OPTIMAL LEADER-FOLLOWER FORMATION CONTROL USING DYNAMIC GAMES

Omid Danesh Shahraki

A thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science
Concordia University
Montréal, Québec, Canada

May 2015

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Omid Danesh Shahraki**

Entitled: **Optimal Leader-follower Formation Control using Dynamic Games**

and submitted in partial fulfillment of the requirements for the degree of

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Ming–Yuan Chen __________________ Chair

Dr. Youmin Zhang __________________ Examiner

Dr. Wei–Ping Zhu __________________ External Examiner

Dr. Chun–Yi Su __________________ Supervisor

Dr. Jamal Bentahar __________________ Co–Supervisor

Approved ________________________________

Chair of Department or Graduate Program Director

Abstract

Optimal Leader-follower Formation Control using Dynamic Games

Omid Danesh Shahraki

Formation control is one of the salient features of multi–agent robotics. The main goal of this field is to develop distributed control methods for interconnected multi–robot systems so that robots will move with respect to each other in order to keep a formation throughout their joint mission. Numerous advantages and vast engineering applications have drawn a great deal of attention to the research in this field.

Dynamic game theory is a powerful method to study dynamic interactions among intelligent, rational, and self-interested agents. Differential game is among the most important subclasses of dynamic games, because many important problems in engineering can be modelled as differential games.

The underlying goal of this research is to develop a reliable formation control algorithm for multi–robot systems based on differential games. The main idea is to benefit from powerful machinery provided by dynamic games, and design an improved formation control scheme with careful attention to practical control design requirements, namely state feedback, and computation costs associated to implementation. In this work, results from algebraic graph theory is used to develop a quasi-static optimal control for heterogeneous leader–follower formation problem. The simulations are provided to study capabilities as well as limitations associated to this approach. Based on the obtained results, a finite horizon open-loop Nash differential game is developed as adaptation of differential games methodology to formation control problems in multi–robot systems. The practical control design requirements dictate state-feedback; therefore, proposed controller is complimented by adding receding horizon approach to its algorithm. It leads to a closed loop state-feedback formation control. The simulation results are presented to show the effectiveness of proposed control scheme.

Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

1.1 Motivation

Mobile robotics is one of two major subsections of broader field of robotics. It has seen a great deal of attention and progress over the last few decades. More recently, thanks to advent of new technologies such as new actuators, sensors and computation units, mobile robots became far more capable than their primitive ancestors and at the same time much cheaper to build and maintain. This has enabled researchers worldwide to join the efforts and triggered new and fruitful research areas.

Over the last decade the topic of team work between multiple robots also known as multi–agent robotics has emerged around the core topic of mobile robotics. As number of mobile robots in research problems increased, consequently the complexity of the problems. The same control, path planning, and perception challenges in a single robot problem multiplied as multiple mobile robots were to participate in a new competitive or collaborative setting, and a whole new level of complexity has been introduced to the research in this field.

Very interesting topics of Multi agent robotics, cooperative control and path planning addresses problems with only a few mobile robots, to the scenarios where tens or even hundreds of homogeneous or heterogeneous mobile robots are involved. This new context necessitate an entirely new machinery to analyse and solve multi –robots problems.

At the beginning researchers attempted to stretch and expand common control methods to multi–robot problems. Linear and non-linear control methods were explored [1] then it was time to optimal control, and Receding Horizon control to be adapted for these new problems [2]. Others saw a virtue in abstraction and application of network science in multi–agent robotics [3] as a more profound view to the dynamics of cooperation and competition. A new insight who has been formed by introduction of algebraic graph theory over the multi–agent robotics and became the dominant approach to lay foundation for further research in mobile robotics.

Among all different methods, an original idea formed around application of Game theory in this context [4]. By considering each robot in a multi–robot scenario as a rational decision maker with biassed individual interests in the final results, one can formulate a game and use game theoretic shades to gain unique insight about local and global interaction between robots and possible outcome of these interactions.

1.2 Game theory in formation control

Nested in economics with wide applications in areas as diverse as political science, biology, psychology and more recently engineering, game theory studies problem of strategic decision making. It provides a valuable insight to the logical aspect of decision making for both humans and non-human decision makers through mathematical modelling of conflict and cooperation among intelligent, rational and self-interested agents.

In the literature, games are classified as static games and dynamic games. In static games the entire game takes place in one instant of the time. All the players make their choice at once, simultaneously, and act based on it. Apparently and dependant on their action each player receives a pay-off [5]. Whereas in dynamic games we think of the game as being played over a number of finite time intervals. In such situation each player changes its strategy based on the action of others, from one interval to the other. This results in a dynamic decision making. Dynamic games offer an ideal machinery to find solutions for a wide range of problems. In nature they are able to capture four aspects of a complex problem naming optimization, multi–agent characteristics, robustness in presence of a dynamic environment, and enduring consequences of decisions. Such a powerful machinery provides a promising mathematical foundation to solve multi–agent robotics problems.

Among the most important subclasses of dynamic games are differential games. Differential game aims to predict behaviour of players dealing with conflicts of interests in context of a dynamically evolving environment. It provides useful analysis about the situation, based on its mathematical formulation. It is called differential games because the dynamic of the players in the game as well as the evolving environment around them are modelled as differential equations. Differential games merged as evolution of game theory and optimal control theory.
Rufus Issacs (1951) was pioneer to work on this set of problems, more than half a century later differential games have a wide application in economics, science and engineering.

While there is a well developed research on pursuit-evasion differential games in economics, engineering applications of differential games are diverse and are mostly formulated as a set of Linear quadratic differential equations known as *Linear Quadratic differential games*. Thankfully, there are mature theoretical development and efficient numerical methods to solve these class of problems.

Differential game is in essence extension of static non-cooperative continuous-kern game theory by adopting optimal control theory's principles for dynamic environments. Optimal control theory addresses the question of optimal solution to the planning problems involving dynamic systems, where the states of the system change over time based on control inputs. According to this definition differential game also can be considered as extension of optimal control theory. Instead of being affected by only one input, dynamic of the system is now controlled by multiple inputs, as each of which is decided by different players. As the result, there is more than one cost function involved in finding the optimal solution, with each player now having a possibly different cost function. Whereas if all the players had the same cost function, problems known as team game, the differential game problem could be mapped down to an optimal control problem. Table 1 places differential games with respect to optimization and game theory.

Table 1: Problem type classification from optimization to differential game.

	Single Player	*Multi Player*
Static	Optimization	Static Game
Dynamic	Optimal Control	Differential Game

In differential games firstly the environment where our players are located is modelled. For this purpose a set of differential or difference equations can be used to mathematically model the environment. These equations capture the effect of actions taken by players over the course of the game on the their surrounding environment. As the second part, agent's objectives are defined which are commonly formalized as cost or utility functions. Throughout the game each agent tries to minimize its cost function, subject to a specific dynamic model of the environment, so that it finds the best strategy in the game. To do so techniques developed in the context of optimal control theory is being used to solve the dynamic game.

In optimal control theory there are two main approaches to find the optimal input. These two approaches are also well adopted to the realm of differential games. Introduced by Bellman in 1950s, *dynamic programming* seeks for solution of optimal control problems as function of current state and the time, providing a closed-loop feedback control. While *Pontryagin's maximum principle* the other major approach to optimal control problems, leads to optimal solutions based on only the time and initial states which results in an open-loop control.

Counterparts of these two ideas in differential games are *Nash equilibrium* and *Stackelberg equilibrium* for non-hierarchical and hierarchical structures, respectively. Thanks to the optimal control theory techniques, one can verify the solution and stability analysis of these games using the same methods used in optimal control.

As described before, differential games provide a powerful analytical tool capable of including optimization and multi–agent characteristics where robustness in presence of a dynamic environment is important and decision making happens in a decentralized fashion. Combination of these features best tailors what is needed to address a problems in multi–agent robotics, where dynamics of the robot can replace the differential equations needed for the game and the robots input are derived by solving the cost function associated to its individual objective.

1.3 Literature Review

This section presents a review of the relevant literature on multi–agent mobile robotics. First a brief review on formation control will be provided, followed by a review on graph theoretic methods to multi–agent systems. Then a review of game theoretic approaches to multi–agent systems will conclude this section.

As humans we perform better in teams. There are also certain objectives that are impossible to achieve without forming a team. The most popular sports are shaped around two teams playing against each other, and almost every successful business venture is based on teamwork. Implementation of teamwork in the realm of robotics can also be potentially very empowering, specially for mobile robots.

One of the staple examples is formation control. As one of the remarkable features of multi–agent robotics, formation control emerged by inspiring from the natural phenomenons like schools of fish, flock of wildebeests and colonies of bacteria. These behaviours leads to better performances in animal world like avoiding danger, saving energy and all in all a better chance of survival [6]. While in engineering its application are diverse including automated highway systems [7], cooperative robot reconnaissance [8], manipulation operation [9], flight formation control [10], [11], [12], satellite clustering [13], distributed sensor networks [14], self reconfiguration MEMS [15], etc.

Leader follower, behaviour based and virtual leader are three main categories based on which formation control has been studied within the literature. In leader follower structure robots are following their neighbours and essentially the leader robots either by keeping a certain the distance with two neighbours, or controlling a distance and an angle with neighbouring robots in the chain. Behaviour based formation control can be achieved by defining certain behaviour protocols for each individual robot that leads to desirable team behaviour. Research has shown that behaviour based formation control well suites uncertain environments but they lack robust theoretical analysis. And finally the virtual leader structure as it is obvious by its name uses a virtual leader robot that drives the formation toward desired position.

In [3] author provide theoretical framework and algorithmic tools to extend theory of graphs from static graphs to dynamic graphs. In a static graph the structure of vertices and edges and weights associated to them is time invariant, whereas in dynamic graphs topology of the interaction among its elements is strongly time and state dependent. This work lays foundation to better understand the intricate stability and performance characteristics of distributed dynamic systems such as distributed space system design. The author introduces the state-driven dynamic graphs and provides insight over invariance and reachability properties of these structures.

A local averaging rule to study discrete-time model of n autonomous agents has been considered in [16], where all moving with the different headings, while maintaining the same speed. Agents located inside or on a circle of specific radius around agent i are considered as its *neighbours* and a local rule decides the heading of agent i based on average of its own heading and heading of its neighbours. This work is based on [17] and Jadbabaie *etal* provide theoretical explanation to it. The system is perceived as a stable switched linear system, but for which there dose not exist a common quadratic Lyapunov function. This is done through a switching signal σ which depends on the models initial heading vector, and radius r of the circle that defines the neighbours.

Olfati Saber *etal* [18] studies convergence analysis of an agreement protocol for a network of integrators with fixed or switched topology and a directed information flow. New concepts are introduced in this work using algebraic graph theory and matrix theory. Here a connection is created among Fiedler eigenvalue and Laplacian of graph and the performance of the agreement protocol.

The problem of distributed formation control for a group of autonomous agents is studied in [19]. Here the objective is for a group of mobile agents such as mobile rovers or UAVs, to reach a specified formation, move around while maintaining the formation and reconfigure from one formation to others. Three formation strategies are proposed cyclic pursuit and formation schemes developed based on inter-neighbour interactions either in an undirected graph or in general directed graph case. In this point there is no kinematic constraints imposed on the agents and they can move around freely. Various game theoretic approaches have been used to address the formation control problem.

A new approach to cooperative control using potential game was proposed in [20], where Consensus problem is modelled as a potential game. Each individual player is assigned to a local objective function, and Nash equilibrium is defined as the action profile who maximizes the potential function, capturing the objective of global planner. Potential games require a detailed alignment between the local objective function and the global objective. And finally, a new class of games called sometimes weakly acyclic games is introduced, which is a weaker notion of potential games.

Authors in [21] formulate consensus problem over leaderless multi–robot structure, with double integrator dynamic, as a cooperative game framework. A decentralized control scheme is developed to achieve consensus over a common value of the agent's output by combining decentralized individual cost function of each agent to form a team cost function. In order to minimize team cost function Pareto optimal solutions are first identified. This minimization problem, subject to dynamic constraint of the robots, forms a standard linear quadratic regulator problem (LQR). The LQR minimization for consensus seeking problem is formulated as a maximization problem subject to a set of Linear Matrix Inequalities(LMIs) after imposing additional constraint to ensure desired consensus. Due to un-uniqueness of Pareto solutions an algorithm is then proposed to numerically solve for the Nash bargaining solution among Pareto frontier solutions. A maximization problem solves for the Nash bargaining solution which results in a unique cooperative strategy for this leader less multi–robot structure.

In [22] Extremum seeking algorithm, originally used in adaptive on-line optimization problem of dynamical systems has been employed to general non-cooperative games. The players generate their actions based solely on the measurement of their individual cost function, whose detailed analytical form may be unknown, and individual cost for each agent can be expresses as both sum of locally defined goals -based on the individual agent's position or action- and a collective goal based on position or action of other agents. By implementing the proposed scheme an adaptive compromise is achieved between this two goals, maintaining connectivity with neighbouring agents and with no need to detailed inter-agent communication or position measurement. Detailed formulation of the proposed method is provided for formation control, rendezvous and coverage control as examples of non cooperative games application in multi–robot problems.

