total index of all Y-Pods

ISBN: / DL:

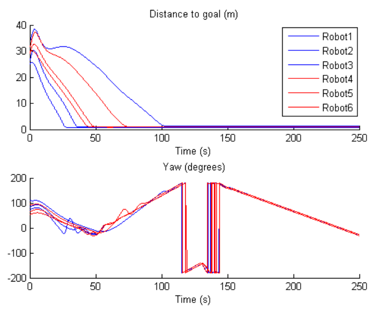
Combined-A case: R=6, Y=2:



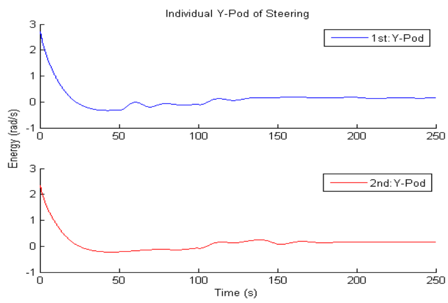
(i) 6-Robots and 2: Y-Pods



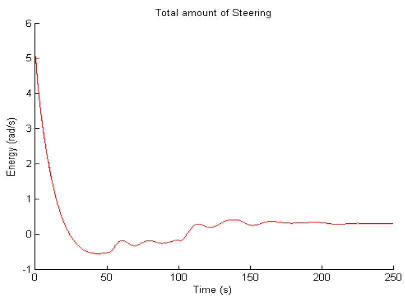
(ii) Speed and steering



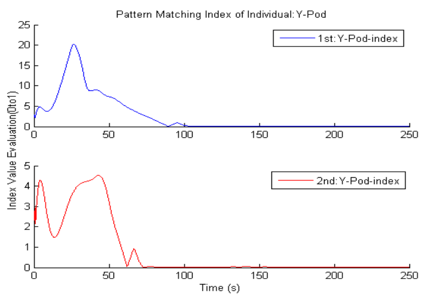
(iii) Distance to goal and Yaw angle θ



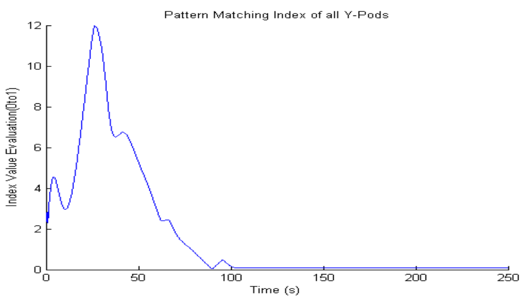
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

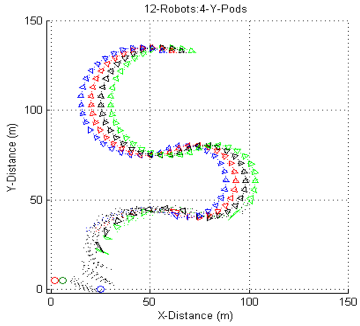


(vi) Index of individual Y-Pod

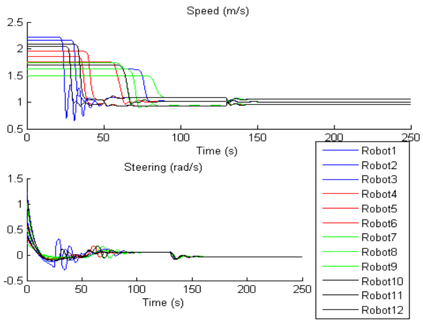


(vii) Total index of all Y-Pods

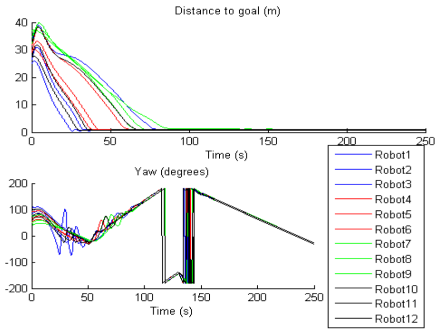
R=12, Y=4:



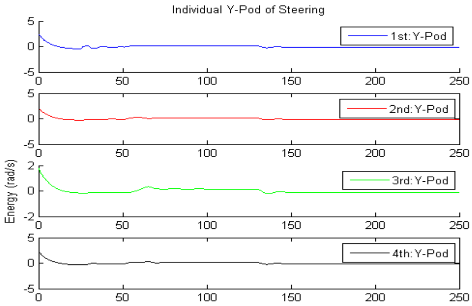
(i) 12-Robots and 4: Y-Pods



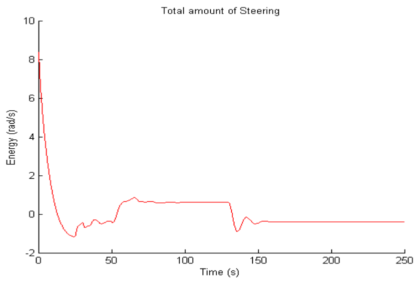
(ii) Speed and steering



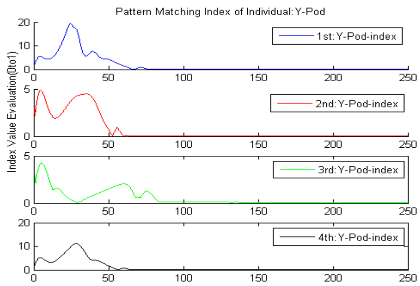
(iii) Distance to goal and Yaw angle θ



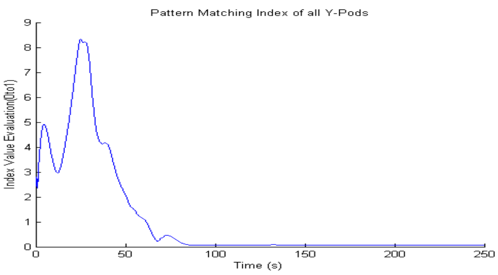
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

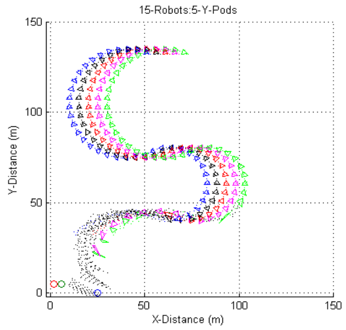


(vi) Index of individual Y-Pod

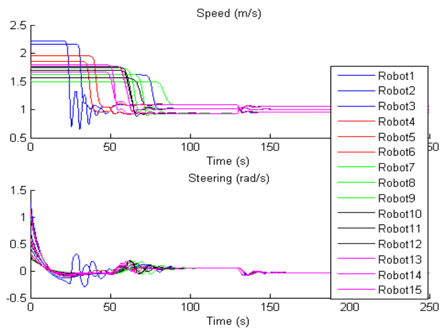


(vii) Total index of all Y-Pods

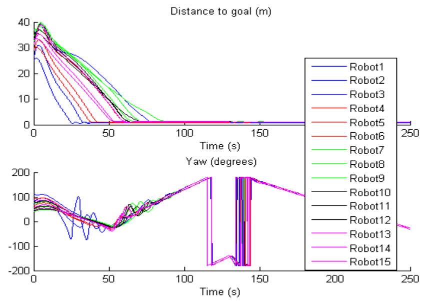
R=15, Y=5:



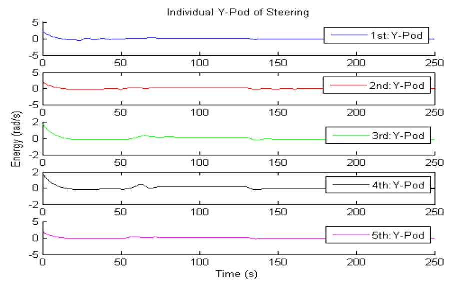
(i) 15-Robots and 5: Y-Pods



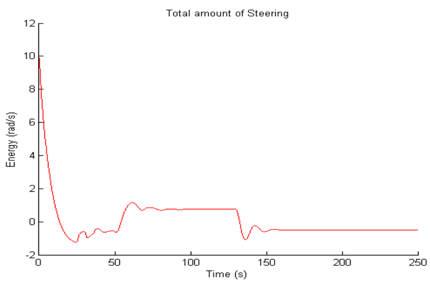
(ii) Speed and steering



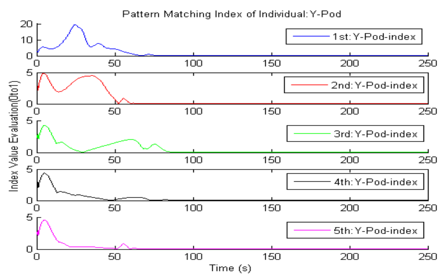
(iii) Distance to goal and Yaw angle θ



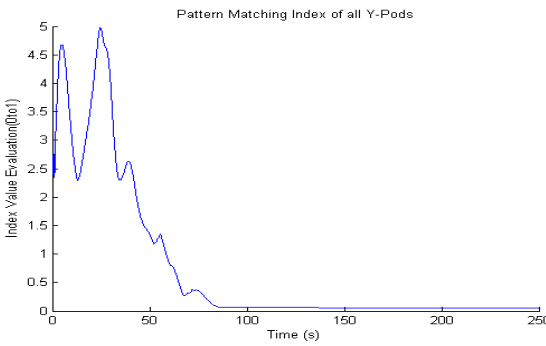
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

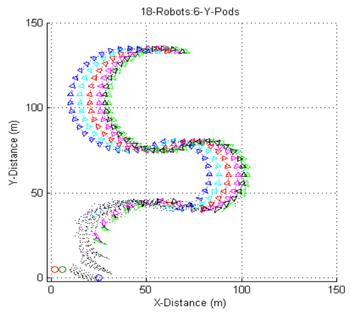


(vi) Index of individual Y-Pod

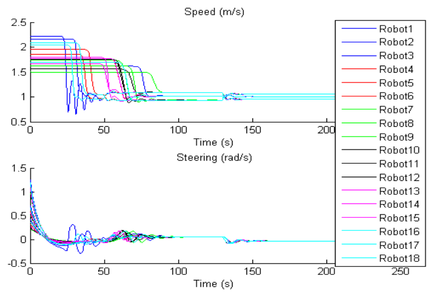


(vii) Total index of all Y-Pods

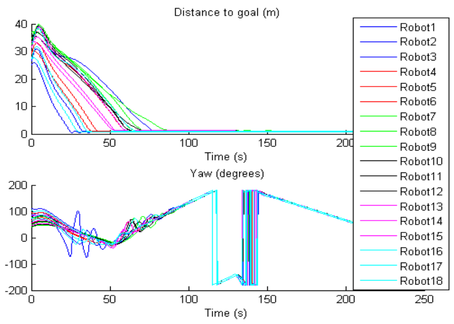
R=18, Y=6:



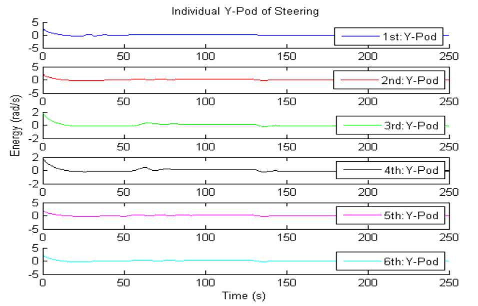
(i) 18-Robots and 6: Y-Pods



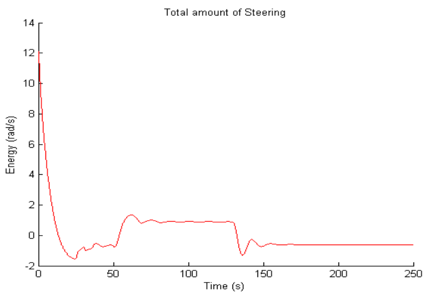
(ii) Speed and steering



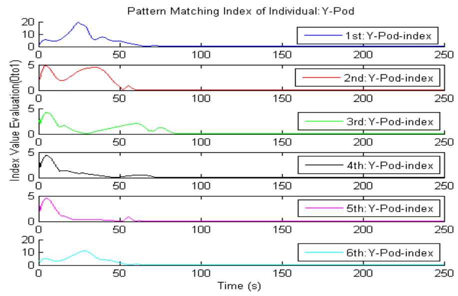
(iii) Distance to goal and Yaw angle θ



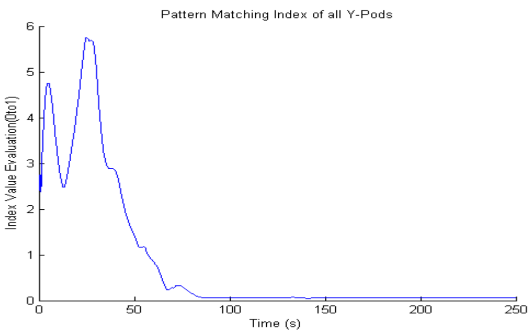
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)



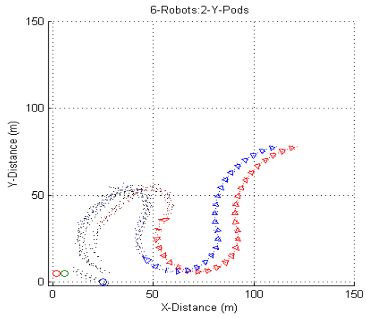
(vi) Index of individual Y-Pod



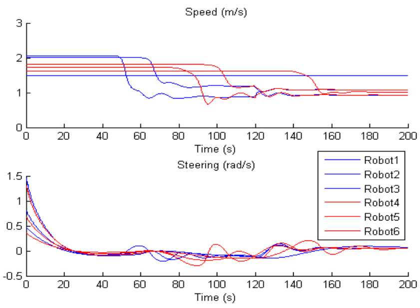
(vii) Total index of all Y-Pods

ISBN: / DL:

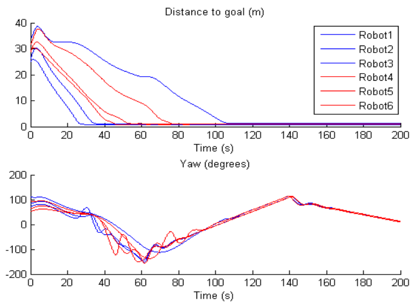
Combined-B case: R=6, Y=2:



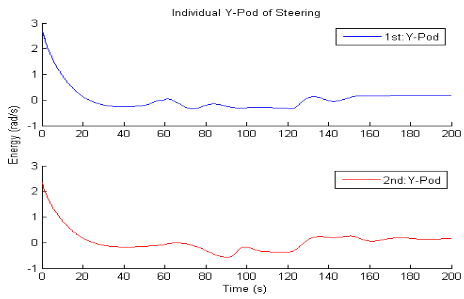
(i) 6-Robots and 2: Y-Pods



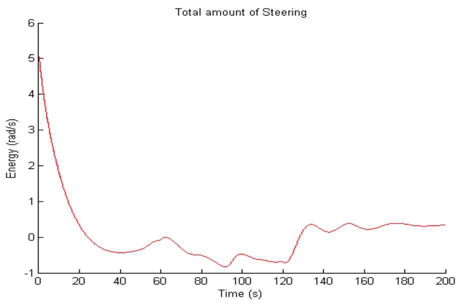
(ii) Speed and steering



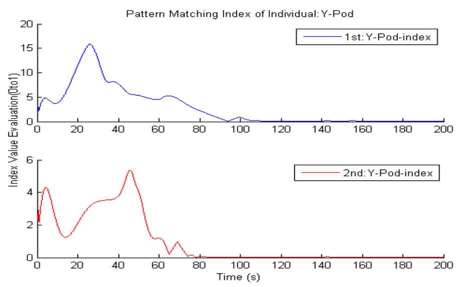
(iii) Distance to goal and Yaw angle θ



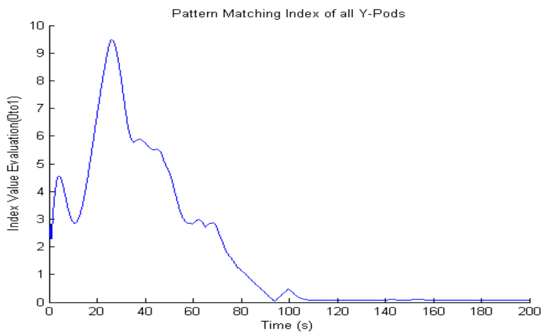
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

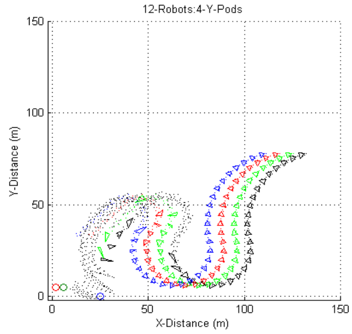


(vi) Index of individual Y-Pod

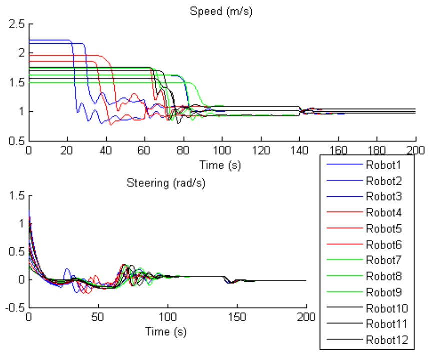


(vii) Total index of all Y-Pods

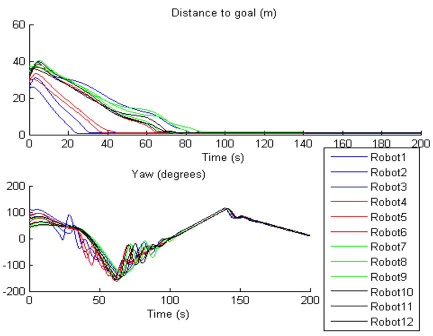
R=12, Y=4:



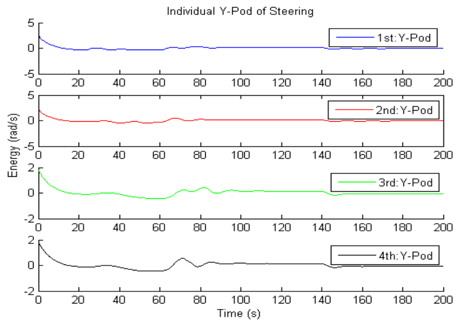
(i) 12-Robots and 4: Y-Pods



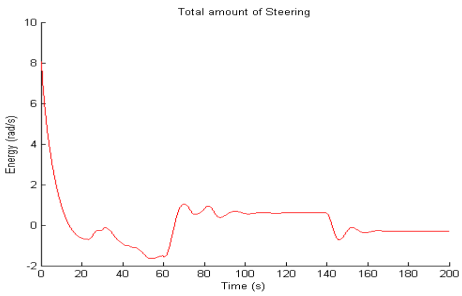
(ii) Speed and steering



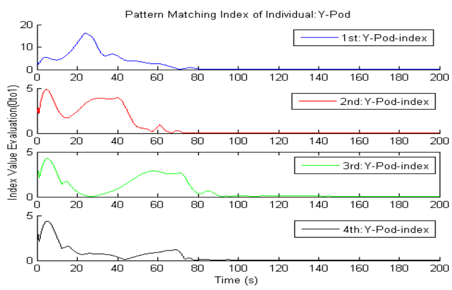
(iii) Distance to goal and Yaw angle θ



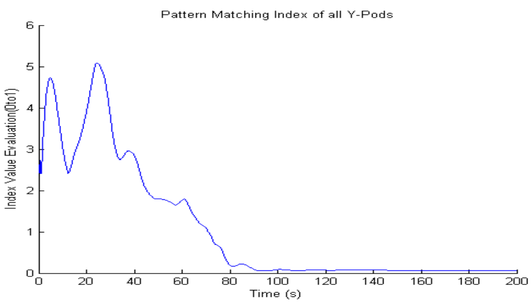
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)



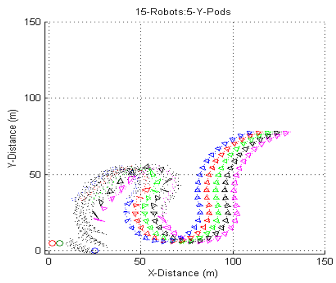
(vi) Index of individual Y-Pod



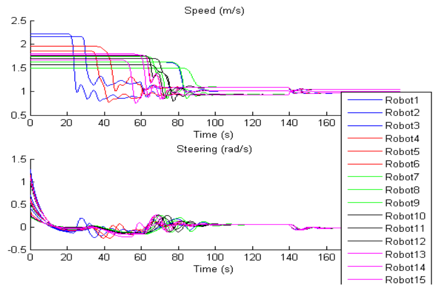
(vii) Total index of all Y-Pods

APPENDIX

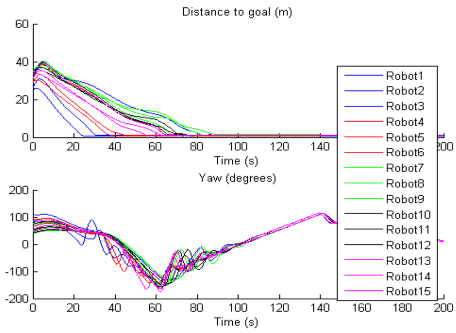
R=15, Y=5:



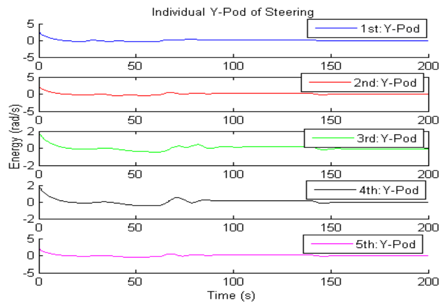
(i) 15-Robots and 5: Y-Pods



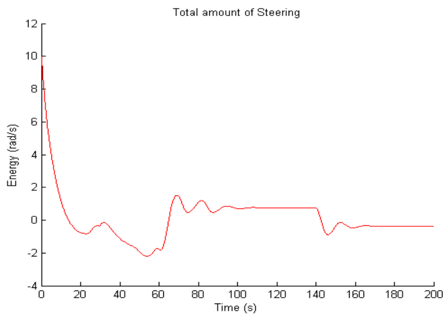
(ii) Speed and steering



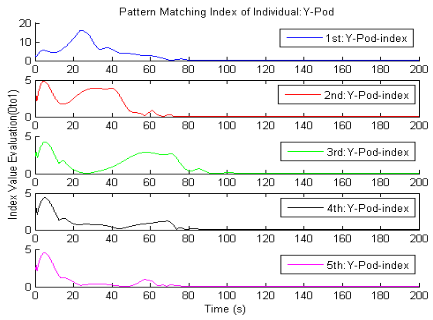
(iii) Distance to goal and Yaw angle θ



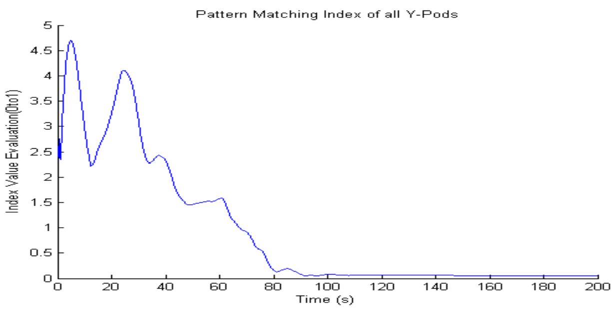
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

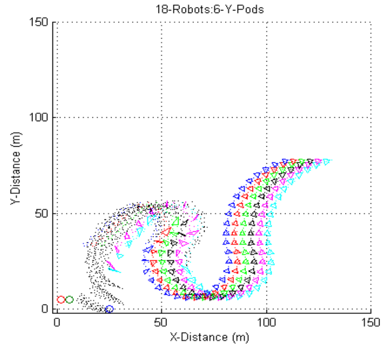


(vi) Index of individual Y-Pod

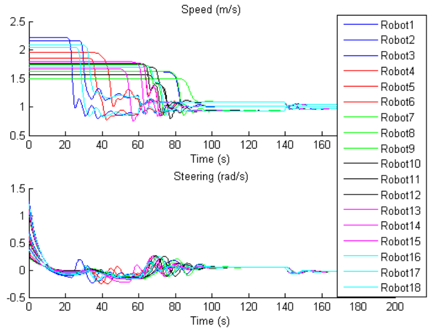


(vii) Total index of all Y-Pods

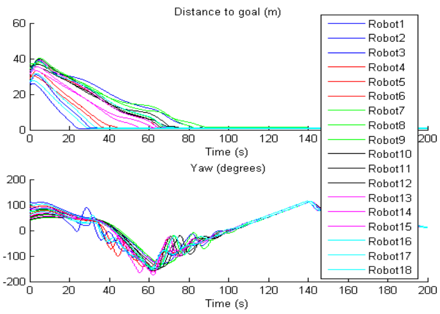
R=18, Y=6:



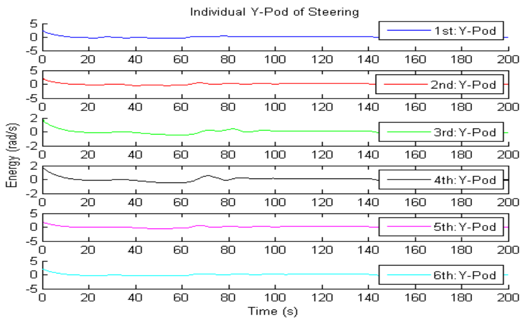
(i) 18-Robots and 6: Y-Pods



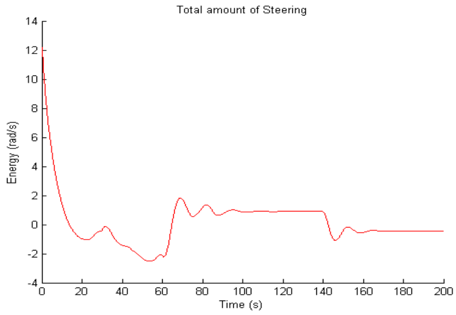
(ii) Speed and steering



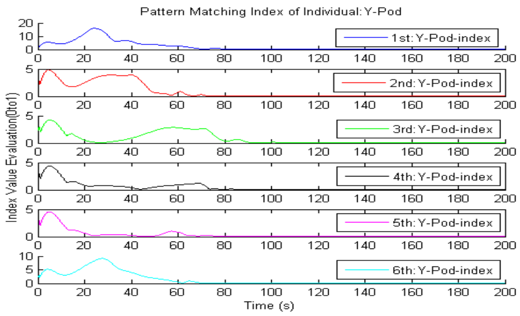
(iii) Distance to goal and Yaw angle θ



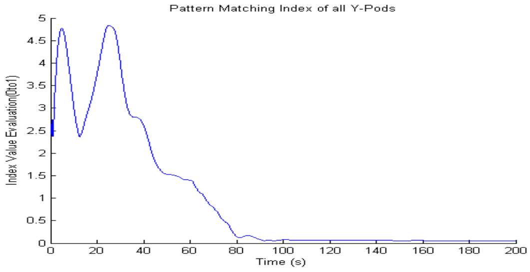
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)

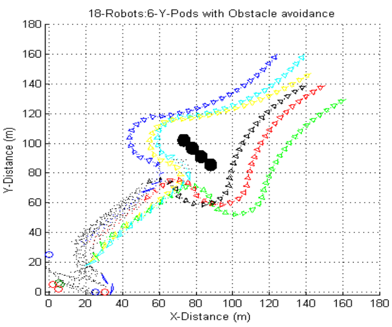


(vi) Index of individual Y-Pod

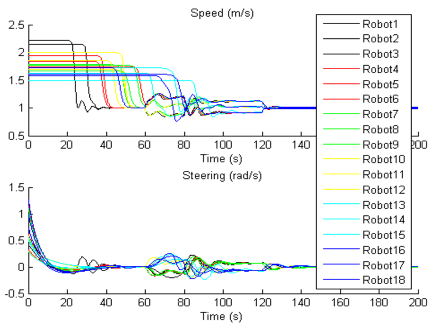


(vii) Total index of all Y-Pods

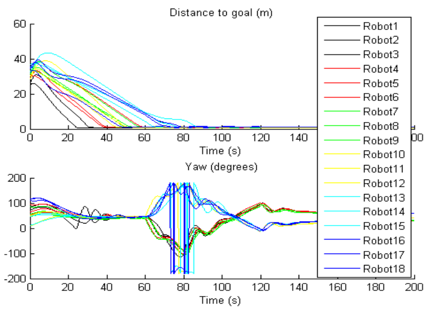
Obstacle avoidance case: <u>R=18, Y=6:</u>



