DECENTRALIZED LEADER FOLLOWER BASED FORMATION CONTROL STRATEGIES FOR MULTIPLE NONHOLONOMIC MOBILE ROBOTS

GAYAN GAMAGE DON

Decentralized Leader Follower Based Formation Control Strategies for Multiple Nonholonomic Mobile Robots

© Gayan Gamage Don, B.Sc.Eng (Hons)

A thesis submitted to the
School of Graduate Studies
in partial fulfillment of the
requirements for the degree of

Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

October, 2009

St. John's, Newfoundland, Canada

Abstract

This thesis develops a hybrid decentralized formation control framework to coordinate multiple mobile robots with nonholonomic constraints. The proposed approach deploys a control theoretic bottom-up approach where, some low level behavior based controllers are coordinated by a discrete event system with supervisory control. The robots are required to navigate in an unstructured environment with a predetermined geometric formation while being adaptable to avoiding obstacles and following walls on the way. The complexity of the environment is handled by a discrete event system with supervisory control. For proper navigation, the multi robot systems are transformed in to flexible leader-follower coordinate structures, where we derive the aforementioned low level behavior based controllers. These controllers being nonlinear due to the nonholonomic nature of the robots involved, are subjected to linearization through nonlinear control techniques of static and dynamic feedback linearization.

Trajectory tracking type formation controllers for nonholonomic mobile robots are also developed and compared against static and dynamic feedback linearized counterparts for performance. The behavior based controllers, collectively known as formation controllers, require the designated leader/leaders robot's state and velocity profiles be known to all of its followers. Hence instead of explicit communication, we use recursive Baysian estimation techniques to estimate the leader robot's state and velocity profiles through the observations taken from sensors local to the robot. We implement and simulate different recursive Baysian estimation techniques to estimate leader robot's state and compare their respective estimation accuracy. The whole conceptual system is implemented through simulation and the results are shown to verify its operation.

Acknowledgments

I would like to express my profound gratitude to the supervisors, Dr. George Mann and Dr. Ray Gosine for their invaluable support, encouragement, supervision and financial support without which this thesis would never have materialized. I really appreciate their immense help and the opportunity bestowed on me to build up my own research career through the exploration of my own research ideas. Moreover I would specially thank Dr. George Mann for his untiring support given to me throughout this period of time. I am also highly thankful to Dr. Andrew Vardy for sharing his ideas and for the valuable suggestions given. Also I would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada and Memorial University of Newfoundland for providing financial assistance for this research. A special mentioning and a note of thanks goes to my friends and colleagues at the Intelligent Systems Lab (ISLab), specially, Migara, Dilhan, Awantha, Suranga and Dilan for their friendship and support and cooperation during my stay from whom I learnt so much.

I am as ever, especially indebted to my parents for bringing me up in times of tribulations and times of triumphs with love and care and giving me the best education a parent can give. Finally I would like to thank my energetic wife and my life parter Punyama Jayasinghe for her constant love, encouragement, support and respect without whom my life would never have been so glamorous and exciting; to whom I dedicate this thesis.

Contents

List of Tables

List of Figures

List of Symbols

v	: linear velocity of the robot
ω	: angular velocity of the robot
dt	: delta t time step
l	: leader robot
s	: follower robot
$\mathbf{X}_k$	: state of the robot at time step k
$\mathbf{\Sigma}_k$	: covariance of the robot state at time step k
$\mathbf{U}_k$	: controls for the robot at time step k
$\mathbf{Z}_k$	: observations taken by the robot at time step k
$\mathbf{M}_k$	: Kalman gain at time step k
$\mathbf{G}_k$	: Jacobian of the state dynamics with respect to state at time step k
$\mathbf{H}_k$	: Jacobian of the observation with respect to state at time step k
$\mathbf{R}_k$	: noise covariance of state dynamics at time step k
$\mathbf{Q}_k$	: noise covariance of observation at time step k
$h(\mathbf{X}_k)$	: calculated observation from the current state at time k
v_t^f	: desired linear velocity at time t
ω_t^f	: desired angular velocity at time t
v_t^s	: linear velocity of a follower robot at time t
ω_t^s	: angular velocity of a follower robot at time t
X_t^f	: desired pose in SE(2) coordinates at time t
X_t^s	: pose of a follower robot in SE(2) coordinates at time t

superscripts

$\widehat{\ }$	: estimated or approximated value of the parameter

List of Abbreviations

RMS	:	Root Mean Square
EKF	:	Extended Kalman Filter
UKF	:	Unscented Kalman Filter
SRUKF	:	Square Root Unscented Kalman Filter
GSPF	:	Gaussian Swarm Particle Filter
GMSPPF	:	Gaussian-Mixture Sigma-Point Particle Filter
UT	:	Unscented Transformation
SPKF	:	Sigma Point Kalman Filter
WSLR	:	Weighted Statistical Linear Regression
CDKF	:	Cumulative Difference Kalman Filter
RV	:	Random Variable
SRUKF	:	Square Root Unscented Kalman Filter
SRCDKF	:	Square Root Central Difference Kalman Filter
MCMC	:	Markov Chain Monte Carlo
MMSE	:	Minimum Mean Square Error
SPPF	:	Sigma point particle filter
ICR	:	Instantaneous Center of Rotation
SISO	:	Single Input Single Output
MIMO	:	Multiple Input Multiple Output
MPC	:	Model Predictive Control
MRS	:	Multi Robot Systems
SRS	:	Single Robot Systems
AHS	:	Automated Highway System

Chapter 1

Introduction

About this chapter: This chapter gives a concise account of the research problem addressed in the thesis. It describes the problem statement, motivation, specific research objectives and the contributions of this thesis in the area of decentralized formation control of multiple nonholonomic mobile robots. Finally the organization of the thesis is outlined.

1.1 Introduction

Multi robot systems are one of the key emerging research areas due to its high potential in many vibrant practical applications. The collective nature of performing a task/tasks makes it more robust than its single robot systems (SRS) counterpart. Some of the advantages of multi robot systems (MRS) over the use of (SRS) are as follows,

- Total system cost may be reduced in many domains by utilizing multiple simple and cheap robots as opposed to a single complex and expensive robot.
- The parallelism and the redundancy of multiple robots increase system efficiency, robustness, and flexibility

- The inherent complexity of some task environments may require the use of multiple robots as the capabilities required are not sufficient to be met by a single robot.

- Ability of networking, fault tolerance and sharing of resources.

MRS has been used in exploration, surveillance, cooperative transportation of large objects [1], [2], cooperative attack and rendezvous and formation control [3], [4], [5] to name a few. In majority of these applications, formation control serves as an essential element in the context of coordination. It plays the role both as a multi robot coordinator and as a multi robot controller for those applications. Being a coordinator and a controller both allows the benefits of resource sharing, parallelism, reliability, fault-tolerance, reconfiguration ability and structural flexibility to be harnessed in to other types of MRS applications. Formation control on multiple robots is inspired partly by the necessity of the nature of the tasks and partly by the formation behavior of schools of fishes or flocks of birds [3], where multiple agents combine their senses for efficient food finding or combining their thrust in the liquid or air to move forward as one pack.

Formation control usage has been explored for search and rescue missions [6], [7], reconnaissance and patrols [8], satellite control, automated highways [9]. It has been observed in [9] that, the flow of traffic can easily be managed; if vehicles in an automated highway systems (AHS) can move as a pack by keeping a desired velocity and specified distances between them. Satellite control and clustering uses formation control to reduce fuel consumption for propulsion and also to increase sensing capabilities. The basic formation control problem consists of the following objectives,

- Multiple robots should maintain a desired geometric formation of varied shapes and sizes. e.g.: triangular, line or column.

- The robot formations should be flexible enough to accommodate the geometric

constraints imposed by the environment such as obstacles, walls and corridors with inter-robot collisions avoidance.

Other challenging issues in formation control, which revolves around the major issues cited above are robot initialization [10], path planning and robot reconfiguration [11]. Many control techniques have been proposed to address the central issues in formation control. These techniques can be categorized as either centralized or decentralized. Centralized strategies consist of one or more designated leader-robots, which control the whole formation. They issue control commands for the other slave-type robots to follow. In decentralized strategies, the control is performed collectively by every robot or by a majority of robots. Boiling down the hierarchy of formation control, there are different types of formation control methodologies that are belonging to either centralized or decentralized strategies. Each method has merits and demerits on them. Some techniques involve leader-follower strategy [12], virtual structure approach [13], behavior-based formation control [3] and consensus based formation control [5]. In the leader follower method there is one or more dedicated leader robots and a set of other robots known as followers. Only the leader/leaders motion must be specified while the followers are required to maintain an inter-robot formation shape in navigation with their leader. The leader-follower classification can build a hierarchy of formations such that the first layer of followers will act as leaders for the second layer of followers and so on [4]. In the virtual structure approach, the whole formation is considered as one rigid body whose virtual center of rotation is taken as the de-facto center of the formation. The fact that the points on a rigid body remains relatively the same (relative to the rigid body under any six degrees of motion), helps to create control algorithms for each agent in a formation by just considering the motion of the rigid body itself [13]. There is no designated leader in this approach, but essentially the rigid body motion must be interpreted by a central entity to each agent in the formation. Behavior based formation control [3] uses motor schemas to define valued vectors to represent

different behaviors including formation maintenance, obstacle avoidance etc. These vectors are combined with desired modulating weights in order to generate motion commands for the robots considered. There had also been instances where the use of subsumption architecture [14] is explored for formation control of multiple robots. Consensus based control [15] uses the consensus algorithm to arrive at a consensus for a desired task. Therefore the formation control problem is also addressed in the sense of arriving at a consensus for a desired formation shape, scale, motion in [16], [17], [15] by multiple robots. The consensus based formation control can be achieved by incorporating formation constraints in the basic consensus algorithm.

1.2 Problem statement

There have been a variety of formation control algorithms developed, both centralized and decentralized over the years. Common centralized formation control strategies include virtual structure based [13], behavior based [3] and leader-follower based [4] strategies. Virtual structure based formation controls [13], require that the virtual structure position be sent out to all the robots in the formation by a central entity and also the nature of the environment must be known a *priori* to calculate the motion of the rigid structure. Moreover, navigation in an obstacle populated environment adamantly require some kind of deformation of this virtual structure, which has not been addressed properly in [13]. The behavior based approaches [3] suffer from finding the correct modulating weights to combine different motor schemas in order to yield the optimum motion commands to the robot. Also, these methods, lack in defining analytically sound vectors for formation maintenance unlike obstacle avoidance and wall following vectors (there is not a single vector but a set of rules which violates the analytical properties of schema vectors). Leader-follower based formation controls ([18], [4], [19]), are more flexible than other types of centralized strategies, simple in

operation, scalable and computationally inexpensive too. Drawbacks are that, they are highly centralized such that the position and the velocity profiles of the leader-robots must be communicated at every sampling interval to the respective followers in the formation.

Decentralized formation control strategies mostly comprise of consensus based formation control strategies ([16], [17], [15]). Although fully distributed, these consensus based controls use heavy bandwidth in communication in order to arrive at a consensus for formation control. Communication delays and certain topological graph constraints (e.g: The robot communication topology graph should at least have one directed spanning tree [20]) make the consensus algorithms more complex and lead to failure as the number of robots increase. Also the effect of noise for these given consensus based formation controllers has not been properly addressed.

In comparison of the centralized and decentralized formation control strategies, it can be concluded that most centralized strategies [4], [13], [3] are flexible and simple to implement. However the fact that these strategies are highly centralized, fails to exploit the inherent parallelism, fault tolerance an redundancies of MRS for the formation control problem. On the other hand decentralized strategies [16], [17], although fully distributed, use heavy bandwidths in communication to arrive at a consensus among all or a majority of the robots in the formation. The research problem addressed in this thesis revolves around the leader-follower formation control paradigm. It is an effort both to exploit the flexibility of the leader-follower structure for formation control of nonholonomic mobile robots and also to make the leader follower control more decentralized. The major challenges in designing a decentralized leader-follower based formation control involves addressing the following key issues.

1.2.1 Problem I: Nonlinear control laws for formation control of nonholonomic mobile robots

It is a known fact that, control laws involving posture stabilization and trajectory tracking of nonholonomic mobile robots have certain restrictions. Posture stabilization of nonholonomic mobile robots via smooth time invariant control laws is impossible (Brockett's theorem [21]). Trajectory tracking involving nonholonomic mobile robots is only possible with smooth time invariant control laws, when there is a guarantee that the trajectory does not come to a standstill [21]. These implications are a direct result of the true-nonlinearity of the nonholonomic mobile robots. The leader-follower based control theoretic formation control approaches involving nonholonomic mobile robots also exhibit similar control constraints as seen for posture stabilization and trajectory tracking problems. In fact the formation control can be thought of as a combination of trajectory tracking and posture stabilization techniques.

There are a number of nonlinear time invariant, time varying or discontinuous [22], [21], [23] control techniques proposed and implemented for posture stabilization and trajectory tracking of nonholonomic mobile robots. The fact that these numerous nonlinear control techniques have not been exploited for formation control of nonholonomic mobile robots is a problem in itself. Because different controllers derived via different nonlinear techniques can exhibit different rates of convergence, variant stability, in short variant performance which can probably build an effective set of newer controllers to address the formation control problem of nonholonomic mobile robots.

1.2.2 Problem II: Comparative study of Leader-follower based control theoretic approaches

Numerous leader-follower based control theoretic formation control approaches have been proposed and implemented, but there is a lack of a comparative study of these different approaches in terms of their flexibility, performance and extendability. Neither is there any extensive experimental validation of the proposed methods for noise tolerance in motion and in observations.

Some research shows the use of Lyapunov theory to establish the stabilizability of the proposed formation controllers [4]. But the real world implementation problems related to platform dynamics, wheel slippage, noises in observation etc. have not been sufficiently evaluated for the proposed controllers through real world experiments.

1.2.3 Problem III: Leader-follower based modularized controllers for formation control

Leader follower based formation control of nonholonomic mobile robots has been addressed in terms of control theoretic approaches [4], [24], [19], behavior based approaches [3] and fuzzy approaches [25]. The control theoretic approaches are the most widely used approaches due to its analytical ability of proof of control laws for both formation maintenance and formation stability. The extensive literature review suggests that, out of all the existing leader-follower based control theoretic approaches, the $l-\psi$ and $l-l$ controllers (see Fig.1.1) proposed by Desai et al. [4] have more flexibility and scalability for formation control applications. It has been shown in [4] that, the basic single-leader, single-follower formation can be effectively scaled to a hierarchy of multiple-leaders and multiple-followers by using the $l-\psi$ and $l-l$ controllers of [4]. It has also been shown in [4] that, multi robot formation navigation in

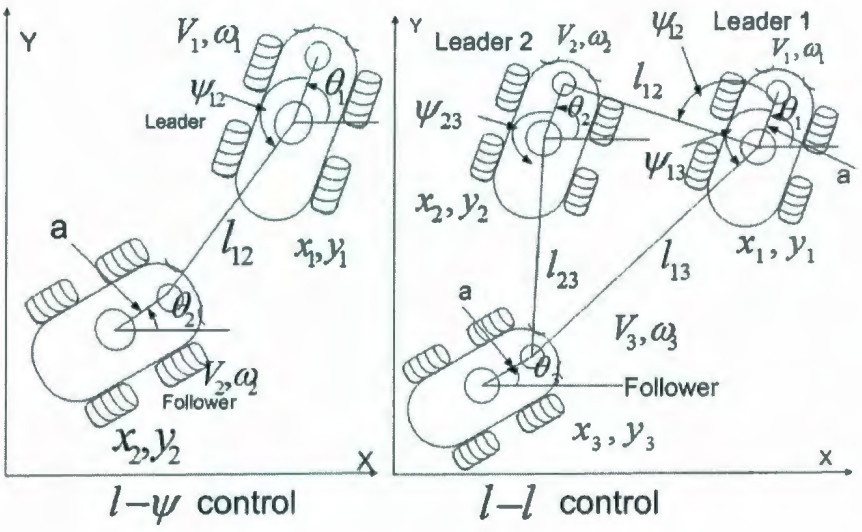


Figure 1.1: $l - \psi$ and l-l controllers

an obstacle populated environment is possible with the use of such controllers.

In all of these suggested controllers ($l - \psi$ and $l - l$ controllers) the follower robots stabilize not their origins*, but an offset from their origins* of the robot platforms to desired geometric poses. Thus these controllers lead to the following problems.

- not fulfilling the real objectives of formation control, which is to stabilize the origins of the robot platform to desired formation locations.
- if the offset from the origin of the platform is not coincident with another wheel of the robot (third castor wheel of a differential drive robot) the formation controllers become unstable.

It is found that, the problem mentioned above is a direct impact of static feedback linearization of the nonholonomic mobile robot systems.

*origin refers to the center of the axel connecting the two differentially driven wheels of a robot.

1.2.4 Problem IV: Decentralized state estimation

The leader-follower controller being a centralized controller deprives the abilities of formation maintenance, formation reconfiguration, dynamic role assignment [11], unless there is a topological communication network coordination with other robots of the system. Communication is prone to noise, delay and interference and specially in a multi robot platform the robots have limited communication abilities too. Hence the transmission of a global state of the system is subjected to many difficulties [26]. On the other hand obtaining local information is cheaper and faster with the use of exteroceptive sensors that are local to a robot, provided the robots are within the sensing range

The leader-follower formation control technique is highly centralized due to information dissemination only from the leader-robots of the system. If the leader robot's states (pose in Euclidean $SE(2)$ coordinate system and linear and angular velocities) can be measured or estimated remotely using exteroceptive sensors of the robot, the leader-follower strategy can be made more decentralized. Measuring the linear and angular velocities and leader robot orientation under noisier sensor observations is really challenging if not impossible. Hence decentralized state estimation is essential in estimating unknown states from the observations acquired from exteroceptive sensors local to the robots.

There have been the use of Extended Kalman Filter (EKF) [4] and Dual Unscented Kalman Filter (DUKF) [18] for decentralized leader robot state estimation in formation control. The experimental results recorded in both [4] and [18] are more or less for constant velocity profiles of the leader robot ([4] - linear and angular velocity are kept constant, [18] - results show the angular velocity of the leader robot is kept at zero while changing the linear velocity). Hence it is found that there is neither

any experimental validation of state estimation for different velocity profiles of the leader robot (keeping linear velocity constant while changing angular velocity, vice versa, changing both velocities etc.) nor any experimental evaluation of the performance of the formation controllers used, under these decentralized state estimation techniques. Also there seems to be no benchmarking on the estimation accuracy of different recursive Baysian filters, which can be used for decentralized leader robot state estimation.

1.2.5 Problem V: Modularized formation control framework

Path planning, formation initialization [10], formation-maintenance, formation reconfiguration, dynamic role assignment [11], obstacle avoidance and inter robot collisions avoidance are some of the essential behaviors of a formation control problem. In order to yield an effective formation control solution, these behaviors must be coordinated in optimum ways under nondeterministic environments. Although the problem of behavior coordination under nondeterministic environments had been addressed for single robots [27], [28], the multi-robot behavior coordination under nondeterministic environment still remains an open research area. The traditional control theory fails in the face of dynamic changes due to its fixed single mode of operation. Thus it highlights the need of a higher-level coordination protocol to handle the switching of the single modes of control theoretic operations [29].

Hence the problem is found to be the lack of a modularized formation control framework that will enable new behaviors be added without a significant alteration of the framework and is also scalable and robust in operation.

1.3 Research objectives

The research focus of this thesis is to address the issues related to problems I-V in section 1.2 above. The current research develops a novel leader-follower based decentralized formation control framework to coordinate multiple nonholonomic mobile robots. The proposed strategy is based on the use of static and dynamic feedback linearization to build low-level behavior based formation controllers which are coordinated by a supervisory controlled discrete event system [30], [28], [31]. The decentralized leader robot state estimation accuracy of different recursive Baysian estimation strategies is also experimentally evaluated. The feasibility of the use of existing nonlinear control techniques to develop a new set of formation controllers is also investigated.

1.3.1 Objectives

In order to achieve the proposed research goals, the following objectives are identified.

- **Objective I:** Development of a novel decentralized leader-follower based formation control framework to coordinate multiple nonholonomic mobile robots that will address the issues related to problems IV and V in section 1.2 above. Supervisory control of discrete event systems is exploited for the coordination control problem of the framework while decentralized state observation is employed to estimate leader robot's state variables.

- **Objective II:** Use of dynamic feedback linearization to develop formation controllers to address the issue related to problem III in section 1.2 above.

- **Objective III:** A comparative study to benchmark different leader-follower based formation controllers in terms of performance, noise tolerance that will address the issues related to problem II in section 1.2 above.

- **Objective IV:** A comparative study to benchmark different decentralized state

estimation techniques for estimation accuracy which will address the issues related to problem IV in section 1.2.

- **Objective V:** Development of robust formation controllers for nonholonomic mobile robots through different nonlinear control techniques that addresses the issues related to problem I in section 1.2.

1.4 Contributions of the thesis

The resulting contributions of this thesis are highlighted as follows:

1. Contributions from Objective I and II:

   (a) Development of a novel hybrid formation control framework for multiple nonholonomic mobile robots to navigate in an unstructured environment.

   - Dynamic feedback linearized formation controllers for I.) single robot navigation II.) leader-follower based formation control of multiple mobile robots in unstructured environments. These include controllers for elementary behaviors, (e.g: obstacle avoidance) and controllers for combined-behaviors, (e.g: Wall following with goal navigation). Some elementary behaviors for e.g: formation control, can be combined with wall following or obstacle avoidance by relaxing some formation-constraints
   - Similar static feedback linearized formation controllers to overcome the structural singularity of its dynamic feedback linearized counterparts (when the robot linear velocity is dropping to zero).
   - Use of supervisory control of discrete event systems to model the coordination of different behaviors of formation control.

2. Contributions from Objective III and V:

(a) Development and simulation of trajectory tracking type leader-follower formation keeping controllers for nonholonomic mobile robots, and their comparison with dynamic and static feedback linearized counterparts.

3. Contributions from Objective IV:

(a) Development and simulation of decentralized leader robot state estimation through different recursive baysian filters and particle filters and their comparison against state estimation accuracy.

1.5 Thesis organization

Chapter 2 Provides the fundamental concepts necessary for the research performed in this thesis. This chapter reviews some nonlinear control techniques and outlines the merits and demerits of the existing formation control techniques and derives the case for the research of this thesis.

Chapter 3 Develops the leader-follower dynamic feedback linearized controller and trajectory tracking based formation controllers for nonholonomic mobile robots and compares their performance against one another and also with the static feedback linearized formation controller of [4].

Chapter 4 Development and simulation of the decentralized Leader robot state estimation techniques through different flavors of recursive baysian estimation filters including the particle filter and the comparison of their state estimation accuracy.

Chapter 5 Development and simulation of a hybrid formation controller framework for multiple robot navigation in an unstructured environment. Formation controllers are developed for obstacle avoidance and wall following for both single leader robots

and multiple follower robots. It also develops a sample discrete event system which can be used to coordinate different behaviors of formation control effectively.

Chapter 6 provides concluding remarks, discussion and presents some future work.

Chapter 2

Literature Review

About this chapter: This chapter will first review the nonholonomic unicycle drive robot and it's controllability to stabilize itself to a given pose and about a feasible trajectory. Next some fundamentals and concepts of nonlinear feedback linearization will be presented. Following will be a review of the existing formation control strategies for the nonholonomic mobile robots. Existing behavior coordination techniques for the formation control problem will be described later. Finally some existing works on decentralized robot state estimation methodologies to estimate the pose and velocity of a designated robot from another robot will be explained.

2.1 Nonholonomic Mobile Robots: Fundamentals

Legged and wheeled robots are the most common types of mobile robots. But the wheeled mobile robots are the most commonly used ones attributing to their simpler, cheaper and faster characteristics [14]. There are four different basic wheel types that can be fixed in multiple to a body that makes the structure of a mobile robot. standard wheel, castor wheel, Swedish wheel, spherical wheel are the types of wheels which can be fixed to the body of the robot. Differential drive robot (Unicycle robot) normally has two standard wheels fixed with a castor wheel (P3DX robot). The

wheels are assumed to have,

- Single point of contact
- No sliding or skidding
- Plane of wheel vertical to ground

These robots have two constraints to motion, namely the rolling and the sliding constraint. Rolling constraint is the component of motion in the wheel direction which is equivalent to the roll speed and since no skidding is assumed the component of motion orthogonal to the wheel direction is taken as zero [14]. The sliding constraint to the motion in differentially driven robots makes the degree of mobility [14] two while the degree of steerability [14] (number of independent steerable wheels that yields a valid ICR) is zero. It also makes the differentially driven robots nonholonomic drive robots (Sliding constraint is a nonholonomic constraint). Using the rolling constraint and the sliding constraint of a differentially driven two standard wheel mobile robot, we can derive the dynamic equation for the i^{th} robot as,

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \\ 0 \end{pmatrix} v_i + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega_i \tag{2.1}$$

Where (x_i, y_i) is the robot pose in Cartesian coordinate system while θ_i is the robot orientation. v_i and ω_i are the linear and rotational velocities respectively.

Controllability of the Unicycle robot at a point

Equation 2.1 is driftless (no motion takes place under zero inputs) and the number of commanded inputs (v_i, ω_i) is lesser than the number of states (x_i, y_i, θ_i) of the system. Hence the linearized system of Eq.(2.1) does not satisfy the controllability rank condition [21] for control systems. As a result the above system becomes uncontrollable.

Linearized system of Eq.(2.1) at an arbitrary pose of $q_e = [x_e, y_e, \theta_e]$ (system being driftless makes any arbitrary q_e an equilibrium point of the system under zero inputs) results in,

$$\dot{q}_e = \begin{pmatrix} \cos\theta_e & 0 \\ \sin\theta_e & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{2.2}$$

Where v and ω are the linear and angular velocities of the robot. The controllability matrix of Eq.(2.2) has rank two and is lesser than the number of generalized coordinates (3). Hence the system is not controllable. But the Lie algebra rank condition (accessibility rank condition) of [32], [33] is fulfilled by the unicycle system of Eq.(2.1). Hence the given system is controllable in some sense. Accessibility rank condition for Eq.(2.1) is shown below.

System Eq.(2.1) can be written as:

$$\dot{q}_i = g_1 v_i + g_2 \omega_i \tag{2.3}$$

where $g_1 = [\cos\theta_i \quad \sin\theta_i \quad 0]^T$ and $g_2 = [0 \quad 0 \quad 1]^T$. Then

$$rank[g_1 \quad g_2 \quad [g_1, g_2] \quad [g_1, [g_1, g_2]] \quad [g_2, [g_1, g_2]]...] = \text{Number of states}$$

$$Rank \begin{pmatrix} \cos\theta_i & 0 & \sin\theta_i \\ \sin\theta_i & 0 & -\cos\theta_i \\ 0 & 1 & 0 \end{pmatrix} = 3$$

where $[g_1, g_2] = \frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2$ is the Lie bracket of g_1 and g_2.

Stabilizability of the Unicycle robot at a point

According to the Brockett's theorem [34], a necessary condition for the smooth sta-

bilizability of a driftless regular system is that the number of inputs be equal to the number of states of the system. Since this is violated with the unicycle type robot, a smooth time invariant control law fails to stabilize the robot asymptotically to a point [34]. Hence time variant or discontinuous control laws are needed to asymptotically stabilize the unicycle type robot to a given point. Time variant or discontinuous control laws employed to stabilize unicycle robots to a given point are shown in [35], [21]. It has also been established in [21] that the controllability of the unicycle robots about feasible trajectories (trajectories satisfying nonholonomic motion constraints) are possible with time variant or discontinuous control laws. The use of static and dynamic feedback linearization [32] for point stabilization and trajectory tracking of unicycle robots has been presented in [21].

2.1.1 Nonlinear Feedback Linearization

Nonlinear feedback linearization is used to linearize the nonlinear systems using feedback. A differential geometrical approach [32] used to feedback linearize a given single input single output (SISO) system is presented here. A general (SISO) nonlinear system is given by the following.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}$$

Here x is state vector, y is system output and u is one dimensional input vector. It can possibly be linearized by the combination of a change of coordinates and a state feedback [32], [36].

Nonlinear feedback linearization procedure according to [32]

1. Differentiate y until input u appears.

2. The following results will yield,

$$\begin{aligned} y &= h(x) = L_f^0 h \\ \dot{y} &= L_f^1 h + L_g(h)u = L_f^1 h \quad with \quad L_g h = 0 \\ \ddot{y} &= L_f^2 h + L_g(L_f^1 h)u = L_f^2 h \quad with \quad L_g(L_f^1 h) = 0 \\ .. &= .. \\ y^r &= L_f^r h + L_g(L_f^{r-1} h)u = v \quad with \quad L_g(L_f^{r-1} h) \neq 0 \end{aligned}$$

Where $L_f^k h = \frac{\partial L_f^{k-1} h}{\partial x} \times f(x)$ and $L_g(L_f^{k-1} h) = \frac{\partial L_f^{k-1} h}{\partial x} \times g(x)$ with $L_f^0 h = y = h(x)$.

The relative degree of the nonlinear system is the number of times the output y has to be differentiated before the control u appears. Hence the relative degree for the above SISO system is r.

If y^r is defined as $y^r = L_f^r h + L_g(L_f^{r-1} h)u = \alpha(x) + \beta(x)u = v$
where $\alpha(x) = L_f^r h$, $\beta(x) = L_g(L_f^{r-1} h)$ with $\beta(x) \neq 0$ and v is called the synthetic input. A linear controller can be designed for the above system with $v = \alpha(x) + \beta(x)u$ [32],

$$u = \frac{1}{\beta(x)}[-\alpha(x) + v] \tag{2.4}$$

Any linear method can be used to design v as given below.

$$v = -\sum_{k=0}^{r-1} c_k L_f^k(h) = -c_0 y - c_1 \dot{y} - c_2 \ddot{y}.. \tag{2.5}$$

If the relative degree is equal to the number of states of the system, then the system is turned into Brunowsky [32] form, which is linear and controllable. If the relative degree is less than the number of states of the system then there is internal dynamics present [32]. The internal dynamics is the part of the system dynamics which is un-

observable. Sometimes they make the system unstable. Thus the internal dynamics are analyzed in the sense of zero dynamics to make things simpler [32]. The multiple input multiple output(MIMO) nonlinear system formulation is an extension from the SISO geometric feedback linearization formulation.

Multiple input multiple output (MIMO) feedback linearization

Considering a system where the number of inputs is equal to the number of outputs: (The following is obtained from [32])

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x)u_i \\ y &= [h_1 \quad h_2..h_m]^T \quad with \quad y_k^{r_k} = L_f^{r_k}(h_k) + \sum_{i=1}^{m} L_{g_i}(L_f^{r_k-1}(h_k))u_i \end{aligned}$$

Here x- state vector, u_i - i^{th} one dimensional input vector (there are m one dimensional input vectors), y- $m \times 1$ dimensional output state vector and $y_k = k^{th}$ output channel. Also $L_f^{r_k}(h_k) = \frac{\partial L_f^{r_k-1}h_k}{\partial x} \times f(x)$ and $L_{g_i}(L_f^{r_k-1}(h_k)) = \frac{\partial L_f^{r_k-1}h_k}{\partial x} \times g_i(x)$ with $L_f^0(h_k) = y_k = h_k$, where r_k is the relative degree of each output channel k for some $L_{g_i}(L_f^{r_k-1}(h_k)) \neq 0$.