Author in [4] adopts linear quadratic differential game to formation control problem of a multi–robot system with double integrator dynamics. A linear quadratic Nash controller is first developed, then receding horizon method is used to provide state–feedback feature to the proposed controller to guarantee practical control design demands.

1.4 Scope of the Thesis

In this research we aim to develop a improved formation control algorithm for multi–robot systems based on dynamic games. The main idea is to benefit from individualized outlook to agent's interest which is provided by dynamic games, and design an improved formation control scheme with careful attention to practical control design requirements, namely state feedback and computation costs associated to controller implementation.

In this work, results from algebraic graph theory is first used to develop a quasi-static optimal control for heterogeneous leader–follower formation problem. Simulations are provided to study capabilities as well as limitations associated to this approach. Steamed of this results a finite horizon open-loop Nash differential game was developed as adaptation of differential games methodology to formation control problems in multi–robot systems. Practical control design requirements dictates state-feedback, therefore proposed controller was complimented by adding receding horizon approach to its algorithm, leading a closed loop state-feedback formation control. Simulation results are presented to show the effectiveness of the proposed control scheme.

1.5 Organization of the Thesis

This thesis is organized as follows. Chapter 1 contains introduction to the research topic addressed in the thesis, research objectives, contributions and organization of the thesis are presented in the first chapter. Background topics on game theory, graph theory, and optimal control theory is introduced in chapter 2. In Chapter 3 Optimal Quasi-Static Formation Control is discussed, and simulation results are provided for network of six agents to verify the effectiveness of proposed method. In chapter 4 differential game technique is adopted to formation control of networked mobile agents, and a open loop controller is suggested based on an open loop information structure. Chapter 5 brings state feedback feature to differential game approach through Receding Horizon Nash formation control, and simulation results are provided to prove the concept. Conclusion and future works are discussed in the final chapter, Chapter 6.

Chapter 2

2 Background Material

Technical challenges are numerous in development of distributed multi–agent networks especially distributed multi–robot systems. While on board computational power, communication and sensing capabilities of each mobile robot are limited, they have to work together towards a specific mission. Objectives of such mission are being achieved at the system level as individual agents collaborate. In this introductory chapter, we tend to introduce basics of graph theory, game theory and optimal control theory as they lead us to better understand multi–agent robotics problems.

Graph theory provides a convenient structure to understand the intrinsic properties of interactions within a finite or infinite set of elements [23]. Graph theory approach to multi–agent networks is through network abstraction. Communication networks such as wireless communication among robots in a team or an active sensing of the position of other agents through vision based or range sensors are examples of these information exchange systems that we aim to abstract into equivalent graphs. In graph theory we tend to consider the interaction geometry of these networks to analyse and synthesis multi–agent systems. [24].

Game theory provide an excellent mathematical tool to analyse and predict behaviour of individuals in a social context [25]. Examples of such a self-interested rational agents can be Human beings, biological entities, nodes in a power grid or robots working together. They can be involved in a cooperative on non-cooperative game, seeking their own benefits through interaction with other agents. Based on the variety of features involved in description of the problem, games can be classified mainly into static and dynamic games. In static games everything happens simultaneously in one step of the time. While in dynamic games agent make decision in different steps of the time as long as the game continues [5]. Dynamic games can be viewed as extension of optimal control problem to multi–player context.

Optimal control theory addresses the problem of finding the optimal input with respect to some constraints, to control a system along a desired state [26]. Pontryagin's Minimum Principle and Dynamic Programming are two main approaches in optimal control which can be very well adopted in dynamic games as Open-loop information games and state-feedback information games.

Throughout the rest of this chapter we take an in-depth look into these very powerful tools to better understand multi–agent robotics. We start with Optimal Control, then we precede with Graph theory and we finish with Dynamic Game theory.

2.1 Optimal Control

In this subsection we review optimal control theory in context of linear time-invariant systems with quadratic cost function. This section is mainly based on [27] and [28].

Consider the linear time-invariant system:

$$\dot{x}(t) = Ax(t) + Bu(t),\; x(t_0) = x_0, \tag{1}$$

where $x(t) \in R_n$ is the state of the system, and $x(t_0)$ the initial given state of the system. $u(.) \in U$ is the vector of admissible control inputs. One objective could be to find a control function $u(t)$ defined on $[t_0, T]$ which derives the state to small neighbourhood of zero at time T. This is known as *Regulator problem.* If the state of the system presents a set of desired trajectory for a dynamic system, like a mobile robot or a set of economic variables to which revenues are attached, and u represents investment actions, the objective might be to control the value of these variables as quickly as possible towards some desired level which is known as *Tracking Problem.* In fact if the system is controllable in both problem settings the objective can be accomplished in an arbitrary short time interval. However to accomplish this, one may need a very large control action when the time interval is short. Usually in both economics and physical systems, the use of a large control action during a short time interval is not feasible. On the other hand, linear models are often used as an approximation to real system. Using a very large control action might usually drive the system out of the region where the given linear model is valid. Having mentioned this two, it is obvious that using large inputs are not recommended.

Given these considerations it seems reasonable to consider the following quadratic cost function

$$J = \int_0^T x^T(t)Qx(t) + u^T(t)Ru(t)dt + x^T(T)Q_T x(T), \tag{2}$$

where without loss of generality the matrices Q, R and Q_T are assumed to be symmetric. Moreover, we assume that matrix R is positive definite, this account for the fact that we do not allow for any arbitrary large control inputs. The matrices Q, R and Q_T can be used to discriminate between two distinct goals, on one hand some objectives and on the other hand to attain this objectives with as little as possible control action. Usually the matrix Q_T expresses the relative importance attached to the final value of the state variable. Since R is assumed to be positive definite, this problem is generally called the regular linear quadratic control problem. It can be shown that if matrix R is indefinite, the control problem has no solution. The solution of this problem is closely related to the existence of a symmetric solution to the following matrix **Riccati differential equation (RDE)**

$$\dot{K}(t) = -A_T K(t) - K(t)A + K(t)SK(t) - Q, \quad K(T) = Q_T, \tag{3}$$

where $S : BR^{-1}B^T$. The fact that we are looking for the symmetric solution to this equation follows from Q_T being symmetric. This implies that if $K(.)$ is a solution of RDE. By taking the transposition of both sides of equation, $K^T(.)$, it satisfies RDE with the same boundary value.

The linear quadratic control problem presented in here has for every initial state x_0, a solution if and only if the Riccati differential equation has a symmetric solution $K(.)$ on $[0, T]$. If the linear quadratic control problem has a solution, then it is unique and the optimal control in feedback form is

$$u^*(t) = -R^{-1}B^T K(t)x(t), \tag{4}$$

whereas in open-loop form is

$$u^*(t) = -R^{-1}B^T K(t)\Phi(t, 0)x_0, \tag{5}$$

with ϕ the solution of the transition equation

$$\dot{\phi}(t, 0) = (A - SK(t))\phi(t, 0), \quad \phi(0, 0) = I. \tag{6}$$

The solution of the Riccati differential equation can be found by solving a set of linear differential equations. To elaborate this, consider the following non-symmetric matrix Riccati differential equation

$$\dot{\phi}(t, 0) = (A - SK(t))\phi(t, 0), \quad \phi(0, 0) = I, \tag{7}$$

where K, $Q \in R^{m\times m}$, $D \in R^{m\times m}$, $A \in R^{n\times n}$ and $Q \in R^{n\times m}$. The solution of

Riccati differential equation then is closely connected with the set of linear differential equations below

$$\begin{bmatrix} \dot{U}(t) \\ \dot{V}(t) \end{bmatrix} = \begin{bmatrix} A & -S \\ -Q & -D \end{bmatrix} \begin{bmatrix} U(t) \\ V(t) \end{bmatrix}. \tag{8}$$

In brief, if U, V is a solution pair of equation (7) with U nonsingular on the interval $[0, T]$. Then $K(t) = VU^{-1}$ is a solution of Riccati differential equation on $[0, T]$. Conversely if $K(t)$ is a solution of RDE on $[0, T]$ and $U(.)$ is a fundamental solution of

$$\dot{U} = (A - SK(t))U(t), \tag{9}$$

then the pair $U(t)$, $V(t) := K(t)U(t)$ is solution of the RDE on $[0, T]$.

2.2 Graph Theoretic methods

Review of graph theory provided in this section is mainly based on [24] and [29]. A graph g consists of a set of vertices denoted by V, a vertex set with n element is represented by

$$V = \{v_1, v_2, ..., v_n\},$$

and a set of edges E as a particular 2-element subset of $[V]^2$. This set consist of elements of the form $\{v_i, v_j\}$ such that $i, j = 1, 2, ..., n$ and $i \neq j$. Sometimes we refer to vertices and edges of g as $V(g)$ and $E(g)$. A graph is inherently a set theoretic object; however, it can conveniently be represented graphically. the graphical representation of g consist of *dots* and *lines* between v_i and v_j when $v_iv_j \in E$. When an edge exists between two vertices, we call them *adjacent*. The *neighbourhood* $N(i) \subseteq V$ of vertices is defined as the set $\{v_j \in V \mid v_iv_j \in E\}$, that is set of all vertices adjacent to V_i. Subsequently, a *path* of length m in graph g is defined as a sequence of distinct vertices

$$v_{i0}, v_{i1}, ..., v_{im}, \tag{10}$$

such that for $k = 0, 1, ..., m-1$, the vertices v_{ik} and v_{ik+1} are adjacent.

We call the graph g *connected* when for every pair of vertices in $V(g)$, there is a path that has them as end vertices. Otherwise the graph is *disconnected*. A component is a subset of the graph associated with a minimal partitioning of the vertex set, such that each partition is connected. Based on this definition a connected graph has only one component. Consider a graph $g = (V, E)$ and a subset of vertices $S \subseteq V$. A subgraph S consist of vertices in the subset S of $V(g)$ and edges in g that are incident to vertices in S. If a function $w : E \to R$ is given that associates a value to each edge, the resulting graph $G = (V, E, w)$ is a *weighted graph.*

When the edges in a graph are given directions the resulting interconnection is a *directed graph*, denoted by $D = (V, E)$. Notions of adjacency, neighbourhood, subgraph and connectedness are easily extended to directed graphs. A digraph is strongly connected if for every pair of vertices there is a directed path between them. a digraph is called weakly connected if it is connected when considered as a graph. Graphs can also be represented in matrix form. For a graph g, the *degree* of a vertex $d(v_i)$ is equal to the number of vertices that are adjacent to vertex v_i in g. The *degree matrix* of g is a diagonal matrix, containing the vertex-degrees of g on the diagonal that is

$$\Delta(g) = \begin{bmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_n) \end{bmatrix}. \tag{11}$$

The *adjacency matrix* $A(g)$ is the symmetric $n \times n$ matrix encoding of the adjacency relationship in the graph g, in that

$$[A(g)]_{ij} = \begin{cases} 1 & if \quad v_i v_j \in E \\ 0 & otherwise \end{cases} \tag{12}$$

For a graph with arbitrary oriented edges, the $n \times m$ *Incidence matrix* $D(g)$ is defined as

$$D(g) = [d_{ij}], \quad where \quad d_{ij} = \begin{cases} -1 & if \quad v_i \, is \, the \, tail \, of e_j \\ 1 & if \quad v_i \, is \, the \, head \, of e_j \\ 0 & otherwise \end{cases} \tag{13}$$

As it is obvious incidence matrix captures both adjacency and orientation in the graph. Since every edge has one tail and one head, the incidence matrix is a column sum zero matrix.

Another important matrix representation of graph is the *graph Laplacian*, $L(g)$. It can be defined as

$$L(g) = \delta(g) - A(g), \tag{14}$$

in case of an undirected graph, where $\delta(g)$ is the degree matrix of the graph g and $A(g)$ associated adjacency matrix. Another way to form the graph *Laplacian* is defined as

$$L(g) = D(g)D(g)^T, \tag{15}$$

where $D(g)$ is the incidence matrix associated to a graph with oriented edges. Either way both approaches form the same matrix which will be symmetric and positive semi-definite matrix with rows sum to zero. For a weighted graph we can also consider a graph Laplacian of form

$$L(g) = D(g)WD(g)^T, \tag{16}$$

where W is an $m \times m$ diagonal matrix, with $w(e_i)\, i = 1, ..., m$, on the diagonal, and $D(g)$ the incidence matrix. Graph Laplacian has an important role in *Algebraic graph theory*. It suffices to mention few more related conclusions one can draw using graph Laplacian, for example real eigenvalues of Laplacian matrix can be ordered as

$$\lambda_1(g) \leq \lambda_2(g) \leq ... \leq \lambda_n(g),$$

With $\lambda_1(g) = 0$. It has been proven that graph g is connected if and only if $\lambda_2(g) \geq 0$. A cell C is a subset of the vertex set $V = [n]$. A *Partition* of a graph is the a grouping of its node set into different cells. An $r-partition$ π of V, with cells $C_1, ..., C_r$, is said to be *equitable* if each node in C_j has the same number of neighbours in C_i, for all i, j. Let b_{ij} be the number of neighbours in C_j of a node in C_j of a node in C_i. The directed graph, potentially containing self-loops, with the cells of an equitable $r-partition$ π as its nodes and b_{ij} edges from the ith to the jth cells of π, is called the *quotient* of g over π, and is denoted by g/pi.