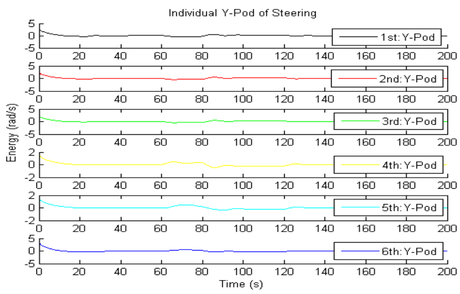
(i) 18-Robots and 6: Y-Pods



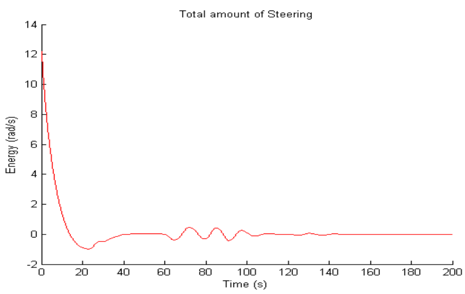
(ii) Speed and steering



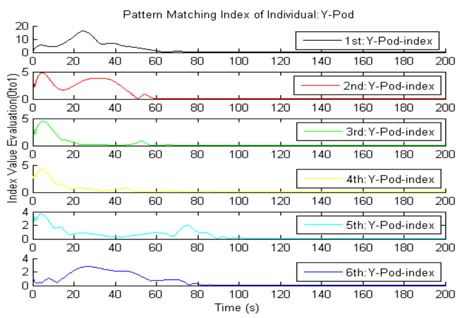
(iii) Distance to goal and Yaw angle θ



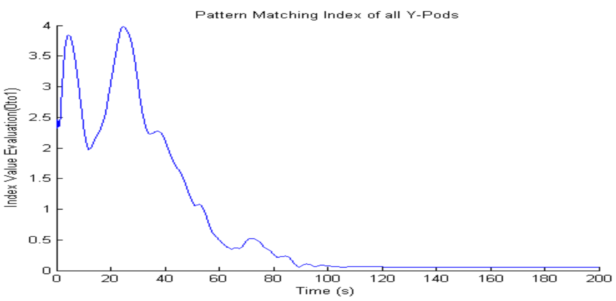
(iv) Individual Y-Pod steering



(v) Amount of energy (Total steering)



(vi) Index of individual Y-Pod



(vii) Total index of all Y-Po

ISBN: / DL:

Bibliography

Aicardi, M., Casalino, G., and Indiveri, G. (2001). Closed loop control of 3D under actuated vehicles via velocity fields tracking. *IEEE/ASME International Conference on Advanced Intelligent and Mechatronics Proceedings*, Italy.

Alvin, S. (2001). On golden spirals the subtlety of their symmetry. 3D assembler morphing architecture. *Annual Symposium DARPA.*

Ando, H., Suzuki, I., and Yamashita, M. (1995). Formation and aggregation problems for synchronous mobile robots with limited visibility. *IEEE International Symposium on Intelligent Control,* pp.453-460.

Antonelli, G. (2013).Interconnected dynamic systems: An overview on distributed control. *IEEE Control System Magazine*.

Arkin, R. (1999). Behavior-based robotics. The MIT Press, Cambridge, MA.

Balch, T. and Arkin, R. (1998). Behavior based formation control for multi robot teams. *IEEE Transaction on Robotics and Automation*, 14(6):926-939.

Balch, T.and Hybernett, M. (2000). Social potentials for scalable multi robot formation. *IEEE International Conference on Robotics and Automation*, pp73-80.

Barfoot, T.D. and Clark, C.M. (2004). Motion planning for formations of mobile robots. *Robotics and Autonomous Systems*, 46, pp, 65-78.

Barraquand, J. and Latombe, J.C. (1989). On non-holonomic mobile robots and optimal maneuvering. *Revue d Intelligence Artificial,* 3(2), pp77-103.

Beard, R.W., Lawton, J., and Hadaegh, F.Y. (2000). A feedback architecture for formation control. *American Control Conference*, pp.4087-4091

Bekey, G. and Khoshnevis, B. (1998). Centralized sensing and control of multiple mobile robots. *In Computers and Industrial Engineering*, pp503-506.

Belta, C. and Kumar, V. (2014). Abstraction and control of groups of robots. *IEEE Transactions on Robotics and Automation,* Vol 20, pp865-875.

Beni, G. and Wang, J. (1989). Swarm intelligence. *In proceeding of the 7th Annual Meeting of the Tokyo,* Japan, pp425-428.

Berder, C.M. (1954). Equation descriptive of fish schools and other animal's aggregations. *Ecology*, 35, pp361-370.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, New York, USA.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (2000). Inspiration for optimization problems from social insect's behaviour. *Nature,* 406, pp39-42.

Bonabeau, M. and Meyer, C. (2001). Swarm intelligence: A whole new way to think about business. *Harward Business Review*, 79(5):106-114.

Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on System Man and Cybernetics*, Vol, 19, no 5, pp, 1179-1187.

Brucstein, A., Wagner, L., and Lindenbaum, M. (2000). Determinism and randomness as complementary approaches to robotic exploration of continuous unknown domains. *The International Journal of Robotics Research*, 19(1), pp12-31.

Camazine, S. (1998). Trophallaxis and the regulation of pollen foraging by honey bees, *Apidogie*, 29, pp13-126.

Cao, Y.U., Fukunaga, A.S., and Khang, A.B.(1997). Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*. Vol.4, pp1-23.

Cao, Z., Tan, M., Wang, S., Fan, Y., and Zhang, B. (2002). The optimization research of formation control for multiple mobile robots. *World Congress on Intelligent Control and Automation*, pp1270-1274.

Cao, Z., Xie, L., Zhang, B., Wang, S., and Tan, M. (2003). Formation constrained multi robot system in unknown environments. *IEEE International Conference on Robotics and Automation,* pp735-740.

Chaimowicz, L. and Kumar, V. (2004). Framework of scalable control of swarms of unmanned ground vehicles with unmanned aerial vehicles. *International Conference on Robotics and Remote Systems for Hazardous Environments.*

Chaimowicz, L., Michael, N., and Kumar, V. (2005). Controlling swarms of robots using interpolated implicit functions. *IEEE International Conference on Robotics and Automation.* pp2487-2492.

Chiaverini, S. and Antonelli, G. (2003). Kinematic control of a platoon of autonomous vechicles. IEEE Intenational Conference on Robotics and Automation, pp1285-1292.

Cheng, J., Chen, W., and Nagpal, R. (2005). Robust and self-repairing formation control for swarms of mobile agents. *AAAI,* pp59-64.

Chiaverini, S., Antonelli, G., and Arrichiello, F. (2005). The null space based behavioural control for mobile robots. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp1257-1262.