If there is a $m \times m$ matrix $J(x)$ such that,

$$J(x) = \begin{pmatrix} L_{g_1}(L_f^{r_1-1}(h_1)) & ... & L_{g_m}(L_f^{r_1-1}(h_1)) \\ ... & ... & ... \\ L_{g_1}(L_f^{r_m-1}(h_m)) & ... & L_{g_m}(L_f^{r_m-1}(h_m)) \end{pmatrix} \tag{2.6}$$

The $J(x)$ in Eq.(2.6) is called the decoupling matrix. If $J(x)$ is assumed to be non-singular and if we define,

$$y^r = \begin{pmatrix} \frac{d^{r_1} y_1}{dt^{r_1}} \\ \dots \\ \frac{d^{r_m} y_m}{dt^{r_m}} \end{pmatrix} \quad and \quad l(x) = \begin{pmatrix} L_f^{r_1}(h_1) \\ \dots \\ L_f^{r_m}(h_m) \end{pmatrix}$$

Both y^r and $l(x)$ are $m \times 1$ vectors. Hence we get,

$$y^r = l(x) + J(x)u = v \tag{2.7}$$

v is a $m \times 1$ vector and can be designed using linear techniques as in the SISO case above. Then the control becomes [32].

$$u = J^{-1}(v - l)$$

Below we show how the static and dynamic nonlinear feedback linearization can be used to feedback linearize the unicycle system of Eq.(2.1) and drive the robot to a desired (x^d, y^d) Cartesian pose. Static and dynamic feedback linearization will be used to derive controllers necessary for formation control in the coming chapters. This serves as an explanation of the methods used.

Static feedback linearization for point stabilization of unicycle robots

The static feedback linearization for generalized SISO and MIMO systems has been explained in [21]. For Eq.(2.1): the Jacobian Matrix (Eq.(2.6)) derived through MIMO feedback linearization is singular. Hence it can be made non-singular through changing the output state (measurements) (x_i, y_i) to an offset from the current output

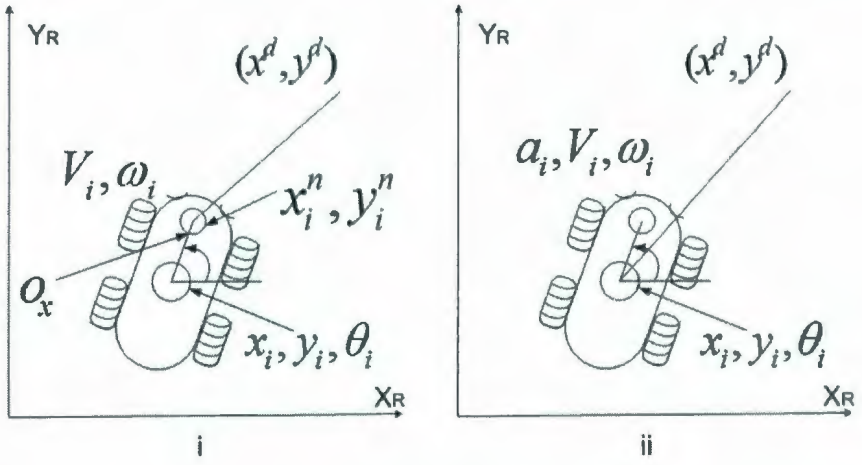


Figure 2.1: Point (x^d, y^d) stabilization of nonholonomic mobile robots via (i.) static feedback linearization (ii.) dynamic feedback linearization

state (current output state is the the origin of the robot). It results in,

$$\begin{pmatrix} x_i^n \\ y_i^n \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} o_x \\ o_y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{2.8}$$

o_x and o_y are offsets from the *origin* of the robot-coordinate system in X_R and Y_R directions respectively. And (x_i, y_i, θ_i) are the current output state coordinates in the global-coordinate system while (x_i^n, y_i^n) are the newest output state (measurement) coordinates in the global-coordinate system (see Fig.2.1 (i.)). Differentiation of the newer output state coordinates with respect to time, results in a nonsingular dynamic system which is readily controllable.

$$\begin{pmatrix} \dot{x}_i^n \\ \dot{y}_i^n \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -o_x\sin\theta_i - o_y\cos\theta_i \\ \sin\theta_i & o_x\cos\theta_i - o_y\sin\theta_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} \tag{2.9}$$

Using the method of MIMO nonlinear feedback linearization above applied to Eq.(2.9), a controllable system can be derived, which stabilizes the unicycle robot to a given

(x^d, y^d) Cartesian pose. It should be noted that this particular controller does not stabilize the origin of the robot to a desired pose. Instead it stabilizes the defined offset from the origin to the desired pose.

Dynamic feedback linearization for point stabilization of unicycle robots

The dynamic feedback linearization for generalized SISO and MIMO systems has been explained in [21]. Since the decoupling matrix $J(x)$ of Eq.(2.6) is singular for a unicycle type robot, dynamic extension or dynamic feedback linearization can be used to linearize the unicycle system.(see Fig.2.1 (ii.))

$$\dot{x_i} = x_3 \cos\theta_i \quad \dot{y_i} = x_3 \sin\theta_i \quad \dot{x_3} = a_i \quad \dot{\theta_i} = \omega_i \tag{2.10}$$

where $x_3 = v_i$ and a_i is the linear acceleration of the system, (x_i, y_i, θ_i) is the pose in Euclidean $SE(2)$ coordinate system and (v_i, ω_i) are the linear and angular velocities respectively. Taking $\mathbf{z} = [z_1 \quad z_2]^T = [x_i \quad y_i]$ as the output of the system, and through differentiation of Eq.(2.10) we get,

$$(\ddot{\mathbf{z}}) = \begin{pmatrix} \ddot{x_i} \\ \ddot{y_i} \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -v_i \sin\theta_i \\ \sin\theta_i & v_i \cos\theta_i \end{pmatrix} \begin{pmatrix} a_i \\ \omega_i \end{pmatrix} \tag{2.11}$$

The new Jacobian matrix is nonsingular as long as the axle is moving. Thus the system has a singularity at $v_i = 0$. Again using the method of MIMO nonlinear feedback linearization, we can derive a controllable system which stabilizes the unicycle robot to a given (x^d, y^d) pose, but this time with a singularity at $v_i = 0$.

2.2 Formation Control Methods

The basic objective of a formation control problem is to keep a predetermined geometric formation as much as possible while being adaptable to the changes of the environment. Either a detection of an obstacle or a wall needs a swift reaction from the robots involved in the formation such that the obstacles must be avoided or the walls must be followed. In doing so, the hard constraints of formation keeping should be relaxed and once the obstacles are avoided the robots can re-enter in to the formation. Centralized and decentralized strategies have been proposed and implemented to address the formation control problem. In centralized methods there is a designated leader/leaders robot which essentially communicates its position, speed and other information to a set of other robots called followers and the followers move in to desired patterns with formation maintenance techniques. Some of the common centralized formation maintenance techniques are leader-follower [4], behavior-based [3] and virtual-structure [13].

Decentralized methods are thought of as implementing a formation pattern through a collective approach, where all robot poses and speeds are considered in building the formation. Consensus based formation control strategies [16], [17], [15] are examples to decentralized formation control. If decentralized state estimation is used in estimating the leader/leaders position and speed (in the leader-follower centralized formation control strategy) from the exteroceptive sensors of the robot without explicit communication, then that formation strategy becomes decentralized too. Robot initialization - geometric position assignment to robots which are initially in arbitrary locations [10], obstacle avoidance, inter robot collision avoidance and dynamic role assignment [11] are some of the challenging issues in formation control. For this thesis, the problems of obstacle avoidance and wall following issues are considered with formation control. The existing formation control strategies with their merits and

demerits are explained below in a nutshell.

2.2.1 Consensus based formation control

Consensus algorithm has been used in many contexts, where there is a need for group consensus or agreeing to do or allow some form of a task. The rendezvous problem [37] is the easiest form of consensus for multi vehicle cooperative control. It is given by,

$$\dot{x}_i = -\sum_{k=1}^{n} a_{ij}(x_i - x_j) \tag{2.12}$$

Single integrator dynamics for the agent i is considered. x_k is the position of the agent k and there are n agents in the domain. a_{ij} is a weight representing the reliability of information obtained through communication between the robots i and j. When Eq.(2.12) is executed for n number of point masses under no disturbance, the point masses will all arrive at the initial center of the configuration simultaneously (e.g: If three robots were placed initially at arbitrary positions, they will arrive at the center of it's initial triangle simultaneously). Formation stabilization, formation maneuvering through consensus algorithms are achieved by including formation constraints to Eq.(2.12). Most formation stabilization techniques assume that the robot positions and the shape of the formation to be maintained are pre-known. Hence each robot initializes its information state by proposing a formation center and a consensus algorithm is used to agree on a common formation center by all or a majority of robots [16], [20]. Decentralized formation maneuvers are also modelled through consensus based control strategies [38], where an event such as a detection of an obstacle will make the group of robots maneuver away from the obstacle either by shrinking or expanding the formation. Decentralized formation maneuvers can also be executed via the consensus algorithms (by incorporating the formation maneuvering needs in the basic consensus equations). The basic consensus equation is in the continuous

mode of operation but apparently it has been extended to discrete time operation too. Consensus based control strategies are robust since there are no designated leaders. Every robot has to contribute towards generating a solution. It increases fault tolerance but suffers from heavy bandwidth usage since every robot or a certain section of robots must communicate with each other for the consensus control strategy to converge. Communication delays and certain topological graph constraints (e.g.: The robot communication topology graph should at least have one directed spanning tree [20]) makes it more vulnerable as the number of robots increase. Also the effect of noises to these given algorithms of formation control has not been properly addressed.

2.2.2 Behavior based formation control

Different behaviors are fused together including the formation maintenance behaviors to enable a robotic team to reach navigational goals, avoid obstacles and also to maintain a desired formation shape [3], [39]. The behaviors are represented as motor schemas, which generate desired force vectors for each stimulus from the environment. The resulting control action is a weighted vector from those different schema vectors. [3] proposes such a behavior based approach by combining obstacle avoidance, inter vehicle collision avoidance, goal seeking and formation keeping behaviors through their schema generated vectors. What's new in this approach, is the definition of a formation maintenance behavior through a new schema-vector. [3] accomplish the formation maintenance in two steps; 1.) detect-formation position 2.) maintain formation. In order to detect formation position, [3] use a unit center referenced or a leader-referenced or a neighbor-referenced strategy, each of which needs the location of the other robots in the formation. Hence [3] uses inter-robot communication for this matter. Ones the formation position is known, the maintain-formation behavior generates motor commands to drive the robot to the desired formation position.

The maintain-formation motor schema is established through the concatenation of maintain-formation-speed and maintain-formation-steer motor schemas. These low level schemas are implemented through a set of rules which has numerous parameters. Hence this definition of a motor schema exhibits problems of correct value selection for these parameters in the set of rules. Also, such a motor schema definition will always drive the robot to an approximate pose instead of the real desired geometric pose. It can always be seen that, in order to keep a good formation one needs not only the feed forward command but also some feedback action too. A common difficulty for behavior based approaches of finding the correct modulating weights for different behaviors applies to this given solution in [3] as well.

Social potential fields [40] have also been used in developing behavior based formation control techniques. Social potential fields [40] are simple artificial force laws between pairs of robots or robot groups. They incorporate both attraction and repulsion forces. A single robot's motion is controlled by the resultant artificial force of attraction and repulsion imposed by other robots of the system. Here also representing a formation behavior is not mathematically sound, it will possibly give an approximate solution to formation maintenance but has some advantages of large scale robot group deployment. [41] describes the use of artificial potential trenches to represent the desired geometric formation in navigation. Each robot is made to follow along the deepest regions of the potential field and are themselves distributed accordingly to mimic the formation shape. This method also suffers from similar drawbacks seen above including use of extensive communication among all the robots.

2.2.3 Virtual structure based formation control

A virtual structure based formation control method is presented in [13]. It is observed that the relative positions of points in a rigid body remain fixed under any 6 degrees of motion in the space. [13] uses this idea to develop a virtual structure based formation control method. In [13], a virtual rigid structure is translated and rotated to mimic the required movement of the formation and control laws are derived to send the real robots to their fixed positions in the virtual structure. Disadvantages of this method are that the virtual structure position must be sent to all the robots by a central entity and the nature of the environment must be known a *priori* in order to calculate the motion of the rigid structure. Obstacle avoidance probably needs some deformation of the virtual structure, which has not been addressed in [13].

Formation control through generalized coordinates has been proposed in [42]. Here, the formation is characterized by some generalized coordinates which include the robots position, its orientation and its shape with respect to a formation reference point. Control laws are developed for asymptotic tracking of trajectories resulting from the motion of these generalized coordinates. Although this method resembles a flexible virtual structure, all robots need their desired position information from a central entity. In Chapter 3, the development of a combined leader-follower virtual structure based formation control approach for nonholonomic mobile robots is presented. It is based on the trajectory tracking techniques for nonholonomic mobile robots. The feed-forward command for a virtual structure on the trajectory is derived from the current state and velocity profiles of a designated leader robot. The feedback action on the error of desired pose to the current is committed by some existing trajectory tracking controllers [22], [23] for nonholonomic mobile robots.

2.2.4 Leader-follower based formation control

A vision based formation control framework which builds the basis for this research is identified in [4]. It develops low level controllers based on static feedback linearization for formation control of nonholonomic mobile robots. Instead of communication [4] uses decentralized state estimation of the designated leader through an Extended Kalman Filter (EKF) for driving the followers in to formation. [4] uses two types

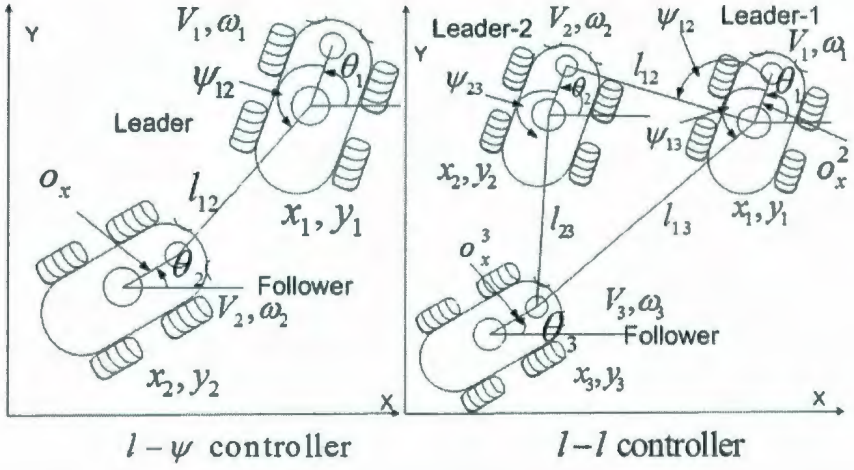


Figure 2.2: $l-\psi$ and l-l controllers

of feedback controllers for maintaining formations of multiple robots. Fig.2.2 shows these two controllers. The first controller in Fig.2.2: $(l-\psi)$ is used to maintain a desired length l^d and a desired relative angle ψ^d between the leader and the follower as shown for differential drive mobile robots. The $l-l$ controller in Fig.2.2 is a three robot formation controller where one robot (leader 2) follows another robot (leader 1) using the $l-\psi$ controller while another third robot (follower) is controlled to follow the two leaders with l_{13}^d and l_{23}^d distances. Both of these controllers use static input/output feedback linearization to yield linear controllable systems to drive the robots to desired values. Hence as explained in Chapter 1, these controllers does not stabilize the origin of the followers to desired values, but some offsets from the origin. In Chapter 3, the undesirable effects of stabilizing offsets from the origin of differential

drive robots to desired values instead of the real origin (if the offset is not coincident on the third castor wheel of the robot) are shown. Instead of stabilizing offsets from the origin to desired values [43] develops a similar $l-\psi$ type basic formation controller through dynamic feedback linearization to stabilize the origin to desired values. (This controller was initially developed in this research and found later that it has already been formulated in [43]).

The $l-\psi$ dynamic feedback linearized based basic formation controller of [43] processes a structural singularity. Hence both static [4] and dynamic [43] feedback linearized formation controllers are combined in this research, in order to overcome the structural singularity and to achieve an effective formation control solution. It is also shown in this research, that this particular $l-\psi$ controller can be used with some modifications for single robot navigation too whereas [4] and [43] uses it only for formation maintenance. We also find that the effect of noise for these controllers given in [4], [43] has not been properly investigated, nor is there is any comparison of these controllers in terms of performance, noise tolerance etc. Inspired from the $l-l$ controller in [4], this research also develops extended formation controllers both through static and dynamic feedback linearization means, to occupy wall following and obstacle avoidance capabilities in formation control.

Another leader follower based formation control of a team of nonholonomic mobile robots using omnidirectional vision is described in [19]. By specifying the desired motion of the followers in the image plane, [19] translates the control problem to a visual servoing task. In order to estimate the state and velocity profile of the leader, [19] uses the rank constraint on the omnidirectional optical flows across multiple frames in the image plane of each follower. One problem of this approach is that the leader state estimation through omnidirectional optical flow is prone to error when establishing

the leader's linear and rotational velocities and it is more computationally involving too.

2.2.5 Fuzzy based formation control

The use of fuzzy logic for formation control is shown in [25]. Fuzzy rules are written specifically to maintain a column formation and are generalized to other formation types using virtual leaders so as having many column formations (with virtual leaders) inside any type of formation. There are linguistic "If and Then" sets of rules which control the robot rotation and linear velocity upon receiving laser measurements to maintain formation and to avoid obstacles. A higher level fuzzy coordination layer coordinates these formation maintenance and obstacle avoidance behaviors. The general problems encountered by the use of fuzzy systems such as difficulty of tuning the membership functions, modeling complexity increase due to any increase in the rule base, lack of a mathematical explanation of controllability, observability and approximate solutions when there is the possibility for near accurate solutions make the fuzzy based formation solution an unfavorable choice.

2.2.6 Model Predictive Control based formation control

A model predictive control (MPC) algorithm (called first state contractive-MPC) is proposed in [24] to address issues of trajectory tracking, point stabilization and formation control of nonholonomic mobile robots. This work claims to have obtained locally optimized controls at every sampling interval for formation navigation which demands less control energy than other control techniques. [44] uses model predictive control as a local controller to increase the overall formation performance under noisier inter-robot communication. Here the correlation of the quality of information to the formation performance is investigated.

All MPC schemes consist of a state prediction horizon and an optimum local control u_k calculation at time k (this calculation depends on the state predictions and is obtained through a performance index which consists of penalties and rewards). Hence the MPC algorithms have to calculate these predictions and solve for the optimum control values online. Such a computation is computationally expensive. Also modeling a performance index under different disturbances from the environment with rewards and penalties is cumbersome.

2.3 Behavior coordination in formation control

The core behaviors of the formation control consists of formation keeping, obstacle avoidance and wall following. There can also be supportive behaviors of robot initialization, formation switching and dynamic role assignment. Those behaviors have to be well coordinated in order to yield an optimized formation control algorithm. [4] provides a gross controller switching strategy for nonholonomic robots in formation navigation. The $l-\psi$ and $l-l$ controllers of [4] in Fig.2.2 are used for obstacle avoidance and formation keeping purposes and are coordinated by a coordination rule set. It's a hard coded rule set of "If and Then's" which lacks scalability as the number of robots increases. Adding new behaviors through new controllers will also be difficult without major modifications of the rule set.

A higher-level coordination layer to coordinate formation control behaviors based on a fuzzy logic control is proposed in [25]. The fuzzy formation maintenance and fuzzy obstacle avoidance controllers proposed in [25] are again coordinated by another higher-level fuzzy layer. The general problems of fuzzy based control apply to the given solution as well. Tuning the membership functions, modeling complexity increase due to any increase in the rule base, lack of a mathematical explanation of

controllability, observability and then to the stability of the system are the problems of the usage of a fuzzy coordination layer in [25]. Behavior based formation control [3] proposes the combination of different behavior schemas for mobile robot formation control such that the end action is a result of action coordination. Different schemas for obstacle avoidance, wall following and formation maintenance are to be combined together using modulating weights. Finding these correct modulating weights is a prime problem of any behavior based approach [45]. There are implementations in the written literature through the use of fuzzy context dependant blending or fuzzy discrete event systems to finding these weights [46]. But the use of fuzzy systems suffers from the same problems outlined above for fuzzy coordination layer in fuzzy based formation control above.

This research proposes the use of discrete event systems with supervisory control as a building tool for the coordination protocol of the multi robot formation control problem. Supervisory controlled discrete event systems have been used to build a coordination platform for single robot navigation in [28], but they are not being much exploited for coordination problems in the multi robot domain. The use of supervisory control of discrete event systems provides the modelling ease, scalability, reusability for most applications which are based on event-triggered behavior transitions.

2.4 Decentralized state estimation in formation control

There are few research articles which deal with leader-follower based decentralized formation control by estimating the leader robot's state and velocity by the use of Kalman type filters. [4] uses an extended Kalman filter based strategy for leader robot state estimation. The results of estimation in [4] are for constant velocity profiles of

the leader and there are no any experiments of estimation for possibly changing velocity profiles of the leader. [18] deals with an implementation of visual tracking of mobile robots in formation. An image segmentation technique is used in [18] to segment possible leader robots from the images captured and uses decentralized state estimation to estimate segmented leader's state and velocity profiles. For estimation [18] uses a Dual Unscented Kalman Filter (DUKF). Again there is no experiment on state estimation for different velocity profiles of the leader (results show the angular velocity of the leader robot is kept at zero while changing the linear velocity). Moreover the effect of these noisy measurement on the formation controller used has not been experimentally evaluated. Also there seems to be no benchmarking on the estimation accuracy of different recursive Baysian filters, which can be used for leader robot state estimation.

2.5 Summary

- It is evident from the literature review that the leader follower concept is the basis for most of the existing formation control strategies due to its simplicity, scalability via hierarchy of leaders and followers, controllability, stabilizability and flexibility etc.

- It is also found that there is no substantial qualitative analysis on the different leader-follower based formation controllers in the written literature especially for nonholonomic mobile robots.

- Although [4] provides interesting results of two local controllers for formation control, they all stabilize not the origin of the nonholonomic robot, but an offset of the origin to desired values. In Chapter 3, it is shown that stabilizing an offset from the origin (if the offset does not happen to be coinciding with the third castor wheel of the differential drive robot) has some undesirable effects on the

stability of the controller.

- Also there is no benchmarking of possible recursive Baysian type filters for decentralized state estimation of the state and velocity profile of the leader-robot.

- Finally, the multiple behavior coordination problem has not been properly addressed in the domain of formation control, where formation maintenance, obstacle avoidance, wall following has to be effectively coordinated for navigation.

Chapter 3

Leader Follower Based Formation Controllers

About this chapter: This chapter explains the development, simulation and comparison of different leader-follower based formation maintenance strategies for multiple nonholonomic mobile robots. The key contributions of this chapter are,

- Development and simulation of trajectory tracking type formation maintenance controllers.
- Development and simulation of dynamic feedback linearized formation maintenance controllers.
- Comparison of trajectory tracking, static feedback linearized and dynamic feedback linearized formation maintenance controllers in terms of formation accuracy, noise tolerance and smoothness of control inputs using P3AT mobile robots.
- Highlighting the undesirable effects of stabilizing offsets from the origin of the robot frame to desired formation values instead of stabilizing the origin itself to the desired formation values.

3.1 Background

Leader based formation control requires that the followers keep predetermined geometric formation with respect to the leader robot. If the pose of the leader robot is known a *priori*, the desired position of the followers in the Euclidean SE(2) coordinate system can simply be described geometrically. These desired poses of followers become fixed poses in the leader-robot coordinate system. Hence the entire motion for followers can be modelled solely through leader robot dynamics. The nonholonomic motion results in a path, which can be approximated through an accumulation of straight and circular path segments [47]. The motion of any point fixed in an offset to the origin of the leader robot coordinate system results in a similar path to that of when the point is the origin itself. Hence the path resulting from this fixed point is feasible for another nonholonomic mobile robot to track. This research proposes three types of formation controllers for the nonholonomic unicycle robots. Two of such controllers are developed through virtual robot path tracking techniques and another through dynamic feedback linearization. The first controller is based on the approximate linearization of the unicycle dynamics described in [21]. The second controller is based on a Lyaponov-based nonlinear time varying design [22]. Third controller is developed through dynamic feedback linearization. It is also shown here, that the static feedback linearized formation controller described in [4] has flaws in terms of stability. Real time formation control simulation results through P3AT mobile robots in MobileSim/PlayerStage are presented for comparison purposes of the developed controllers in terms of formation accuracy, noise tolerance and smoothness of control inputs.

3.1.1 Target robot: P3AT

The experimental validation of the proposed and existing control schemas are carried out in P3AT mobile robots both in simulation and in the real physical world. P3AT description follows:

Physical Description

P3AT is a four wheel differential drive mobile robot with $50cm$ x $49cm$ x $26cm$ aluminum body with $21.5cm$ diameter drive wheels. Its four motors use 38.3:1 gear ratios and contain 100-tick encoders. On flat floors this robot can move at translational speeds of 0.6 ms^{-1} and rotate at a maximum angular velocity of $\pm 43° s^{-1}$. On flat terrains, it can carry a payload up to 30 kg at slower speeds and these payloads must be balanced appropriately for effective operation of the robot. The three fully charged batteries allow the robot to run for 3-6 hours. P3AT includes a Renesas SH7144 based microcontroller and it has multiple I/O varieties and these user I/O are integrated into the packet structure, accessible through ARIA software. It has 8 forward and 8 rear sonars which can sense obstacles from 15 cm to 7 m. P3AT can be optionally loaded with global position systems (GPS), differential GPS (DGPS), bumpers, gripper, vision, stereo range finders, laser range finders and compass etc.

P3AT controlling Architecture

ARCOS (Advanced Robot Controller Operating System) which runs on the robot embedded computer transfers sonar readings, motor encoder information and other I/O via packets to the PC clients and returns control commands from the clients. The communication to a PC client can be established through (a.) wireless radio modem, (b.) robot-to-laptop connector, (c.) robot-to-desktop tether, (d.) connection to an embedded computer. Using ARIA Robotics API, users can write C/C++ or Java programs to control the robot. The API provides a richer control interface to control

the robot to do mapping, navigation, vision, cooperation and manipulation etc. Also this interface provides the user with lower level control ability of the robot: The linear v and the angular ω speeds can be set to drive the robot or the speeds of left ω_L and right ω_R wheels can be independently set to drive it.

Control constraints

The P3AT robot has saturation levels of the linear and angular velocities and linear and angular accelerations.

$$| v | \leq v_{max} = 0.6ms^{-1}, \qquad | \omega | \leq \omega_{max} = 0.75rads^{-1}$$

$$| a | \leq a_{max} = 0.3ms^{-2}, \qquad | \alpha | \leq \alpha_{max} = 0.8rads^{-2}$$

With these given constraints the controllers need a velocity scaling so as to preserve the path curvature radius originated from the given v and ω. (since the linear and angular velocities are upper bounded, a velocity scaling is used to give the same ratio of $v : \omega$, such that the robot moves in the same path, but now with different linear and angular velocities). Hence the actual commands for the robot are computed through a procedure given below:

$$\Lambda = max\{| v | /v_{max}, \quad | \omega | /\omega_{max}, \quad 1\}$$

If the scaled down linear and angular velocities are v_s and ω_s respectively, we have:

If $(\Lambda == | v | /v_{max})$ then $v_s = sign(v)v_{max}, \quad \omega_s = \omega/\Lambda$
Else If $(\Lambda == | \omega | /\omega_{max})$ then $v_s = v/\Lambda, \quad \omega_s = sign(\omega)\omega_{max}$
Else $v_s = v, \quad \omega_s = \omega$

With this choice, the shape of movement for a particular task will be preserved but tracking targets of inaccessible velocities and distances will be infeasible, which is

rightly justified owing to the mechanical constraints of the system. The MobileSIM simulator also captures the uncertainty of the wheel encoder measurements, communication delays and uncertainty of other measurements etc. Hence a real world simulation can be worked out with the simulator.

3.2 Virtual robot tracking based formation controllers

Tracking a virtual robot path and its desired velocities requires a combination of a nominal feed forward command with a feedback action on the error [48]. In formation control the pose of the virtual robot and its velocities to which the actual designated follower must reach to, is gained through the geometrical relationship of the virtual robot to the actual leader.

3.2.1 Feedforward Command Generation

Assuming that the leader robot's pose at time t is $[x_t \quad y_t \quad \theta_t]^T$ and the velocities being $[v_t \quad \omega_t]^T$, we can describe the desired position of the follower as an offset of o_x units and o_y units from the origin to X and Y directions respectively in the leader robot coordinate system. $(x_t^f, y_t^f, \theta_t^f)$ is the desired pose for a follower robot in the Euclidean $SE(2)$ coordinate system. E.q.3.1 is taken from Fig. 3.1.

$$\begin{pmatrix} \dot{x}_t^f \\ \dot{y}_t^f \\ \dot{\theta}_t^f \end{pmatrix} = \begin{pmatrix} \cos\theta_t & -o_x \sin\theta_t - o_y \cos\theta_t \\ \sin\theta_t & o_x \cos\theta_t - o_y \sin\theta_t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} \tag{3.1}$$

The feed-forward command generation of unicycle robots involves generating desired poses $(x_t^f, y_t^f, \theta_t^f)$ and desired velocities (v_t^f, ω_t^f) at a given time t. The desired poses

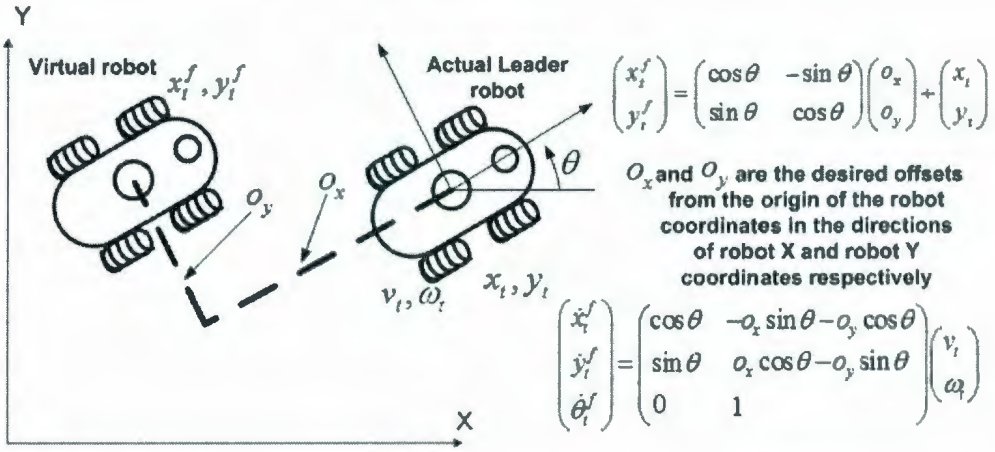


Figure 3.1: Virtual robot representation for tracking based formation control

can be easily taken as shown in Fig. 3.1. The desired linear and angular velocities are taken as in [48], given by Eq.(3.2);

$$v_t^f = \pm\sqrt{\dot{x}_t^f + \dot{y}_t^f} \quad and \quad \omega_t^f = \frac{\ddot{y}_t^f \dot{x}_t^f - \ddot{x}_t^f \dot{y}_t^f}{(\dot{x}_t^f)^2 + (\dot{y}_t^f)^2} \tag{3.2}$$

Here the ω_t^f is derived through defining θ_t^f as;

$$\theta_t^f = atan2(\dot{y}_t^f, \dot{x}_t^f) + k\pi \qquad k = 0, 1 \tag{3.3}$$

$k = 0$ for forward motion and $k = 1$ for backward motion respectively.