In the following we will talk about agreement protocols in multi–agent coordination, meaning when a collective of agent are to agree on a joint state value. Simulation results of a simple agreement protocols is presented to help better illustrate this idea. Agreement protocol consists of n dynamic unites which are connected with each other through some information exchange links. The rate of change in state of each unit is considered to be controlled by sum of its relative states with respect to its neighbouring units. Denoting the scaler state of unit i as $x_i \subset R$, one then has

$$\dot{x}_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)), \quad i = 1, ..., n, \tag{17}$$

where $N(i)$ is the set of units neighbouring unit i in the network [24]. When the adopted notion of adjacency is symmetric, can be represented by

$$\dot{x} = -L(g)x(t), \tag{18}$$

where the positive semi-definite matrix $L(g)$ is the Laplacian of the network g and $x(t) = (x_1(t), ..., x_n(t))^T \in R^n$. Equation (20) is knowns as Agreement dynamics in the literature.

An example of agreement protocol is the *Rendevous* problem, where a collection of mobile agents with single integrator dynamics, are to meet at a single location. This location is not given in advance and the agents do not have access to their global positions. They can solely measure their relative displacement with respect to each other.

In Fig. 1 five agents participating in a rendezvous algorithm are simulated as they are connected through an underlying network. This network can for example represent an ad-hoc wireless network, with range of their inter-agent communication depicted by the red circle around each agent, and two agents are connected by dotted lines if they are within this range. From Fig. $1_{(a)} - 1_{(f)}$ they gradually go through the rendezvous algorithm. As results they are approaching toward each other until they find consensus over their position.

Fig. 2 shows gradual agreement on agents joint state value as different agent process the rendezvous algorithm after $T = 2$ seconds.

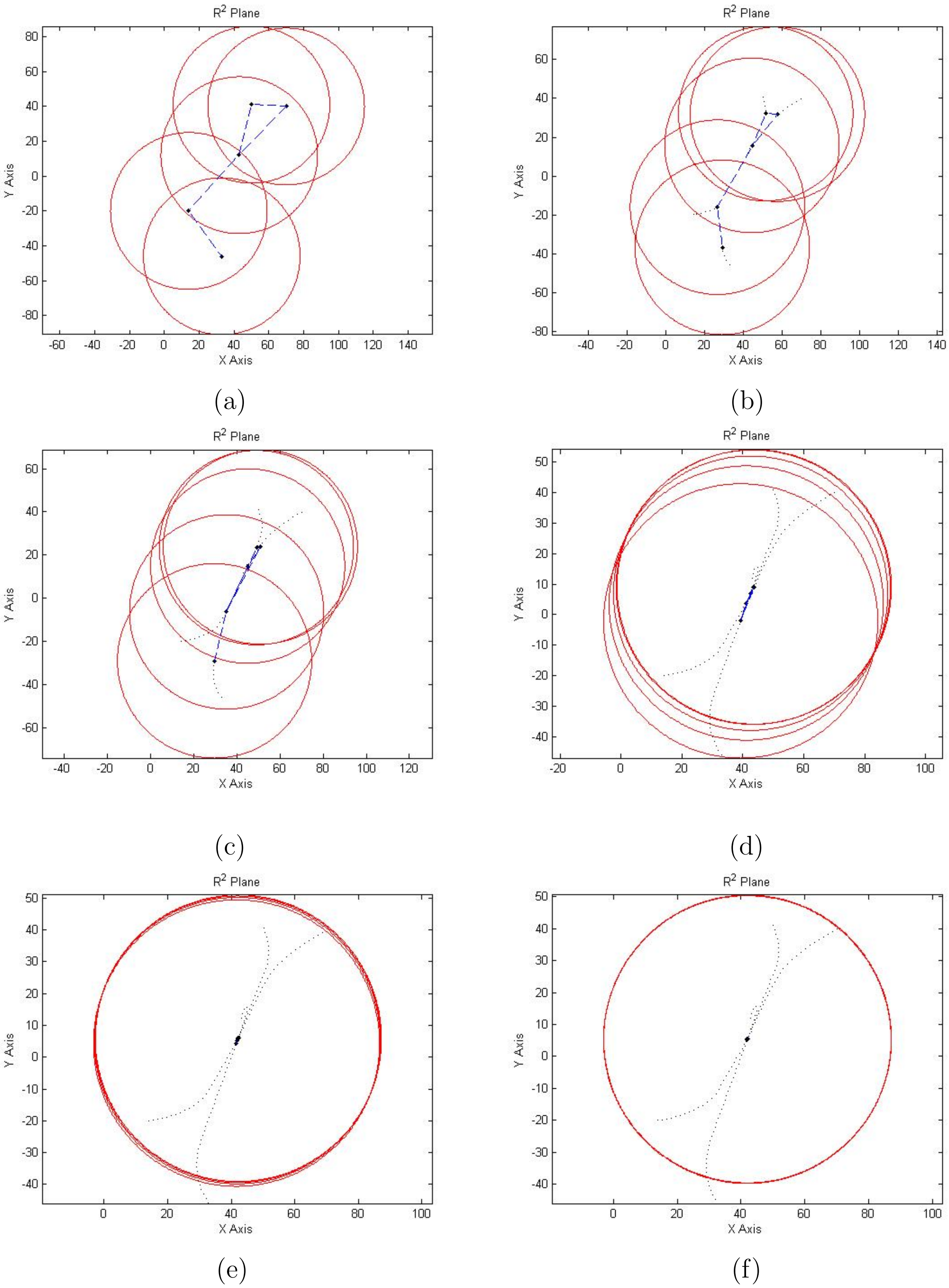

Figure 1: A progression of an agreement protocol to solve the rendezvous problem for five agents.

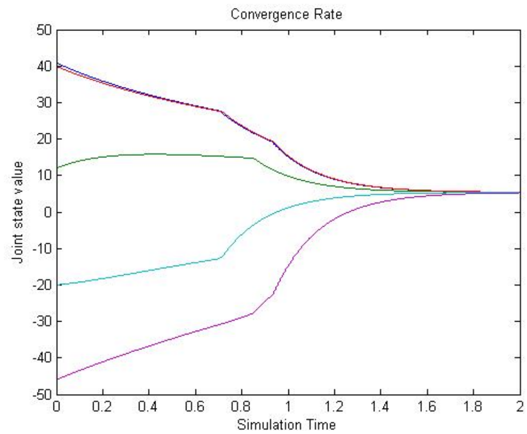

Figure 2: Consensus over joint state value of five robot during the rendezvous problem.

2.3 Differential Game

The brief review of Differential games provided in here is mainly based on [27] and [25]. The differential games framework extends static non-cooperative continuous-kern game theory into dynamic environment by adopting tools, methods, and mode of optimal control theory. Differential games can be viewed as extension of optimal-control problems in two directions: (i) the evolution of state is controlled by a collective input, under control of all the players, and (ii) for each player we have possibly a different objective function as pay-off or cost function. This objective function is defined over time intervals of interest and relevant to the problem. Two main approaches that yield solutions to optimal control problems are *dynamic programming* introduced by Bellman and *maximum principle* introduced by Pontryagin. The former leads to an optimal control that is a function of the state and time, whereas the later leads to the one that is function of time and initial state. These two approaches has also been adapted to differential games. Using the techniques of optimal control theory, not only the solution of differential games can be obtained, but their stability can be analysed.

Players in differential game interact in a dynamic environment in which state of the system and players action directly affect the pay-off of each individual. For example stock holders in a stock market setting are able to make decision to buy or sell shares of a certain company, based on the current and expected future status of the company which is also affected by the over all players (stock holders) decisions. Players themselves could also be dynamic systems, examples of such a players are mobile robotics agents.

The main ingredients of differential games are the state variable, the control variable and action sets of each player, the objective function of the players, the information structure, and the relevant solution concept. As in the case of an optimal-control problem in a differential game setting the state variable evolves over time, driven by players' action. The actions are generated by the strategies of players, mapping from the available information to actions. The differential game is played over time $t \in [0, T]$, where the time horizon of the game could be finite (i.e., $T < \infty$) or infinite (i.e., $T = \infty$). Let N denote the set of players, defined as $N = 1, ..., N$. The state vector for the game is described by $X(.)$, evolving according to equation

$$\dot{X}(t) = F(x(t), a(t)), \tag{19}$$

where $a(t) = [a_1(t), ..., a_N(t)]^T$ is the collection of actions at time t (which can be viewed as a vector), with a_i standing for player i's action, and $i \in 1, 2, ..., N$.
The objective function of a player (i.e., pay-off) is the benefit to be max or minimized in case of a negative pay-off, equivalently. The pay-off function in a differential game can be defined in general as the discounted value of the function's instantaneous pay-off over time. let $U(.)$ denote the instantaneous pay-off function with respect to time t for player i. This instantaneous pay-off for player i is a function of the actions and state variables of all players. The cumulative pay-off is defined as integral of instantaneous pay-off over time, properly discounted, that is

$$j_i = \int_0^T U_i(x(t), a_i(t), a_{-i}(t)) e^{-\rho t} dt, \tag{20}$$

where $a_{-i}(t)$ is the vector of actions of all players except player i, and $\rho > 0$ is the discount factor. Note that to keep the presentation simple, we have not included a cost on the terminal sate here, such as $q_i(x(T))$. For each player, the objective is to optimize this cumulative pay-off by choosing an action $a_i(.)$, i.e., $\underset{a_i}{max}(.) j_i$, more generally by choosing a strategy γ_i.

Here, we need to introduce possible information structures for the players in the game. Even though a higher number of information structure is possible, we will consider here the three most commonly used ones:

- **Open-loop information.** The players have common knowledge of the value of the state vector at initial time $t = 0$, and acquire no further information.

- **Feedback information** At time t, each player has access to the value of the state vector at time t, that is $x(t)$, and no further information.

- **closed loop information** At time t, players have access to the value of the state variables from time 0 to t, namely $x(s)$, $0 \leq s \leq t$, that is, to *Perfect information* on the past and present as far as the state goes.

A mixture of these three information structures are possible with some problems having partially open–loop, feedback, and some Closed-loop information structure. In the context of Nash equilibrium we only consider the open–loop and feedback information structures.

The derivation of the Nash equilibrium for the open–loop structure, involves the solution of N optimal-control problems where, in the generic ith one, the actions of all players except the ith are held fixed as open–loop policies (that is as function of time, and not of state), and maximization of pay-off, $J_i(a_i, a_{-i})$ is carried out with respect to $a_i(.)$. the action variable of player i :

$$\max_{a_i} \quad j_i(a_i, a_{-i}) = \int_0^T U_i(x(t), a_i(t), a_{-i}(t))e^{-\rho t} dt, \tag{21}$$

such that

$$\dot{x}(t) = F(x((t), a(t)), \quad x(0) = x_0, \tag{22}$$

This problem can be solved for each a_{-i} using maximum principle of Pontryagin, discussed earlier in this chapter. The Hamiltonian function is defined as

$$H(x(t), a_i(t), a_{-i}(t), \lambda_i(t)) = e^{-\rho t} U_i(x(t), a_i(t), a_{-i}(t)) + \mu(t) F(x(t), a(t)), \tag{23}$$

where $\mu_i(t)e^{-\rho t}$ is the co-state. A set of necessary conditions for the open-loop solution of the cost function (21) now rise from the maximum principle:

$$\frac{\partial H(x(t), a_i(t), a_{-i}(t), \lambda_i(t))}{\partial a_i(t)} = 0, \tag{24}$$

$$-\frac{\partial H(x(t), a_i(t), a_{-i}(t), \lambda_i(t))}{\partial x_i(t)} = \frac{d\lambda_i(t)}{dt}, \tag{25}$$

given the two-point boundary conditions $x_i(0) = x_0$, $\lambda_i(T) = 0$

For infinite horizon problems, we require in addition that the system be stable under optimal control,

$$\lim_{t \to \infty} x_i(t) = 0,$$

Introducing $\tilde{H}_i := e^{\rho t} H_i$, we obtain a relationship equivalent to the equation, but without the exponential term, and in terms of μ_i, and subject again to the boundary condition $\mu_i(T) = 0$, for all i:

$$-\frac{\partial \tilde{H}(x(t), a_i(t), a_{-i}(t), \lambda_i(t))}{\partial x(t)} + \rho\mu_i(t) = \frac{d\mu_i(t)}{dt}. \tag{26}$$

For the feedback Nash equilibrium, say $\gamma_1^*(x(t), t), ..., \gamma_N^*(x(t), t)$, the underlying optimization problem for each player, say player i, as counterparts of (21) would be

$$max_{\gamma_i(.)} J_i(\gamma_i, \gamma_{-1}^*) = \int_0^T U_i(x(t), \gamma_i(x(t), t), \gamma_{-i}^*(x(t), t)) e^{-\rho t} dt. \tag{27}$$

subject to dynamics (22) with $\mathbf{a}$ replaced by γ.