Chou, Y.M. and Feng, L. (2008). Analysis of formation control and networking pattern in multi-robot systems. A hexagonal formation example. *International Journal of System Control and Communication,* Vol, 1 No 1.

Chuang, Y.L., Huang, Y.R., Orsogna, M.R., and Bertozzi, A.L. (2007). Multi Vehicles Flocking: Scalability of Cooperative Control Algorithms Using Pairwise Potentials. *IEEE International Conference on Robotics and Automation.* pp 2292-2299.

Das, A.K., Fierro, R., Kumar, V., Ostrowski, J.P., Spletzer, J., and Taylor, C.J. (2002a). A vision based formation control framework. *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp813–825.

Desai, J. P. (2001). Modeling multiple teams of mobile robots: a graph theoretic approach. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp381-386.

Desai, J. P. (2002). A graph theoretic approach for modeling mobile robot team formations. *Journal of Robotic Systems,* Vol. 19, pp511-525.

Desai, J. P., Ostrowski, J., and Kumar, V. (1998). Controlling formations of multiple mobile robots. *Proceedings of IEEE International Conference on Robotics and Automation*, pp 2864-2869.

Diestel, R. (1997). Graph Theory. New York. Springer-Verlag.

Dorigo, M. (2005). The cooperation of swarm-bots: Physical interactions in collective robots. *IEEE Robotics & Automation Magazine*, 12(2):21–28.

Dorigo, M. and Gambardella, L.M. (1997). Ant colony system: a cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Comupattion* 1(1), pp53-66.

Dudek, G., Jenkin, M., Milios, E., and D. Wilkes. (1993). A taxonomy for swarm robot. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp441-447.

Dudenhoeffer, D. and Jones, M.P. (2005). A Formation behavior for large scale micro-robot force deployment. *In Proceedings of the Simulation Conference*, pp972-982.

Dunbar, W.B. and Murray, R.M. (2002). Model Predictive Control of Coordinated Multi Vehicle Formation. *41st IEEE Conference on Decision and Control*, pp978-982.

Egerstedt, M. and Xiaoming, H. (2001). Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation,* Vol. 17, pp947-951.

Elkaim, G. H. and Kelbley, R. J. (2006). A lightweight formation control methodology for a swarm of non-holonomic vehicles. *In IEEE Aerospace Conference.*

Erik, V. and Kamesh, S. (2011). Cooperative control of swarms of unmanned aerial vehicles. *49th AIAA Aerospace Sciences Meeting and Aerospace Exposition,* pp4–7.

Everett, H.R. (2003). Robotic security systems. *IEEE Instrumentation & Measurement Magazine,* Vol. 6, pp30-34.

Fahlbusch, S., Fatikow, S., Seyfried, J., and Axel, B. (1999). Flexible micro robotic system miniman: Design, actuation principle and control. In *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Atlanta, GA, USA.*

Farinelli, A., Iocchi, L., and Nardi, D. (2004). Multi-robot systems: A classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics*, pp2015-28.

Fatikow, S. and Rembold, U. (1993). Principles of micro actuators and their applications. *IAPR Workshop on Micromachine Technologies and Systems (MTS)*, pp 108–117.

Fatikow, S., Seyfried, J., Fahlbusch, S., Axel, B., and Schmoeckel, F. (2000). A flexible microrobot-based micro assembly station. *Journal of Intelligent and Robotic Systems*, 27(1-2):135–169.

Fierro, R. and Das, A. K. (2002). A modular architecture for formation control.*Third International Workshop on Robot Motion and Control*. pp 285-290.

Fierro, R., Das, A.K., Kumar, V., and Ostrowski, J.P. (2001). Hybrid control of formations of Robots.*IEEE International Conference on Robotics and Automation*, pp.157–162.

Fredrik, G.(2001). Sensor fusion for accurate computation of yaw rate and absolute velocity. *SAE International*, SAE 2001-01-1064.

Fredslund, J. and Mataric, M. J. (2002). A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation,* Vol. 18, pp837-846.

Fujibayashi, K., Murata, S., Sugawara, K., and Yamamura, M. (2002). Self-organizing formation algorithm for active elements. *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, pp416-421.

Ge, S. S. and Fua, C. H. (2005). Queues and Artificial potential trenches for millirobot formations. *IEEE Transactions on Robotics and Automation,* Vol. 21, pp 646-656.

Ge, S. S., Fua, C.H., and Lim, K. W. (2004). Multi-robot formations: queues and artificial potential. *International Conference on Robotics and Automation*, pp3345-3350.

Godenstein, S., Large, E., and Metaxas D. (1999) .Non-linear dynamical system approach to behavior modelling. *The Visual Computer*, pp349–364.

Goldberg, D. and Mataric, M.J. (2000). Robust behaviour-based control for distributed multi-robot collection tasks. *IEEE Transactions on Robotics and Automation.*

Gonzales, L., Koestler, A., Nguyen, M., Petitt, J., Braunl, T., and Boeing, A. (2004). The autonomous underwater vehicle initiative. *IEEE Conference on Robotics, Automation and Mechatronics*, pp446–451.

Hackeloeer, A., Klasing, K., Krisp, J.M., and Meng, I. (2013). Comparison of point matching techniques for road network matching. *8th International Symposium on Spatial Data Quality,* 30 May -1June, Hong Kong.

Hirota, K., Kuwabara, T., Ishida, K., Miyanohara, A., Ohdachi, H. Ohsawa, T., Takeuchi, W., Yubazaki, N., and Ohtani, M. (1995). Robots moving in formation by using neural network and radial basis functions. *IEEE International Joint Conference on FuzzySystems and Fuzzy Engineering Symposium*, pp 91-94.

Hollis, R., Butler, Z., and Rizzi A. (2000). Cooperative coverage of rectilinear environments. In IEEE International Conference on Robotics and Automation, pP 2722 – 2727.

Huberman, B.A. and Glance, N.S. (1993). Evolutionary Games and Computer Simulations. *PNAS,* USA pp7716-7718.

Hugli, H., Tieche, F., and Facchinetti, C. (1998). Multi-layered hybrid architecture to solve complex tasksof autonomous robots. *Journal on Artificial Intelligence Tools*,pp 6-8.

Hsu, H. and Liu, A. (2005). Multiagent-based Multi-team formation control for mobile robots. *Journal of Intelligent and Robotic Systems,* Vol. 42.

Indiveri, G. (1999). Kinematic time-invariant contol of 2D non-holonomic vehicle. *38th Conference on Decision and Control.*

Jacobsen, H. (1998). A generic architecture for hybrid intelligent systems. *In IEEE World Congress on Computational Intelligence*, pages 709–714.

Jadbabaie, A., Jie, L., and Morse, A. S. (2003). Coordination of group's mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control,* Vol. 48, pp988-1001,

Jeanne, R.L. (1986). The evolution of the organization of work in social insects.*Monit Zool* Italy 20:119-133.

Jeong, D. and lee, K. (2014). Dispersion and line formation in artificial swarm intelligence. *Collective Intelligence.*

Jones, J.L. (2006). Robots at the tipping point: the road to iRobot Roomba*. IEEE Robotics &Automation Magazine,* Vol. 13, pp76-78.