It is found that the angular velocity of the leader ω_t is same as the desired angular velocity of any virtual follower. In order to directly use the angular velocity of the leader robot ω_t as the desired velocity of the virtual follower, we prove below that the definition of θ_t^f in Eq.(3.3) is equal to the orientation of the leader robot θ_t.

Theorem: Given an initial posture $[x_0^s \quad y_0^s]^T$ and a desired trajectory $[x_t^f \quad y_t^f]^T$ at time t, there is a unique associated state trajectory $q_t^f = [x_t^f \quad y_t^f \quad \theta_t^f]^T$ which can

be computed algebraically as,

$$\theta_t = \theta_t^f = atan2(\dot{y}_t^f, \dot{x}_t^f) + k\pi \qquad k = 0, 1 \tag{3.4}$$

$k = 0$ for forward motion and $k = 1$ for backward motion respectively.

proof: Through differentiation of $\theta_t^f = atan2(\dot{y}_t^f, \dot{x}_t^f) + k\pi$ of Eq.(3.4), we get,

$$\omega_t^f = \frac{\ddot{y}_t^f \dot{x}_t^f - \ddot{x}_t^f \dot{y}_t^f}{(\dot{x}_t^f)^2 + (\dot{y}_t^f)^2} \tag{3.5}$$

Assuming o_x and o_y are constants with $\dot{v}_t$ and $\dot{\omega}_t$ both being zero, substitution of the values of Eq.(3.1) and its differentiated values of $\ddot{x}_t^f, \ddot{y}_t^f, \dot{x}_t^f$ and $\dot{y}_t^f$ in Eq.(3.5) results in,

$$\omega_t^f = \omega_t \frac{((o_x^2 + o_y^2)\omega_t^2 - 2o_y v_t \omega_t + v_t^2)}{((o_x^2 + o_y^2)\omega_t^2 - 2o_y v_t \omega_t + v_t^2)} = \omega_t \quad with \quad v_t \neq 0 \tag{3.6}$$

Where v_t and ω_t are the linear and angular velocities of the leader robot respectively. Hence $\int \omega_t^f dt = \int \omega_t dt = \theta_t^f = \theta_t$. Note that ω_t^f is not defined for when $v_t = 0$. The only time the follower experiences $v_t = 0$ is when the leader robot is at rest. In order to overcome this discontinuity at $v_t = 0$, we propose that the follower may switch to a similar posture stabilization control routine at $v_t = 0$, to move to the desired pose with respect to the leader $\square$.

Hence the feed forward commands developed are,

$$v_t^f = \pm\sqrt{\dot{x}_t^f + \dot{y}_t^f} \quad and \quad \omega_t^f = \omega_t = \frac{\ddot{y}_t^f \dot{x}_t^f - \ddot{x}_t^f \dot{y}_t^f}{(\dot{x}_t^f)^2 + (\dot{y}_t^f)^2} \tag{3.7}$$

Trajectory tracking needs to combine the feed forward commands generated in this section with an action on the feedback error. Below, we describe two such existing

controllers which achieve this objective using the feed forward commands generated here. The resulting action from these controllers will be formation maintenance for nonholonomic mobile robots.

3.2.2 Approximate linearization based formation controller

If the state tracking error is defined as in [23],

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos\theta_t^s & \sin\theta_t^s & 0 \\ -\sin\theta_t^s & \cos\theta_t^s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_t^f - x_t^s \\ y_t^f - y_t^s \\ \theta_t^f - \theta_t^s \end{pmatrix} \tag{3.8}$$

Where $[x_t^f, y_t^f, \theta_t^f]$ is the desired pose and $[x_t^s, y_t^s, \theta_t^s]$ is the actual follower pose at time t. Through a nonlinear transformation of the velocity inputs of the follower, the new velocity commands $[v_t^n\ \omega_t^n]$ have the following relationship. (v_t^s, ω_t^s) is the linear and angular velocity of the follower robot respectively.

$$\begin{pmatrix} v_t^s \\ \omega_t^s \end{pmatrix} = \begin{pmatrix} v_t^f \cos e_3 - v_t^n \\ \omega_t^f - \omega_t^n \end{pmatrix} \tag{3.9}$$

Then the error dynamics become,

$$\dot{e} = \begin{pmatrix} 0 & \omega_t^f & 0 \\ -\omega_t^f & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} e + \begin{pmatrix} 0 \\ \sin e_3 \\ 0 \end{pmatrix} v_t^f + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_t^n \\ \omega_t^n \end{pmatrix} \tag{3.10}$$

Through linearizing Eq.(3.10) around the reference trajectory one obtains a *linear time varying system.* If a linear feedback law is defined as in [23]:

$$\begin{pmatrix} v_t^n \\ \omega_t^n \end{pmatrix} = \begin{pmatrix} -k_1 e_1 \\ -k_2 sign(v_t^f) e_2 - k_3 e_3 \end{pmatrix} \tag{3.11}$$

Where the choice of gains is (see [23]),
$k_1 = k_3 = 2c_1\sqrt{(v_t^f)^2 + (\omega_t^f)^2}$, $k_2 = c_2 \mid v_t^f \mid$ where $c_1 \in (0,1)$ and $c_2 > 0$, one can substitute the controls of Eq.(3.11) to the linearized system around the desired trajectory of Eq.(3.10) to obtain,

$$\begin{aligned} v_t^s &= v_t^f \cos(\theta_t^f - \theta_t^s) + k_1((x_t^f - x_t^s)\cos\theta_t^s + (y_t^f - y_t^s)\sin\theta_t^s) \\ \omega_t^s &= \omega_t^f + k_2 sign(v_t^f)((y_t^f - y_t^s)\cos\theta_t^s - (x_t^f - x_t^s)\sin\theta_t^s) + k_3(\theta_t^f - \theta_t^s) \end{aligned} \tag{3.12}$$

$(x_t^f, y_t^f, \theta_t^f)$ is the desired pose and $(x_t^s, y_t^s, \theta_t^s)$ is the current follower pose in Euclidean $SE(2)$ coordinate system at time t. (v_t^f, ω_t^f) is the desired velocity at time t and (v_t^s, ω_t^s) is the follower robot velocity input at time t. With these control signals and the feed-forward desired velocities and desired pose generated in section "Feed-Forward Command Generation" we simulate and run 5 P3AT type robots with a designated leader to desired formations. Afterwards these controls will be implemented in 2 physical P3AT mobile robots, one being the leader and the other, the follower.

Simulation Results

The simulation uses 5, P3AT type mobile robots as followers with one designated P3AT leader robot. The gains for followers are taken as $c_1 = 0.9$ and $c_2 = 15$. The simulation spans 4 different courses for the leader robot.

- leader moves with constant (v_l^c, ω_l^c).
- changing angular velocities of the leader (v_l^c, ω_l^d) while keeping the linear velocity a constant.
- constant angular velocity with a changing linear velocity (v_l^d, ω_l^c).
- both linear and angular velocities are changing (v_l^d, ω_l^d)..

The norm of the formation errors and the quality of the driving inputs are considered for comparison criteria for the formation controllers developed in this section. The formation geometry described above, in terms of offsets of o_x and o_y in the respective X and Y leader robot coordinate system from its origin is converted to a new polar geometric system(Fig. 3.2) for comparison ease with the remaining controllers. Thus the formation geometry can be described by,

- Distance from the leader to the follower: d_{ls}
- Orientation of the follower's location in the leader robot coordinate system: β_{ls}
- Relative orientation difference of the leader and follower: θ_{ls}

The formation geometry from the above variables can be calculated as,

$d_{ls} = \sqrt{(x_l - x_s)^2 + (y_l - y_s)^2}$

$\beta_{ls} = -\theta_l + \pi + atan2(y_l - y_s, x_l - x_s)$

$\theta_{ls} = \theta_l - \theta_s$

subscript l stands for the leader and s stands for the follower. (x_l, y_l, θ_l) is the pose of the leader while (x_s, y_s, θ_s) is the pose of the follower in the Euclidean $SE(2)$ coordinate system. Hence the formation errors can be described as $e_d = d^d - d_{ls}$, $e_\beta = \beta^d - \beta_{ls}$ and $e_\theta = \theta^d - \theta_{ls}$. (d^d, β^d, θ^d) are the desired formation values in the new polar coordinate system. An example velocity course with a constant linear velocity and a changing angular velocity of the leader is shown in Fig. 3.3, while the resulting formation errors by the application of approximate linearization based formation control are depicted in Fig. 3.4. For clarity of images, we only depict the formation errors of two followers out of 5 followers in the simulation. The follower-1's desired formation geometry is $(d_{ls}^{desired} = 1\text{m}, \beta_{ls}^{desired} = \frac{\pi}{2}\text{rads}, \theta_{ls} = 0\text{rads})$ and the other depicted follower-4's desired formation variables values are $(d_{ls}^{desired} = \sqrt{8}\text{m}, \beta_{ls}^{desired} = \frac{-2\pi}{3}\text{rads}, \theta_{ls} = 0\text{rads})$. It is observed that the distance-d_{ls} and the bearing-β_{ls} errors converge almost to zero, while the relative orientation difference-θ_{ls} error stays small

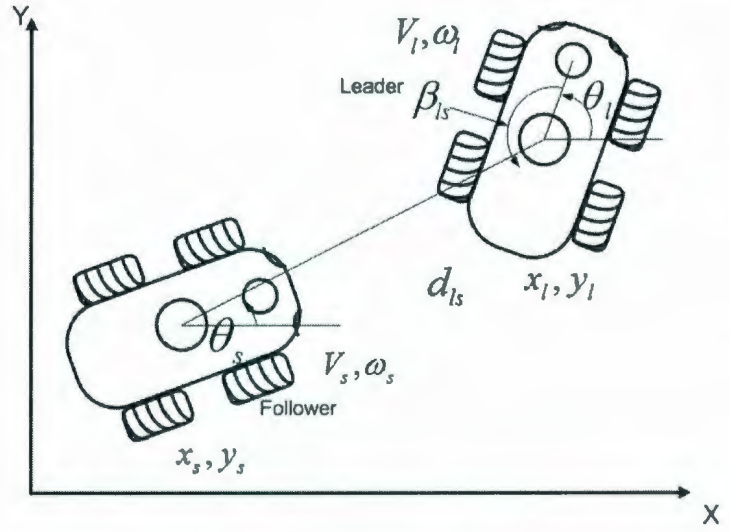


Figure 3.2: Formation geometry in the new coordinates system

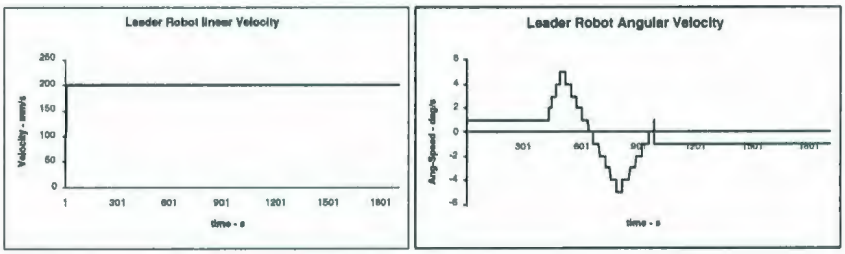


Figure 3.3: Example velocity course for the leader robot (v_l^c, ω_l^d)

bounded around zero. When the desired bearing is $\pm\frac{\pi}{2}$, the relative orientation error θ_{ls} goes almost around zero while for other desired bearing values, the θ_{ls} error stays bounded around zero. It can be attributed to the fact that the generated trajectory is very consistent with the leaders path and its velocities for a follower robot whose bearing is $\pm\frac{\pi}{2}$, while for other bearing values the generated trajectory from the above feed-forward command generation is roughly consistent. The velocity profiles for these two followers are depicted in Fig. 3.5.

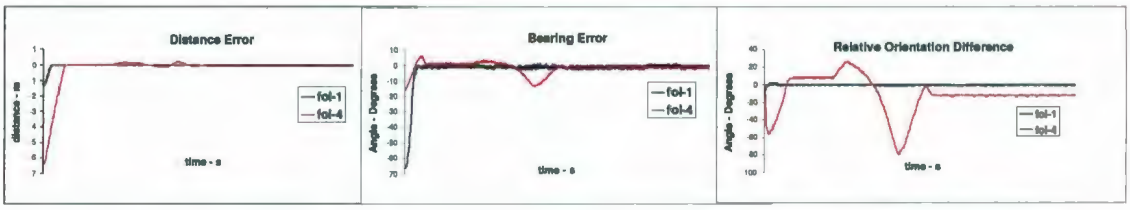


Figure 3.4: Formation errors for two follower robots:Blue color represents follower robot-1's and the red color represents follower robot-2's formation errors

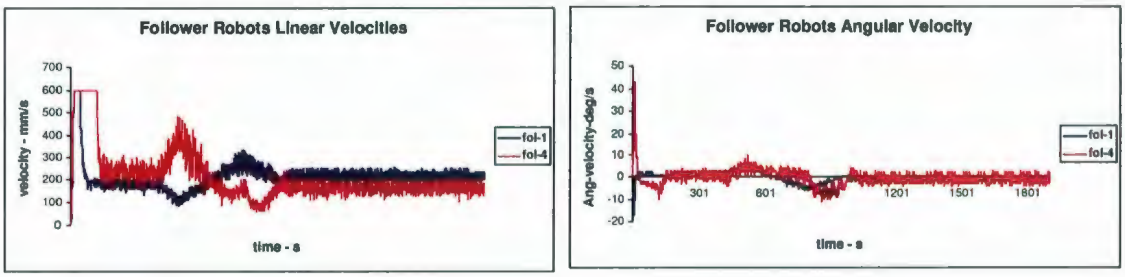


Figure 3.5: Linear and angular velocities for the two follower robots:Blue color represents follower robot-1 and the red color represents follower robot-2

3.2.3 Lyapunov function based nonlinear formation controller:

For the same error dynamics given in Eq.(3.10), If the linear and angular velocity controls are defined as in [22],

$$\begin{pmatrix} v_t^n \\ \omega_t^n \end{pmatrix} = \begin{pmatrix} -k_1(v_t^f, \omega_t^f)e_1 \\ -\bar{k}_2 v_t^f \frac{\sin e_3}{e_3} e_2 - k_3(v_t^f, \omega_t^f)e_3 \end{pmatrix} \tag{3.13}$$

(v_t^s, ω_t^s) are the linear and angular velocities of the follower robot. (v_t^f, ω_t^f) are the desired linear and angular velocities. $\bar{k}_2 > 0$. $k_1(v_t^f, \omega_t^f)$ and $k_3(v_t^f, \omega_t^f)$ are positive continuous gain functions. (e_1, e_2, e_3) are as in Eq.(3.10). Equation 3.13 becomes a controller based on a Lyapunov function of,

$V = \frac{\bar{k}_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2}$ with $\dot{V} = -k_1\bar{k}_2 e_1^2 - k_3 e_3^2 \leq 0$

When v_t^f, ω_t^f and its derivatives are bounded and if $v_t^f \nrightarrow 0$ and $\omega_t^f \nrightarrow 0$ as $t \rightarrow \infty$,

the above control in Eq.(3.13) globally asymptotically stabilizes the origin $e \equiv 0$ [22]. Using similar choices of the gains as in the approximate linearized formation controller, If we choose, $k_1 = k_3 = 2c_1\sqrt{(v_t^f)^2 + (\omega_t^f)^2}$ where $c_1 \in (0,1)$ and $k_2 = b > 0$, the application of the controls of Eq.(3.13) to the error dynamics of Eq.(3.10) results in,

$$\begin{aligned} v_t^s &= v_t^f \cos(\theta_t^f - \theta_t^s) + k_1((x_t^f - x_t^s)\cos\theta_t^s + (y_t^f - y_t^s)\sin\theta_t^s) \\ \omega_t^s &= \omega_t^f + k_2 v_t^f \tfrac{\sin(\theta_t^f - \theta_t^s)}{\theta_t^f - \theta_t^s}((y_t^f - y_t^s)\cos\theta_t^s - (x_t^f - x_t^s)\sin\theta_t^s) + k_3(\theta_t^f - \theta_t^s) \end{aligned} \tag{3.14}$$

$(x_t^f, y_t^f, \theta_t^f)$ is the desired pose and $(x_t^s, y_t^s, \theta_t^s)$ is the current follower pose in Euclidean $SE(2)$ coordinate system at time t. (v_t^f, ω_t^f) is the desired velocity at time t and (v_t^s, ω_t^s) is the follower robot velocity input at time t.

Simulation Results

The simulation is again carried out with five P3AT type mobile robots for the same path courses of the leader as above in "Approximate linearization based formation control" with the same starting positions for the followers and the leader and with gains of: $c_1 = 0.9$ and $c_2 = 15$. Again for clarity of images, we only depict the formation errors of two followers out of five followers in the simulation in Fig. 3.6. The follower-1's desired formation geometry again is $(d_{ls}^{desired} = 1\text{m}, \beta_{ls}^{desired} = \frac{\pi}{2}\text{rads}, \theta_{ls} = 0\text{rads})$ and the other depicted follower-4's desired formation variables values are $(d_{ls}^{desired} = \sqrt{8}\text{m}, \beta_{ls}^{desired} = \frac{-2\pi}{3}\text{rads}, \theta_{ls} = 0\text{rads})$. It is again observed that

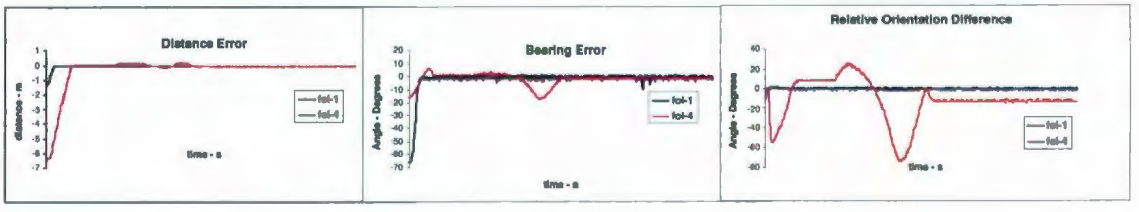


Figure 3.6: Formation errors for two follower robots with nonlinear control:Blue color represents follower robot-1's and the red color represents follower robot-2's formation errors

the distance-d_{ls} and the bearing-β_{ls} errors converge almost to zero, while the relative orientation difference-θ_{ls} error stays small bounded around zero. It is seen that when the desired bearing is $\pm\frac{\pi}{2}$, the relative orientation error θ_{ls} again goes almost around zero while for other desired bearing values, the θ_{ls} error stays bounded around zero. The scenario is similar to the case with "Approximate linearization based formation control" above. Since the same feed forward command generation generates the desired trajectory and the desired velocity profiles, the logic of consistent trajectories for bearings of $\pm\frac{\pi}{2}$ and rough consistent trajectories for all other bearing values will govern the behavior of relative orientation difference error over time. The velocity profiles for these two followers with nonlinear control are depicted in Fig. 3.7.

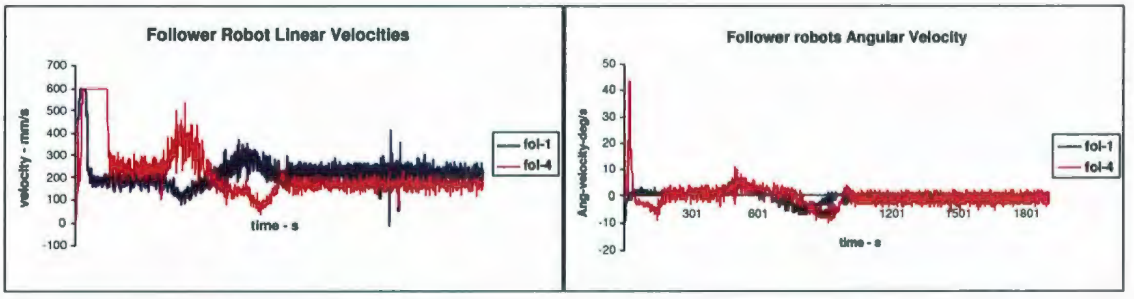


Figure 3.7: Velocity profiles of the two follower robots with nonlinear control::Blue color represents follower robot-1 and the red color represents follower robot-2

In the trajectory tracking type formation controllers the feed-forward command generation happens to be the same while the feed back action on the error is implemented through the approximate linearization of unicycle dynamics and again with a Lyapunov based nonlinear design. The relative orientation error stays bounded for all the bearing values except for $\pm\frac{\pi}{2}$ where the error goes to zero. That is attributed to the fact of consistent and roughly consistent trajectories generated through feed-forward command generation. Another significant conclusion is that as the desired distance of the follower gets farther and farther away from the leader robot, any small sudden rotational displacement of the leader requires a much larger displacement of

the followers' positions where the demanded linear and angular velocities for such manoeuvres fall outside the velocity limits of a real robot.

3.3 Static and Dynamic feedback linearized formation controllers

This section presents two other formation controllers whose error coordinates are transformed to a new coordinate system as shown in Fig. 3.8. The static feedback linearized formation controller is based on [4] and the dynamic feedback linearized controller is developed in this research initially (later it is found that the controller has already been formulated in [43]). The formation can be described by,

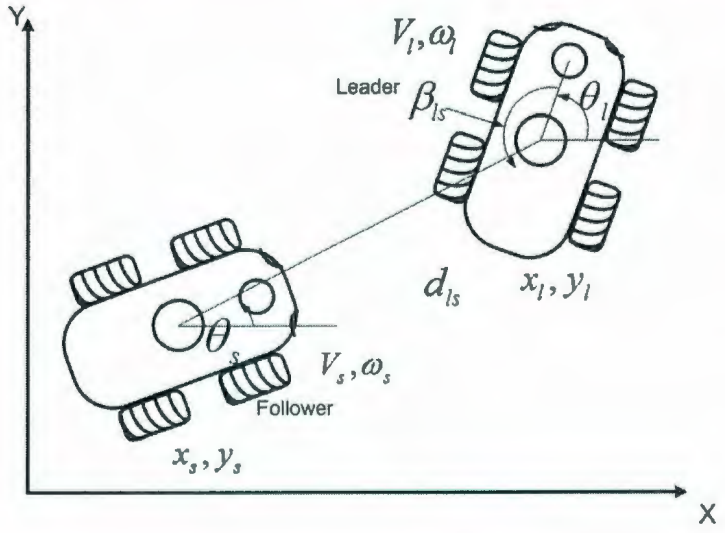


Figure 3.8: Formation Controller in new coordinate system

$$
\begin{aligned}
d_{ls} &= \sqrt{(x_l - x_s)^2 + (y_l - y_s)^2} \\
\beta_{ls} &= -\theta_l + \pi + atan2(y_l - y_s, x_l - x_s) \\
\theta_{ls} &= \theta_l - \theta_s
\end{aligned}
$$

Differentiation of these formation variables results in,

$$\begin{pmatrix} \dot{d}_{ls} \\ \dot{\beta}_{ls} \\ \dot{\theta}_{ls} \end{pmatrix} = \begin{pmatrix} \cos\gamma_{ls} & 0 \\ \frac{-\sin\gamma_{ls}}{d_{ls}} & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} v_s \\ \omega_s \end{pmatrix} + \begin{pmatrix} -\cos\beta_{ls} & 0 \\ \frac{\sin\beta_{ls}}{d_{ls}} & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_l \\ \omega_l \end{pmatrix} \tag{3.15}$$

where $\theta_{ls} = \theta_l - \theta_s$ is the relative orientation between the leader and follower. $\gamma_{ls} = \theta_{ls} + \beta_{ls}$, while $u_l = [v_l \quad \omega_l]$ is the exogenous input by the leader robot to the system. $u_s = [v_s \quad \omega_s]$ is the follower's driving inputs. The decoupling matrix (decouples control variables from state variables) in this context is singular.

3.3.1 Static feedback linearized formation controller

By performing a change on the current output state vector of the follower (changes the current output state by an offset valued vector to refer to another location on the robot except the origin) [32], [36], [34], we get,

$$\begin{pmatrix} x_s^n \\ y_s^n \end{pmatrix} = \begin{pmatrix} \cos\theta_s & -\sin\theta_s \\ \sin\theta_s & \cos\theta_s \end{pmatrix} \begin{pmatrix} o_x \\ o_y \end{pmatrix} + \begin{pmatrix} x_s \\ y_s \end{pmatrix} \tag{3.16}$$

o_x and o_y are offsets from the *origin* of the follower robot-coordinate system in X_S and Y_S directions respectively. And (x_s, y_s) are the current output state vector coordinates in the global-coordinate system while (x_s^n, y_s^n) are the newest output state vector coordinates in the global-coordinate system. The resulting new formation variables are given below.

$$\begin{aligned} d_{ls} &= \sqrt{(x_l - x_s^n)^2 + (y_l - y_s^n)^2} \\ \beta_{ls} &= -\theta_l + \pi + atan2(y_l - y_s^n, x_l - x_s^n) \\ \theta_{ls} &= \theta_l - \theta_s \end{aligned}$$

For simplicity we make the offset $o_y = 0$ in Eq.(3.16). Hence the new measured outputs x and y happen to lie on the X axis of the follower robot coordinate system with an offset of o_x units from it's origin. Differentiation of the newer output coordinates with respect to time results in a nonsingular dynamic system which is readily controllable [4].

$$\begin{pmatrix} \dot{d}_{ls} \\ \dot{\beta}_{ls} \\ \dot{\theta}_{ls} \end{pmatrix} = \begin{pmatrix} \cos\gamma_{ls} & o_x \sin\gamma_{ls} \\ \frac{-\sin\gamma_{ls}}{d_{ls}} & \frac{o_x \cos\gamma_{ls}}{d_{ls}} \\ 0 & -1 \end{pmatrix} \begin{pmatrix} v_s \\ \omega_s \end{pmatrix} + \begin{pmatrix} -\cos\beta_{ls} & 0 \\ \frac{\sin\beta_{ls}}{d_{ls}} & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_l \\ \omega_l \end{pmatrix} \quad (3.17)$$

By applying nonlinear static feedback linearization, the inputs of the follower robot are given by,

$$\mathbf{u_s} = G_1^{-1}(\, k(z_{ls}^d - z_{ls}) - F_1 \mathbf{u_l}) \quad (3.18)$$

Where $z_{ls} = [d_{ls} \quad \beta_{ls}]^T$ is the system output, $k = [k_1 \quad k_2]^T > 0$ are the controller gains, while $z_{ls}^d = [d_{ls}^d \quad \beta_{ls}^d]^T$ are the desired relative distance and bearing of the follower robot from the leader robot. $u_l = [v_l \quad \omega_l]$ is the exogenous input by the leader robot to the system while $u_s = [v_s \quad \omega_s]$ is the follower's driving inputs. F_1 and G_1 are given by,

$$G_1 = \begin{pmatrix} \cos\gamma_{ls} & o_x \sin\gamma_{ls} \\ \frac{-\sin\gamma_{ls}}{d_{ls}} & \frac{o_x \cos\gamma_{ls}}{d_{ls}} \end{pmatrix}, \qquad F_1 = \begin{pmatrix} -\cos\beta_{ls} & 0 \\ \frac{\sin\beta_{ls}}{d_{ls}} & -1 \end{pmatrix}$$

It has been proved in [4] that by applying the above follower controls, the system outputs $[d_{ls} \quad \beta_{ls}]$ exponentially converge to the desired values of $[d_{ls}^d \quad \beta_{ls}^d]^T$. And by using theory of perturbed systems [32], [36] it has also been proved that $\| \theta_{ls} \| \leq \delta$ for small $\delta \geq 0$ as $t \to \infty$. Another significant fact is that when $[d_{ls} \quad \beta_{ls}]$ converge to the desired values, it means that the new output state vector calculated with an offset of o_x units from the origin converges to the desired location, but not the origin of the robot coordinate system.

Simulation Results

The simulation, equivalent to the ones already performed for the previous controllers in this chapter is yet again performed for this controller with gains of: $k_1 = 0.9$ and $k_2 = 0.9$. The formation errors for two followers out of five followers in the simulation is shown in Fig. 3.9. The follower-1's desired formation geometry again is ($d_{ls}^{desired} = 1\text{m}, \beta_{ls}^{desired} = \frac{\pi}{2}\text{rads}, \theta_{ls} = 0\text{rads}$) and the other depicted follower-4's desired formation variables values are ($d_{ls}^{desired} = \sqrt{8}\text{m}, \beta_{ls}^{desired} = \frac{-2\pi}{3}\text{rads}, \theta_{ls} = 0\text{rads}$). The velocity

Figure 3.9: Formation errors for two follower robots with static feedback linearized control:Blue color represents follower robot-1's and the red color represents follower robot-2's formation errors

profiles for these two followers with nonlinear static feedback linearization are depicted in Fig. 3.10. As can be seen from Fig. 3.9, relative orientation error stays bounded as

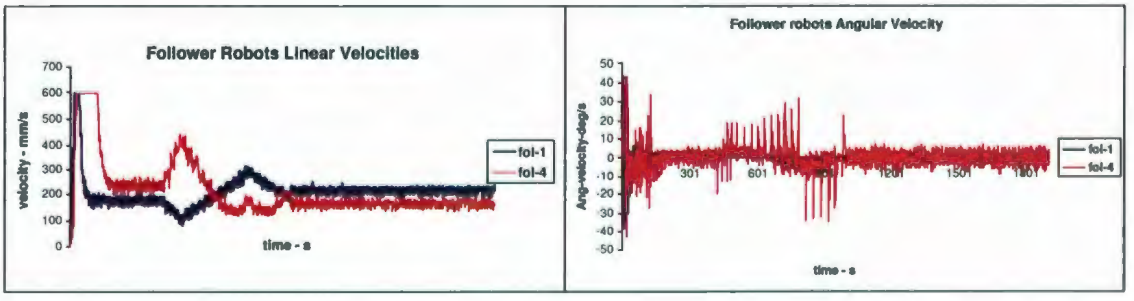


Figure 3.10: Velocity profiles for the two followers with static feedback linearized control:Blue color represents follower robot-1 and the red color represents follower robot-2

proved in [4]. The shown course of the leader is obtained through changing angular velocity while keeping the linear velocity at a constant. Angular velocity of the follower-4 seems quite noisy when compared to the angular velocities generated from previous controllers for the same follower.