The tool used in this case is dynamic programming, and particularly HJB equation. If $V_i(x, t)$ denotes the cost-to-go function associated with player i, assuming that it is jointly continuously differentiable in x and t, we have as a sufficient condition for a feedback Nash equilibrium solution the following set of coupled PDEs:

$$-\frac{\partial v_i(x, t)}{\partial t} = \max_{a_i} \left[\frac{\partial v_i(x, t)}{\partial x} F(x, a_i, \gamma_{-i}^*(x(t), t)) + e^{-\rho t} U_i(x(t), a_i, \gamma_{-i}^*(x(t), t)) \right], \tag{28}$$

with boundary conditions $V_i(x, T) \equiv 0$, for $i = 1, ..., N$. The a_i that maximize the right-hand side of the *HJB PDE*s above are clearly function of both x and t and they constitute the feedback Nash equilibrium solution $\gamma_1^*(x(t), t), ..., \gamma_N^*(x(t), t)$ in the differential game.

Chapter 3

3 Optimal Quasi-Static Formation Control

3.1 Introduction

Application of algebraic graph theory in the area of multi–agent robotics has been a very popular topic [30], [16], [31], [19], [32], and to a large extend it became the dominant research approach in this field [33], [3], [18], [34], [35], [34].

Abstracting away the complexities of underlying information exchange networks between robots, and representing it as directed or undirected graphs in one way and diluting the complex dynamics of each robot to rather simple models such a double integrator dynamics helped researchers to focus more on large scale problems and brought a significant contribution to the field. Nonetheless, this body of work is mainly concentrates on the leaderless or homogeneous scenarios. Parallel to this research trend, studies has been done on heterogeneous formation including the leader–follower, string stability and virtual leader based control schemes. The combination of these two approaches brings together two seemingly separate outlooks and provides a power tool to design controllers for formation tasks. This is discussed in this chapter as results from algebraic graph theory is used to control a heterogeneous leader–follower formation problem.

In this chapter we will discuss a heterogeneous formation control problem based on algebraic graph theory. Here we assume no limitation on leaders movements as we allow some agents to directly influence other agent's behaviour as a leader, and they can also have access to global information of the system.

This idea was first introduced by Meng Ji [36] where he used the analogy of a herd with sheep and herding dogs to better explain this approach, loosely called autonomous sheep herding problem [37]. In other words, this problem comes down to how should the herding dogs move in order to manoeuvre the herd toward some desired location, and what are the desired characteristics of sheep to react accordingly. Fig. 3 is very interesting representation of this ideas.

Figure 3: A cartoon representation of the Formation control

Based on the numbers of leaders and the network topology a sufficient condition for controllability, i.e. condition where leaders can move followers to any desired position, can be proposed. This condition helps to select leaders in a way that renders the system controllable.

Later, we propose to use this feature to drive a differential game approach to control several group of robots each representing one component of a larger graph.

In the following quasi-static optimal formation control is studied and simulation results are presented to confirm the effectiveness of developed control scheme.

3.2 Problem Statement

Consider combination of robots with limited sensing and maneuver abilities with another set of robots, agile and well equipped with sensors. To design a decentralized formation control for such a non-homogeneous combination of robots is subject of discussion in this section.

Assuming that the dynamic along each dimension can be decoupled, let $x_i \in \mathbb{N}$, $i = 1, 2, ..., N$, be the position vector of the ith agent, and let $x = [x_1, x_2, ..., x_N]^T$ be the state vector of the group of agents, where N is the total number of agents. Considering widely adopted consensus control strategy for driving the system to a common point (the rendezvous problem) is given by [16]

$$\dot{x}_i = \sum_{j \in N(i)} (x_j - x_i), \tag{29}$$

where $N(j)$ encodes the neighbouring status of the agents with respect to agent i. By considering each agent as nodes of a graph and the edges between the nodes as representative of communication links among neighbouring robots in the system, as discussed in previous chapters, we can add extra insight to the problem.
Adopting graph theoretic methods we can rewrite the equation (29) as

$$\dot{x} = -L(g)x, \tag{30}$$

where $L(g)$ is the graph Laplacian for g, for the definition of $L(g)$ and related properties, refer to chapter 2, section 2.

Considering two different subsets for the agents involved in the multi–agent scenario, one subset with superior sensing and communication abilities taking the role of leaders in the problem and the remaining agents as follower subset. As result the state of the agents x can be divided into two parts, the state of the leaders x_l and those of the followers x_f, where the subscripts l and f represent *leaders* and *followers* subsequently. Therefore number of agents can be written as summation of leaders and followers $N = N_l + N_f$. The Laplacian can also be partitioned as

$$L(g) = \begin{bmatrix} L_f & l_{fl} \\ l_{fl}^T & L_l \end{bmatrix}, \tag{31}$$

where $L_f \in \mathbb{R}^{N_f \times N_f}$, $L_l \in \mathbb{R}^{N_l \times N_l}$ and $L_{fl} \in \mathbb{R}^{N_f \times N_l}$. One can easily form the Laplacian as the product of incidence matrices,

$$L(g) = D(g^\lambda)D(g^\lambda)^T, \tag{32}$$

where λ is an arbitrary orientation assignment to the edges of the graph, $D \in \mathbb{R}^{N \times M}$ is the incidence matrix, $N = |V(g)|$ and $M = |E(g)|$. By using incidence matrix to represent the Laplacian we get,

$$Ł_f = D_f D_f^T,\ L_l = D_l D_l^T,\ l_{fl} = D_f D_l^T. \tag{33}$$

The rendezvous control law in [29] averages the the contribution from all neighbours and it can be used to define a basis for the movement of the followers. In the other words, we will choose to let

$$\dot{x}_f = -L_f\, x_f - l_{fl}\, x_l, \tag{34}$$

Which allows us to continue with the rest of the control scheme.
Theorem 3.1 Given fixed leader positions X_l, the equilibrium point under the followers dynamics in (34) is

$$x_f = -L_f^{-1}\, l_{fl}\, x_l, \tag{35}$$

which is globally asymptotically stable.

3.2.1 Controllability Analysis of the Leader–Follower

In this section we will discuss the controllability issue, and based on [31] we will provide sufficient condition for controllability of such a leader–follower multi–agent system.
Proposition 3.1 The system $(l_f, l_{fl})$is controllable if $\mathcal{G}$ is connected and $N(D_l) \in N(D_f)$.
Proof: For the system (l_f, l_{fl}), the condition in **Preposition 3.1** translates to $v_i(l_f) \notin N(D_f)$, $\forall v_i \in spec(l_f)$, or $D_l D_f^T v_i \neq 0 \forall v_i \in spec(l_f)$. Thus if $N(D_l) \subseteq Im(D_l^T)^\perp = N(D_f)$, the system is controllable.

Note that as a consequence of Theorem 3.1, we have a constructive way of assigning leadership roles to agents in order to ensure controllability.

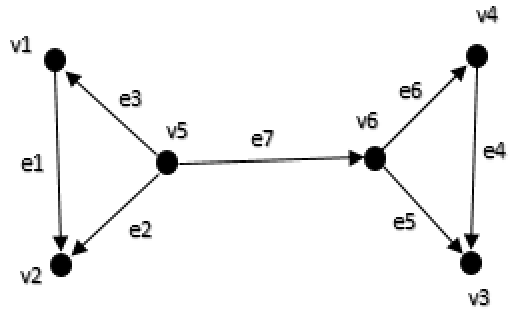

Figure 4: An example graph used for choosing leader

Given a network topology, we first find the null space of D, then select the appropriate rows of D and stack them into new matrix such that the null space of the new matrix is contained in $N(D)$. As an example, consider the directed graph in Fig. 4, where we have

$$D = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}, \tag{36}$$

with

$$N(D) = span \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{37}$$

From the incidence matrix we directly see that by choosing any single agent as follower and the remaining five as leaders, Theorem 3.2 will be satisfied.

It is worth noticing that this sufficient condition is conservative. In many cases we can find configurations with less leaders, which is still controllable. For instance in Fig. 4, we can choose nodes v_1, v_2, v_3, and v_4 as leaders and the system is still controllable.

3.2.2 Optimal Control of Quasi-Equilibrium Process

Now that we studied the question of controllability of the system the discussion moves to the question of how to control the system in order to move it from one equilibrium to the next in a finite time span.

For convenience, we use x instead of x_f, and u instead of x_l in the following equations. Moreover we can replace $-l_f$ with A and $-l_{fl}$ with B. Using this new notations we can rewrite equation (34) as

$$\dot{x} = Ax + Bu, \tag{38}$$

Leaders have no constraints to move therefore

$$\dot{u} = v, \tag{39}$$

where v is the control input. Quasi-static equilibrium for a fixed u will be

$$x = -A^{-1}Bu, \tag{40}$$

The problem here is a quasi-static equilibrium process problem, that means moving (x, y) from an initial point to a final point while satisfying (40). Besides we want to reach to the final point in finite amount of time , so we define the performance function as follows

$$J = \frac{1}{2}\int_0^T (x^T P x + v^T Q v) dt, \tag{41}$$

where $P \succeq 0$ and $Q \succ 0$. The optimal control problem can be formulated as

$$\min_v \quad J. \tag{42}$$

It is well known that such a problem has a solution if the pair (A, B) is controllable, however we can remove this condition as will be discussed in continue. Without loss of generality let consider the standard controllable decomposition as

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{x}_u \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{x}_u \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} u, \tag{43}$$

where x_c and x_u are the controllable and uncontrollable parts respectively. Now, given a fixed u^e, where the superscript e denotes equilibrium, the quasi-static equilibrium is given by

$$0 = \begin{bmatrix} A_{11}x_c^e + A_{12}x_u^e + B_1 u^e \\ A_{22}x_u^e \end{bmatrix}. \tag{44}$$

Since A is invertible and also A_{22}, this means that $x_u^e = 0$. Therefore the quasi-static process will naturally drive $x_u = 0$ to $x_u(T) = 0$ and we can concentrate our attention to the rest of the system

$$\dot{x}_c = A_{11}x_c + A_{12}x_u + B_1 u, \tag{45}$$

whereas $x_u(t) = 0$, on the interval $[0, T]$, we only have

$$\dot{x}_c = A_{11}x_c + B_1 u, \tag{46}$$

and (A_{11}, B_1) is a controllable pair, the point–to–point mapping is always possible. Now we precede with solving for the analytical solution by forming the Hamiltonian

$$\begin{aligned} H &= \tfrac{1}{2}(x^T P x + v^T Q v) + \lambda^T (Ax + Bu) + \mu^T v \\ &= \tfrac{1}{2}(x^T A^T P A x + 2X^T A^T P B u + u^T B^T P B u \\ &\quad + v^T Q v) + \lambda^T (Ax + Bu) + \mu^T v, \end{aligned} \tag{47}$$

where λ and μ are the co-states. The first order necessary optimality condition then gives

$$\begin{aligned} \tfrac{\partial H}{\partial v} &= v^T Q + \mu^T = 0 \to v = -Q^{-1}\mu, \\ \dot{\lambda} = -(\tfrac{\partial H}{\partial x})^T &= -APAx - A^T PAx - A^T PBu - A^T\lambda, \\ \dot{\mu} = -(\tfrac{\partial H}{\partial u})^T &= -B^T PAx - B^T PAx - B^T PBu - B^T\lambda, \end{aligned} \tag{48}$$

in other words, by letting $z = \left[x^T, u^T, \lambda^T, \mu^T\right]^T$, we obtain the following equation

$$\dot{z} = Mz, \tag{49}$$

where

$$M = \begin{bmatrix} A & B & 0 & 0 \\ 0 & 0 & 0 & -Q^{-1} \\ -A^T PA & -A^T PB & -A^T & 0 \\ -B^T PA & -B^T PB & -B^T & 0 \end{bmatrix}, \tag{50}$$

let the initial state be given by

$$z = \left[x_0^T, u_0^T, \lambda_0^T, \mu_0^T\right]^T. \tag{51}$$

Now, the problem is to select λ_0 and μ_0 in such a way that, through this choice, we get