Jones, C. V., Mataric, M., Sukhatme, G., Lerman, K., Koenig, S., and Mitra, U. (2004). Aformal design methodology for coordinated multi-robot systems. *IEEE/ASME.*

Junmei, W., Romain, M.W., James, W.C., Peter, K., and David, A.C. (2004). Development and testing of general amber force filed. *Journal of Computational chemistry*, Wiley Periodicals.25:1157-1174.

Kalantar, S. and Zimmer, U.R. (2004). Contour shaped formation control for autonomous underwater vehicles using canonical shape descriptors and deformable models. *MTS/IEEE TECHNO-OCEAN*, Vol.1, pp296-307.

Kazadi, S., Wen, A., and Volodarsky, M. (2009). Reliable Swarm design: Control and Automation. *17th Mediterranean Conference*, pp1301-1306.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5:90–98.

Kobayashi, F., Tomita, N., and Kojima, F. (2003). Re-formation of mobile robots using genetic algorithm and reinforcement learning. *SICE Annual Conference*, pp2902-07.

Koo, T. J. and Shahruz, S. M. (2001). Formation of a group of unmanned aerial vehicles (UAVs). *Proceedings of the American Control Conference*, pp69-74.

Kostelnik, P., Samulka, M., and Janosik, M. (2002). Scalable multi-robot formations using local sensing and communication. *Proceedings of the 3rd International Workshop on Robot Motion and Control*, pp319-324.

Kowalczyk, W. (2002). Target assignment strategy for scattered robots building formation. *Third International Workshop on Robot Motion and Control*, pp181-185.

Lawton,J and Beard,R.W.(2000). Synchronized Multiple Spacecraft Rotations. *Automatica,* Vol. 38, pp. 1359-1364

Lawton, J. R., Beard, R.W., and Young, B. J. (2003). A decentralized approach to formation Maneuvers. *IEEE Transactions on Robotics and Automation,* Vol. 19, pp933-941.

Leonard, J.J. and Durrant-Whyte, H.F. (1991). Simultaneous map building and localization for an autonomous mobile robots. *Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop* 1442–1447.

Leonard, N. E. and Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups. *40th IEEE Conference on Decision Control,* pp 2968–2973.

Lewis, M. A. and Tan, K. H. (1997). High Precision Formation Control of Mobile Robots Using Virtual Structures. *Autonomous Robots,* Vol. 4.

Lietmann, G. (1981). On the efficiency of non-linear control in uncertain systems. *ASME Journal of Dynamic Systems, Measurements and Control,* 95-102.

Lindhe, M., Ogren, P., and Johansson, K. H. (2005). Flocking with obstacle avoidance: A new distributed coordination algorithm based on voronoi partitions. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp1785-1790.

Li, X., Xiao, J., and Cai, Z. (2005). Stable flocking of swarms using local information. *IEEE International Conference on Systems, Man and Cybernetics, pp3921*–3926.

Liu, B., Zhang, R., and Shi, C. (2006). Formation control of multiple behavior-based robots. *International Conference on Computational Intelligence and Security*, pp. 544-547.

Lorenzo, S., Cristian, S., and Cesare, F. (2011). Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation.*Auton robot*: 385-387.

ISBN: / DL:

Mariottini, G. L. Morbidi, F., Prattichizzo, D., Pappas, G. J., and Daniilidis, K. (2007). Leader-Follower formations: uncalibrated vision-based localization and control. *IEEE International Conference on Robotics and Automation*, pp2403-2408.

Mataric, M.J. (1992). Designing emergent behaviors: From local interactions to collective intelligence. *International Conference on Simulation of Adaptive Behavior.*

Mataric, M., Nilsson, M., and Simsarin, K.T.(1995). Cooperative multi-robot box-pushing. *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp556-561.

McLurkin, J. (2004). Stupid robot tricks: Behavior-based distributed algorithm library for programming swarms of robots. Master's Thesis, Massachusetts Institute of Technology, Cambridge, M.A.

Michael, N., Belta, C., and Kumar, V. (2006). Controlling three dimensional swarms of robots. *IEEE International Conference on Robotics and Automation*, pp964-969.

Michael, B., Mondada, F., Dorigo, M., Nolfi, S., Baldassarre, G., Trianni. V. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Transaction on System, Man and Cybernetics*, 37:224–239.

Michaud, F., Letourneau, D., Guilbert, M., and Valin, J. M. (2002). Dynamic robot formations using directional visual perception. *IEEE/RSJ International Conference on Intelligent Robots and System*, pp2740-2745.

Milutinovic, P. and Lima. D. (2006). Modeling and optimal centralized control of a large-size robotic population. *IEEE Transactions*, 22(6):1280-1285.

Miyawaki, F., Masamune, K., Suzuki,S., Yoshimitsu, K., and J. Vain. (2005). Scrub nurse robot system-intraoperative motion analysis of a scrub nurse and timed-automata for surgery. *IEEE Transactions on Industrial Electronics*, Vol. 52, pp 1227-1235.

Monteiro, S. and Bicho, E. (2002). A dynamical systems approach to behavior-based forma -tion control.*IEEE Conference on Robotics and Automation*, pp2606-2611.

Monteiro, S., Vaz, M., and Bicho, E. (2004). Attractor dynamics generates robot formations: from theory to implementation. *Proc. of IEEE International Conference on Robotics and Automation.* 2582-2587.

Moshtagh, N., Jadbabaie, A., and Daniilidis, K. (2006). Vision-based control laws for distributed flocking of non-holonomic agents. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp 2769-2774.

Muniganti, P. and Oller, A. (2010). A Survey on mathematical models of swarm robotics. *Journal of physical agents.*

Muniganti, P. and Oller, A. (2015). Y-Pod formation of swarm robotics using amber force fields. *12th International Conference on Informatics in Control, Automation and robotics, ICINCO*, pp320-327.

Nawaz, S., Hussain, M., Watson, S., Trigoni, N. (2010). An Underwater Robotic Network for Monitoring Nuclear Storage Pools. *Sensor Systems and Software*, 24,236–255.

Nouyan, S., Grob, R., Bonani, M., and Dorigo, M. (2009). Teamwork in self organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 695-711.

Ogunsanya, A. (1986). Graph theory in intra-urban traffic flow estimation. *GeoJournal*, 12(3):334–336.

Olfati-Saber, R. and Murray, R. M. (2002). Graph rigidity and distributed formation stabilization of multi-vehicle systems. *Proceedings of the 41st IEEE Conference on Decision and Control*, pp2965-2971.

Oller, A., and Garcia, M.A. (2002). Using a linear model to evaluate performance of non-holonomic paths: The shooting-ball as a case study. *FIRA*.

Padgham, L and Winikoff, W. (2004). Developing intelligent agent systems: A Practical guide. John Wiley and Sons, Ltd.

Parker, L.E. (1998). Alliance: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14 (2), pp220-240.

Parker, L.E. (2008). Distributed Intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1).

Parker, L., Schneider, F., and Schultz, A. (2005). Multi-robot systems: from swarms to intelligent automata Vol.111, New York: Springer.

Parrish, J.K. and Hammer, W.M.(1997). Animal groups in three dimensions. Cambridge University Press.

Parrish, J. K., Viscido, S. V., and Grunbaum, D. (2002). Self-organized fish schools: An examination of emergent properties. Biological Bulletin, 202 (3), 296–305.