3.3.2 Dynamic feedback linearized formation controller

The singularity of the decoupling matrix of 3.15 can be removed through the dynamic extension [32], [36], [34] such that v_s of the follower robot is taken as a dynamic state of the system. The first integrator of v_s is taken as a control variable for the follower along side ω_s.

$$\begin{aligned} \xi_s &= v_s \\ \dot{\xi}_s &= a_s \end{aligned} \tag{3.19}$$

a_s is the linear acceleration of the follower and ξ_s is the dynamic extension to v_s. Substituting the new variables to 3.15 results in,

$$\begin{pmatrix} \dot{d}_{ls} \\ \dot{\beta}_{ls} \\ \dot{\theta}_{ls} \end{pmatrix} = \begin{pmatrix} \cos\gamma_{ls} & 0 \\ \frac{-\sin\gamma_{ls}}{d_{ls}} & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \xi_s \\ \omega_s \end{pmatrix} + \begin{pmatrix} -\cos\beta_{ls} & 0 \\ \frac{\sin\beta_{ls}}{d_{ls}} & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_l \\ \omega_l \end{pmatrix} \tag{3.20}$$

Differentiating 3.20 with respect to time results in,

$$\ddot{z}_{ls} = G_2(z_{ls}, \theta_{ls}, \xi_s)\hat{\mathbf{u}}_\mathbf{s} + F_2(z_{ls}, \theta_{ls}, \xi_s)\hat{\mathbf{u}}_\mathbf{l} + \mathbf{L} \tag{3.21}$$

Again $\dot{\theta}_{ls} = \omega_l - \omega_s$ and $z_{ls} = [d_{ls} \quad \beta_{ls}]^T$ is the system output and $\hat{u}_l = [a_l \quad \omega_l]$ are the exogenous input by the leader robot to the system where a_l is the linear acceleration of the leader and ω_l is the angular velocity of the leader. $\hat{u_s} = [a_s \quad \omega_s]$ is the follower's driving inputs and a_s is its linear acceleration while ω_s is the angular velocity. And G_2, F_2 and $\mathbf{L}$ are given as,

$$G_2 = \begin{pmatrix} \cos\gamma_{ls} & \xi_s \sin\gamma_{ls} \\ \frac{-\sin\gamma_{ls}}{d_{ls}} & \frac{\xi_s \cos\gamma_{ls}}{d_{ls}} \end{pmatrix}, \quad F_2 = \begin{pmatrix} -\cos\beta_{ls} & -\xi_s \sin\gamma_{ls} \\ \frac{\sin\beta_{ls}}{d_{ls}} & \frac{-\xi_s \cos\gamma_{ls}}{d_{ls}} \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} v_l \dot{\beta}_{ls} \sin\beta_{ls} - \xi_s \dot{\beta}_{ls} \sin\gamma_{ls} \\ \frac{\xi_s \dot{d}_{ls} \sin\gamma_{ls}}{d_{ls}^2} + \frac{v_l \dot{\beta}_{ls} \cos\beta_{ls}}{d_{ls}} - \frac{v_l \dot{d}_{ls} \sin\beta_{ls}}{d_{ls}^2} - \frac{\xi_s \dot{\beta}_{ls} \cos\gamma_{ls}}{d_{ls}} - \dot{\omega}_l \end{pmatrix}$$

Hence the singularity is overcome with the dynamic extension. Yet there exists a potential singularity, when $\xi_s = v_s = 0$. That is when the axel of the mobile robot is not moving. This is perhaps a structural singularity to nonholonomic unicycle type mobile robots [48]. In order to overcome this singularity, we use only a naive approach that resets the state of ξ_s once the velocity of the axel falls below a lower threshold. This can be established through imposing a constraint for the followers as : the linear velocity of the follower $v_s > ||v_s^{lower}||$ where v_s^{lower} is the lower threshold, a smaller positive value. If v_s falls in between $-v_s^{lower}$ and $+v_s^{lower}$ from any feedback controls, then we reset it to $-v_s^{lower}$ or $+v_s^{lower}$ depending on which side (negative or positive) the follower velocity decreased from. Thus it results in a bounded velocity input with isolated discontinuities with respect to time. By applying nonlinear dynamic feedback linearization to a closed loop system of Eq.(3.21), the control variables of the followers can be given by,

$$\hat{\mathbf{u}}_\mathbf{s} = G_2^{-1}(C - F_2\hat{\mathbf{u}}_\mathbf{l} - \mathbf{L}) \tag{3.22}$$

Here C is the auxiliary control input given by $C = [c_1 \quad c_2]^T$,

$$\ddot{z}_{ls} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \ddot{d}_{ls}^d + k_1(\dot{d}_{ls}^d - \dot{d}_{ls}) + k_2(d_{ls}^d - d_{ls}) \\ \ddot{\beta}_{ls}^d + k_1(\dot{\beta}_{ls}^d - \dot{\beta}_{ls}) + k_2(\beta_{ls}^d - \beta_{ls}) \end{pmatrix} \tag{3.23}$$

And $z_{ls}^d = [d_{ls}^d \quad \beta_{ls}^d]^T$ are the desired relative distance and bearing of the follower robot from the leader robot. (k_1, k_2, k_3, k_4) are controller gains. It is seen that by applying the inputs 3.22 to the closed loop system resulting from 3.23, the outputs $[d_{ls} \quad \beta_{ls}]$ exponentially converge to the desired values $[d_{ls}^d \quad \beta_{ls}^d]$. Hence in order to prove the given system is stable, it is sufficient show that the orientation error θ_{ls} remains

bounded as $t \rightarrow \infty$. Here we assume that, leader robots $v_l > 0$ and $\| \omega_l \| \leq W_{max}$. For the follower $v_{max} \geq v_s > v_s^{lower}$, $\| a_s \| \leq a_{max}$ and $\| \omega_s \| \leq W_{max}$. Hence by using the stability theory of perturbed systems [36], it can be shown that,

$$\| \theta_{ls} \| \leq \delta \text{ for small } \delta \geq 0 \text{ as } t \rightarrow \infty.$$

Thus the formation controller in (3.21,3.22) is stable.

Simulation Results

The same simulation performed for the controllers above is performed for this controller with gains of: $k_i = 0.9$ for $i = (1, .., 4)$. The formation errors over time for selected two followers out of five followers in the simulation is shown in Fig. 3.11. The follower-1's desired formation geometry again is ($d_{ls}^{desired} = 1\text{m}, \beta_{ls}^{desired} = \frac{\pi}{2}\text{rads}, \theta_{ls} = 0\text{rads}$) and the other depicted follower-4's desired formation variables values are ($d_{ls}^{desired} = \sqrt{8}\text{m}, \beta_{ls}^{desired} = \frac{-2\pi}{3}\text{rads}, \theta_{ls} = 0\text{rads}$). The initial starting velocities of both follower robots are 0.01m/s. The velocity profiles for these selected two

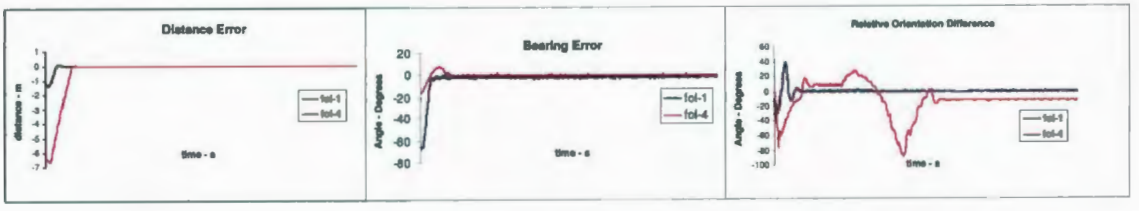


Figure 3.11: Formation errors for two follower robots with dynamic feedback linearized control:Blue color represents follower robot-1's and the red color represents follower robot-2's formation errors

followers with nonlinear dynamic feedback linearization are depicted in Fig. 3.12. Again it is seen that the relative orientation error stays bounded as proved in [4]. The shown velocity course of the leader is obtained through changing angular velocity while keeping the linear velocity at a constant. It is also observed that the linear and angular velocities are much smoother than the velocities obtained by the previous illustrated controllers.

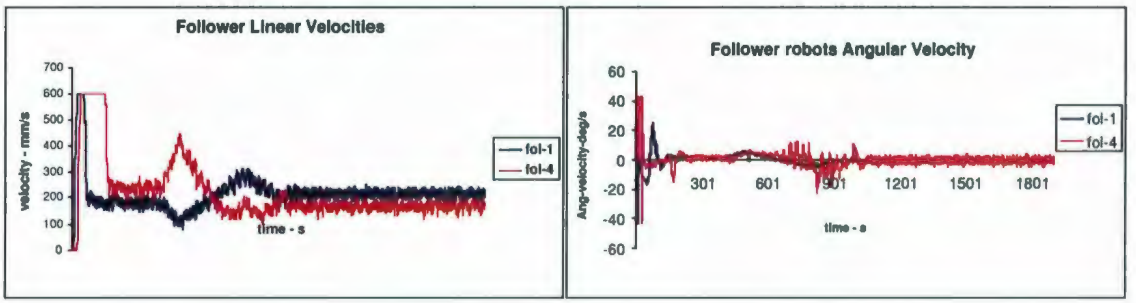


Figure 3.12: Velocity profiles for the two followers with dynamic feedback linearized control:Blue color represents follower robot-1 and the red color represents follower robot-2

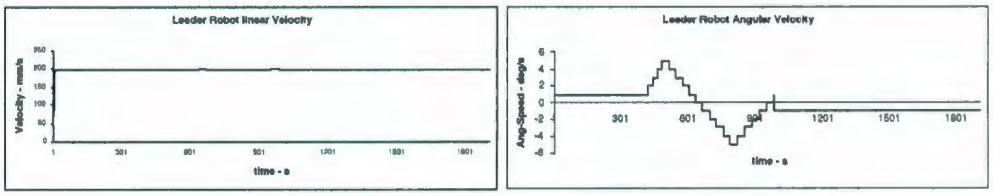
3.3.3 Formation Controller Comparison

For different courses of velocity profiles of the leader robot, five P3AT type follower robots are driven to desired formations using the different formation controllers explained earlier in this chapter. The starting 2-D $SE(2)$ poses and starting velocities of the follower robots are kept the same through out all simulations involving the different controllers and different courses of velocity profiles yielded by the leader robot. In all of these simulations, for the first $t_p = 400$ time units, the leader robot is driven with constant linear v_l and a constant angular ω_l velocity. The time above is more than enough to bring all the followers in to desired formations. Once t_p time is passed the **RMS** error of formation variables $(d_{ls}^{error}, \beta_{ls}^{error})$ and (θ_{ls}^{error}) are recorded for each robot in the formation. For each different velocity profiles of the leader robot, we calculated the **RMS** error of each formation variable per one follower for comparison purposes. Thus giving a holistic error indicator metric for the whole formation. The courses of velocity profiles of the leader in which the followers were run in to different geometric formations are given in Fig. 3.13. And the follower robots desired formation variables are given in Table 3.1 The different formation controllers implemented and simulated in this chapter are listed as controller 1 to 4. They are given as,

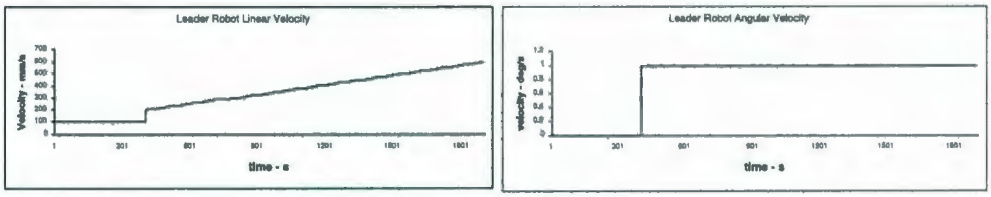
- controller 1 - Approximate linearized trajectory tracking type formation con-

	follower-1	follower-2	follower-3	follower-4	follower-5
d_{ls}^d	1	1	$\sqrt{8}$	$\sqrt{8}$	3
β_{ls}^d	$\frac{\pi}{2}$	$\frac{-\pi}{2}$	$\frac{3\pi}{4}$	$\frac{-3\pi}{4}$	π
θ_{ls}^d	0	0	0	0	0

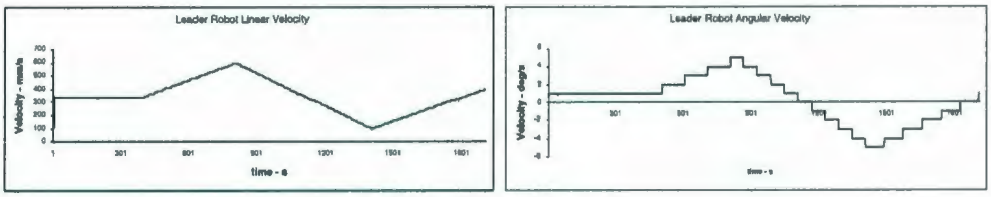
Table 3.1: Desired formation variable values for the 5 P3AT follower robots



3.13.1: velocity profile 1



3.13.2: velocity profile 2



3.13.3: velocity profile 3

Figure 3.13: Leader robot Velocity Profiles

troller

- controller 2 - Lyapunov based trajectory tracking type formation controller
- controller 3 - Static feedback linearized $l - \beta$ type formation controller
- controller 4 - Dynamic feedback linearized $l - \beta$ type formation controller

Fig. 3.14 shows that six, Pioneer-3-AT type mobile robots are run to formations with a designated leader in the middle, by the different formation controllers 1-4, developed. The given example figure of Fig. 3.14 is for a type-1 velocity profile of the leader. The

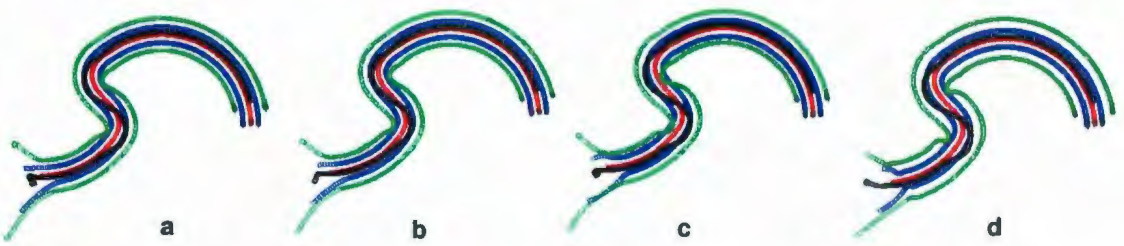


Figure 3.14: 5- P3AT follower robot formation drive with a P3AT leader robot using (a.)controller-1 (b.)controller-2 (c.)controller-3 (d.)controller-4

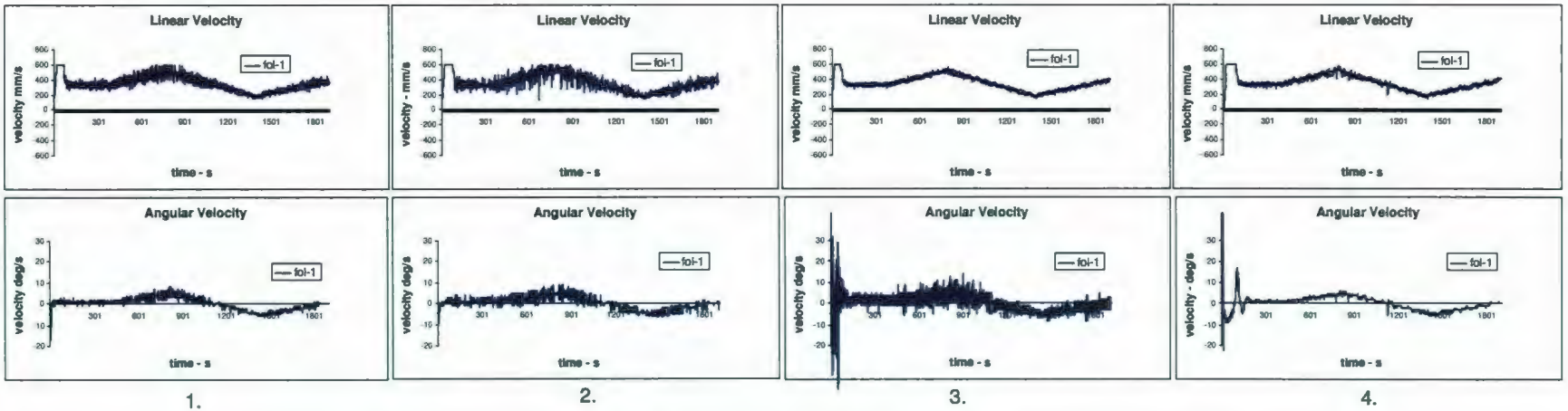
resulting formation RMS errors per follower robot per unit time are given in Table 3.2. Individual $d_{ls}^{error}, \beta_{ls}^{error}, \theta_{ls}^{error}$ as well as holistic $\frac{d_{ls}^{error}+\beta_{ls}^{error}+\theta_{ls}^{error}}{3}$ errors are given for each velocity profile of the leader. And lastly the average individual and holistic RMS error component are given for each controller. From these results it's sufficient

RMS errors	d_{ls}^{error}	β_{ls}^{error}	θ_{ls}^{error}	formation error
Leader velocity profile 1				
controller 1	0.0533	2.5741	18.3964	7.0080
controller 2	0.0718	3.2726	**18.2408**	7.1951
controller 3	**0.0128**	**0.8470**	19.0622	6.6407
controller 4	0.0166	1.0526	18.6247	**6.5646**
Leader velocity profile 2				
controller 1	0.0423	2.0728	**5.3498**	2.4883
controller 2	0.5493	3.8823	8.1171	4.1829
controller 3	**0.0284**	**1.4150**	5.5565	**2.3333**
controller 4	0.0361	1.8062	5.3780	2.4068
Leader velocity profile 3				
controller 1	0.3345	10.3393	24.1732	11.6157
controller 2	**0.3100**	12.0413	**22.1300**	**11.4938**
controller 3	0.6143	**8.9711**	24.9390	11.5081
controller 4	0.8511	10.3688	24.7626	11.9942
Average Error value for all the above velocity profiles				
controller 1	**0.1434**	4.9954	**15.9731**	7.0373
controller 2	0.3104	6.3987	16.1626	7.6240
controller 3	0.2185	**3.7444**	16.5192	**6.8274**
controller 4	0.3013	4.4092	16.2551	6.9885

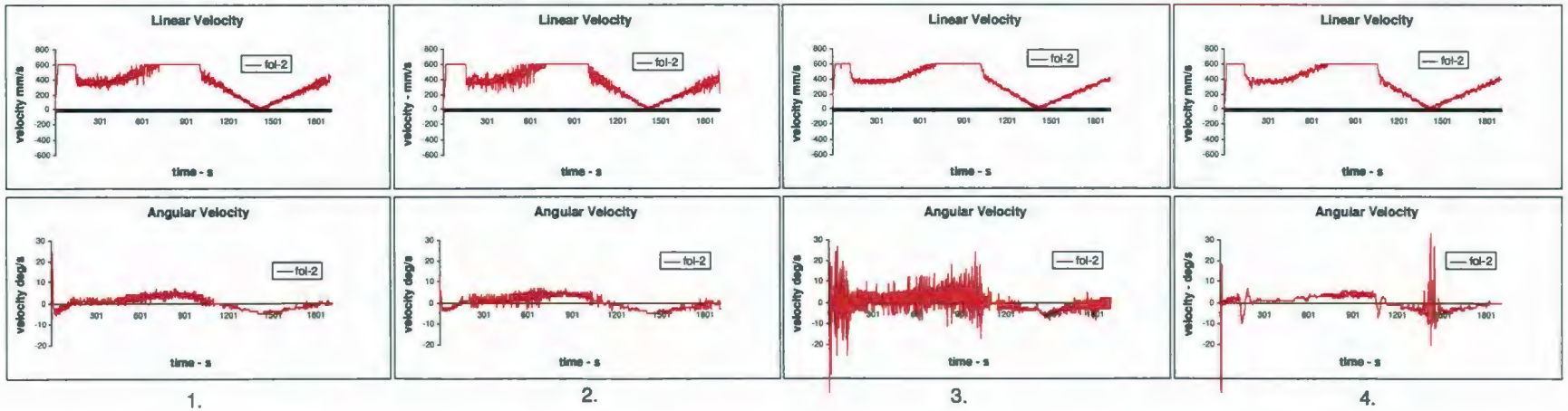
Table 3.2: RMS formation error values for the different formation controllers developed above

to conclude that the static feedback linearized controller outperforms it's counter-

parts with a narrow margin. It minimizes β_{ls}^{error} to a much lesser value than anyone else but suffers from a high θ_{ls}^{error} value more than the rendered respective value from other controllers. Another flaw of this controller is that the given formation control law does not stabilize the origin of the robot, instead an offset point from the origin. Thus the comparison of these values may be somewhat controversial. On the other hand, both approximate linearized and Lyapunov based nonlinear trajectory tracking type controllers keep the θ_{ls}^{error} at a possible minimum. The approximate linearized formation controller also tries to keeps d_{ls}^{error} at much more lower values throughout the simulation than the others. The Lyapunov based nonlinear formation controller has shown better performance with varying v_l and ω_l. The dynamic feedback linearized controller is next better to the static feedback linearized controller. It keeps pretty decent error ratings for each formation variable and for the average errors. If it had not been for the discontinuity at $v_s = 0$ the error ratings of this controller would have been more improved. These results are taken by setting the gains of the controllers to arbitrary values belonging to the regions of convergence in the selected controllers and through many trials with real world noises and communication delays (leader information is communicated to followers at each Δt time unit). Hence the comparison is not 100% accurate, but rather expressive, informative and somewhat reliable. From the velocity profiles of followers $1-5$ in figures 3.15 to 3.17, it can be concluded that both 1.) approximate feedback linearized, and 2.)nonlinear Lyapunov based controllers exert much oscillation in their respective linear velocities but quite stable in rendering the angular velocities. Static and dynamic feedback linearized controllers on the other hand renders a much smoother linear velocity profile for all the followers. But the static feedback linearized controller has more noisy angular velocity profile(highly oscillating), whereas the dynamic feedback linearized controller has a very smooth angular velocity profile across all followers, more smoother than its counterparts. (Note: follower 2 in controller- 4 of figure 3.16 angular velocity shoots

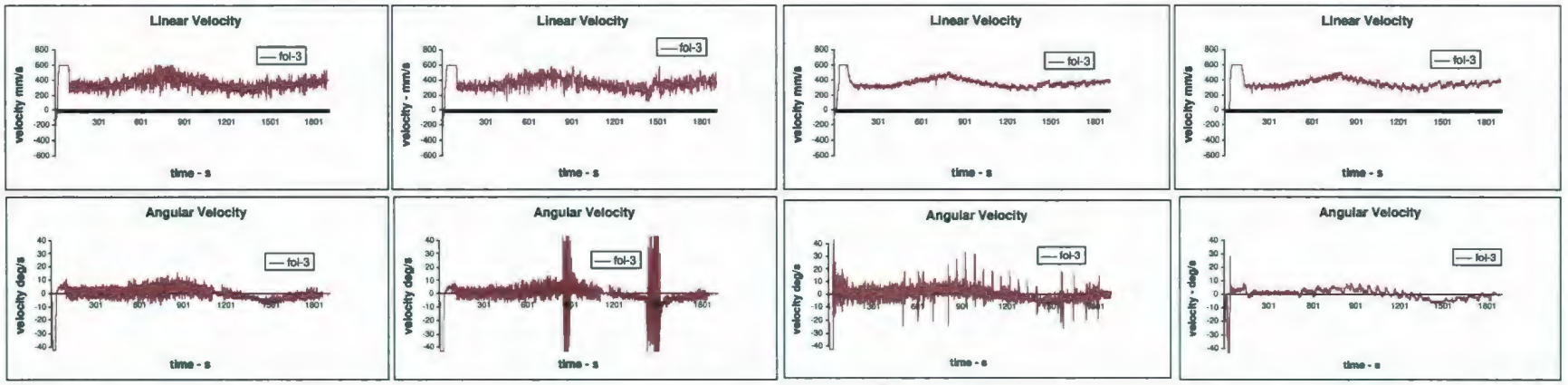


3.15.1: follower-1 velocity profile for (1.)controller-1 (2.)controller-2 (3.)controller-3 (4.)controller-4



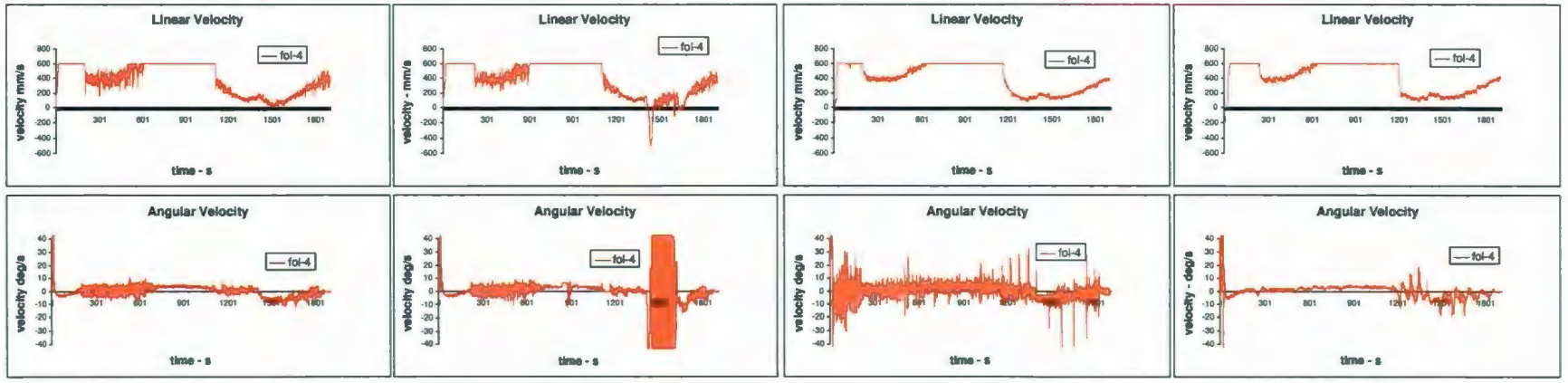
3.15.2: follower-2 velocity profile for (1.)controller-1 (2.)controller-2 (3.)controller-3 (4.)controller-4

Figure 3.15: Linear and Angular velocity profiles for Follower 1 and 2



1. 2 3 4

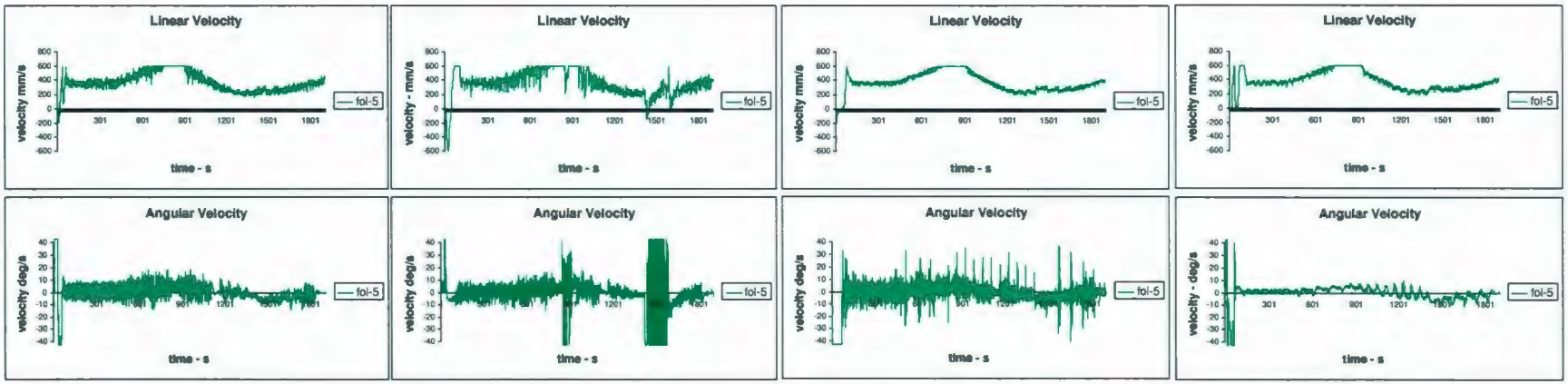
3.16.1: follower-3 velocity profile for (1.)controller-1 (2.)controller-2 (3.)controller-3 (4.)controller-4



1. 2 3 4

3.16.2: follower-4 velocity profile for (1.)controller-1 (2.)controller-2 (3.)controller-3 (4.)controller-4

Figure 3.16: Linear and Angular velocity profiles for Follower 3 and 4



1. 2. 3. 4.

3.17.1: follower-5 velocity profile for (1.)controller-1 (2.)controller-2 (3.)controller-3 (4.)controller-4

Figure 3.17: Linear and Angular velocity profiles for Follower 5

at around $t = 1600$. This happens when its linear velocity reaches zero - the singularity point and its linear velocity is reset through a naive approach as mentioned above under the dynamic feedback linearized controllers. Hence as a result of this discontinuity, the overshoot happens.). All in all considering all the given controllers, the dynamic feedback linearized controller outperforms all of its counterparts but suffers from the singularity when its linear velocity $v_s = 0$.

3.4 Summary

Leader follower based formation maintenance controllers were developed and compared with each other for formation error and input smoothness. Formation errors of these controllers for arbitrary velocity profiles of the leader robots were recorded with P3AT type robots. Results suggest that all of the controllers perform well, to keep the formation errors of nonholonomic robots as small as possible. The smoothness of the control variables (v_s, ω_s) for the follower robots, from these different controllers suggest that dynamic feedback linearized inserts much smoother controls than others, but suffers from a structural singularity when $v_s = 0$. The static feedback linearized formation controller has highly oscillating angular velocity, partly attributed to the inflexibility of the mechanical system to stabilize not the origin, but an offset point from the origin to desired values and partly due to the magnification of noises. The odometry an gyro readings calculate the state (x_i, y_i, θ_i) of the robot to the origin or to a fixed location in the robot frame. Any translation from these values to the new offset from the origin (static feedback linearization needs the output state vector to be at an offset from the origin) adds more noise to the new output state. For example assuming that only the measured θ_i is corrupted by noise, the translation of 3.24 with (x_i, y_i, θ_i) results in both (x_i^n, y_i^n) corrupted with noise too.

$$\begin{pmatrix} x_i^n \\ y_i^n \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} o_x \\ o_y \end{pmatrix} + \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{3.24}$$

Chapter 4

Leader Robot State Estimation by the Followers

About this chapter: This chapter explains possible leader-robot state (robot pose (x, y, θ) and linear and angular velocities) estimation techniques by the followers of the system without explicit communication. It is an effort to experimentally validate state estimation accuracies of different recursive Baysian estimation techniques for a variety of movements of the leader robot. The effect of different filter estimations towards the formation control of multiple nonholonomic mobile robots is also studied. The recursive Baysian state estimation filters are implemented through the REBEL Matlab interface [49] with real world noises in sensor observations and motion dynamics. They are tested against a hybrid formation controller developed by combining static and dynamic feedback linearized formation controllers of chapter 3 above. These controllers are combined in order to,

- Avoid the potential singularity of the dynamic feedback linearized formation controller when the follower robot's linear velocity is zero: $v_s = 0$
- To harness the better accuracy of dynamic feedback linearized formation controller (see results in chapter 3)

The hybrid formation controller is implemented in such a way that,

- When the follower velocity $v_s < ||v_{threshold}||$ for a small $v_{threshold}$, static feedback linearized formation controller is invoked for controlling the formation.
- When $v_s \geq ||v_{threshold}||$ dynamic feedback linearized formation controller is invoked to control the formation.