$$u(T) = t_t, \quad x(T) = -A^{-1} B u_T x_T, \tag{52}$$

in order to achieve this, we partition the matrix exponential in the following way

$$E^{MT} = \begin{bmatrix} \phi_{xx} & \phi_{xu} & \phi_{x\lambda} & \phi_{x\mu} \\ \phi_{ux} & \phi_{uu} & \phi_{u\lambda} & \phi_{u\mu} \\ \phi_{\lambda x} & \phi_{\lambda u} & \phi_{\lambda\lambda} & \phi_{\lambda\mu} \\ \phi_{\mu x} & \phi_{\mu} & \phi_{\mu\lambda} & \phi_{\mu\mu} \end{bmatrix}, \tag{53}$$

we can find the initial conditions of the co-state by solving

$$\begin{bmatrix} x_T \\ x_t \end{bmatrix} = \begin{bmatrix} \phi_{xx} & \phi_{xu} & \phi_{x\lambda} & \phi_{x\mu} \\ \phi_{ux} & \phi_{uu} & \phi_{u\lambda} & \phi_{u\mu} \end{bmatrix} \begin{bmatrix} x_0 \\ u_0 \\ \lambda_0 \\ \mu_0 \end{bmatrix}, \tag{54}$$

now let have

$$\Phi_1 = \begin{bmatrix} \phi_{xx} & \phi_{xu} \\ \phi_{ux} & \phi_{uu} \end{bmatrix}, \Phi_2 = \begin{bmatrix} \phi_{x\lambda} & \phi_{x\mu} \\ \phi_{u\lambda} & \phi_{u\mu} \end{bmatrix}, \tag{55}$$

which leads to

$$\begin{bmatrix} \lambda_0 \\ \mu_0 \end{bmatrix} = \phi_2^{-1} \left(\begin{bmatrix} x_T \\ u_T \end{bmatrix} - \phi_1 \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \right), \tag{56}$$

since we are considering a quasi-static process, we have

$$x_0 = -A^{-1}Bu_0, \tag{57}$$

$$x_T = -A^{-1}Bu_T, \tag{58}$$

and consequently the initial conditions of the co-states become

$$\begin{bmatrix} \lambda_0 \\ \mu_0 \end{bmatrix} = -\phi_2^{-1}\Psi \begin{bmatrix} u_0 \\ u_T \end{bmatrix}, \tag{59}$$

where

$$\Psi = \begin{bmatrix} \phi_{xx}A^{-1}B - \phi_{xu} & -A^{-1}B \\ \phi_{xu}A^{-1}B - \phi_{uu} & I \end{bmatrix}, \tag{60}$$

this point-to-point process always have a unique solution; therefore, invertability of Φ_2 is guaranteed as discussed earlier.

3.3 Simulation

The simulation results of the implemented algorithm are presented in this section. This will help us to spot the limitation of the proposed algorithm and provide better insight to fully understand its capabilities, as well as insufficiencies. First a single integrator dynamic is simulated and results are given. To take further step, the same algorithm is expanded to cover formation control for a group of six agents, including three leaders followed by three followers to perform a triangle shape formation.

3.3.1 Single Integrator Model

As discussed throughout the chapter there are elements associated to problem formulation. The quasi-static dynamic of a system is given by

$$\dot{x} = -x - u, \tag{61}$$

and P and Q are set to be both equal to 1. The follower starts the simulation from $x_0 = 1$, and the leader from $u_0 = -1$ and the desired final position for the leader and follower is set to be $x_T = -1$, $u_T = 1$ respectively with $T = 2$.

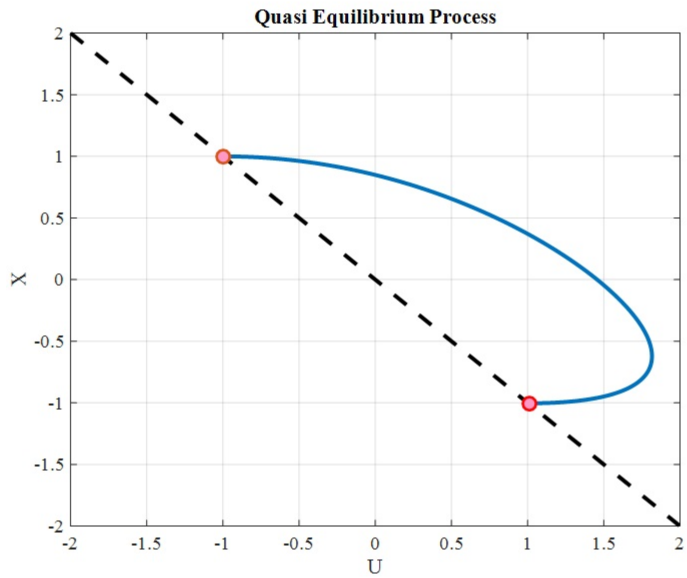

Figure 5: Quasi-Static process.

As shown in Fig. 5 the system starts at $(1, -1)$ and slowly moves to the new equilibrium at $(-1, 1)$ under the optimal control scheme. The dash-doted line depicts the subspace $\{(x, u)\,|x = -A^{-1}Bu\}$, while the solid line is the actual trajectory of the system under the optimal control law.

It is evident that presented formation control based on quasi-static optimal control is capable of the defined task. Now we are interested to verify its capabilities for a larger system.

3.3.2 Formation Control of Six Agents

Simulation results for a formation of six robot agents are presented in this section. Fig. 6 depicts the formation shape and respective position of leaders and followers. As can be seen, three leaders depicted by black circles, in a triangle shape, surround three followers which they form a triangle themselves.

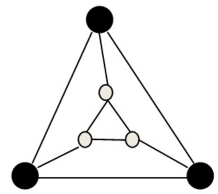

Figure 6: Formation graph topology (black circles: leaders, white circles: followers).

A communication network connects leaders to each other and to their neighbouring followers. This communication network can obviously be captured by a graph. The incident matrix associated to graph representing underlying network between these agents is

$$D = \begin{bmatrix} -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}, \tag{62}$$

It is expected that based on the proposed formation control, followers, under the influence of leaders, will move from an equilibrium to another one within a finite time envelop. We set the simulation time to be $T = 5$ seconds. Leaders and followers start to move from initial position $X_l(0) = \{(0, -1), (1, 1), (-1, -1)\}$.

Under the quasi-static optimal formation control, agents will move from initial position to a defined final position while maintaining their formation. It should be noted that there are no limitation on movement of leader agents. Moreover, followers will adopt consensus control strategy based on [29] during the simulation process. In Eq. (41), P and Q are set to be identity matrices with appropriate dimensions.

Three different final positions are chosen to test the ability of formation control to keep the agents together, while moving them towards the following:

- $X_{l1}(T) = \{(-1, -10), (1, -9), (1, -11)\}$,
- $X_{l2}(T) = \{(9, 0), (11, -1), (11, 1)\}$,
- $X_{l3}(T) = \{(8, -10), (-8, 12), (10, -8)\}$.

In Figs. 7–9 leaders are represented by square, triangle and diamond shapes, while followers are presented as circle, star and asterisk. They start from their initial position, colour coded with black, and move toward their final position shown in red. Fig. 7 is related to robots motion in $X - Y$ plane associated to a translation from their initial position to $X_{l3}(T) = \{(-1, -10), (1, -9), (1, -11)\}$.

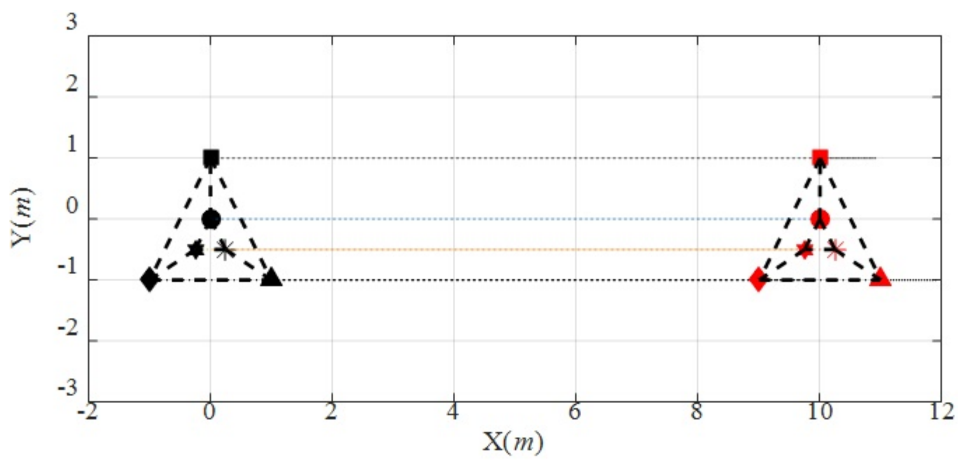

Figure 7: Leader follower formation for six agents. Translation in x direction.

In Fig. 8 the robots start from initial position $X_l(0) = \{(0, -1), (1, 1), (-1, -1)\}$ to $X_{l2}(T) = \{(9, 0), (11, -1), (11, 1)\}$. This requires leaders to perform a 90-degree rotation, and translation in x direction.

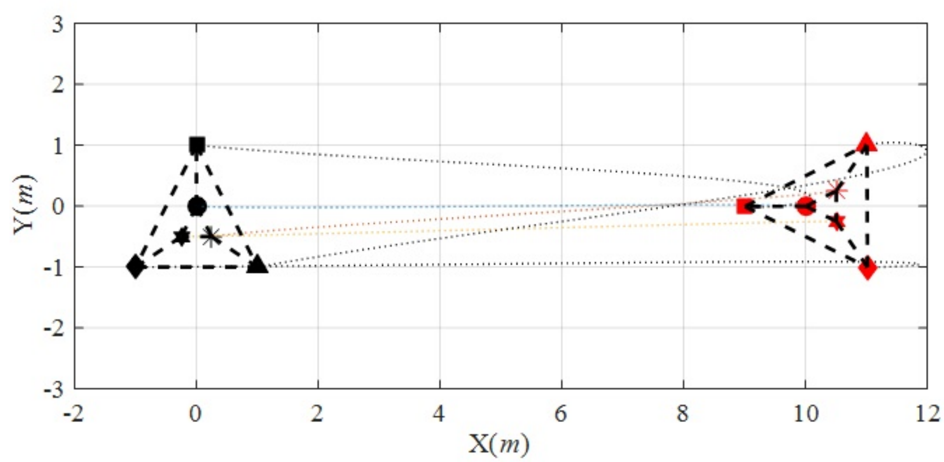

Figure 8: Leader follower formation for six agent.

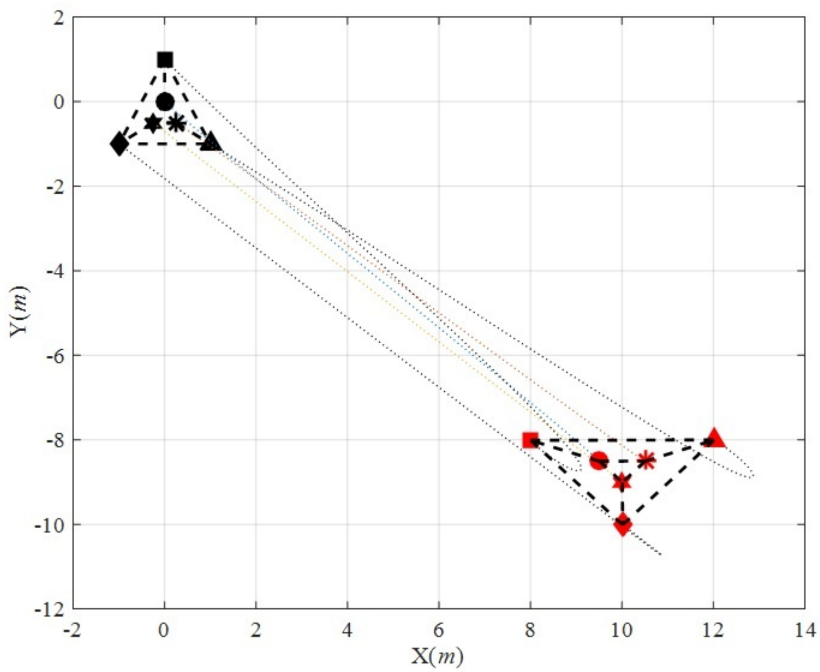

Figure 9: Leader follower formation for six agent.

As can be seen in Fig. 9 during the simulation followers maintain their relative position with respect to each other and leaders. Leaders go through simultaneous translation and rotation to move the entire formation to the defined final location at $X_{l3}(T) = \{(8, -10), (-8, 12), (10, -8)\}$. In simulation results an overshoot can be seen as we follow the movements of the leaders from initial position toward the end of the simulation. This can be explained by reminding the assumption made at the beginning of this chapter to consider no limitation on leaders movements.

3.4 Summary

In Chapter 3 we discussed an optimal control technique to maintain formation between leaders with unrestricted motion, and followers reflecting based on a general consensus strategy.