Ramchurn, S.D., Huynh, D., and Jennings, N.R. (2004). Trust in multi-agent systems. *The Knowledge Engineering Review*, 1(7):1-25.

Reif, J. H. and Wang, H. (1995). Social potential fields: a distributed behavioral control for autonomous robots. *Workshop on Algorithmic Foundations of Robotics*, pp331-345.

Reynolds, C.W. (1987). Flock, herds and schools: A distributed behavioural model. *In Computer Graphics, SGGRAPH-87 Conference Proceedings,* 21(4):25-34.

Rosen, B. and saalfeld, A. (1985). Match criteria for automatic alignment. *Proceedings of Auto-Carto,* Vol 7.

Rothermich, J., Ecemiş, M.İ., and Gaudiano, P. (2005). Distributed Localization and Mapping with a Robotic Swarm. *Swarm Robotics*, Vol. 3342, pp58-69.

Russell, S.J. and Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice Hall.

Saaj, C. M., Lappas, V., and Gazi, V. (2006). Spacecraft Swarm Navigation and Control Using Artificial Potential Field and Sliding Mode Control. *IEEE International Conference on Industrial Technology*, pp2646-2651.

Sahin, E. (2005). Swarm robotics: From source of inspiration to domains of applications In Swarm Robotic. *Lecture Notes in Computer Science*, Springer-Verlag, 10-20.

Sahin, H. and Guvenc, L. (2007). Household robotics: autonomous devices for vacuuming and lawn moving. *IEEE Control Systems Magazine*, Vol. 27, pp20-96.

Scheuer, A. and Fraichard, T. (1997). Continuous-curvature path planning for the carl-like vehicles. *International Conference on Intelligent Robots and Systems,* 997-1003.

Seyfried, J., Szymanski, M., Bender, N., Ramon, E., Michael, T., and Worn, H. (2005). The I-SWARM project: *SAB International Workshop*, pages 70–83.

Shao,J., Yu, J., and Wang, L. (2006). Formation Control of Multiple Biomimetic Robotic Fish. *IEEE/RSJ Intelligent Robots and Systems*, pp.2503-2508.

Shao, J., Xie, G., and Wang, L. (2007). Leader-following formation control of multiple mobile vehicles, *IET Control Theory & Applications,* Vol. 1, pp545-552.

Shaw, E. (1962). The schooling of fishes. *American Science Society*, 206, 128-138.

Shiller, Z. and Lu, H-H. (1990). *Robust computation of path constrained time-optimal motion . IEEE Conference on Robotics and Automation*, 144-149.

Shmilovici, A., Ramkddam, F., Lopez, B., and De la Rosa, J.L. (2004). Measuring progress in multi robot research with rating methods - the RoboCup example. *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 34, pp1305-1308.

Siemens, R. (2010). Fundamentals of motion control. 1st edition. Prentice Hall.

Sisto, M. and Gu, D. (2006). A Fuzzy Leader-Follower Approach to Formation Control of Multiple Mobile Robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp2515-2520.

Spears, W.M., Spears,D.F., Hamann, J.C., and Heil, R. (2004). Distributed, Physics-Based Control of Swarms of Vehicles. *Autonomous Robots*, Vol 17,137–162.

Spry, S. and Hedrick, J. K. (2004). Formation control using generalized coordinates. In *43rd IEEE Conference on Decision and Control*, pp 2441-2446.

Steve, W. (2010). 3D assembler morphing architectures. *Extra Ordinary Technology Conference,* Marriott Pyramid North, NM. Albuquerque.

Sugawara, K. and Watanabe, T. (2002). Swarming robots-foraging behavior of simple multi robot system. *International Conference on Intelligent Robots and System*, 2702-2707.

Sugihara, K. and Suzuki, I. (1996). Distributed algorithms for formation of geometric patterns with many mobile robots. *Robot Systems,* Vol. 13.

Sugiyama, H., Tsujioka, T., and Murata, M. (2006). QoS routing in a multi-robot network system for urban search and rescue. *International Conference on Advanced Information Networking and Applications.*

Tan, K.H. and Lewis, M. A. (1996). Virtual structures for high-precision cooperative mobile robotic control. *IEEE/RSJConference on IntelligentRobots and Systems*, pp132-139.

Tanford, C. and Reynolds, J. (2004). Nature's robots: A history of proteins. Oxford: Oxford University Press.

Tanner, H.G., Pappas, G.J. and Kumar, V. (2004). Leader-to-formation stability', *IEEE Transaction on Robotics and Automation*, Vol. 20, No. 3, June, pp443–455.

Vetulani, Z. (2002). Decision making for a robocup multi-agent system. *In Proceedings of the 3rd International Workshop on Robot Motion and Control,* pp193–200.

Vidal, R., Shakernia, O., and Sastry, S. (2003). Formation control of non-holonomic mobile robots with omnidirectional visual servings and motion segmentation. *IEEE International Conference on Robotics and Automation,* pp 584-589.

Wang,T., Srivatsa, M., Agarwal, D., and ling, L. (2010). Spatio-temporal patterns in network events. *ACM Context*, Philadelphia, USA.

Weimerskirch, H., Martin, J., Clerquin, Y., Alexandre, P., and, Jiraskova, S. (2001). Energy saving in flight formation. *Nature*, Vol 413,697–698.

Weiss, G. (1999). Multi agent system: a modern approach to distributed AI. MIT Press.

Woolridge, M. (2001). Introduction to Multi-agent Systems. John Wiley & Sons, Inc. New York, USA.

Wooldridge, M. (2009). An introduction to multi agent systems. John Wiley, 2nd edition.

Worn, H., Seyfried, J., Fahlbusch, S., and Schmockel, F. (2000). Flexible micro robots for micro assembly tasks. In *International Symposium on Micro mechatronics and Human Science Nagoya*, pages 22–25.

Wyman, C., Gini, M., Rybski, P., and Stoeter, S. (1997). A cooperative multi-robot approach to the mapping and exploration of mars. *AAAI-97 Proceedings*, pp798–799.

Yao, J., Ordonez, R., and Gazi, V. (2006). Swarm Tracking Using Artificial Potentials and Sliding Mode Control.*45th IEEE Conference on Decision and Control*, pp. 4670-75.

Yamaguchi, H., Arai, T., and Beni, G. (2001). A distributed control scheme for multiple robotic vehicles to make group formations, *Robotics and Autonomous Systems,* Vol. 36, pp125-147.

Yamakita, M. and Saito, M. (2004). Formation control of SMC with multiple coordinate Systems. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp1023-1028.

Zelinski, S., Koo, T. J., and Sastry, S. (2003). Optimization-based formation reconfiguration planning for autonomous vehicles. *IEEE International Conference on Robotics and Automation*, pp3758-3763.

Zhang, F., Goldgeier, M., and Krishnaprasad, P. S. (2003). Control of small formations using shape coordinates. *IEEE International Conference on Robotics and Automation*, pp. 2510-2515.

Zheng, X., Jain, S., Koenig, S., and Kempe, D. (2005). Multi-robot forest coverage. *IEEE International Conference on Intelligent Robots and Systems*, pp3852-3857.