The $v_{threshold}$ is set to an arbitrary small value in the experiments run for this thesis. In fact $v_{threshold}$ must have been set by considering the dynamic properties such as accelerations, decelerations of the robot. The simulated results of state estimations by different recursive Baysian filters and formation maintenance accuracies through the application of these different state estimations are tabulated for comparison purposes. The key contributions of this chapter are,

- Experimental validation of state estimation accuracies of different recursive Baysian estimation techniques for a variety of movements of the leader robot. (Different filters include: extended Kalman filter-(EKF), unscented Kalman filter-(UKF), central difference Kalman filter-(CDKF), square root unscented Kalman filter-(SRUKF), square root central difference Kalman filter-(SRCDKF), sigma point particle filter-(SPPF)).
- Experimental validation of the effect of different state estimations towards formation maintenance problem.

4.1 Background

In order to reach desired formations, the followers need consistent data about their leader-robot's pose and velocities over the respective sampling time periods. It is noted in Chapter 1 that the use of communication to obtain such information is

subjected to noise, delay, interference and resource constraints and that such a communication makes the formation control solution a highly centralized one too. On the other hand obtaining local information through exteroceptive sensors of a robot is cheaper and faster than using communication [26]. Hence, If the leader robot's states can be measured or estimated remotely using these exteroceptive sensors of the robot, the leader-follower strategy can be made more decentralized. The state of the leader robot includes linear and angular velocities and Euclidean ($SE2$) pose: (x, y, θ). Measuring the linear and angular velocities and leader robot orientation under noisier sensor observations is challenging and perhaps impossible. Therefore decentralized state estimation is essential in estimating unknown states from the observations acquired from exteroceptive sensors local to the robots.

The use of EKF [4] and dual unscented Kalman filter (DUKF) [18] have been reported for decentralized leader robot state estimation in formation control. The experimental results recorded in [4] are for constant velocity profiles of the leader robot (e.g: leader robot's linear and angular velocity are kept constant in estimation). Similar experimental results of leader robot state estimation are shown in [18] where the angular velocity of the leader robot is kept at zero while changing the linear velocity. In reality linear and angular velocities of robots are both changing over time. Hence it is found that there is a lack of experimental validation of state estimation for possibly changing velocity profiles (linear and angular) of the leader robot. The effect of such estimations towards the formation control solutions provided have also not been experimentally evaluated. The extensive literature also reveals that there seems to be no benchmarking on the estimation accuracy of different recursive Baysian filters, which can be used for decentralized leader robot state estimation.

Although EKF based state estimation is the default solution for most state estimation

problems, it's solutions suffer from;

- Disregard of probabilistic spread of the system states and noises during initialization of system equations (linearization expands the distribution around only a single point)
- Taylor series expansion holds only for first order accuracy of mean and covariance of the distribution.

In order to overcome these shortcomings the idea is to use deterministic sampling approaches that circumvent the calculation of analytical derivatives. Filters which uses a deterministic sampling strategy are collectively known as Sigma-Point Kalman filters. Some examples include UKF, CDKF, SRUKF and SRCDKF. This chapter tries to experimentally validate the state estimation accuracies of the above filters especially for the formation control problem. The effect of different estimations (incurred through different estimation filters) towards the formation stability of the hybrid formation controller explained in "About this chapter" section is also investigated. Matlab REBEL filter interface is used to implement the different types of filters with the appropriate dynamic and observational models while a custom catered Matlab multi-robot simulation platform is built for multi robot simulation. For all of the simulations, noises are introduced to both observations and motion dynamics of the robots so as to reflect a real world scenario. The follower-robots are assumed to be localized to some degree of accuracy while the sampling time for the robots is taken as 0.5 *seconds*.

4.2 Estimation through Kalman type filters

Assuming that the leader robot's state evolves as a gaussian distribution, the use of EKF and its varieties, to estimate hidden states from the known observations are

exploited. The discrete dynamics of the leader robot evolve as,

$$\begin{pmatrix} \hat{v}^l_{k+1} \\ \hat{\omega}^l_{k+1} \\ \hat{x}^l_{k+1} \\ \hat{y}^l_{k+1} \\ \hat{\theta}^l_{k+1} \end{pmatrix} = \begin{pmatrix} v^l_k + p^v_n \times dt \\ \omega^l_k + p^\omega_n \times dt \\ x^l_k + v^l_k \times \cos\theta^l_k \times dt + \frac{1}{2}(p^v_n \times \cos\theta^l_k - v^l_k \times \omega^l_k \times \sin\theta^l_k) \times dt^2 \\ y^l_k + v^l_k \times \sin\theta^l_k \times dt + \frac{1}{2}(p^\omega_n \times \sin\theta^l_k + v^l_k \times \omega^l_k \times \cos\theta^l_k) \times dt^2 \\ \theta^l_k + \omega^l_k \times dt + \frac{1}{2} \times p^\omega_n \times dt^2 \end{pmatrix} \tag{4.1}$$

Superscript l stands for the leader robot. Subscript k stands for the current time step while $k+1$ stands for the immediate next time step: $k+1 = k+dt$ where dt is the sampling time period. v & ω are the linear and angular velocities of the robot while x, y & θ is the pose of the robot in the Euclidean-$SE2$ coordinate system. $[p^v_n \quad p^\omega_n]^T$ is the process noise with a covariance $[\sigma^v_n \quad 0; 0 \quad \sigma^\omega_n]$. The observation model includes a range and a bearing measurement to the leader taken from the follower and a measurement of the relative orientation difference between the leader and the follower. The measurement model including the relative orientation measurement is,

$$\begin{pmatrix} \hat{z}^1_{k+1} \\ \hat{z}^2_{k+1} \\ \hat{z}^3_{k+1} \end{pmatrix} = \begin{pmatrix} \sqrt{(\hat{x}^l_{k+1} - \hat{x}^f_{k+1})^2 + (\hat{y}^l_{k+1} - \hat{y}^f_{k+1})^2} + o^1_n \\ atan2(\hat{y}^l_{k+1} - \hat{y}^f_{k+1}, \quad \hat{x}^l_{k+1} - \hat{x}^f_{k+1}) + o^2_n \\ \hat{\theta}^l_{k+1} - \hat{\theta}^f_{k+1} + o^3_n \end{pmatrix} \tag{4.2}$$

Superscript f stands for the follower and $[o^1_n \quad o^2_n \quad o^3_n]^T$ is the observation noise with a covariance of $[\sigma^1_n \quad 0 \quad 0; 0 \quad \sigma^2_n \quad 0; 0 \quad 0 \quad \sigma^3_n]$.

4.3 Extended Kalman filter based state estimation

Using the robot dynamics of 4.1 applied to the EKF algorithm given in Appendix A (the variables of EKF algorithm are calculated as),

$$\mathbf{G}_{k+1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \cos\theta_k dt - \frac{1}{2}\omega_k \sin\theta_k dt^2 & -\frac{1}{2}v_k \sin\theta_k dt^2 & 1 & 0 & -v_k \sin\theta_k dt - \frac{1}{2}v_k\omega_k \cos\theta_k dt^2 \\ \sin\theta_k dt + \frac{1}{2}\omega_k \cos\theta_k dt^2 & \frac{1}{2}v_k \cos\theta_k dt^2 & 0 & 1 & v_k \cos\theta_k dt - \frac{1}{2}v_k\omega_k \sin\theta_k dt^2 \\ 0 & dt & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_{k+1} = \begin{pmatrix} dt & 0 \\ 0 & dt \\ \frac{1}{2}\cos\theta_k dt^2 & 0 \\ \frac{1}{2}\sin\theta_k dt^2 & 0 \\ 0 & \frac{1}{2}dt^2 \end{pmatrix} \begin{pmatrix} \sigma_n^v & 0 \\ 0 & \sigma_n^\omega \end{pmatrix} \begin{pmatrix} dt & 0 \\ 0 & dt \\ \frac{1}{2}\cos\theta_k dt^2 & 0 \\ \frac{1}{2}\sin\theta_k dt^2 & 0 \\ 0 & \frac{1}{2}dt^2 \end{pmatrix}^{\mathbf{T}}$$

$$\mathbf{H}_{k+1} = \begin{pmatrix} 0 & 0 & \frac{\hat{x}^l_{k+1} - \hat{x}^f_{k+1}}{d} & \frac{\hat{y}^l_{k+1} - \hat{y}^f_{k+1}}{d} & 0 \\ 0 & 0 & -\frac{\hat{y}^l_{k+1} - \hat{y}^f_{k+1}}{d^2} & \frac{\hat{x}^l_{k+1} - \hat{x}^f_{k+1}}{d^2} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \textit{and} \quad \mathbf{Q}_{k+1} = \begin{pmatrix} o_n^1 & 0 & 0 \\ 0 & o_n^2 & 0 \\ 0 & 0 & o_n^3 \end{pmatrix}$$

where $d = \sqrt{(\hat{x}^l_{k+1} - \hat{x}^f_{k+1})^2 + (\hat{y}^l_{k+1} - \hat{y}^f_{k+1})^2}$ while superscript l and f stands for the leader and follower respectively. Here $\mathbf{G}_{k+1}$ is the Jacobian of robot dynamics with respect to the robot pose. $\mathbf{H}_{k+1}$ is the Jacobian of robot-observations with respect to the robot pose. $\mathbf{R}_{k+1}$ and $\mathbf{Q}_{k+1}$ are the process and observation covariances. For more information about the extended Kalman filter see Appendix A (Extended Kalman filter).

Fig. 4.1 shows the EKF leader state estimation and its resulting formation error when applied to the hybrid formation controller (the hybrid formation controller is explained in the "about" section of this chapter). Fig. 4.1 shows that the linear velocity estimation is better than angular velocity estimation. Angular velocity estimation has a time lag as well as more distortions. The estimated values

for range and bearing measurements are well smoothed while the estimated relative orientation has some distortions. The results in Fig. 4.1 are taken with a process noise covariance of $[(0.003)^2 \quad 0;0 \quad (0.1\pi/180)^2]$ for linear and angular velocities respectively along the matrix diagonal and for an observation covariance of: $[(0.1)^2 \quad 0 \quad 0;0 \quad (5\pi/180)^2 \quad 0;0 \quad 0 \quad (5\pi/180)^2]$. Along the diagonal of this matrix being the variances of range, bearing and relative orientation measurements. The leader's state is estimated for a number of trials of different paths incurred through different velocity profiles. The average root mean square (RMS) errors of estimation are tabulated in Table 4.1 for comparison purposes. The different velocity profiles

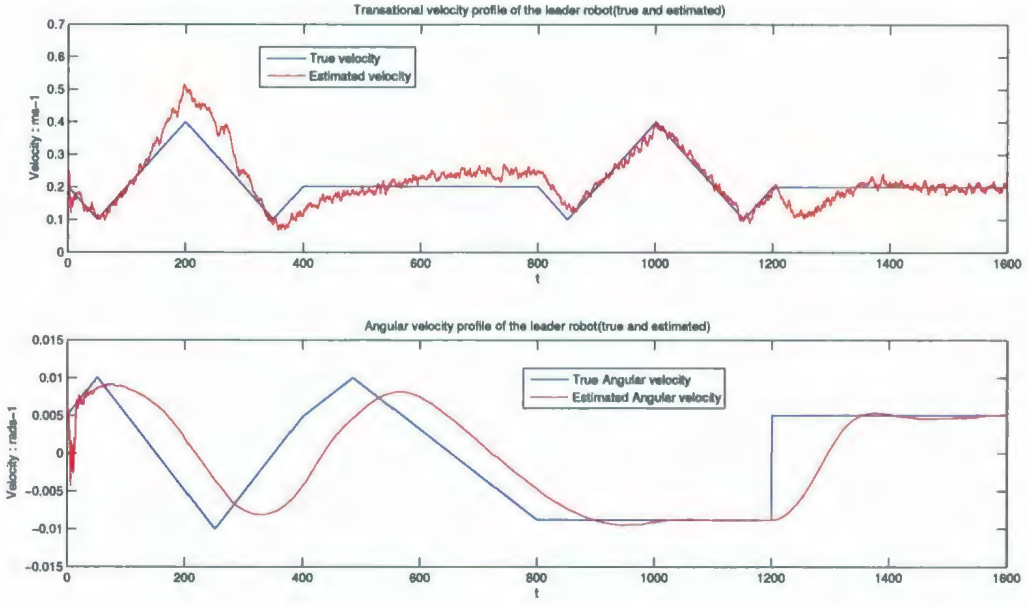
	$v^c\omega^c$	$v^d\omega^c$	$v^c\omega^d$	$v^d\omega^d$	Mix	$Average_{error}$
v_l^{rms}	9.1e-3	0.0136	0.0333	0.0604	0.0399	0.0313
ω_l^{rms}	5.136e-4	2.0036e-4	4e-3	6.2e-3	4.6e-3	3.1e-3
d_l^{rms}	0.0330	0.0355	0.1098	0.1789	0.1246	0.0964
β_l^{rms}	0.0170	0.0177	0.1899	0.3069	0.2161	0.15
θ_l^{rms}	6.3e-3	4.7e-3	0.1466	0.2170	0.1586	0.107

Table 4.1: RMS error of estimated and true state values of the lead robot

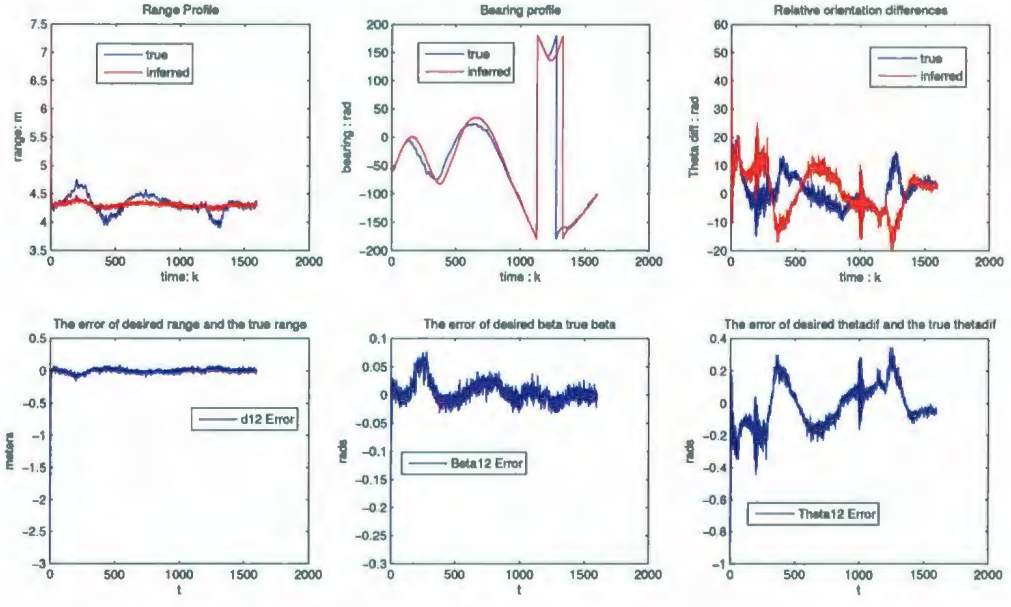
shown in Table. 4.1 are given below;

- $v^c\omega^c$ - constant linear and angular leader robot velocities
- $v^d\omega^c$ - constant angular but changing linear velocity of the leader robot
- $v^c\omega^d$ - constant linear but changing angular velocity of the leader robot
- $v^d\omega^d$ - linear and angular velocities of the leader robot are changing over time.
- Mix - All velocity profiles of the leader above are combined together

The average RMS estimation error is taken over averaging individual errors for different velocity profiles of the leader robot. The formation error propagation under EKF estimated leader robot state variables is also shown in Fig. 4.1.



4.1.1: true and estimated velocity profiles for the leader robot



4.1.2: True and estimated errors of the state and the formation error of the leader robot

Figure 4.1: EKF-State estimation with orientation measurement

4.4 Sigma Point Kalman filter based state estimation

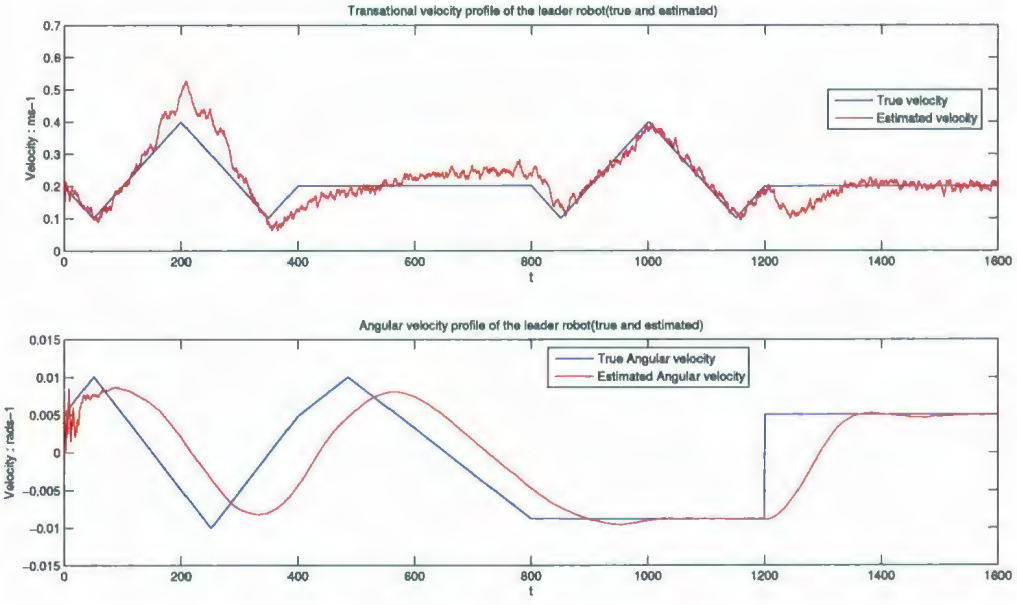
4.4.1 Unscented Kalman filter(UKF) based state estimation

The UKF is a sigma type Kalman Filter (See Appendix A - UKF for more details), which uses a deterministic sampling strategy where it carefully chooses only a sample points in the gaussian prior distribution that captures the true mean and the covariance of the entire distribution. When propagated through any nonlinear function the posterior distribution holds for third order accuracy in a Taylor series expansion whereas EKF only holds for the first order accuracy of mean and covariance of the same distribution. Hence UKF estimation is supposed to be more accurate than the EKF and have the same computational complexity too [50]. Fig. 4.2 shows the results in somewhat similar sense to the EKF results. But the accuracy is improved as reflected by the results of Table 4.2. The formation error propagation under UKF estimated leader robot state variables is also shown in Fig. 4.2. The results are taken

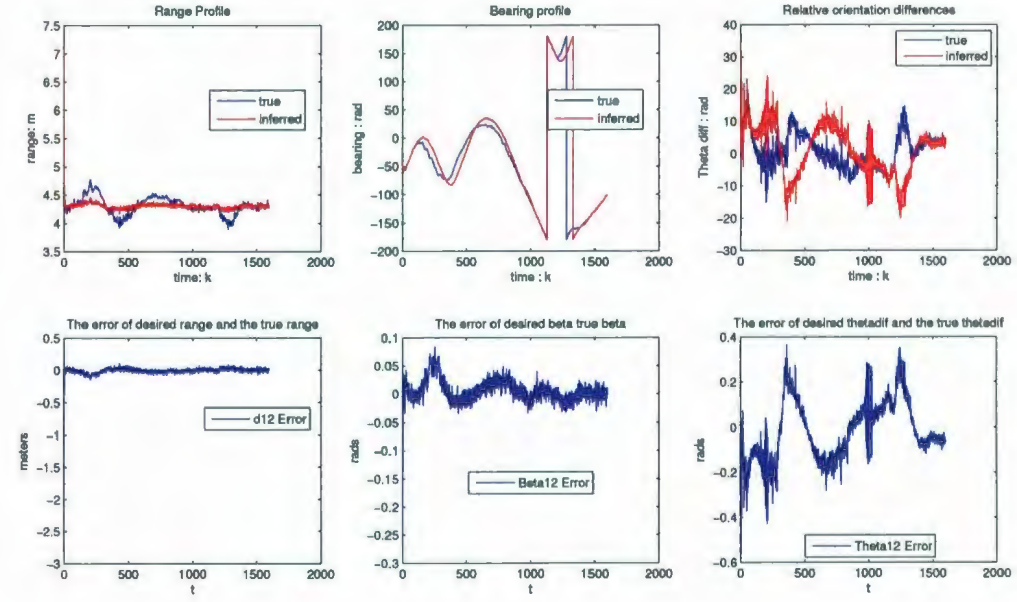
	$v^c\omega^c$	$v^d\omega^c$	$v^c\omega^d$	$v^d\omega^d$	Mix	$Average_{error}$
v_l^{rms}	8.9e-3	0.014	0.0336	0.0603	3.92e-2	0.0312
ω_l^{rms}	2.0474e-4	3.6084e-4	3.9e-3	6.1e-3	4.41e-3	2.995e-3
d_l^{rms}	0.0344	0.031	0.105	0.1629	1.235e-1	0.09136
β_l^{rms}	0.0171	0.0173	0.179	0.3086	0.2142	0.14724
θ_l^{rms}	5.7e-3	4.5e-3	0.144	0.2160	0.1581	0.10566

Table 4.2: RMS error of estimated and true state values of the lead robot

for the same velocity profiles of the leader robot as in EKF based state estimation above. It is seen that, even with a few number of sample runs the UKF estimation is better than EKF based estimation.



4.2.1: true and estimated velocity profiles



4.2.2: True and estimated error of the state and the formation error of the leader robot

Figure 4.2: UKF-State estimation with orientation measurement

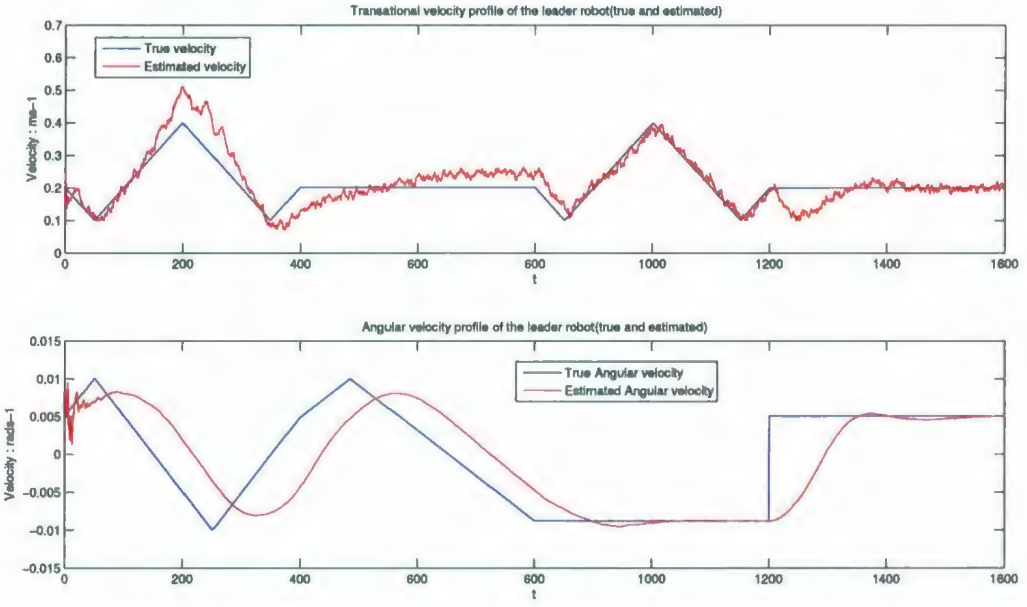
4.4.2 Central difference Kalman filter (CDKF)-based state estimation

Central difference Kalman filter is another Sigma point Kalman filter which uses the Sterling's interpolation formula to replace the first and second order derivatives of the Taylor series with numerically evaluated central differences. (See Appendix A-CDKF for more details). Its performance is superior to EKF and marginally better than the UKF [50].

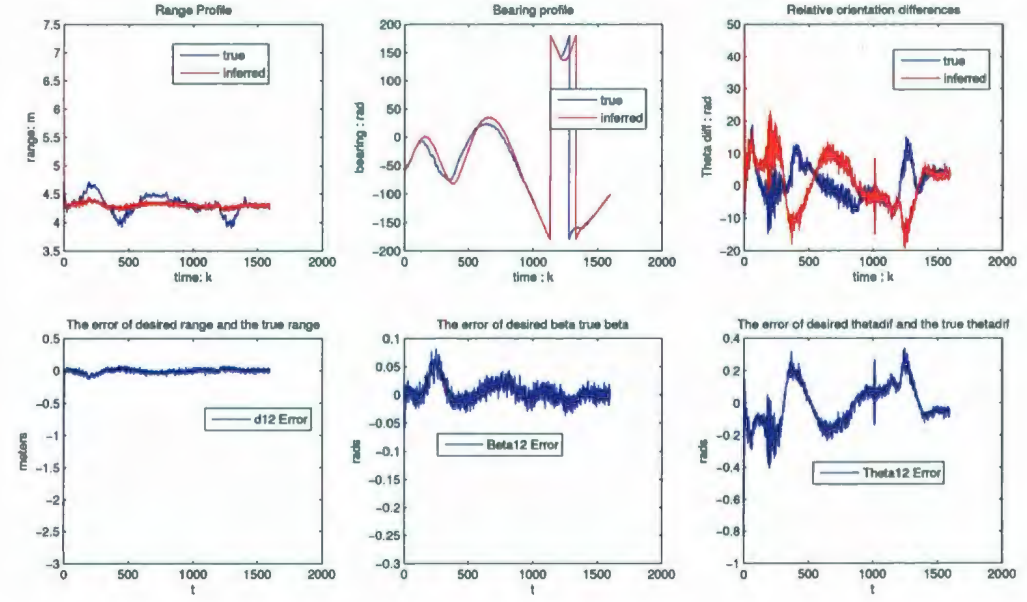
	$v^c\omega^c$	$v^d\omega^c$	$v^c\omega^d$	$v^d\omega^d$	Mix	$Average_{error}$
v_l^{rms}	8.6e-3	0.014	0.0337	0.0604	3.99e-2	0.03132
ω_l^{rms}	2.1714e-4	3.9e-4	3.9e-3	6.1e-3	4.6e-3	3.041e-3
d_l^{rms}	0.0317	0.0339	0.1115	0.1729	1.247e-1	0.09494
β_l^{rms}	0.0148	0.0163	0.186	0.3088	0.2185	0.14888
θ_l^{rms}	6.1e-3	3.9e-3	0.1475	0.2179	0.1586	0.1068

Table 4.3: RMS error of estimated and true state values of the lead robot

From Table 4.3, it is evident that CDKF results are better than EKF, but closer to the results of UKF. These results are just from five test runs and many test runs will prove the marginal superiority of the CDKF over UKF. The noise covariance matrices were chosen same as those values used for the extended kalman filter. The time update happens every 0.5 seconds and is more than enough for the whole filter to estimate the posterior. The time taken for the update is quite higher than the EKF due to the calculation of square root of the weighted covariance matrix [50].



4.3.1: true and estimated velocity profiles



4.3.2: True and estimated error of the state and the formation error of the leader robot

Figure 4.3: CDKF-State estimation with orientation measurement

4.4.3 Formation control through Square-Root Sigma-Filter based state estimation

SRUKF and SRCDKF are square root Sigma point Kalman filters. One of the computational bottlenecks for the UKF and the CDKF is the calculation of the square root of the weighted covariance matrices at each time step to form the sigma point sets [50]. The square root Kalman type filters derive the square root form of the weighted covariance matrices efficiently using QR decomposition, Cholesky factor updating and efficient pivot based least squares [51]. They carry numerical stability and also lesser computational complexity and have the same or marginal accuracy with UKF and CDKF type filters.

Leader robot state estimation via SRUKF and SRCDKF were also experimentally evaluated. Although the estimation results of these two methods are not shown in this thesis, they both resulted in results similar to those of CDKF and UKF. But the computational cost experienced was lesser.

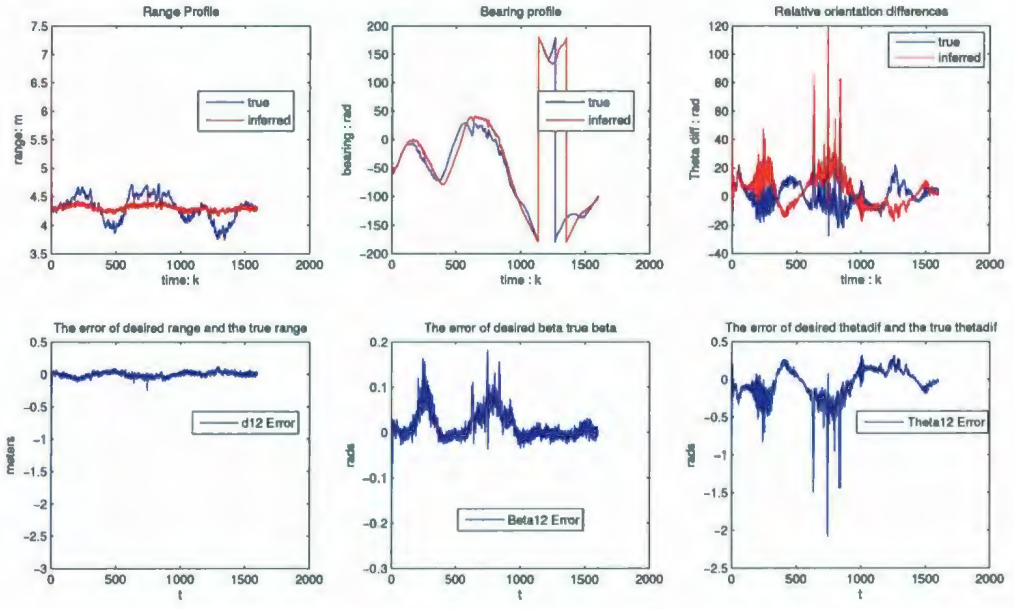
4.5 Particle filter-based state estimation

Particle filter being a sequential Monte Carlo method represents the complete distribution of the state using sequential importance sampling and re-sampling. The advantage of the Particle filter over the many Kalman type filters is that it does not assume the state distribution to be Gaussian or linear (See Appendix A - Particle filter). Hence for many real world applications Particle filters can be used for estimation of hidden states of the systems. Particle filtering has two major stages: the sampling step and the resampling step. The sampling step chooses some particles which captures the true mean and the covariance of the prior distribution. Once the particles are sent through the dynamic system, their respective variances gets

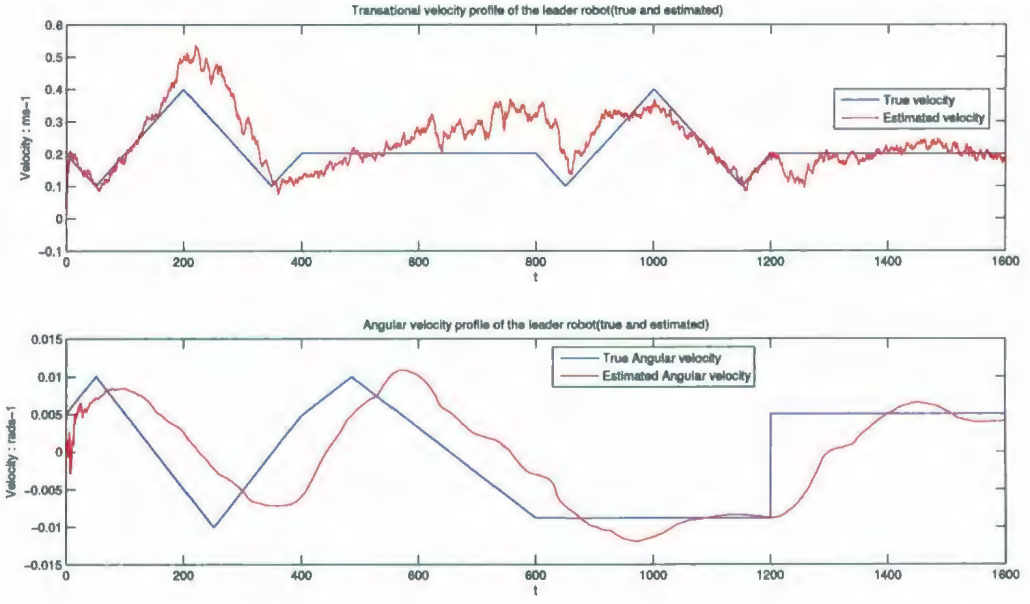
increased and then the new measurements are used to refresh the posterior distribution to approximate it to the true distribution. Then a resampling strategy makes the highly likelihood particles to the true state of the system evolve to the next step of the filter. This is established through removing the lower likelihood particles and making multiple copies of the highly likelihood particles such that the number of particles in the distribution remain the same.