This algorithm is capable of taking predefined heterogeneous formation from an initial state to desired terminal state within a finite time, however forming the desired graph topology from a random agents layout was not considered in this algorithm. Moreover aside from dual classification of agents to leader and followers, none of the individual differences or specified interests of the agents were taken into account.

Simulation results of the proposed controller for two different layouts of agents proves the reliability of this approach to maintain the formation regardless of drastic difference in initial and desired terminal state of robots in the formation mission. An additional insight provided by the simulation is the fact that the quasi–static optimal formation control is not computationally demanding, and rather simple to implement. This is an interesting feature, making a comparison with the algorithm proposed in the following Chapters.

Chapter 4

4 Differential Game Outlook

4.1 Introduction

In Chapter 2, Section 3 we briefly introduced dynamic games and we defined the boundaries between static games, optimization, optimal control, and dynamic games. In Chapter 3 we discussed formation control of heterogeneous multi–robot system based on Quasi-static optimal control algorithm. Now in this Chapter we introduce differential games as one of the most important subclasses of dynamic games and we adopt differential games to the problem of formation control in multi–robot systems.

Differential games theory is used to solve formation control problem through open loop information structure. This is mainly based on work reported in [4] and [5], where differential game is suggested as a more flexible approach to formation control design.

In formulation of the formation control, one can create an identical team objective for each individual robot so that a common team objective forms as formation goal. This means, all the robots will have the same identical interests throughout the formation control. Another more realistic scenario would be to include individual robots interests, steamed by different dynamics or states, in the global team objective. In this way formation control of multiple robots can become subject to game theoretical reasoning. By using a tracking cost function for the leader robot and a formation keeping cost function for the rest of the team, one can incorporate individual interests in distributed control and have cost functions that are only related to the neighbouring robots and not necessarily the entire team. In that regime each robot in formation control can be modelled as self-interested, rational agent participating in a non cooperative game, where formation control strategy is defined as the Nash equilibrium of the designed game. As all the agents comply with Nash equilibrium, knowing that they will not gain more by unilateral deviation from equilibrium point, formation control is guaranteed for the multi–robot system.

In order to introduce powerful machinery of dynamic games, one has to start with optimal control theory. It provides optimal solution to planning problems involving dynamic systems, where the states of the system change over time, based on the control inputs. According to this definition differential game also can be considered as extension of optimal control theory. Here, instead of being affected by only one input, dynamic of the system is now controlled by multiple inputs, derived by different players. This formulation results in having more than one cost function to be optimized for finding optimal solution, with each player now having a possibly different cost function from the others. Whereas, if all the players had the same cost function the differential game problem could be mapped down to an optimal control problem.

In differential games we aim to predict behaviour of players, involved in a dynamically evolving environment. In this context players can be involved in competition over conflicting interests or cooperation toward a joint objective, known as non-cooperative and cooperative games. Differential game provides useful insight through mathematical formulation. It is called differential game as we model both dynamic of player's action, and their surrounding environment using differential equations. Formation control can be formulated by considering each robot, in a multi–robot scenario, as a rational decision maker with biassed individual interests in the final results. This way one can formulate a game, and use game theoretic shades to gain unique insight about local and global interaction between robots and possible outcome of these interactions.

The formation control problem in this chapter consists of mobile robots with double integrator dynamics, typically chosen because these models can capture the equations of motion for broad range of vehicles. Each robot is interacting through communication channels, only available in between neighbouring robots. In a more abstract way, each mobile robot can be conceived as a node of a graph with communication channels depicted as edges of the graph, which provides a powerful analytical tool to study the effect of local interactions of robots on the overall performance of the network.

Thanks to graph theory, and considering the linear dynamics of the robots, formation control cost function can be modelled as linear–quadratic Nash differential game. In order to find its strategy to interact with other robots, each robot has to minimize the linear quadratic cost function associated to this Nash differential game. This essentially comes down to solving a coupled asymmetric Riccati Differential equation.

Based on the nature of the information available, two possible structure can be considered. Open loop information, and state feedback information. The former, state–feedback structure, although provides more information and leads to better results, is computationally and analytically intricate.

On the other hand, using open-loop information structure in the game is a favourable option. That is due to its analytical tractability for problems with linear differential equations models, and quadratic cost functions. In the open–loop information structure we assumes that the only information available to the players are their present states and the model structure. Meaning that at the beginning of the of the game all players simultaneously make decision about their strategies for the entire period of the game.

In the following adaptation of differential games to formation control problem is studied.

4.2 Formation Model

4.2.1 Robots Dynamic

One of the main applications of formation control is in the area of mobile robotics. Where formation control schemes can provide solution to multi–vehicle problems.

4.2.2 Multi Agent Network Model

Consider a team of m mobile robots with double integrator dynamics. For robot i with n-dimensional coordinates $q_i \in R^n$, the state and control vectors are $z_i(t) = \left[q_i(t)^T, \dot{q}_i(t)^T\right]^T \in R^{2n}$, and $u_i(t) \in R^n (i = 1, ..., m)$. The robot dynamics therefore are

$$\dot{z}_i = az_i + bu_i, \tag{63}$$

where $a = \begin{bmatrix} 0 & I_{(n)} \\ 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ I_(n) \end{bmatrix}$.

Matrix $I_{(n)}$ is the identity matrix of dimension n. State and control vectors of robot i are $u_i(t) \in U$, and $z_i(t) \in Z$, respectively.
By concatenating the states of all m robots of the team into a vector $z = \left[z_1^T, ..., z_m^T\right]^T$ $\in R^{2nm}$, the team dynamics are

$$\dot{z} = Az(t) + \sum_{i=1}^{m} B_i u_i(t), \quad t \geq 0, \tag{64}$$

where $A = I_{(m)} \otimes a$ and $B_i = [0, ..., 1, ..., 0]^T \otimes b$. The operator $\otimes$ represents the Kronecker product. let $z_i^d = \left[(q_i^d)^T, (\dot{q}_i^d)^T\right]^T$be the desired state vector for robot i. The desired team state vector is then represented as $z^d = \left[(z_1^d)^T, ..., (z_m^d)^T\right]^T \in R^{2nm}$. The desired state z_i^d should also have the same dynamics as the multi–robot system dynamic in equation (63)

$$\dot{z}_i^d = az_i^d + bu_i^d, \tag{65}$$

and by concatenating the state function we have

$$\dot{z}^d(t) = Az^d(t) + \sum_{i=1}^{m} B_i u_i^d(t), \quad t \geq 0, \tag{66}$$

In order to optimize the performance a convexity assumption is necessary for optimization algorithms.
Assumption 1(Convexity Assumption): U is a compact and convex subset of R^n containing the origin in its interior, and Z is a convex, connected subset of $\in R^{2n}$ containing z_i^d in its interior, for every i.

In the next section we discuss the information exchange network among the robots.

4.2.3 Formation Graph

The information exchange network between the robots in a multi–robot problems can be captured by forming the graph of underlying information structure [24]. A node in a graph corresponds to a robot and the edge between the nodes captures the dependences of the interconnections. A direct graph $g = (V, E)$ consists of a set of vertices $V = \{v_1, ..., v_m\}$, indexed by the robots in a team, and a set of edges $E = \{(v_i, v_j) \in V \times V\}$, containing ordered pairs of distinct vertices. Assuming the graph has no loops, i.e., $(v_i, v_j) \in V$ implies $v_i \neq v_j$. A graph is connected if for any vertices $(v_i, v_j) \in V$, there exist a path of edges in E from v_i to v_j. An edge-weighted graph is a graph in which each edge is assigned a weight. The edge (v_i, v_j) is associated with the weight $W_{ij} \geq 0$. Graph connectivity is necessary condition to be able to control the formation.

Assumption 2 (Connectivity Assumption): Graph $\mathcal{G}$ is connected.

The incidence matrix D of a directed graph g is the $\{0, \pm 1\}$-matrix with row and columns indexed by vertices of V and edges of E, respectively, such that the uvth entry of D is equal to 1 if the vertex u is the head of the edge v, -1 if the vertex u is the tail of the edge v, and 0, otherwise. If graph g has m vertices and $|E|$ edges, then incidence matrix D of the graph g has order $m \times |E|$ [38].

The cohesion and separation of formation control is defined by the desired distance vector $d_{ij}^d = z_i^d - z_j^d$ between two neighbours v_i and v_j. The formation error vector is defined as $z_i - z_j - d_{ij}^d$ for edge (v_i, v_j). Let $\hat{D} = D \otimes I_{(2n)}$. From the definition of the incidence matrix, we know the whole team formation can be expressed in a matrix form

$$\sum_{ij \in E} W_{ij} ||z_i - z_j - d_{ij}^d|| = (z - z^d)^T \hat{D} \hat{W} \hat{D}^T (z - z^d) = ||z - z^d||^2_{\hat{D}\hat{W}\hat{D}^T}, \tag{67}$$

where $\hat{W} = w \otimes I_{(2n)}$ and $w = diag\,[W_{ij}]$ is a diagonal weight matrix with dimension $|E|$. Here we will use $||z - z^d||^2_{\hat{D}\hat{W}\hat{D}^T}$ for $(z - z^d)^T \hat{D} \hat{W} \hat{D}^T (z - z^d)$. Following [24], we define the Laplacian of a graph g as L

$$L = DWD^T, \tag{68}$$

for a directed graph g, the Laplacian L is symmetric and positive semi-definite.

For real value matrices X, Y, U, V with appropriate dimensions, the Kronecker product has the following properties:

$$(X \otimes Y)^T = (X^T \otimes Y^T),$$

$$(X \otimes Y)(U \otimes V) = (XU) \otimes (YV), \tag{69}$$

based on these properties we have

$$\hat{L} = \hat{D}\hat{W}\hat{D}^T = (D \otimes I_{(2n)})(W \otimes I_{(2n)})(D \otimes I_{(2n)}^T) = L \otimes I_{(2n)}, \tag{70}$$

$\hat{L}$ is also symmetric and positive semi-definite. The team formation error is rewritten as follows:

$$\sum_{(ij)\in E} W_{ij}||z_i - z_j d_{ij}^d|| = ||z - z^d||_{\hat{L}}^2. \tag{71}$$

4.2.4 Formation Cost Function

The finite horizon cost function control for robot i can be expressed as follows:

$$J^i(u) = g^i(T, z(T)) + \int_0^T C^i(\tau, z(\tau), u(\tau))d\tau, \tag{72}$$

$$\begin{aligned} g^i(T, z(T)) &= \sum_{(ij)\in E} W_{ij}||z_i(T) - z_j(T)d_{ij}^d||^2, \\ C^i(\tau, z(tau), u(\tau)) &= \sum_{(ij)\in E} \mu_{ij}||z_i(\tau) - z_j(\tau)d_{ij}^d||^2 + \sum_{(ij)\in E} ||u_j(\tau)||_{R_{ij}}^2, \end{aligned} \tag{73}$$

where T is the infinite time horizon and $\mu \geq 0, R_{ij} \geq 0, (i = 1, ..., m)$ are the weight parameters. The cost function (73) can be transformed into a standard linear-quadratic form

$$\begin{aligned} g^i(T, z(T)) &= ||z(T) - z^d(T)||_{k_{if}}^2, \\ C^i(\tau, z(\tau), u(\tau)) &= ||z(\tau) - z^d(\tau)||_{Q_i}^2 + \sum_{(i,j)\in E} ||u_j(\tau)||_{R_{ij}}^2, \end{aligned} \tag{74}$$

where $K_{if} = \hat{L}_{if} = \hat{D}\hat{W}\hat{D}^T$, $\hat{W}_{if} + W_{if} \otimes I_{(2n)}$, $W_{if} = diag\,[W_{ij}]$, $Q_i = \hat{L}_i = \hat{D}\hat{W}_i\hat{D}^T$, $\hat{W}_i = W_i \otimes I_{(2n)}$, $W_i = diag\,[\mu_{ij}]$. K_{if} and Q_i are symmetric and positive semi-definite. The formation cost functions are used to design controllers, which can control robots to have the desired distances d_{ij}^d. To track a specific trajectory z_l^d, the leader robot L should track z_l^d. Thus, the cost function of the leader robot should include a linear-quadratic standard tracking term

$$g^l(T, z(T)) = ||z(T) - z^d(T)||^2_{k_{lf}} + ||z_l(T) - z^d_l(T)||^2_{k_{lf}} = ||z(T) - z^d(T)||^2_{\acute{k}_{lf}},$$

$$\begin{aligned} C^l(\tau, z(\tau), u(\tau)) &= ||z(\tau) - z^d(\tau)||^2_{Q_l} + ||z_l(\tau) - z^d_l(\tau)||^2_{q_l} + ||u_l(\tau)||^2_{R_{ll}} \\ &= ||z(\tau) - z^d(\tau)||^2_{\acute{Q}_l} + ||u_l(\tau)||^2_{R_{ll}}, \end{aligned} \tag{75}$$

where $K_{lf} = Diag\,[W_l]$, $q_l = Diag\,[\mu_l]$, $\acute{K}_{lf} = K_{lf} + diag\,[0, ..., k_{lf}, ..., 0]$ and $\acute{Q}_l = Q_l + diag\,[0, ..., q_l, ..., 0]$. $\acute{K}_{lf}$ and $\acute{Q}_l$ are also symmetric and positive semi-definite. The leader robot can use $k_{lf} = 0$ and $Q_l = 0$, which means the leader robot only tracks the desired trajectory without taking the formation error into account. In such situation, it's the follower robots who keep the formation by following the leader with a fixed distance.