4.5.1 Formation control through Sigma-Point particle filter (SPPF) based state estimation

Particle depletion is a major problem in the particle filter [51]. Particle depletion of the Particle filter makes the filter diverge. Hence moving all the particles to the highly likely region from the current observations will improve the robustness of the particle filter in the resampling step. Using an EKF generated Gaussian approximation to the optimal proposal, one is able to move these particles to the high likelihood areas. This can be accomplished by using a separate EKF to generate and propagate a Gaussian distribution for each particle proposal distribution. The idea is to use the EKF equations at time step k to generate the mean and the covariance of the importance distribution for each particle from time step $k-1$. Then the i^{th} particle from this distribution is redrawn. This way the chosen particles happen to fall in the highly likelihood regions of the distribution. Using the sigma point filters over the EKF, an effective proposal distribution can be taken for each particle. These are known as SPPF (sigma point particle filters). There is a significant computational burden from this approach where a separate Sigma point Kalman filter is to be maintained and also the increment of the number of the particles will require an extra computational demand. Leader state estimation using SPPF (CDKF based Particle filter) for the formation control is given in Fig. 4.4. The formation errors for a number of sample runs are shown in Table 4.4. The errors in Table 4.4 show that, SPPF type estimation



4.4.1: true and estimated velocity profiles and leader robot path



4.4.2: True and estimated error of the state and the formation error of the leader robot

Figure 4.4: SPPF-State estimation

	$v^c\omega^c$	$v^d\omega^c$	$v^c\omega^d$	$v^d\omega^d$	Mix	$Average_{error}$
v_l^{rms}	0.0106	0.0148	0.05	0.0673	0.0662	0.04178
ω_l^{rms}	4.0076e-4	8.4007e-4	4.2e-3	6.4e-3	5.2e-3	3.4081e-3
d_l^{rms}	0.0336	0.0394	0.1709	0.167	0.1853	0.11924
β_l^{rms}	0.0175	0.0185	0.3092	0.31	0.3217	0.19538
θ_l^{rms}	6.7e-3	7.8e-3	0.1985	0.22	0.2434	0.13528

Table 4.4: RMS error of estimated and true state values of the lead robot

is less accurate, when compared with Kalman-filter type estimations.

4.6 Summary

Different Kalman type filters and Particle filters were developed and simulated to estimate the the pose and the velocity profile of leader robots in formation control. It is evident from the experimental results above, that out of all the recursive Baysian filters tested, the Sigma Point Kalman filters have the better accuracy and the performance over the EKF and SPPF type estimations. Hence this thesis recommends the use of Sigma point Kalman filters (Especially the square root Sigma Point Kalman filters e.g: SRUKF, SRCDKF, which reduces the computational complexity of UKF and CDKF respectively) for the estimation of the leader robot's state variables which can then be used for formation control of multiple nonholonomic mobile robots.

Chapter 5

Hybrid Formation Control Framework for Multiple Nonholonomic Mobile Robots

About this chapter: This chapter develops a novel leader-follower based formation control framework to coordinate multiple nonholonomic mobile robots. The framework focuses on multi robot navigation in an unstructured environment. Designated leader robots are made to navigate to particular goal points of interest with obstacle avoidance and wall following capabilities. A set of follower-robots keeps predetermined geometric formation shapes with these designated leader-robots while also being adaptable to the constraints imposed by obstacles in the environment. In order to achieve proper navigation, a set of behavior based low-level continuous controllers are developed while a higher-level discrete event system [30], [28], [31] manages the dynamic interaction with the external environment. The basic formation controllers developed through static and dynamic feedback linearization in chapter-3 will be used along with newer extended versions of formation controllers to handle different formation behaviors needed. These extended formation controllers will again be developed

using,

1. dynamic feedback linearization [32], [36], [48]
2. static feedback linearization combined with some modification of the state measurement coordinates [32], [36], [34]

Both types of controllers are necessary because 1.) to avoid the structural singularity of the dynamic feedback linearized controllers when linear velocity is zero 2.) to exploit the better performance of the dynamic feedback linearized controllers. Hence two families of controllers are developed to tackle different elementary, (e.g.:obstacle avoidance) as well as secondary, (e.g.: formation control with obstacle avoidance) behaviors required. In both types of controllers the coordination of the required behaviors are handled by a discrete event system with supervisory control. The key contributions of this chapter are,

1. Development of a novel hybrid formation control framework for multiple nonholonomic mobile robots to navigate in an unstructured environment.
    - Dynamic feedback linearized formation controllers for I.) single robot navigation II.) leader-follower based formation control of multiple mobile robots in unstructured environments. These include controllers for elementary behaviors, (e.g.: obstacle avoidance) and controllers for combined-behaviors, (e.g.: Wall following with goal navigation). Some elementary behaviors for e.g.: formation control, can be combined with wall following or obstacle avoidance by relaxing some formation constraints
    - Similar static feedback linearized formation controllers to overcome the single singularity (robot linear velocity dropping to zero) of its dynamic feedback linearized counterparts.
    - Use of supervisory control of discrete event systems to model the coordination of different behaviors of formation control.

5.1 Background

Earlier approaches to leader-follower formation control with navigation had been addressed in [12], [4]. Static feedback linearization based $l-\psi$ and $l-l$ controllers are developed in [4] in order to maintain the formation and to build hierarchy of leader followers respectively. But as shown in Chapter 3 above, the static feedback linearized $l-\psi$ and $l-l$ controllers do not stabilize the follower robot's origin to desired formation locations. Hence these control laws are neither robust nor do they achieve the real objective of formation control, which is to stabilize the robot origins to desired formation locations. This research overcomes such shortcomings via the use of dynamic feedback linearization to build formation controllers that are effective in performance. The research also uses static feedback linearized formation controllers with their dynamic counterparts in order to avoid the singularity of the dynamic feedback linearized controllers. The basic $l-\psi$ formation controller is used in [4] only to maintain the formation whereas this thesis shows the use of that controller for single robot navigation in unstructured environments. In addition we develop a new set of static as well as dynamic feedback linearized extended formation controllers for the follower robot formation navigation in unstructured environments.

Formation behaviors include formation maintenance, obstacle avoidance, wall following etc. Coordinating these different behaviors of formation control is itself challenging owing to the uncertainty of real environments. Such behavior coordination for the formation control problem had been addressed in [4] with a gross controller switching strategy. It lacks modelling ease, reusability and scalability. This thesis exploits the use of supervisory control of discrete event systems [30], [28] to coordinate such formation behaviors. Supervisory control of discrete event systems is an alternative design paradigm especially catered to model the dynamic and synchronous changes of a system. The dynamic interactions are modelled as events, which are controllable

and uncontrollable in nature, for e.g: in robot navigation, detecting an obstacle is an uncontrollable event whereas avoiding the obstacle is a controllable event. The supervisory control in the discrete event system exploits this controllability feature of events, to enable or disable them in such a way that the system robustly interacts with the dynamic environment.

5.2 Leader Robot Navigation

It is shown in this section that the basic static feedback linearized Eq. (3.18) and dynamic feedback linearized Eq.(3.22) formation controllers shown in Chapter 3 can be utilized for the navigation of a single mobile robot. They can be used for obstacle avoidance as well as for wall following such that the chattering effects get minimized. The leader robot in the formation is supposed to navigate from a given location to a goal location along some sub goals on its way. It can be accomplished by performing a Voronoi decomposition on a given map and using A* algorithm [52] to find the sub goals which incur a minimum cost path from a starting point to a goal location. The leader robot is navigated to the end goal via these sub goals. Every time the leader robot approaches a sub goal the robot is turned to the next sub goal at a distance d from the sub goal which is equivalent to the maximum turning radius of the robot (for the P3AT robot the value is 500 mm). When navigating along these consecutive sub goals, the leader robot is expected to avoid static obstacles and follow the walls in the given environment. The static obstacle avoidance procedure using the basic formation controllers is explained below.

5.2.1 Single obstacle avoidance

For single robot obstacle avoidance, an obstacle is considered as a virtual lead-robot (see Fig.5.1), whose heading is in the direction of the next sub goal. The heading is

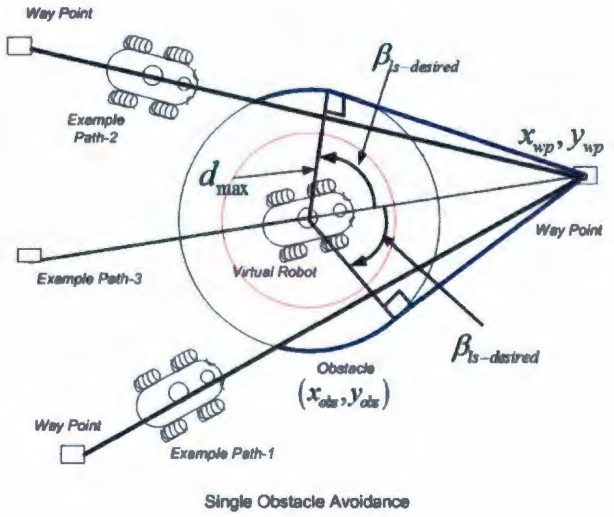


Figure 5.1: Single robot obstacle avoidance with goal navigation

given as,

$$\alpha_{head} = atan2(y_{wp} - y_{obs}, x_{wp} - x_{obs}) \tag{5.1}$$

Also the obstacle has been extended to a circle of radius d_{max}. The basic formation controllers shown in section 3.3 of Chapter 3 can be applied to avoid obstacles by taking the follower(shown in Fig.5.1) as the actual leader-robot and the leader(shown in Fig.5.1) as the obstacle for this context. Once a real lead-robot approaches the obstacle boundary of d_{max} (maximum turning radius of the robot), the static or dynamic formation controllers (shown in section 3.3 of Chapter 3 by Eq.(3.18) and Eq.(3.22) respectively)can be used to drive it with a desired $z_{ls}^d = [d_{max} \quad \beta_{ls}^d]^T$ where,

$$\beta_{ls}^d = \cos^{-1} \frac{d_{max}}{\sqrt{((y_{wp} - y_{obs})^2 + (x_{wp} - x_{obs})^2)}} \tag{5.2}$$

The control law makes the robot to keep a constant d_{max} distance from the obstacle and once the robot arrives at β_{ls}^d, it can safely return to goal navigation. For static obstacle avoidance, the virtual leader's exogenous inputs to the system are made zero. If the obstacle has a motion, the exogenous inputs can be estimated by decentralized

state estimation (explained in Chapter 4). Otherwise the exogenous inputs are taken as zero. When the initial relative bearing (the bearing which can be seen by the virtual robot of the location of the actual leader robot) of the single robot on the surface of the obstacle is $\pm\pi$, the two types of controllers can make the robot travel to either direction of the obstacle. Hence, when the robot arrives (as in path 3 of Fig. 5.1, the most feasible direction is chosen by considering, the current pose of the robot and if the robot heading is coincidental with the virtual (obstacle) heading then a hysteresis is added to move the robot in either of the directions momentarily followed by the inputs from the basic formation controllers.

5.2.2 Clustered obstacle avoidance and wall following for leader robot

Clustered obstacle avoidance

Clustered obstacles can be identified as a set of overlapped obstacles as in Fig. 5.2 (a). While *obstacle-1* in Fig.5.2 (a) is being avoided by the above single obstacle avoidance strategy, the robot comes to $\mathbf{P_1}$ and identifies a second obstacle. To avoid *obstacle-2*, the robot keeps d_{max} (maximum turning radius of the robot) distance from the obstacle and tries to go in the *shortest path* to the way point with the above strategy. The problem arises, when the shortest path around the obstacle overlaps with the the previous avoided obstacles. This problem can be overcome by changing the heading of the virtual leader robot of *obstacle-2* to as heading along the straight line connecting the two obstacles from *obstacle-1* to 2, which is,

$$\theta_h = atan2(y^2_{obs} - y^1_{obs}, x^2_{obs} - x^1_{obs}) \tag{5.3}$$

Then drive the real-robot along the longest path of obstacle avoidance to just above $\mathbf{P_2}$, which is $\mathbf{P_3}$. Once $\mathbf{P_2}$ is passed, the system switches to the earlier obstacle

avoidance strategy. The β_{ls}^{d} for point $\mathbf{P_2}$ can be calculated as,

$$\beta_{ls}^{d} = \pi + atan2(y_{way}^{2} - y_{obs}^{2}, x_{way}^{2} - x_{obs}^{2}) - \theta_{h} \tag{5.4}$$

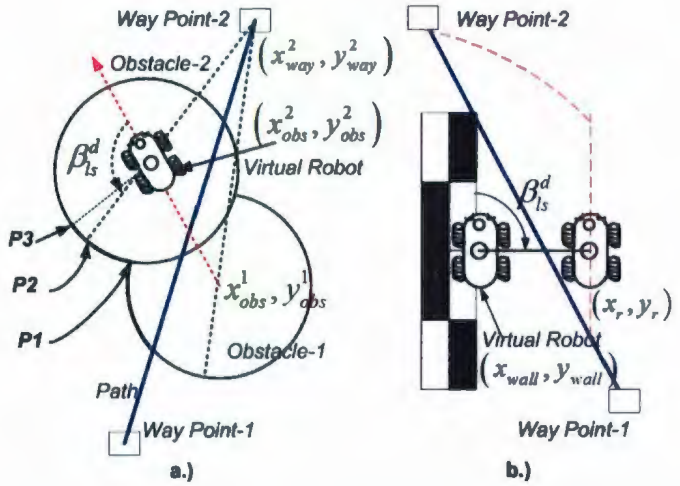


Figure 5.2: (a) clustered obstacle avoidance (b) wall following by the leader robot

Wall following

Once a wall is detected, if a virtual leader is made to slide along the wall with a fixed velocity, the real robot can be made to follow it with a constant relative distance of d_{wall}^{max} and an angle of β_{ls} as in Fig. 5.2 (b). Assuming the distance sensors are fixed on the front of the robot, if a wall is detected on the left distance sensors, the heading of the virtual leader robot is taken as pointing towards the right-most scans of the wall from the left-most. When an obstacle is detected on the right distance sensors, vice versa. Moreover for a left most scan $\beta_{ls}^{d} = -\frac{\pi}{2}$ while for right most scans $\beta_{ls}^{d} = \frac{\pi}{2}$. The location on the wall where the shortest scan-distance is recorded initially, is taken as the virtual robot's position (x_{wall}, y_{wall}). When a wall is detected straight on the front of the real robot, the previously defined obstacle avoidance strategy (section 5.2.1 and 5.2.2) is activated. Sample path navigation for a nonholonomic mobile robot is shown in Fig.5.3. The static and dynamic feedback linearized formation controllers

are combined to harness the potential of the dynamic feedback linearized controller and to avoid its singularity when the linear velocity of the axel is zero. The results show that the chattering effect of obstacle avoidance and wall following is lesser than when it's done in a naive approach.

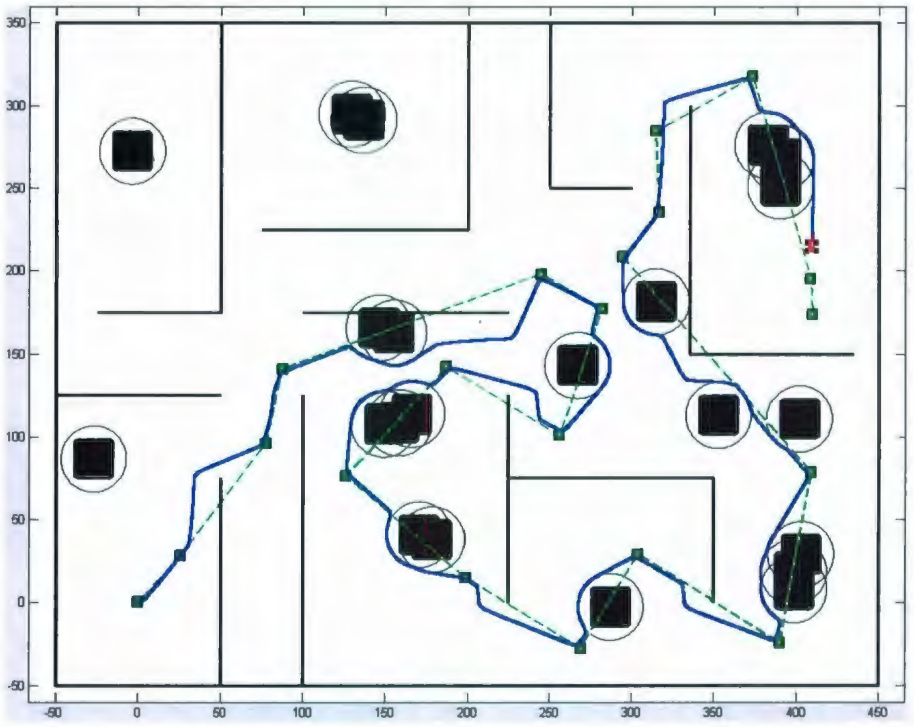


Figure 5.3: Leader robot simulation in an office layout with walls and obstacles: Blue path is the actual robot path, green squares are way points, black squares marked with boundaries of black circles are obstacles while black lines are walls

5.3 Multiple Robot Navigation with Formation Control

Motivated from the work shown in [4], two multi robot formation controllers based on a three robot structure for avoiding obstacles and following walls are presented in this section. The resulting dynamic equations for these configurations will carry a similar singularity seen in the basic leader follower formation controller (see Chapter 3 section 3.3 Eq.(3.15)). Hence both dynamic and static feedback linearization will

be used to linearize these robot configurations and build two families of controllers. These two families of controllers will be used in conjunction in order to avoid the structural singularity seen for the dynamic feedback linearized formation controllers and to achieve a better performance (the basic dynamic feedback linearized controller was better in performance).

5.3.1 Static feedback linearized extended formation controllers

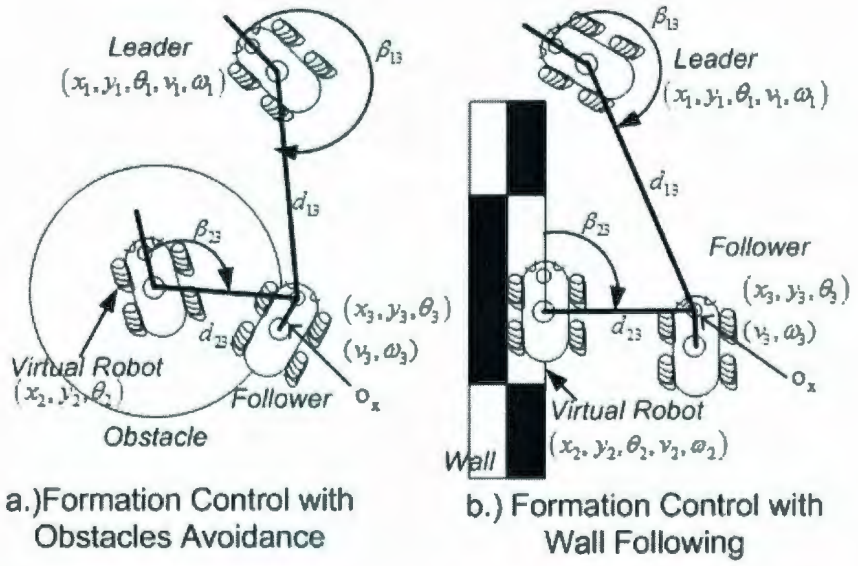


Figure 5.4: (a) obstacle avoidance with formation control (b) wall following with formation control

The followers of the system keep a tight formation with the leader by generating motion commands through the basic static feedback linearized formation controller [4]. However once obstacles or walls are encountered all of the formation constraints can no longer be met at the same time. Hence keeping a desired relative bearing is relaxed while still keeping a desired distance from the leader. (See Appendix B. static feedback linearized controller derivation for more information)

Obstacle avoidance with formation control

Obstacle avoidance and wall following are achieved through a three-robot formation structure. One is the real leader, another the follower and the other is the obstacle or the wall. The kinematics for obstacle avoidance while keeping a desired distance (Fig.5.4 (a)) with the leader is given as,

$$\dot{z}_{dll} = G_1(\gamma_{13}, \gamma_{23}, a)\mathbf{u_3} + F_2(\beta_{23})\mathbf{u_2} + F_1(\beta_{13})\mathbf{u_1} \tag{5.5}$$

where $z_{dll} = [d_{13} \quad d_{23}]^T$ is the system output. $u_1 = [v_1 \quad \omega_1]$ is one exogenous input by the real-leader robot to the system and if the obstacle's motion parameters can be estimated $u_2 = [v_2 \quad \omega_2]$ can be used as another exogenous input to the system while $u_3 = [v_3 \quad \omega_3]$ is the real follower's driving inputs. d_{13} and d_{23} are the relative distances from the real-leader and the obstacle to the follower respectively.

$$G_1 = \begin{pmatrix} \cos\gamma_{13} & o_x \sin\gamma_{13} \\ \cos\gamma_{23} & o_x \sin\gamma_{23} \end{pmatrix}, \qquad F_2 = \begin{pmatrix} 0 & 0 \\ -\cos\beta_{23} & 0 \end{pmatrix}$$

$$F_1 = \begin{pmatrix} -\cos\beta_{13} & 0 \\ 0 & 0 \end{pmatrix}, \quad where \quad \begin{aligned} \gamma_{13} &= \beta_{13} + \theta_{13} \\ \gamma_{23} &= \beta_{23} + \theta_{23} \end{aligned}$$

Also $\theta_{13} = \theta_1 - \theta_3$, $\theta_{23} = \theta_2 - \theta_3$. $\beta_{13} = -\theta_1 + \pi + atan2(y_1 - y_3, x_1 - x_3)$ and $\beta_{23} = -\theta_2 + \pi + atan2(y_2 - y_3, x_2 - x_3)$ are calculated as in the basic formation controller above. Through nonlinear feedback linearizarion, motion commands for the follower are,

$$\mathbf{u_3} = G_1^{-1}(\, c(z_{dll}^d - z_{dll}) - F_1\mathbf{u_1} - F_2\mathbf{u_2}) \tag{5.6}$$

$c = [c_1 \quad c_2]^T > 0$ are controller gains, while $z_{dll}^d = [d_{13}^d \quad d_{23}^d]^T$ are the desired relative distances from the leader and the obstacle. It is seen that the closed loop system is stable and converges to z_{dll}^d arbitrarily fast. This controller is similar to the leader obstacle control controller in [4] except for the extension of dynamic or inter-robot

collision avoidance capability through the substitution of colliding robot's motion parameters through $\mathbf{u_2}$.

Wall following with formation control

Wall following with formation (Fig.5.4 (b)) is done in the same way as in the previous case of obstacle avoidance with the addition of keeping β_{23} at a desired value. Depending on the heading of the wall it can be $\pm\frac{\pi}{2}$. Also we make ω_2 of the virtual leader zero, such that it can only slide along the heading of the wall with a v_2 only. Dynamics of the system are,

$$\dot{z}_{wll} = G_2(z_{wll}, \beta_{23}, \gamma_{13}, \gamma_{23}, a, d_{23})\mathbf{u} + F_3(\beta_{13})\mathbf{u_1} \tag{5.7}$$

$$G_2 = \begin{pmatrix} \cos\gamma_{13} & o_x \sin\gamma_{13} & 0 \\ \cos\gamma_{23} & o_x \sin\gamma_{23} & -\cos\beta_{23} \\ \frac{-\sin\gamma_{23}}{d_{23}} & \frac{o_x\cos\gamma_{23}}{d_{23}} & \frac{\sin\beta_{23}}{d_{23}} \end{pmatrix}, F_3 = \begin{pmatrix} -\cos\beta_{13} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Also $z_{wll} = [d_{13} \quad d_{23} \quad \beta_{23}]^T$ is the system output. $u = [v_1 \quad \omega_1]$ is the only exogenous input by the real leader. And $u = [v_3 \quad \omega_3 \quad v_2]^T$ are follower's inputs followed by the virtual leader's linear velocity. Through feedback linearization,

$$\mathbf{u} = G_2^{-1}(\, c(z^d_{wll} - z_{wll}) - F_3\mathbf{u_1}) \tag{5.8}$$

$c = [c_1 \quad c_2]^T > 0$ being controller gains, while $z^d_{wll} = [d^d_{13} \quad d^d_{23} \quad \beta^d_{23}]^T$ are the desired settings of the system. The location on the wall where the initial wall detection scans got the shortest distance, is taken as the virtual robot's starting pose. And by subsequent usage of Eq.(5.8), we derive motion commands for the virtual leader on the wall (v_2) and the follower $[v_3 \quad \omega_3]^T$ to follow the wall. Also the virtual leader is stopped where the wall ends, to switch to some other navigation task. It is seen that the closed loop system Eq.(5.8) is stable and converges to z^d_{wll} arbitrarily fast. The

given static feedback linearized controller is an extended derivation from the three robot shape control configuration in [4]. Here we use our own formulated controller for following the walls while in [4] a similar controller is used as a hierarchical formation building block (shape control) where robot-2 keeps formation with robot-1 while robot-3 tries to keep the formation using robot-1 and robot-2.

5.3.2 Dynamic feedback linearized extended formation controllers

Dynamic feedback linearization is used to derive controllers for the same configurations for which we derived static feedback linearized controllers in the earlier subsection. We saw in the earlier chapters the performance of the dynamic feedback linearized controller is better than its static counterpart, and that it can be used to stabilize the origin of the robot to a desired formation pose as opposed to stabilizing an offset from the origin to a desired pose (as done in the static feedback linearized controller). Hence static feedback linearized controller is used only as a tool to overcome the singularity posed by the dynamic extension when the axel's linear velocity falls to zero. (See Appendix B. Dynamic Feedback Linearized Controller derivation for more information)

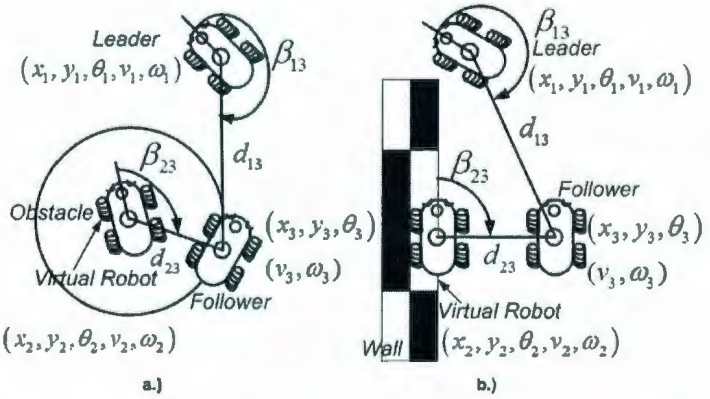


Figure 5.5: (a) obstacle avoidance with formation control (b) wall following with formation control

Obstacle avoidance with formation control

As in the static feedback linearized case earlier, obstacle avoidance and wall following is achieved through a three-robot formation structure. Again one robot is the real leader, another the follower and the other is the obstacle or the wall. The dynamic feedback linearized controller for obstacle avoidance while keeping a desired distance (Fig.5.5 (a)) with the leader is given as,

$$\ddot{z}_{dll} = G_3(\gamma_{13}, \gamma_{23}, v_3)\mathbf{u_3} + F_4(\beta_{13})\mathbf{u_1} + F_5(\beta_{23})\mathbf{u_2} + \mathbf{P} \tag{5.9}$$

where $z_{dll} = [d_{13} \quad d_{23}]^T$ is the system output. $u_1 = [a_1 \quad \omega_1]$ is one exogenous input by the real leader robot to the system and a_1 is its linear acceleration. And if the obstacle's motion parameters can be estimated $u_2 = [a_2 \quad \omega_2]$ can be used as another exogenous input to the system while $u_3 = [a_3 \quad \omega_3]$ is the real-follower's driving inputs. a_2, a_3 are the linear accelerations of the virtual robot and the actual follower. d_{13} and d_{23} are the relative distances from the real leader and the obstacle to the follower respectively.

$$G_3 = \begin{pmatrix} \cos\gamma_{13} & v_3\sin\gamma_{13} \\ \cos\gamma_{23} & v_3\sin\gamma_{23} \end{pmatrix}, F_4 = -\begin{pmatrix} \cos\beta_{13} & v_3\sin\gamma_{13} \\ 0 & 0 \end{pmatrix}$$

$$F_5 = -\begin{pmatrix} 0 & 0 \\ \cos\beta_{23} & v_3\sin\gamma_{23} \end{pmatrix} \qquad \gamma_{13} = \beta_{13} + \theta_{13}$$

$$\gamma_{23} = \beta_{23} + \theta_{23}$$

$$\mathbf{P} = \begin{pmatrix} v_1\dot{\beta_{13}}\sin\beta_{13} - v_3\dot{\beta_{13}}\sin\gamma_{13} \\ v_2\dot{\beta_{23}}\sin\beta_{23} - v_3\dot{\beta_{23}}\sin\gamma_{23} \end{pmatrix} \qquad \theta_{13} = \theta_1 - \theta_3$$

$$\theta_{23} = \theta_2 - \theta_3$$

$\beta_{13} = -\theta_1 + \pi + atan2(y_1 - y_3, x_1 - x_3)$ and $\beta_{23} = -\theta_2 + \pi + atan2(y_2 - y_3, x_2 - x_3)$ are calculated as in the static feedback linearized extended formation controller above. Through nonlinear feedback linearizarion, motion commands for the follower are,

$$\mathbf{u_3} = G_3^{-1}(-c(\dot{z_{dll}}) - k(z_{dll} - z_{dll}^d) - F_4\mathbf{u_1} - F_5\mathbf{u_2} - \mathbf{P}) \tag{5.10}$$

$k = [k_1 \quad k_2]^T > 0$ and $c = [c_1 \quad c_2]^T > 0$ being controller gains, while $z^d_{dll} = [d^d_{13} \quad d^d_{23}]^T$ are the desired relative distances from the leader and the obstacle to the follower. It is seen that the closed loop system is stable and converges to z^d_{dll} arbitrarily fast.

Wall following with formation control

Wall following with formation (Fig.5.5 (b)) is done in the same way as in the previous case of obstacle avoidance with the addition of keeping β_{23} at a desired value. Depending on the heading of the wall it can be $\pm\frac{\pi}{2}$. Also we make ω_2 of the virtual leader zero, such that it can only slide along the heading of the wall with an a_2 and v_2 only. Kinematics of the system are,

$$\dot{z}_{wll} = G_4(z_{wll}, \beta_{23}, \gamma_{13}, \gamma_{23}, v_3, d_{23})\mathbf{u} + F_6(\beta_{13})\mathbf{u_1} + \mathbf{L} \tag{5.11}$$

$$G_4 = \begin{pmatrix} \cos\gamma_{13} & v_3\sin\gamma_{13} & 0 \\ \cos\gamma_{23} & v_3\sin\gamma_{23} & -\cos\beta_{23} \\ \frac{-\sin\gamma_{23}}{d_{23}} & \frac{v_3\cos\gamma_{23}}{d_{23}} & \frac{\sin\beta_{23}}{d_{23}} \end{pmatrix}, F_6 = \begin{pmatrix} -\cos\beta_{13} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} v_1\dot{\beta_{13}}\sin\beta_{13} - v_3\sin\gamma_{13}(\dot{\beta_{13}} + \omega_1) \\ v_2\dot{\beta_{23}}\sin\beta_{23} - v_3\sin\gamma_{23}(\dot{\beta_{23}} + \omega_2) \\ \frac{\dot{d_{23}}(v_3\sin\gamma_{23} - v_2\sin\beta_{23})}{d^2_{23}} + \frac{v_2\dot{\beta_{23}}\cos\beta_{23} - v_3\cos\gamma_{23}(\dot{\beta_{23}} + \omega_2)}{d_{23}} - \dot{\omega_2} \end{pmatrix}$$

Where $z_{wll} = [d_{13} \quad d_{23} \quad \beta_{23}]^T$ is the system output. $u_1 = [a_1 \quad \omega_1]$ is the only exogenous input by the real leader, and $u = [a_3 \quad \omega_3 \quad a_2]^T$ are follower's inputs followed by the virtual leader's linear acceleration. Through feedback linearization,

$$\mathbf{u} = G_4^{-1}(-k(\dot{z_{wll}}) - c(z_{wll} - z^d_{wll}) - F_6\mathbf{u_1} - \mathbf{L}) \tag{5.12}$$

$k = [k_1 \quad k_2]^T > 0$ and $c = [c_1 \quad c_2]^T > 0$ are controller gains, while $z^d_{wll} = [d^d_{13} \quad d^d_{23} \quad \beta^d_{23}]^T$ are the desired settings of the system. The location on the wall where the initial wall

detection scans recorded the shortest distance is taken as the virtual robot's starting pose. And by subsequent usage of Eq.(5.12) we derive motion commands for the virtual leader on the wall (a_2) and the follower $[a_3 \quad \omega_3]^T$ to follow the wall. Also the virtual leader is stopped where the wall ends, to switch to some other navigation task. It is seen that the closed loop system Eq.(5.11),Eq.(5.12) is stable and converges to z^d_{wll} arbitrarily fast.

5.4 Discrete Event Systems Modelling

The behavior coordinations of the robots are formulated by discrete event systems (DES) with supervisory control. Here we develop separate discrete event systems for both the lead-robot and its followers. Continuous dynamics models developed for different behaviors in the thesis are taken as controllable action events of the DES models. Any other constraints specified will be handled by modelled supervisors. The primitive DES systems for a leader robot and for follower robots can be described by (Fig.5.6). We also assume that the robot obstacle avoidance and wall following

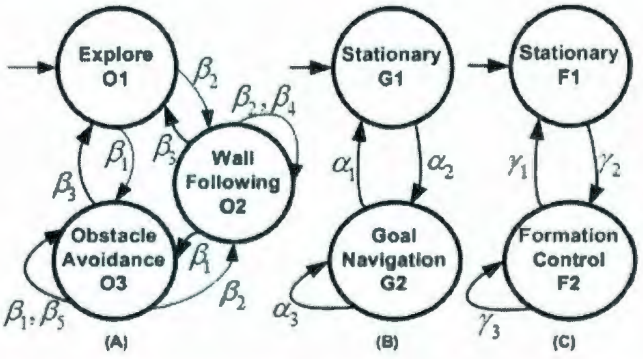


Figure 5.6: DES models for primitive behaviors. (A) obstacle avoidance and wall following, (B) goal navigation behavior, (C) formation control behavior

can not be active at the same time. Therefore the precedence is given to obstacle avoidance when there is both wall following and obstacle avoidance to tackle. For this context, the contact point of walls are considered as obstacles.

DES model for obstacle avoidance and wall following

Set of states $Q : \{Explore - O_1, Wall\ following - O_2, Obstacle\ avoidance - O_3\}$

Set of events $\Sigma : \{detect\ obstacle - \beta_1, detect\ wall - \beta_2, detect\ freespace - \beta_3,$
$move(wallFollow) - \beta_4, move(obstacleAvoidance) - \beta_5\}$

supervisory controllable events $\Sigma_c = \{\beta_4, \beta_5\}$

DES model for goal navigation behavior

Set of states $Q : \{Stationary - G_1, Goal\ navigation - G_2\}$

Set of events $\Sigma : \{goal\ reached - \alpha_1, goal\ computed - \alpha_2, move(to\ goal) - \alpha_3\}$

supervisory controllable events $\Sigma_c = \{\alpha_2, \alpha_3\}$

DES model for formation control behavior

Set of states $Q : \{Stationary - F_1, Formation\ control - F_2\}$

Set of events $\Sigma : \{leader\ lost - \gamma_1, leader\ detected - \gamma_2, keep\ formation - \gamma_3\}$

supervisory controllable events $\Sigma_c = \{\gamma_3\}$

5.4.1 Leader-Robot Behavior Coordination

Once the intermediate sub goal locations are found by applying A* algorithm to the given Voronoi decomposed map, the leader robot is to be navigated to the end goal via the sub goals while avoiding obstacles and following walls on the way. As explained above, every time the leader robot approaches a sub goal the robot is turned to the next sub goal at a distance d from the current sub goal which is equivalent to the maximum turning radius of the robot. In order to develop the holistic leader-robot navigation system, the primitive obstacle avoidance and goal navigation DES models above are combined together using parallel composition. For the DES model in (Fig.5.7), the controllable events are $\{\alpha_2, \alpha_3, \beta_4, \beta_5\}$ and the supervisor developed

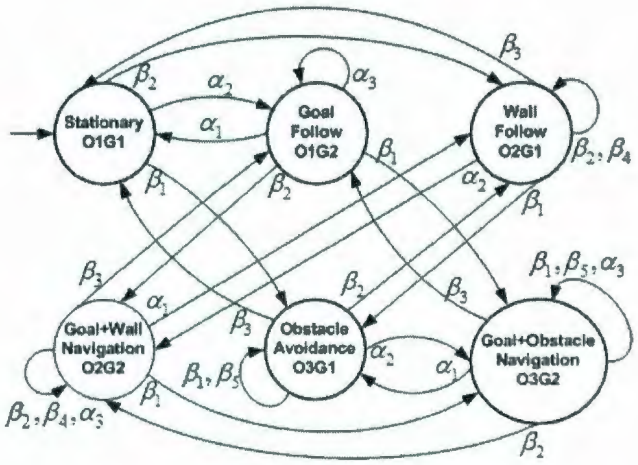


Figure 5.7: Leader robot DES model

to enable or disable these controllable events is given below. '1' stands for enabling, '0' stands for disabling and 'x' stands for not caring the given controllable event. In

State	O_1G_1	O_1G_2	O_2G_1	O_2G_2	O_3G_1	O_3G_2
Σ_c	xxxx	xxxx	0x1x	x01x	0xx1	x1x1

Table 5.1: $\Sigma_c = \{\alpha_2, \alpha_3, \beta_4, \beta_5\}$

states O_2G_1 and O_3G_1, the event goal computed-α_2 is disabled since the goal computation happens when there is no obstacles or walls near the robot. Wall avoidance-β_4 is enabled in both O_2G_1 and O_2G_2 and the continuous dynamics of wall following procedures described in the leader robot navigation Section (5.2.2) is applied to follow the walls. Since β_4 is enabled in O_2G_2, event move to goal-α_3 is disabled to make sure that no two controllable events exist in one state. For this system we have not defined a combined control methodology for wall following and goal navigation at the same time. Hence O_2G_2 degenerates to a control of wall following only. In state O_3G_1, the event of pure obstacle avoidance-β_5 is handled by a reactive obstacle avoidance procedure as in [27]. But in O_3G_2 both events α_3 , β_5 are enabled and a new event is introduced to combine both goal navigation with obstacle avoidance described in the leader robot navigation section (5.2).

5.4.2 Multiple Follower-Robots Coordination

Through the parallel composition of the elementary discrete event systems of obstacle avoidance and formation control, a new complex DES model is built as shown in Fig. 5.8. The followers of the lead robot follow their leader while avoiding obstacles and following walls. The supervisor to control the follower DES model is given below,

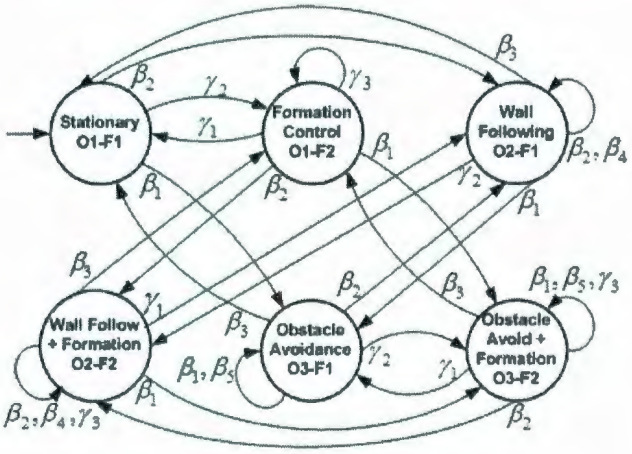


Figure 5.8: Follower robots DES model

State	O_1F_1	O_1F_2	O_2F_1	O_2F_2	O_3F_1	O_3F_2
Σ_c	xxx	1xx	x1x	11x	xx1	1x1

Table 5.2: $\Sigma_c = \{\gamma_3, \beta_4, \beta_5\}$

In state O_1F_2, the event formation control-γ_3 is enabled since it's a state of pure formation control and the leader based basic formation dynamics controllers are used to follow the leader in a given geometric formation. The event wall follow-β_4 is enabled in O_2F_1 to follow the walls when the robot is near a wall and the communication to the leader robot is lost. Since the communication with the leader is lost, the individual wall following procedure of Section 5.2.2 is applied to follow the walls. But in O_2F_2, the state where both the wall following and formation control becomes active events γ_3 and β_4 are enabled to introduce a new event which incorporates both wall following and formation keeping actions and the dynamics of section 5.3 (5.3.1

and 5.3.2 wall following with formation control) are used to handle that event. In state O_3F_1 event obstacle avoidance-β_5 is triggered and the reactive obstacle avoidance procedure explained in [27] is again applied to avoid only the obstacles. The formation control does not become active in this state. In O_3F_2, both obstacle avoidance and formation control actions become active and the supervisor enables both 'formation keep' and 'obstacle avoidance' events and introduce a new event, which combines both the actions in to one continuous dynamics model given again in section 5.3 (5.3.1 and 5.3.2 Obstacle avoidance with formation control) above.

5.5 Simulations

The simulations were carried out in Matlab environment (Fig.5.9) and also using player-erstage/mobilesim (Fig.5.12). Voronoi decomposition is used to find feasible path segments in a test bed of an office layout map with walls and obstacles. And A* algorithm was run to get shortest paths from different starting poses to different destinations and the intermediate way points are recorded. Then the leader-robot is driven along the way point-based path segments as shown in (Fig.5.9). The red robot is the leader while green squares are the way points. The connecting straight line segments of these way points sometimes overlap with existing walls and obstacles. Hence the different behaviors of wall following and obstacle avoidance strategies explained above were used along with the dedicated lead-robot DES model for successful leader robot navigation. As explained above the static and dynamic feedback linearized extended formation controllers were used to drive the robots in the simulation. Through the many experiments run, it is observed that the chattering effect caused by the leader-robot behavior transition is minimized due to the introduction of combined behavior controllers e.g.: goal navigation with obstacle avoidance. In the simulations, we only focussed on static obstacle avoidance, and these obstacles are

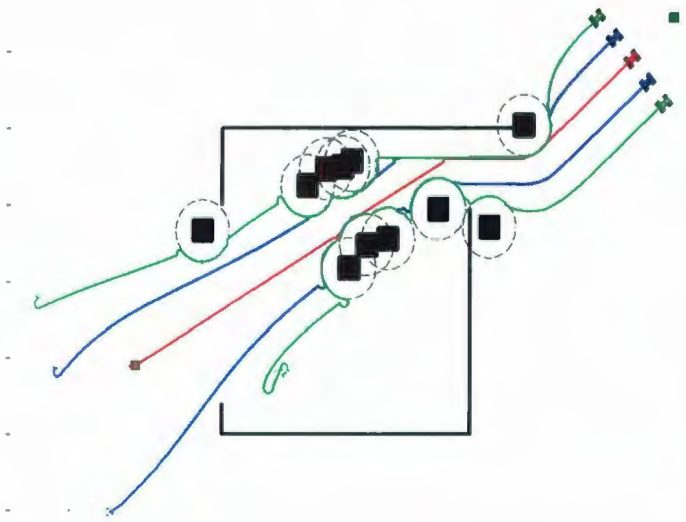


Figure 5.9: Multi Robot Simulation-1: keeping a line formation: Red robot is the leader, blue and green robots are its followers (done in Matlab environment)

avoided in the shortest path possible, to the next way point. Cluttered obstacles were also successfully evaded without any significant chattering effect. The wall following procedure leads more or less a straight path following the wall, again minimizing the chattering effects. The supervisory controlled discrete event system does the transitioning well enough to cater to the dynamic changes of the environment.

We also simulated multiple follower robots with their respective leader robot. They were run in predetermined geometric formations while also coordinating other behaviors through the respective DES model with the assistance of the designated low level controllers. The system was tested with different geometric shapes of wedge, diamond, horizontal lines and triangular shapes with arbitrary starting points for the followers. Obstacle avoidance and wall following while keeping a desired distance to the leader was also tested for different shapes cited above. We observed that as long as the leader robot does not make sudden rotations, the system manages to avoid the walls or obstacles and navigate effectively. The upper bound ω_{max} for rotational velocities of the leader robots make sure the sudden manoeuvres does not occur. Also

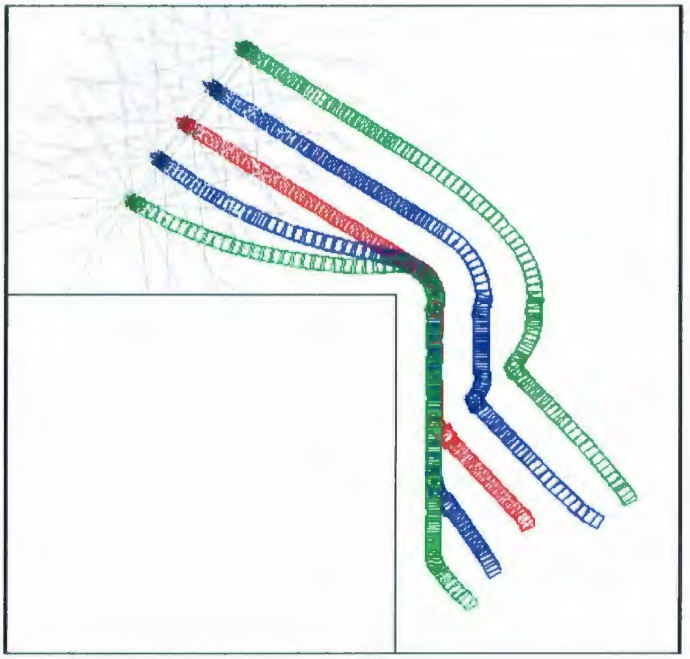


Figure 5.10: Multi Robot Simulation-2: keeping a line formation: Red robot is the leader, blue and green robots are its followers (mobilesim/playerstage simulation)

we have made the followers in such a way that once a sudden manoeuvre of the leader is detected, the followers depend on their own controls without any exogenous input from the leader until the leader stabilizes. Fig.5.11 shows how the errors of the formation, namely relative distance, bearing and orientation errors propagate over time when navigating in an obstacle populated environment. We see that the error of relative distance approaches zero and stays there over time throughout the whole simulation. It is because in avoiding obstacles and following walls, we made the controller so as to only keep a desired relative distance while other constraints were put in place when there was no obstacle or wall. Fig. 5.12 shows the DES transitioning sequence for the same simulation. In to the latter part of this transitioning diagram all the followers stay in pure formation keeping the state at $O1F2$, and is reflected by the fact that in Fig.5.11 all of the relative errors of the followers are zero or near zero to the end. Again the important observation is that the chattering effect is minimized in the whole system, due to the introduction of combined behaviors of for ex: wall

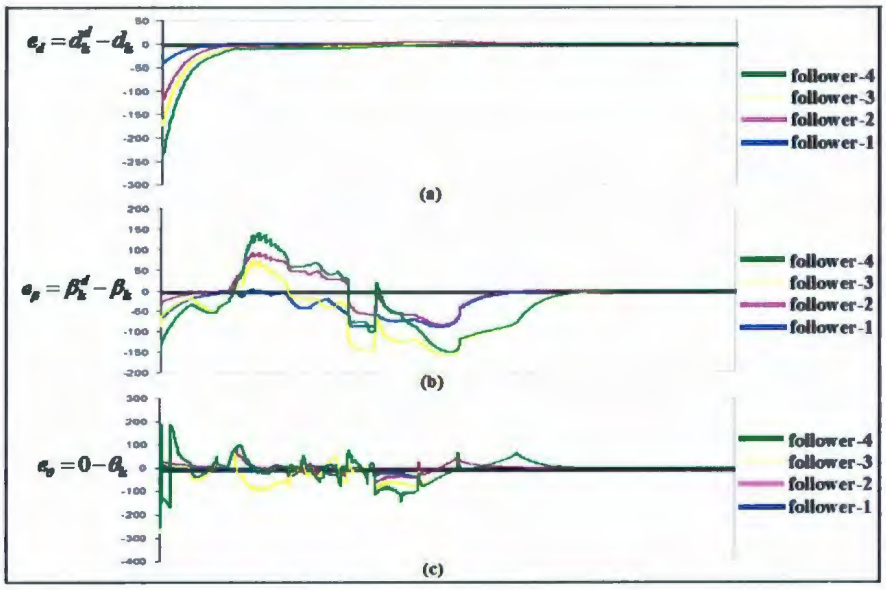


Figure 5.11: Relative distance, bearing and orientation error projection for the followers with respect to the leader robot over time: Robots are driving in a line formation avoiding obstacles and following walls in a sample simulation: (a) $e_d = d_{ls}^d - d_{ls}$ (b) $e_\beta = \beta_{ls}^d - \beta_{ls}$ (c) $e_\theta = 0 - \theta_{ls}$

following with formation control etc. Fig.5.12 shows this minimization of chattering scenario where for both the leader and for the set of followers, the transition between the states of their respective DES models are infrequent.

5.6 Summary

In this chapter, extended formation controllers, based on both static and dynamic feedback linearization were developed to handle formation navigation in an unstructured environment. It is shown that the basic formation controllers can be utilized for single robot navigation too while more complex controllers are developed to handle multi robot navigation. Static and dynamic feedback linearized controllers are combined in an effort to again harness the potential of the dynamic feedback linearized controller and also to avoid it's singularity when the axel is not moving. Discrete event systems with supervisory control were developed to handle the dynamic inter-

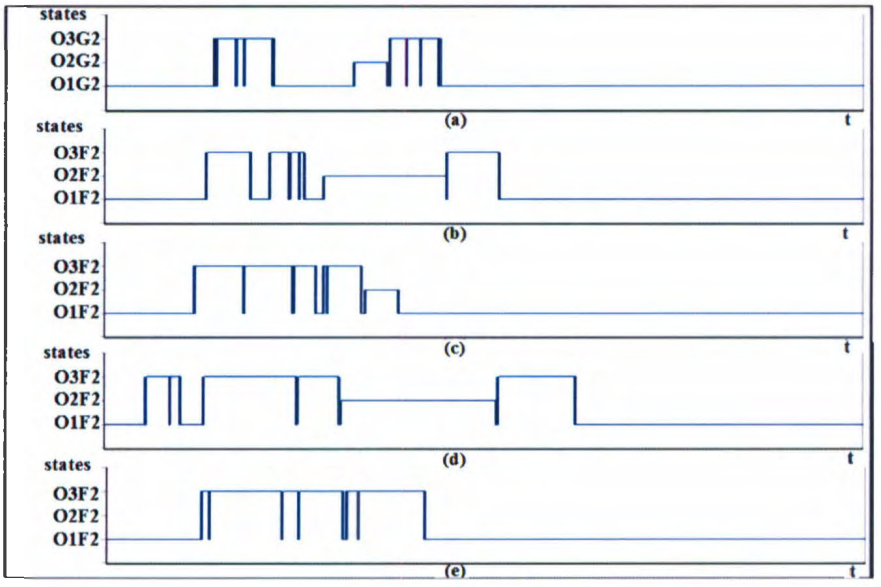


Figure 5.12: State transition diagrams of the DES models for a subset of states: Robots are driving in a line formation avoiding obstacles and following walls in a sample simulation: (a) Leader robot (b) follower-1 (c) follower-2 (d) follower-3 (e) follower-4

action with the environment for both the leader and follower robots in the formation. DES provides a platform to model the dynamic interaction with the environment in a structured way that is both reusable and scalable. The system is simulated in Mobilesim and Matlab environments and the results suggest that the proposed algorithms are effective in formation navigation of a set of multiple mobile robots.

Chapter 6

Conclusion

6.1 Overview

This research develops a decentralized hybrid-framework for formation control of multiple nonholonomic mobile robots. The framework is based on a leader-follower based control theoretic bottom-up approach. The bottom layer of this framework consists of low-level controllers which are derived by nonlinear control methods to handle elementary (obstacle avoidance) and combined (formation control with wall following) robot behaviors for nonholonomic mobile robots in navigation. The upper layer works as a coordinator to model the dynamic interaction with the external environment and is developed through supervisory control of discrete event systems. The current implementation of the framework supports single robot navigation including obstacle avoidance and wall following, multi-robot formation maintenance, multi-robot obstacle avoidance or wall following with formation maintenance. These different behaviors for nonholonomic mobile robots are implemented through low-level controllers. Due to the nonlinear nature of these controllers static and/or dynamic feedback linearization are used to linearize them and make them controllable. There is also a comparison of different types of leader-follower based formation maintenance con-

trollers for nonholonomic mobile robots in the research. These formation maintenance controllers are derived in the research through the application of different nonlinear control techniques. it's found that the dynamic feedback linearized formation maintenance controller is more effective than the other controllers, but suffers from a structural singularity when the robot's linear velocity is zero. Hence it's proposed that the dynamic feedback linearized formation maintenance controller be combined with its static counterpart to avoid the singularity and to achieve effective formation maintenance. The state and the velocity information of leader robots are necessary for the low-level controllers of the followers to work. It's found that communicating this information is not always possible due to resource constraints or limited communication abilities etc. Hence this research also exploits the use of decentralized state estimation techniques to estimate the leader robot's state and velocity profiles without explicit communication. Different recursive Baysian and particle filter type estimators are implemented and compared for estimation accuracy. The proposed holistic systems are implemented through simulations using Mobilesim/Playerstage and Matlab environments to validate their usability.

6.2 Contributions to research

The resulting contributions of this thesis are given below:

6.2.1 Development of a novel hybrid formation control framework for multiple nonholonomic mobile robots

Behavior based low-level controllers for formation control of multiple nonholonomic mobile robots

Multi robot navigation with formation maintenance in an unstructured environment can be thought of as a collection of behaviors. These behaviors can be competitive

or cooperative in nature. Cooperative behaviors can be simultaneously executed together while competitive behaviors must be executed one at a time, according to some priority scheduling. Multi robot navigation with formation maintenance known as formation control also has competitive and cooperative behaviors. This research exploits the idea of a formation control framework which breaks down the formation control problem in to a set of competitive and cooperative behaviors and uses a control theoretic bottom-up approach to design low-level controllers for each such behavior which are together managed by a higher-level coordinating mechanism.

The research uses a leader-follower formation concept for its simplicity, flexibility and scalability over some other formation control strategies. There, we find two types of robots which are designated as leaders and its followers and the thesis addresses the problem of leader robot navigation and multi robot formation control for nonholonomic mobile robots in the formation control framework. Earlier approaches to leader-follower based formation control have been addressed in [4]. They propose two types of feedback controllers for maintaining formations of multiple nonholonomic robots. The first one is the $l-\psi$ controller which maintains a desired length l^d and a desired relative angle ψ^d between the leader and the follower. The second controller is the $l-l$ controller which has a three robot structure where, one robot follows its leader with a $l-\psi$ controller while a third robot follows the latter two robot's with the $l-l$ controller with desired l_{13}^d and l_{23}^d distances. These two types of controllers are developed using static feedback linearization, which results in stabilizing not the origin of the respective nonholonomic robot, but an offset from the origin to desired formation values. As shown in the thesis, stabilizing an offset from the origin, to desired formation values has an undesirable effect on the controls (the inputs to the robot become more oscillating) of the robot and is also not the real objective of formation maintenance (The real objective is to stabilize the origin of the robot frame

to the desired formation values rather than stabilizing an offset from the origin to desired values).

The static feedback linearization-based $l - \psi$ controller in [4] and the dynamic feedback linearized $l - \psi$ controller of [43] are combined together in this research to 1.) overcome the structural singularity of the dynamic feedback linearized controller 2.) to achieve a stable formation maintenance. In addition to maintaining the formation with the $l - \psi$ controller, this thesis also illustrate the use of it for single robot navigation too, whereas [4] and [43] use it only for formation maintenance. Single robot obstacle avoidance and wall following procedures with the $l - \psi$ controller are developed in this research. A set of extended formation controllers is also developed in this thesis, for obstacle avoidance and wall following by follower robots in formation navigation. Both dynamic and static feedback linearized controllers are developed for these controllers and are used in conjunction in order to avoid a structural singularity of the dynamic feedback linearized controllers when the robot velocity is zero.

Use of supervisory control of discrete event systems to model dynamic interaction with the environment

The low-level behavior-based formation controllers for both the leader and the followers have to be coordinated in order to interact with the dynamic environment properly. This research models and develops a discrete event system managed by supervisory control for such dynamic interaction with the external environment for the formation control problem . Discrete event systems provide a modular framework to coordinate the actions of the robots required in a dynamic environment. They also provide ease of modelling, scalability and reusability. Thus new behaviors can be occupied in the system without much of a hassle.

6.2.2 Development of trajectory tracking type leader-follower based formation keeping controllers

The research also exploits the use of trajectory-tracking controllers of nonholonomic unicycle type robots to design and develop formation keeping controllers. Trajectory tracking combines a feed-forward command for a desired pose and a velocity along a trajectory with a feedback action on the error. In the formation maintenance problem, the desired trajectories of followers can be derived through leader state and velocity information at a given time. It can thus be used as a mode of feed-forward command generation. This research uses two existing trajectory tracking controllers for unicycle robots, in order to design controllers for formation maintenance. One is derived through approximate linearization of the unicycle dynamics and [23] and the other through a nonlinear design based on a particular Lyapunov function [22]. A comparison of these two formation keeping controllers with the earlier developed dynamic $l - \psi$ formation keeping controller [43] and the static feedback linearized $l - \psi$ controller of [4] is also carried out in this thesis. Here it is found that the the trajectory tracking type formation keeping controllers including the dynamic feedback linearized controller performs better than the static feedback linearized controller. Hence it can be concluded that the dynamic feedback linearized controller has the best performance out of all except for its structural singularity (when linear velocity is zero).

6.2.3 Decentralized leader robot state estimation

The followers depend on accurate measurements of its leaders pose and velocity to activate feedback controls to maintain the formation. Use of communication has problems of constrained coverage, resource shortage etc. Hence instead of using communication to send leader's information, [4] and [18] use recursive Baysian filters to

estimate leader's pose and velocity using local sensors. [4] uses an EKF approach while [18] uses a dual unscented Kalman filter type estimation. This research tries to address two problems of the possible usage of estimation methodologies to estimate leader robot's pose and the velocity. One is the effect of noise margins of the estimated leader robot's pose and velocities, on the formation controllers used. The other is a benchmarking of different recursive Baysian type filters for leader robot state estimation to decide which recursive Baysian filter has the best estimation accuracy. Estimation accuracies are compared for different recursive Baysian type filters (EKF, UKF, CDKF, SRUKF, SRCDKF) and also for a particle filter (SPPF). The square root sigma-type Kalman filters are found to have the best performance in terms of performance and accuracy. The effect of noise margins of these estimations on the dynamic feedback linearized controller (combined with its static counterpart to avoid the singularity) is also tested, and it is found that, this controller is robust even with a substantially higher noise margin.

6.3 Further recommendations

The research only addresses the implementation of a few sets of basic behaviors in the formation control framework. It will indeed be necessary to include more functionality into the framework in order to design a holistic formation control system. Some behaviors of interest will surely be robot initialization, dynamic role assignment (e.g: rotation of leadership to the most suitable robot in the MRS), shape deformations to effectively avoid obstacles, object manipulation abilities etc. This research combines the static and dynamic feedback linearized formation maintenance controllers to avoid the structural singularity of the latter and it is found that this approach has some drawbacks. The drawbacks are:

- Static and the dynamic feedback linearized controllers are combined as;

- For all $v_s < |v_{threshold}|$, the static feedback linearization based controllers are used. For other values of vs dynamic feedback linearization based controllers are used (v_s is the linear velocity of the robot and $\pm v_{threshold}$ are the boundary values used for switching between the dynamic and static feedback linearized controllers.