In the following, the weight matrices in the cost function are denoted as K_{if} and Q_i for both leader robots or follower robots. From the state equations (63) and (64) and the cost functions (73) and (74), it can be seen that the formation control is a linear-quadratic tracking problem. By using error state and control vectors, the formation control is viewed as a linear-quadratic regulating problem with $z(t)$ as the state vector and $u(t)$ as the control vector in the following presentation.

4.3 Finite Horizon Open-Loop Nash Differential Game

4.3.1 Nash Differential Games

Each robot in a team can be considered as a self-interested rational agent participating in a differential game. As the state equation of the differential game we have robots dynamic in equation (63) with the initial condition z_0 such that

$$\begin{aligned} \dot{Z}(t) &= Az(t) + \sum_{i=1}^{m} B_i u_i(t) \\ &= Az(t) + Bu(t), \\ z(0) &= z_0, \quad t \geq 0, \end{aligned} \tag{76}$$

where $B = [B_1, B_2, ..., B_m]$, and $u = \left[u_1^T, u_2^T, ..., u_m^T\right]^T$. The cost function J^i is known to each player. In order to find their strategies with respect to other players in the game, each robot has to minimize its own cost function, to find their control input. In the case that all the player have the same cost function, the game will reduce to to a team game, and what happens essentially is that differential game problems comes down to an optimal control problem. Since in our cost functions (73) and (74) the states and control signals are coupled together through the cost function and they are affected by neighbouring robots a different approach should be considered. When players have different cost functions, the optimal control scheme used for one cost function dose not apply to the others, instead the Nash equilibrium have to be found. A collection of strategies for all players in a game that is the best response strategy each player has to with respect to the other players strategies, is called Nash equilibrium. None of the players in the game can gain higher benefits by unilaterally changing its strategy while others keep the strategy decided through Nash equilibrium and as agents participating in the game are assumed to be rational they all seek their best interests and comply with the collective strategies associated to the Nash equilibrium. A collection of strategies $\bar{u}_i(t)$, $(t \geq 0, i = 1, ..., m)$ constitutes a Nash equilibrium if and only if the following inequalities are satisfied for all $u_i(t) \in U$,$(t \geq 0, i = 1, ..., m)$:

$$j^i(\bar{u}_1, ..., \bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}, ..., \bar{u}_m) \leq j^i(\bar{u}_1, ..., \bar{u}_{i-1}, u_i, \bar{u}_{i+1}, ..., \bar{u}_m),\ i = 1, ..., m. \tag{77}$$

There are two main types of information structure in differential games:

- Open–loop information structure.
- State–feedback information structure.

In open-loop information structure each player computes its equilibrium strategy at the beginning of the game solely based on the initial state $z(0)$ and no state feedback is available during the entire control period.

In state-feedback form all the players make their decision based on the current state $z(t)$ and constantly update their decisions based on the evolution of the states. This way state-feedback provides more information with respect to the open-loop information structure and that enables players to make more reasonable decisions throughout the game. However, the analytical complexity associated to state-feedback information structure motivates us to take another approach to design our controller. Using open-loop information structure in combination with receding horizon approach, one can achieve a state-feedback controller: *Receding Horizon Nash control*. In this scheme each of the players in the game computes its own Nash equilibrium strategy at each time instant, but only follows this strategy for one step. in the next step, players compute a new Nash equilibrium by considering the final state of the previous step as their initial state to compute their strategy, and this procedure repeats again.

4.3.2 Linear Quadratic Open-Loop Equilibria

Under the open-loop information structure for a Nash game, the derivation of open-loop Nash equilibria is closely related to the problem of jointly solving m optimal control problem [5]. According to Pontryagin's minimum principle, the condition for an open loop Nash equilibrium for two players games are given in [5] this results can be generalized straightforward to games with m players.

Theorem 4.1: For a m-robot formation control defined as a finite horizon open-loop Nash differential game by (76) and (77), let there exist a solution set $(K_i, t = 0, ..., m)$ to the coupled Riccati differential equations

$$\dot{K}_i = -A^T K_i - K_i A - Q_i + K_i \sum_{j=1}^{m} S_j K_j,$$

$$K_i(T) = K_{if}, \tag{78}$$

where $S_i = B_i R_{ii}^{-1} B_i^T$. Then the formation control has a unique open-loop Nash equilibrium solution for e very initial state as follows:

$$\bar{u}_i(t) = -R_{ii}^{-1} B_i^T K_i(t) \Phi(t, 0) z(0), \tag{79}$$

$$\begin{aligned} \dot{\phi}(t,0) &= \left(A - \sum_{i=1}^{m} S_i K_i(t) \right) \phi(t,0) \\ &= A_{cl}(t)\phi(t,0), \\ \phi(0,0) &= I, \end{aligned} \tag{80}$$

where $A_{cl} = \sum_{i=1}^{m} S_i K_i(t)$ is the closed-loop system matrix. It is easily verified that $z(t) = \phi(t, 0)z_0$. The closed loop system is

$$\dot{z}(t) = A_{cl}(t)z(t), \quad t \geq 0 \tag{81}$$

Remark 1: Due to state assumption 1 (convexity) and 2 (connectivity), the cost function J^i are strictly convex functions of u_i for all admissible control functions u_j, $j \neq i$ and for all z_0. This implies that the conditions following from minimum principle are both necessary and sufficient.

Based on Theorem 1, the solvability of the coupled Riccati differential equation (78) is vital to the finite horizon Nash equilibrium solution. In the following, a necessary and sufficient condition is established for the solvability of the coupled Riccati differential equations.

Define

$$M = \begin{bmatrix} A & -S_1 & \ldots & -S_m \\ -Q_1 & -A^T & 0 & 0 \\ \vdots & \cdots & \ddots & \vdots \\ -Q_m & 0 & 0 & -A^T \end{bmatrix}, \tag{82}$$

and

$$H(T) = \begin{bmatrix} I_{(2nm)} & 0 & \ldots & 0 \end{bmatrix} e^{-Mt} \begin{bmatrix} I_{(2nm)} \\ K_{1f} \\ \vdots \\ K_{mf} \end{bmatrix}, \tag{83}$$

It follows from the results in [5] that the analytic solution of the closed-loop system is

$$Z(T) = \begin{bmatrix} I_{(2nm)} & 0 & \ldots & 0 \end{bmatrix} e^{-Mt} \begin{bmatrix} I_{(2nm)} \\ K_{1f} \\ \vdots \\ K_{mf} \end{bmatrix} H^{-1}(T)z(0). \tag{84}$$

Reference [5] provides an approach to judge if the solution exists for two-player games. This result can be generalized straightforward to m player games. Based on this theorem with m players, the formation control problem has the following result. *Theorem 2:* For a m-robot finite horizon formation control defined as a finite horizon open-loop Nash differential game by (76) and (77), the coupled Riccati differential (78) has a solution for every initial state z_0 on $[0, T]$ if and only if matrix $H(T)$ is invertible.

$Proof$: The formation control of multiple robot systems[13], it is known that $[A, B]$ is stabilizable. As the Laplacian is symmetrical and positive semi-definite $\hat{H}_i \geq 0$ and $\hat{L}_{if} \geq 0$, the symmetrical Riccati differential equations

$$\dot{P}_i = -A^T P_i - P_i A - Q_i + P_i S_i P_i, P_i(T) = K_{if}, \tag{85}$$

have a symmetrical solution P_i on $[0, T]$, for all $i = 1, ..., m$. This result combining with that $H(T)$ is invertible proves coupled Riccati differential equations (78) has a solution for every initial state z_0 on $[0, T]$, as indicated in the Theorem 7.1 and comments for m players game in [5].

Remark 2: The matrix M consist of $(m+1) \times (m+1)$ blocks. $e^{-(MT)}$ also has the same block structure. Denoting by $W_{ij}(T)$ as the ijth block of $e^{-(MT)}$, we have $H(T) = W_{11}(T)$. The invertibility of $H(T)$ depends on M and T. Different T leads to different invertibiltity of $H(T)$. In the finite receding horizon Nash control discusses in next section. T is the length of control horizon. The selection of T in the receding horizon control should guarantee that $H(T)$ is invertible.

4.4 Summary

In chapter 4 we introduced another approach to formation control based on differential games. Using notion of dynamic games within the context of formation control brought us flexibility to incorporate individual interests of each robot in the algorithm, and weight communication links with variety that represent the actual hierarchy of inter-agent communications.

However this was achieved through an open-loop controller, built based on the open-loop information structure as discussed. Meanwhile state feedback is an essential element of a practical control scenario and that renders strategies based on open-loop information structures impractical and insufficient.

In the next Section we propose a solution to this problem through the Receding Horizon methodology.

Chapter 5

5 Receding Horizon Nash Formation Control

5.1 Introduction

In Chapter 3 an optimal control technique is introduced to maintain formation between leaders with unrestricted motion, and followers reflecting based on a general consensus strategy. This algorithm is able to move formation from an initial state to desired terminal state within an finite time. However, forming the desired graph topology from a random agents layout was not considered in this algorithm, and none of the individual differences or specified interests of the agents were taken into account.

In chapter 4 we introduced another approach to formation control based on differential games theory. Using notion of dynamic games within the context of formation control brought us flexibility to incorporate individual interests of each robot in the algorithm, and weight communication links with variety that represent the actual hierarchy of inter–agent communications. However this was achieved through an open–loop controller, built based on a open–loop information structure as discussed. Meanwhile, state feedback is an essential element of a practical control scenario. This renders strategies based on open-loop information structures impractical, and insufficient.

To resolve lack of state feedback, and to be able to use open–loop information structure devised in chapter 4, one can combine open–loop Nash differential game with receding horizon method to provide the state feedback, necessary for practical control design. This new algorithm is proposed in this chapter as *receding horizon Nash control* scheme. This idea has been exploited in differential zero–sum games reported in [39] and [40]. It works in such way that at the beginning of the game all the robot states are read and the first control signal is generated by solving an open–loop Nash controller, at the next step this procedure repeats again with new initial condition after dt time which leads to receding horizon state feedback control.

In the following receding horizon Nash formation control is studied, and simulation results are presented for a triangle formation of four robots to confirm the effectiveness of this control scheme.

5.2 State-Feedback Formation Control

Assuming the current time instant is t and the current state vector is $z(t)$. At each time instant, the receding horizon control uses $z(t)$ as the initial state vector to find the finite horizon open-loop Nash equilibrium $\bar{u}(t)$ based on the following cost function:

$$J^i(t, z(t), u(t)) = g^i(t+T) + \int_t^{t+T} C^i(\tau, z(\tau), u(\tau))d\tau, \tag{86}$$

the receding horizon control signal is defined as

$$\begin{aligned} u_i^*(t, z(t)) &= \bar{u}(t) \\ &= -R_{ii}B_i^T K_i(t)z(t), \end{aligned} \tag{87}$$

As the control signal $u_i^*(t, z(t))$ depends on the current state $z(t)$, the receding horizon Nash control is a state feedback control. The existence condition of the receding horizon Nash control is the same as those of the finite horizon open-loop Nash control, discussed in Chapter 4, section 4.3.2., meaning the receding horizon Nash control exist for every initial state Z_0 if and only if matrix $H(T)$ in equation (83) is invertible.

The receding horizon Nash control needs to continuously check whether or not the closed-loop system is stable. The closed loop system with the receding horizon Nash control $u_i^*(t, z(t))$ is

$$\dot{z}(t) = \left(A - \sum_{j=1}^{m} S_i K_i(0)\right) z(t)$$

$$= A_{cl}(0)z(t), \tag{88}$$

where the closed loop system matrix is as follow

$$A_{cl}(0) = A - \sum_{j=1}^{m} S_i K_i(0), \tag{89}$$

The following results can be made about receding horizon Nash control.

i) The formation control defined as a finite horizon Nash differential game (76), (86) has a receding horizon Nash control for every initial state z_0 if and only if matrix $H(T)$ is invertible.

ii) As long as all the eigenvalues of $A_{cl}(0)$ have negative real parts, the receding horizon Nash control is asymptotically stable.

5.2.1 Distributed Control

Any proposed controller algorithm in order to be considered for multi–robot system should be distributed. In the following this feature is discussed for our Receding horizon Nash formation controller.