  Finding the optimum $v_{threshold}$ is a problem. $v_s = 0$ can occur when v_s is increasing from a negative value to a positive value or vice versa and the acceleration and/or deceleration of the robot at any given time is not the same. Hence there can be no constant value for $v_{threshold}$. (For ex: for a v_s increasing from a negative value to a positive value, if the acceleration is high, a smaller $-v_{threshold}$ will result in reaching $v_s = 0$ before any switching can happen). Also in a real world scenario the accelerations/decelerations are changing over time. Predicting them is impossible. The least possible in determining a good $v_{threshold}$ involves only in assessing the accelerations/ decelerations and any other mechanical constraints at a given time.

- Combining these two controllers breaks the smoothness of operation. One controller tries to stabilize the origin of the robot frame to desired formation values and the other is trying to stabilize an offset from the origin to desired formation values. This discontinuity of operation can cause problems when manoeuvring an object or a wall. There is the possibility that this discontinuity can overshoot an obstacle or a wall. Hence it's recommended that when avoiding an obstacle or a wall, only the use of static feedback linearized controller is preferred.

Given these drawbacks, it will be necessary that the structural singularity in the dynamic feedback linearized controller be tackled properly in future research. Also the formation scalability problem is not properly addressed in the thesis and any future research should substantially investigate it.

Sensor observations and control actions are subjected to natural noise in a real environment. DES is not the ideal solution to tackle the problem of dynamic interaction under noisier observations. Thus this research proposes the investigation of the possible usage of probabilistic discrete event systems with supervisory control (PDES) [53], [54] instead of DES in our framework. We also find that the decentralized state estimation is more challenging than that of communication. Hence we also recommend the investigation of easier and more accurate alternative methods of estimation or communication including custom catered communication networks such as AD-HOC networks or sensor networks [55].

6.4 Contributed papers

6.4.1 conference papers

1. Gayan W. Gamage, George K.I. Mann, Raymond G. Gosine, 2010, "Leader follower based formation control strategies for nonholonomic mobile robots: Design, Implementation and Experimental Validation", *Submitted to 2010 American Control Conference (ACC2010)*, to be held in July 2010, Baltimore, Maryland, USA.

2. Gayan W. Gamage, George K.I. Mann, Raymond G. Gosine, 2009, "Discrete Event Systems based Formation Control Framework to Coordinate Multiple Nonholonomic Mobile Robots", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2009, St. Louis, Missouri, USA.

3. Gayan W. Gamage, George K.I. Mann, Raymond G. Gosine, 2009,"Formation Control of Multiple Nonholonomic Mobile Robots Via Dynamic Feedback Linearization", *14th International Conference on Advanced Robotics (ICAR)*, June

2009, Munich, Germany.

4. Gayan W. Gamage, George K.I. Mann, Raymond G. Gosine, 2009, "A Hybrid Control Strategy for Multiple Mobile Robots with Nonholonomic Constraints", *22nd Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2009, St. John's, NL, Canada.

Appendix A

Kalman Filters and Particle Filter

Extended Kalman Filter

$\text{EKF}(\mathbf{X}_k, \mathbf{\Sigma}_k, \mathbf{U}_{k+1}, \mathbf{Z}_{k+1})$

$\hat{\mathbf{X}}_{k+1} = g(\mathbf{U}_{k+1}, \mathbf{X}_k)$

$\hat{\mathbf{\Sigma}}_{k+1} = \mathbf{G}_{k+1}\mathbf{\Sigma}_k\mathbf{G}_{k+1}^T + \mathbf{R}_{k+1}$

$\mathbf{M}_{k+1} = \hat{\mathbf{\Sigma}}_{k+1}\mathbf{H}_{k+1}^T(\mathbf{H}_{k+1}\hat{\mathbf{\Sigma}}_{k+1}\mathbf{H}_{k+1}^T + \mathbf{Q}_{k+1})^{-1}$

$\mathbf{X}_{k+1} = \hat{\mathbf{X}}_{k+1} + \mathbf{M}_{k+1}(\mathbf{Z}_{k+1} - h(\hat{\mathbf{X}}_{k+1}))$

$\mathbf{\Sigma}_{k+1} = (\mathbf{I} - \mathbf{M}_{k+1}\mathbf{H}_{k+1})\hat{\mathbf{\Sigma}}_{k+1}$

return $\mathbf{X}_{k+1}, \mathbf{\Sigma}_{k+1}$

Sigma Point Filters

Major shortcomings of EKF based estimation are 1.) Disregard of probabilistic spread of the system states and noises during initialization of system equations (linearization expands the distribution around only a single point) 2.) Taylor series expansion holds only for first order accuracy of mean and covariance of the distribution. In order to overcome these shortcomings the idea is to use deterministic sampling approaches that circumvent the calculation of analytical derivatives. Filters which uses a deterministic sampling strategy are collectively known as Sigma-Point Kalman fil-

ters. These together use weighted statistical linear regression in order to calculate the terms needed for the Kalman update rule. WSLR is a method in which the non-linear function of a random variable is linearized using a linear regression between r points drawn from the prior distribution of the RV and the true nonlinear functional evaluations of these points. This tends to be more accurate than linearizing through a truncated Taylor series expansion around a single point since the method outlined takes into account the statistical properties of the prior distribution. For a nonlinear function of the form $y = g(x)$ evaluated at r points of (x_j, y_j) where $y_j = g(x_j)$,

$$\bar{x} = \sum_{j=1}^{r} w_j x_j$$
$$\hat{P_{xx}} = \sum_{j=1}^{r} w_j (x_j - \bar{x})(x_j - \bar{x})^T$$
$$\bar{y} = \sum_{j=1}^{r} w_j y_j$$
$$\hat{P_{yy}} = \sum_{j=1}^{r} w_j (y_j - \bar{y})(y_j - \bar{y})^T$$
$$\hat{P_{xy}} = \sum_{j=1}^{r} w_j (x_j - \bar{x})(y_j - \bar{y})^T$$

w_j are r linear regression weights such that $\sum_{j=1}^{r} w_j = 1$. The idea is to find a linear regression of the form $y = Ax + b$ such that it minimizes a cost function $J = E(\phi(e_j))$. Point wise linearization errors are $e_j = y_j - Ax_j - b$ with covariances $P_{ee} = \hat{P}_{yy} - A\hat{P}_{xx}A^T$ and the error function is the vector dot product. Hence the cost function reduces to the sum of squared error. $\{A, b\} = argmin \sum_{j=1}^{r} (w_j e_j e_j^T)$ and the following solution holds, $A = \hat{P}_{xy}^T \hat{P}_{xx}^{-1} \quad b = \bar{y} - A\bar{x}$.

Unscented Kalman Filter

Unscented Kalman filter is a SPKF type filter which chooses the sigma points (points in the distribution which are supposed to capture the true mean and the covariance) using the rationale "Sigma points" must be chosen so that they capture the most important statistical properties of the prior random variable X" [50]. That can be achieved by choosing the points according to a constraint equation $m(<$

$X, w >, r, p(x)) = 0$ where $< X, w >$ are the sigma points X_i and weights w_i for $i = 1, .., r$. It is also possible to satisfy this constraint while having some degree of freedom for the choice of points through minimizing another cost function of the form $c(< X, w >, r, p(x))$. This cost function serves the purpose of incorporating statistical features of x which are desirable but should not necessarily be met. The statistical information captured by UKF are the first and second order moments of p(x). Number of points used in EKF is $r = 2L + 1$ where L is the dimension of the state x. The sigma points and the weight used for the EKF are given as,

$$X_0 = \bar{x} \qquad w_0^m = \frac{\lambda}{L+\lambda}$$

$$X_i = \bar{x} + (\sqrt{(L+\lambda)P_x})_i \quad i = 1, ..., L \qquad w_0^c = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta)$$

$$X_i = \bar{x} - (\sqrt{(L+\lambda)P_x})_i \quad i = L+1, ..., 2L \qquad w_i^m = w_i^c = \frac{1}{2(L+\lambda)}, i = 1, ..., 2L$$

λ is a scaling parameter given by $\lambda = \alpha^2(L + k) - L$. α is a small positive value usually set to $(1e - 2 \leq \alpha \leq 1)$. It describes the spread of the sigma points around the prior mean $\bar{x}$. k again is a secondary scaling parameter usually set to 0 or $3 - L$. β is used to incorporate any extra knowledge about the prior distribution and $(\sqrt{(L+\lambda)P_x})$ is the i^{th} row or column of the weighted matrix square root of the covariance of P_x [50] [56].

Central Difference Kalman Filter

Based on the Sterlings interpolation formula where the Taylor series 1st and 2nd order derivatives are replaced by the numerically evaluated central divided differences. Taylor series is ,

$$y = g(x) = g(\bar{x} + \delta_x) = g(\bar{x}) + \nabla g \delta_x + \frac{1}{2}\nabla^2 g \delta_x^2 + \frac{1}{3!}\nabla^3 g \delta_x^3 + ...$$

1st and 2nd order terms can be replaced as

$\nabla g \approx \frac{g(x+h\delta_x)-g(x-h\delta_x)}{2h}$, $\nabla^2 g \approx \frac{g(x+h\delta_x)+g(x-h\delta_x)-2g(x)}{h^2}$

Even though the above method does not explicitly use WSLR, it is shown in [57] that the resulting Kalman filter implicitly uses the WSLR. The number of sigma points needed for the CDKF is again $2L+1$ where L is the dimension of state x. The points are chosen as,

$X_0 = \bar{x}$ $\qquad$ $w_0 = \frac{h^2-L}{h^2}$

$X_i = \bar{x} + (\sqrt{h^2 P_x})_i \quad i = 1, ..., L$ $\qquad$ $w_i = \frac{1}{2h^2}, i = 1, ..., 2L$

$X_i = \bar{x} - (\sqrt{\sqrt{h^2 P_x}})_i \quad i = L+1, ..., 2L$

It is shown in [58] that CDKF has marginal accuracy over UKF in replacing the higher order terms in the Taylor series expansion. But all of the SPKF family filters have better performance over the EKF, but marginally different accuracy among different SPKF type filters [58]. Having one scaling parameter h in CDKF as opposed to 3 parameters in UKF makes the CDKF filter easier to use than the UKF.

Particle Filter Algorithm

The general particle filter algorithm is given below [50],

1. *Initialization:* $k = 0$

   - For $i = 1, ..., N$ draw (sample) particle $x_0^{(i)}$ from the prior $p(x_0)$.

2. For $k = 1, 2, ...$

   a *Importance sampling step*

   - For $i = 1, ..., N$ sample $x_k^{(i)} \sim \pi(x_k | x_{k-1}^{(i)}, Y_k)$.
   - For $i = 1, ..., N$ evaluate the importance weights up to a normalizing constant:
     $w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k | x_k^{(i)}) p(x_k^{(i)})}{\pi(x_k | x_{k-1}^{(i)}, Y_k)}$.

- For $i = 1, ..., N$ normalizing the importance weights: $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$

b *Selection step(resampling)*

- Multiply/supress samples $x_k^{(i)}$ with high/ low importance weights $\tilde{w}_k^{(i)}$ respectively. to obtain N random samples approximately distributed according to $p(x_k|Y_k)$.
- For $i = 1, ..., N$ set $w_k^{(i)} = \tilde{w}_k^{(i)} = N^{-1}$.
- (optional) Do a single MCMC move step and add further 'variety' to the particle set without changing their distribution.

c *Output:* The output of the algorithm is a set of samples that can be used to approximate the posterior distribution as follows: $\hat{p}(x_k|Y_k) = \frac{1}{N}\sum_{i=1}^{N} \delta(x_k - x_k^{(i)})$. From these samples, any estimate of the system state can be calculated, such as the MMSE estimate,
$\hat{x}_k = E[x_k|Y_k] \sim \frac{1}{N}\sum_{i=1}^{N} x_k^{(i)}$

Particle depletion of the particle filter makes the filter diverge. Hence moving all the particles to the highly likelihood region from the current observations will improve the robustness of the particle filter in the resampling step. Using a EKF generated Gaussian approximation to the optimal proposal, one is able to move these particles to the high likelihood areas. This can be accomplished by using a separate EKF to generate and propagate a Gaussian distribution for each particle proposal distribution. The idea is to use the EKF equations at time step k to generate the mean and the covariance of the importance distribution for each particle from time step $k-1$. Then we redraw the i^{th} particle from this distribution. This way the chosen particles happen to fall in the highly likelihood regions of the distribution. Using the sigma point filters over the EKF, we can get a good proposal distribution for each particle. These are known as SPPF (sigma point particle filters). There is a significant computational burden from this approach where a separate sigma point

Kalman filter is to be maintained and also the increment of the number of the particles will require an extra computational demand.

Appendix B

Extended Formation Controller Derivation

Static feedback linearized controllers

Three robot extended formation controller derived via static feedback linearization is explained below. Imposing some constraints on the parameters of this three robot dynamic system will let the wall following and obstacle avoidance with formation control feasible. The new coordinates of the follower robot at an offset of o_x and o_y from its origin, in X and Y robot coordinate directions respectively can be calculated as,

$$\begin{pmatrix} x_3^n \\ y_3^n \end{pmatrix} = \begin{pmatrix} \cos\theta_3 & -\sin\theta_3 \\ \sin\theta_3 & \cos\theta_3 \end{pmatrix} \begin{pmatrix} o_x \\ o_y \end{pmatrix} + \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \tag{B.1}$$

(x_3, y_3, θ_3) is the current output state vector in the global-coordinate system of the follower while (x_3^n, y_3^n, θ_3) is the newest output state vector. For simplicity, we assume that $o_y = 0$. Hence the new offset lies on the X axis at o_x units from the origin of the follower robot coordinate system. Writing the dynamic equations for the system

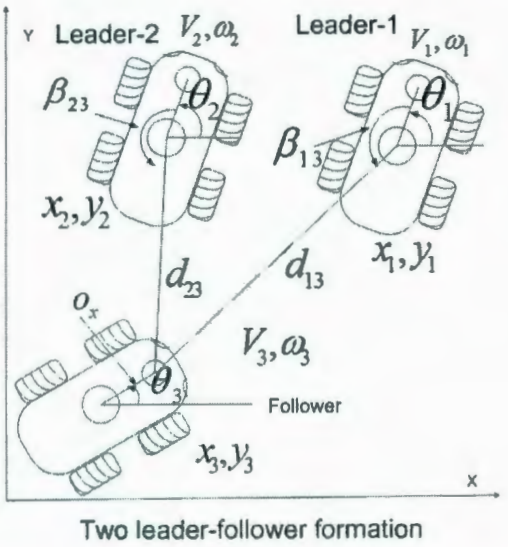


Figure B.1: Three robot formation controller

of Fig.B.1 with the new X and Y values of the follower robot (3rd robot) we get,

$$
\begin{aligned}
d_{13} &= \sqrt{(x_1 - x_3^n)^2 + (y_1 - y_3^n)^2} \\
d_{23} &= \sqrt{(x_2 - x_3^n)^2 + (y_2 - y_3^n)^2} \\
\beta_{13} &= -\theta_1 + \pi + atan2(y_1 - y_3^n, x_1 - x_3^n) \\
\beta_{23} &= -\theta_2 + \pi + atan2(y_2 - y_3^n, x_2 - x_3^n) \\
\theta_{13} &= \theta_1 - \theta_3 \\
\theta_{23} &= \theta_2 - \theta_3
\end{aligned}
$$

Differentiating these with respect to time and simplifying results in,

$$\begin{pmatrix} \dot{d_{13}} \\ \dot{d_{23}} \\ \dot{\beta_{13}} \\ \dot{\beta_{23}} \\ \dot{\theta_{13}} \\ \dot{\theta_{23}} \end{pmatrix} = \begin{pmatrix} \cos\gamma_{13} & o_x \sin\gamma_{13} \\ \cos\gamma_{23} & o_x \sin\gamma_{23} \\ \frac{-\sin\gamma_{13}}{d_{13}} & \frac{o_x \cos\gamma_{13}}{d_{13}} \\ \frac{-\sin\gamma_{23}}{d_{23}} & \frac{o_x \cos\gamma_{23}}{d_{23}} \\ 0 & -1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} v_3 \\ \omega_3 \end{pmatrix} + \begin{pmatrix} -\cos\beta_{13} & 0 \\ 0 & 0 \\ \frac{\sin\beta_{13}}{d_{13}} & -1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ \omega_1 \end{pmatrix} +$$

$$\begin{pmatrix} 0 & 0 \\ -\cos\beta_{23} & 0 \\ 0 & 0 \\ \frac{\sin\beta_{23}}{d_{23}} & -1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_2 \\ \omega_2 \end{pmatrix}$$

For obstacle avoidance, if the leader robot 1 is considered as the actual leader and leader robot 2 as the obstacle, we can let the follower drive with a desired d_{12} and d_{23} with respect to robot 1 and 2. Then the obstacle can be avoided (keeping d_{23}) while keeping a desired distance to the actual leader: d_{13}. For wall following procedure depending on the direction of the wall to follow, we make $\beta_{23} = \pm\frac{\pi}{2}$ in addition to keeping the distances to two leaders at some desired values.

Dynamic feedback linearized controllers

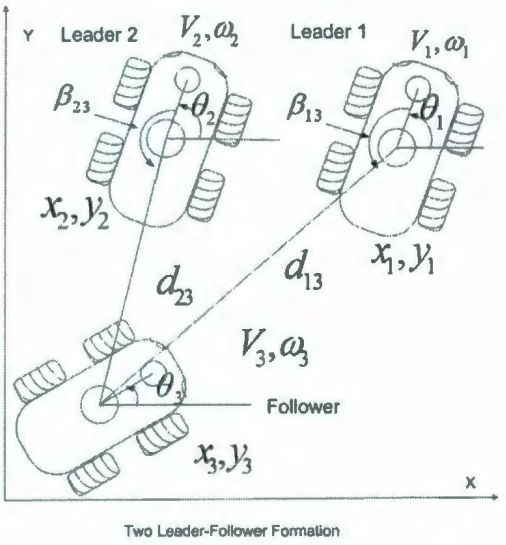


Figure B.2: Three robot formation controller

$$
\begin{aligned}
d_{13} &= \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \\
d_{23} &= \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\
\beta_{13} &= -\theta_1 + \pi + atan2(y_1 - y_3, x_1 - x_3) \\
\beta_{23} &= -\theta_2 + \pi + atan2(y_2 - y_3, x_2 - x_3) \\
\theta_{13} &= \theta_1 - \theta_3 \\
\theta_{23} &= \theta_2 - \theta_3
\end{aligned}
$$

By differentiating the above, we end up with,

$$\begin{pmatrix} \dot{d_{13}} \\ \dot{d_{23}} \\ \dot{\beta_{13}} \\ \dot{\beta_{23}} \\ \dot{\theta_{13}} \\ \dot{\theta_{23}} \end{pmatrix} = \begin{pmatrix} \cos\gamma_{13} & 0 \\ \cos\gamma_{23} & 0 \\ \frac{-\sin\gamma_{13}}{d_{13}} & 0 \\ \frac{-\sin\gamma_{23}}{d_{23}} & 0 \\ 0 & -1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} v_3 \\ \omega_3 \end{pmatrix} + \begin{pmatrix} -\cos\beta_{13} & 0 \\ 0 & 0 \\ \frac{\sin\beta_{13}}{d_{13}} & -1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ \omega_1 \end{pmatrix} +$$
$$\begin{pmatrix} 0 & 0 \\ -\cos\beta_{23} & 0 \\ 0 & 0 \\ \frac{\sin\beta_{23}}{d_{23}} & -1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_2 \\ \omega_2 \end{pmatrix}$$

(Note: This system in Fig. B.2 is different from the earlier dynamic system (static feedback linearized) of Fig. B.1, where the output state vector in Fig. B.2 is the origin itself of the follower as opposed to an offset of o_x from the origin in Fig. B.1). Through applying dynamic extension of $\xi = v_3$ and introducing the time derivative of v_3, which is $\dot{v_3} = \dot{\xi} = a_3$ to the equation above and then differentiating,

$$\ddot{f} = G(f, \dot{f}, \xi)\hat{\mathbf{u_3}} + F_1(f, \dot{f}, \xi)\hat{\mathbf{u_1}} + F_2(f, \dot{f}, \xi)\hat{\mathbf{u_2}} + \mathbf{L} \tag{B.2}$$

$$G = \begin{pmatrix} \cos\gamma_{13} & \xi\sin\gamma_{13} \\ \cos\gamma_{23} & \xi\sin\gamma_{23} \\ \frac{-\sin\gamma_{13}}{d_{13}} & \frac{\xi\cos\gamma_{13}}{d_{13}} \\ \frac{-\sin\gamma_{23}}{d_{23}} & \frac{\xi\cos\gamma_{23}}{d_{23}} \end{pmatrix}, \quad F_1 = \begin{pmatrix} -\cos\beta_{13} & -\xi\sin\gamma_{13} \\ 0 & 0 \\ \frac{\sin\beta_{13}}{d_{13}} & \frac{-\xi\cos\gamma_{13}}{d_{13}} \\ 0 & 0 \end{pmatrix}, F_2 =$$

$$\begin{pmatrix} 0 & 0 \\ -\cos\beta_{23} & -\xi\sin\gamma_{23} \\ 0 & 0 \\ \frac{\sin\beta_{23}}{d_{23}} & \frac{-\xi\cos\gamma_{23}}{d_{23}} \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} v_1\dot{\beta_{13}}\sin\beta_{13} - \xi\dot{\beta_{13}}\sin\gamma_{13} \\ v_2\dot{\beta_{23}}\sin\beta_{23} - \xi\dot{\beta_{23}}\sin\gamma_{23} \\ \frac{\xi\dot{d_{13}}\sin\gamma_{13}}{d_{13}^2} + \frac{v_1\dot{\beta_{13}}\cos\beta_{13}}{d_{13}} - \frac{v_1\dot{d_{13}}\sin\beta_{13}}{d_{13}^2} - \frac{\xi\dot{\beta_{13}}\cos\gamma_{13}}{d_{13}} - \dot{\omega_1} \\ \frac{\xi\dot{d_{23}}\sin\gamma_{23}}{d_{23}^2} + \frac{v_2\dot{\beta_{23}}\cos\beta_{23}}{d_{23}} - \frac{v_2\dot{d_{23}}\sin\beta_{23}}{d_{23}^2} - \frac{\xi\dot{\beta_{23}}\cos\gamma_{23}}{d_{23}} - \dot{\omega_2} \end{pmatrix}$$

$f = [d_{13} \quad d_{23} \quad \beta_{13} \quad \beta_{23}]^T$ is the system output. $u_1 = [a_1 \quad \omega_1]$ and $u_2 = [a_2 \quad \omega_2]$ are the exogenous input by the leader robot 1 and 2 respectively. And $u_3 = [a_3 \quad \omega_3]^T$ are follower's inputs. a_1, a_2 and a_3 are the linear accelerations of robot 1,2 and 3 respectively. The distances d and β's can be set in such a way that the obstacles are avoided and the walls are followed as in the case of static feedback linearization.

Bibliography

[1] D. Rus, B. Donald, and J. Jennings. Moving furniture with a team of autonomous robots. *in Proc. IEEE/RSJ IEEE. Int. Conf. Intelligent Robots and Systems*, pages 556–561, August 1995.

[2] M. Mataric, M. Nilsson, and K. Simsarian. Cooperative multi-robot box pushing. *in Proc. IEEE/RSJ IEEE. Int. Conf. Intelligent Robots and Systems*, pages 235–242, August 1995.

[3] T. Balch and R.C. Arkin. Behaviour-based formation control for multi-robot systems. *IEEE T. on Robotics and Automation*, 14(6):926–939, August 1998.

[4] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzerm, and C.J. Taylor. A vision-based formation control framework. *IEEE T. on Robotics and Automation*, 18(5):813–825, October 2002.

[5] Wei Ren. Consensus based formation control strategies for multi-vehicle systems. *American Control Conference*, June 2006.

[6] R.R. Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics*, 34(2):138153, May 2004.

[7] I.R. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. *IEEE Pervasive Computing*, 4(1):72–78, March 2005.

[8] D. Voth. A new generation of military robots. *IEEE Intelligent Systems*, 19(4):23, August 2004.

[9] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195207, February 1993.

[10] Mathieu Lemay, Francois Michaud, Dominic Letourneau, and Jean marc Valin. Autonomous initializations of robot formations. *IEEE International Conference on Robotics and Automation (ICRA)*, 3:3018– 3023, April 2004.

[11] Yu-Cheng Chen and Yin-Tien Wang. Dynamic role assignment algorithm for robot formation control. *IEEE/ASME international conference on advanced intelligent mechatronics*, pages 1–6, September 2007.

[12] A. Das, James P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE T. on Robotics and Automation*, 17(6):905–908, December 2001.

[13] K.H. Tan and M.A. Lewis. Virtual structures for high precision cooperative mobile robot control. *Auton. Robots*, 4:387–403, Oct 1997.

[14] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.

[15] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of IEEE*, 95(1):215–233, January 2007.

[16] Wei Ren, Randal W. Beard, and Ella M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, April 2007.

[17] W. Ren. Consensus strategies for cooperative control of vehicle formations. *IEEE Control Theory Appl*, 1(2):505–512, April 2007.

[18] Omar A.A. Orqueda and Rafael Fierro. Visual tracking of mobile robots in formation. *American Control Conference*, pages 5940–5945, 2007.

[19] Rent Vidal, Omid Shakemia, and Shankar Sastry. Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. *IEEE International Conference on Robotics and Automation (ICRA)*, 1:584–589, 2003.

[20] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transaction on Automatic Control*, 50(1):121–127, 2005.

[21] A. De Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. *Robot Motion Planning and Control*, 1998.

[22] Claude Samson. Time-varying feedback stabilization of car-like wheeled mobile robots. *International Journal of Robotics Research*, 12(1):55–64, 1993.

[23] C. Canudas de wit, H. Khennouf, C. Samson, and O.J. Sordalen. *Nonlinear Control Design for mobile Robots.* Y.F Zheng Ed. Singapore: World Scientific, 1993.

[24] Xie Feng and R. Fierro. First-state contractive model predictive control of nonholonomic mobile robots. *American Control Conference*, pages 3494–3499, June 2008.

[25] M. Sisto and Dongbing Gu. A fuzzy leader-follower approach to formation control of multiple mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2515–2520, 2006.

[26] M. Rosencrantz, G. Gordon, and S.Thrun. Decentralized sensor fusion with distributed particle filters. *Proceedings of the Conference on Uncertainty in AI (UAI)*, 2003.

[27] Magnus Egerstedt and Xiaoming Hu. A hybrid control approach to action coordination for mobile robots. *Automatica*, 38(1):125–130, 2002.

[28] Jana Kosecka and Ruzena Bajcsy. Discrete event systems for autonomous mobile agents. *Robotics and Autonomous Systems*, 12:187–198, 1994.

[29] Tarek M. Sobh and Beno Benhabib. Discrete event and hybrid systems in robotics and automation. *IEEE Robotics and Automation Magazine*, 4(2):16–19, June 1997.

[30] W.M. Wonham and P.J. Ramadge. Modular supervisory control of discrete-event systems. *Mathematics of Control, Signals and Systems*, 1:13–30, Feb 1988.

[31] Christos G. Cassandras and Stephane Lafortune. *Introduction to discrete event systems.* Springer, 1999.

[32] Alberto Isidori. *Nonlinear Control Systems.* Springer, 1989.

[33] Felix L. Chernousko, Igor M. Ananievski, and Sergey A. Reshmin. *Control of Nonlinear Dynamic Systems: Methods and Applications.* Springer, 2008.

[34] R.W. Brockett, R.S. Millmann, and H.J. Sussmann. *Differential Geometric Control Theory.* Birkhauser, 1983.

[35] C. Canudas de Wit and O.J. Sordalen. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE. Trans on Automatic Control*, 37(11):1791–1797, 1992.

[36] Hassan K. Khalil. *Nonlinear Systems.* Prentice Hall, 1996.

[37] Wei Ren, Haiyang Chao, William Bourgeous, Nathan Sorensen, and YangQuan Chen. Experimental validation of consensus algorithm for multivehicle cooperative control. *IEEE Transaction on Control Systems Technology*, 16(4):745–752, July 2008.

[38] J.R. lawton, R.W. Beard, and B. Young. A decentralized approach to formation maneuvers. *IEEE Transaction on Robotics and Automation*, 19(6):933–941, 2003.

[39] Zhiqiang Cao, Liangjun Xie, Bin Zhang, Shuo Wang, and Min Tan. Formation constrained multi-robot system in unknown environments. *IEEE International Conference on Robotics and Automation (ICRA)*, 1:735–740, September 2003.

[40] J.H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27:171–194, 1999.

[41] Shuzhi Sam Ge, Cheng-Heng Fua, and Khiang Wee Lim. Multi-robot formations: queues and artificial potential trenches. *IEEE International Conference on Robotics and Automation (ICRA)*, 4:3345–3350, April 2004.

[42] Stephen Spry and J. Karl Hedrick. Formation control using generalized coordinates. *in Proceedings of IEEE International Conference on Decision and Control*, pages 2441–2446, December 2004.

[43] G. M. Hassan, K.M. Yahya, and Ihsan ul Haq. Leader-follower approach using full-state linearization via dynamic feedback. *IEEE Conference on Emerging Technologies*, pages 297–305, November 2006.

[44] J. Yan and R.R. Bitmead. Coordinated control and information architecture. *in Proceedings of IEEE International Conference on Decision and Control*, 4:39193923, December 2003.

[45] Ronald C. Arkin. *Behavior based robotics.* MIT Press, 1998.

[46] Rajibul Huq, George K. I. Mann, and Raymond G. Gosine. Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation. *Appl. Soft Comput.*, 8(1):422–436, 2008.

[47] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. A dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, pages 1070–9932, March 1997.

[48] G. Oriolo, A. De Luca, and M. Vendittelli. Wmr control via dynamic feedback linearization: Design, implementation and experimental validation. *IEEE Trans. Contr. Syst. Technol.*, 10(6):835–852, 2002.

[49] Rebel : Recursive bayesian estimation library and toolkit for matlab. *http://choosh.csee.ogi.edu/rebel/.*

[50] Rudolph van der Merwe and Eric Wan. Sigma-point kalman filters for probabilistic inference in dynamic state-space models.

[51] R. Van der Merwe and E. Wan. The square root kalman filter for state and parameter estimation. *International conference on Acoustics, speech and signal processing(ICASSP)*, 2001.

[52] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[53] M. Lawford and W.M. Wonham. Supervisory control of probabilistic discrete event systems. *Circuits and Systems, Proceedings of the 36th Midwest Symposium on*, 1:327–331, August 1993.

[54] Vera Pantelic, Steven Postma, and Mark Lawford. Probabilistic supervisory control of probabilistic discrete event systems. *Automatic Control, IEEE Transactions on*, 54(8), Aug. 2009.

[55] Chong Chee-Yee and S.P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247-1256, Aug. 2003.

[56] Rudolph van der Merwe and Eric Wan. *Kalman Filtering and Neural Networks*. Wiley, 2001.

[57] T. Lefebvre, H. Bruyninckx, and J. De Schutter. A new method for nonlinear transformation of means and covariances in filters and estimators. *IEEE Tran. on Automatic Control*, 47(8), 2002.

[58] M. Norgaard, N. Poulsen, and O. Ravn. New developments in state estimation for nonlinear systems. *Automatica*, 36, 2000.