The receding horizon Nash control signal in equation (87) needs the state vector $z(t)$, which includes all the states from the formation team. However, the weight parameters W_{ij} and μ_{ij} in the Nash game can be selected as Zero for robot i if robot j is not its neighbour. This selection will lead to the following matrix form of Q_i and K_{if} as follows.

$$Q_i = \begin{bmatrix} q_i^{1,1} & \cdots & q_i^{1,j-1} & 0 & q_i^{1,j+1} & \cdots & q_i^{1,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_i^{j-1,1} & \cdots & q_i^{j-1,j-1} & 0 & q_i^{j-1,j+1} & \cdots & q_i^{j-1,m} \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ q_i^{j+1,1} & \cdots & q_i^{j+1,j-1} & 0 & q_i^{j+1,j+1} & \cdots & q_i^{j+1,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_i^{m,1} & \cdots & q_i^{m,j-1} & 0 & q_i^{m,j+1} & \cdots & q_i^{m,m} \end{bmatrix}, \tag{90}$$

$$
K_{if} = \begin{bmatrix}
k_{if}^{1,1} & \cdots & k_{if}^{1,j-1} & 0 & k_{if}^{1,j+1} & \cdots & k_{if}^{1,m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{if}^{j-1,1} & \cdots & k_{if}^{j-1,j-1} & 0 & k_{if}^{j-1,j+1} & \cdots & k_{if}^{j-1,m} \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
k_{if}^{j+1,1} & \cdots & k_{if}^{j+1,j-1} & 0 & k_{if}^{j+1,j+1} & \cdots & k_{if}^{j+1,m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_{if}^{m,1} & \cdots & k_{if}^{m,j-1} & 0 & k_{if}^{m,j+1} & \cdots & k_{if}^{m,m}
\end{bmatrix}, \tag{91}
$$

where $q_i^{u,v}$ or $k_{if}^{u,v}$ is a block with size $(2n) \times (2n)$. Q_i and K_{if} has $m \times m$ blocks. The j-th block row or column consist of m zero blocks. It should be noted that matrix A has a block diagonal structure. Based on this matrix structures, it can be found the j-th block row of solution K_i consist of m zero blocks from the coupled Riccati differential equation (78).

$$
K_i = \begin{bmatrix}
k_i^{1,1} & \cdots & k_i^{1,j-1} & 0 & k_i^{1,j+1} & \cdots & k_i^{1,m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_i^{j-1,1} & \cdots & k_i^{j-1,j-1} & 0 & k_i^{j-1,j+1} & \cdots & k_i^{j-1,m} \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
k_i^{j+1,1} & \cdots & k_i^{j+1,j-1} & 0 & k_i^{j+1,j+1} & \cdots & k_i^{j+1,m} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
k_i^{m,1} & \cdots & k_i^{m,j-1} & 0 & k_i^{m,j+1} & \cdots & k_i^{m,m}
\end{bmatrix}. \tag{92}
$$

Therefore the receding horizon Nash control $u_i^*(t, z(t))$ does not need the state $z_j(t)$ from the non neighbour robot j. If there is more than one robot in the team, which are not the neighbours of robot i, the same conclusion can be made. Thus $u_i^*(t, z(t))$ is a distributed control law.

5.2.2 Receding Horizon Nash Control Algorithm

Let δ denote the control time interval and $0 < \delta < T$. Based on $Z(t)$, each robot computes an open-loop Nash equilibrium solution $\bar{u}(\tau)$ for the period $t \leq \tau \leq t + T$. To indicate that this solution is an open-loop Nash equilibrium solution, and depends on the initial state $z(t)$, it is rewritten as $\bar{u}(\tau, z(\tau))$. The algorithm uses the solution provided by open-loop linear quadratic Nash equilibrium to control robots for the period $[t, t + \delta]$. At the next time instant $t + \delta$, this procedure repeats. The details of the algorithm are listed as follows:

1. Read the current state $z_i(t)$ and all neighbours' state $z_j(t)$.

2. Find the open loop Nash equilibrium solution $\bar{u}(\tau, z(\tau))$ and its state trajectory $\bar{z}(\tau, z(t))$.

3. Construct the receding horizon Nash control $u^*(\tau, z(t))$ based on the open loop Nash equilibrium $\bar{u}(\tau, z(t))$ for the period $[t, t+\delta]$.

4. Use the receding horizon Nash control $u^*(\tau, z(t))$ to control robots. The resulting state trajectory $z^*(\tau, z(t))$ should be

$$z^*(\tau, z(t)) = \bar{z}(\tau, z(t)),\ \tau \in [t, t+\delta)\,,$$

5. Update $t \leftarrow t + \delta$.

6. Loop until the control achieves a satisfying performance.

5.3 Simulation

This section we provide the simulation results for the formation control of a multi–robots system with double integrator dynamics performing a formation mission.

A triangle formation Shape had chosen to be the desired form to be shaped as depicted in the Fig. 10.

Figure 10: Triangle shape formation of a four robot formation.

In the triangle formation, the neighbours of the leader robot 1 are robot 2 and 4, the neighbour of robot 2 are robots 1 and 3, the neighbour of robot 3 is robot 2, and neighbour of robot 4 is robot 1. For $i = 4$ robots with 2-dimensional coordinates we have $q_i = [x, y]^T \in R^2$, the state and control vectors are $z_i(t) = [q_i(t)^T, \dot{q}_i(t)^T]^T \in R^4$ and $u_i(t) \in R^2 (i = 1, ..., 4)$. The robot dynamics are

$$\dot{z}_i = a z_i + b u_i, \tag{93}$$

where

$$a = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, b = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{94}$$

The state and control vector of each robot i are defined as $u_i(t) \in U$, and $z_i(t) \in Z$ respectively. by concatenating the states of all $m = 4$ robots in a team into a vector $z = \left[z_1^T, ..., z_m^T\right] \in R^{16}$, the team dynamics are

$$\dot{z}^d(t) = A z^d(t) + \sum_{i=1}^{m} B_i u_i^d(t), \quad t \geq 0 \tag{95}$$

where

$$A = I_{(m)} \otimes a = \begin{bmatrix} 0 & I_{(2)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{(2)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{(2)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{96}$$

All the entries of this matrix are 2×2 block matrices and we have

$$B1 = [1, 0, 0, 0]^T \otimes b,$$

$$B2 = [0, 1, 0, 0]^T \otimes b,$$

$$B3 = [0,0,1,0]^T \otimes b,$$

$$B4 = [0,0,0,1]^T \otimes b.$$

The operator $\otimes$ is the Kronecker product.

The incidence matrix D of a directed graph g is the $0, \pm 1$ matrix with rows and columns indexed by vertices and edges of the our formation graph. The incidence matrix associated to our formation graph based on Fig. 10 is

$$D = \begin{bmatrix} -1 & 0 & -1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{97}$$

A diagonal weight matrix represent the weighting associated to the edges of the graph with dimension equal to the number of edges in the formation graph

$$W = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \tag{98}$$

let $\hat{D} = D \otimes I_4$, and $\hat{W} = W \otimes I_4$ with $W = diag[w_{ij}]$ then the whole team formation error can be expressed in the matrix form as

$$\sum_{ij \in E} W_{ij} ||z_i - z_j - d_{ij}^d|| = (z - z^d)^T,$$

$$\hat{D}\hat{W}\hat{D}^T(z - z^d) = ||z - z^d||^2_{\hat{D}\hat{W}\hat{D}^T}.$$

Following [41], we define Laplacian of a graph g as $L = DWD^T$ for our formation graph we have

$$L = \begin{bmatrix} 10 & -5 & 0 & -5 \\ -5 & 10 & -5 & 0 \\ 0 & -5 & 5 & 0 \\ -5 & 0 & 0 & 5 \end{bmatrix}. \tag{99}$$

As it can be verified the Laplacian is symmetric and positive semi-definite. Now let define $\hat{L} = \hat{D}\hat{W}\hat{D}^T$. $\hat{L}$ is also symmetric and positive semi-definite. The team formation error therefore can be rewritten as :

$$\sum_{(ij)\in E} W_{ij}||z_i - z_j d_{ij}^d|| = ||z - z^d||_{\hat{L}}^2, \tag{100}$$

the tracking trajectory is assumed to be a circle defined by

$$q^d(t) = [cos(t), sin(t)]^T, \quad t \geq 0, \tag{101}$$

the desired input ($\ddot{q}^d$) of this trajectory will be $[-sin(t), cos(t)]^T$. In the simulation, the triangle formation is to track the circle trajectory. In this context robot 1 is assumed to be the leader robot. The proposed distributed receding horizon Nash controller is tested. the leader robot uses the cost function (73), which includes a formation cost represented by K_{1f}, Q_1 and a tracking cost represented by k_{1f}, q_1. The follower robots use the cost function (72), which only include a formation cost represented by K_{if}, Q_i, $i = 2, 3, 4$.

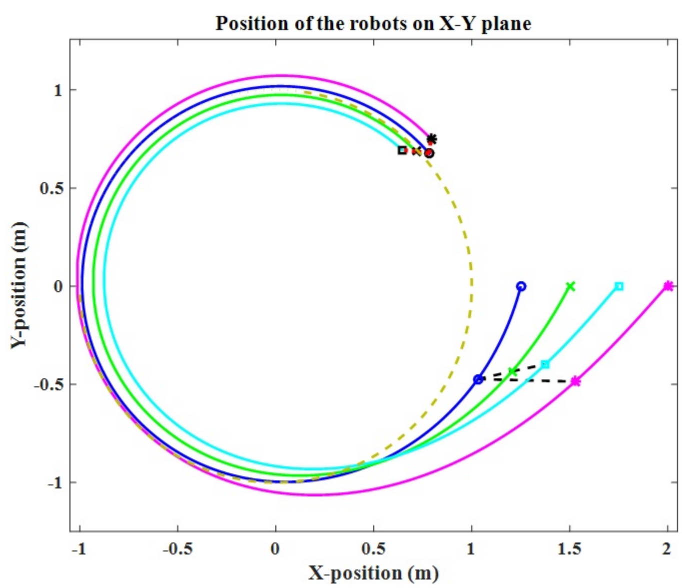

Figure 11: Formation control trajectories of four robots.

Solution of the finite horizon open-loop Nash differential game can be found by using terminal values and iterating backward. The finite horizon length here is $T = 6$ s and the sample time is $\delta = 0.01$ s. Fig. 11 depicts the trajectory of four robot following the formation algorithm. The leader robot 1 uses both tracking and formation cost function. As the results show all four trajectories converge to a triangle shape during the circle tracking.

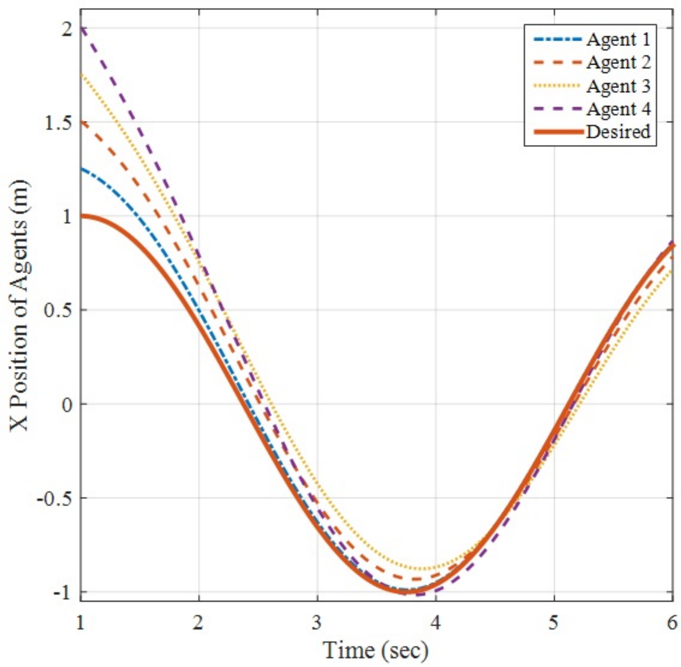

Figure 12: Agents movement in x direction.

Fig. 12 represents the error between robot control signal and tracking control error for the robots on x direction, and all the control signals converge to zero.y position error between robots and their tracking trajectories are shown in Fig. 13. It can be seen that the position error converges to zero, and all robots finally move in the circle trajectory while maintaining triangle shape formation.

5.4 Summary

Chapter 5 addresses lack of state feedback feature in finite horizon open-loop Nash differential game controller proposed in chapter 4. This is achieved through introducing a receding horizon methodology to this problem. Receding horizon Nash formation control was developed as result to provide state–feedback formation control for network of mobile robots. In addition, the proposed method is shown to be decentralized, so that it is possible to be implemented separately on each robot on board processing unit.

Finally results are provided to show the effectiveness of proposed controller. A triangle formation was defined as desired final form. It was shown that both leaders,

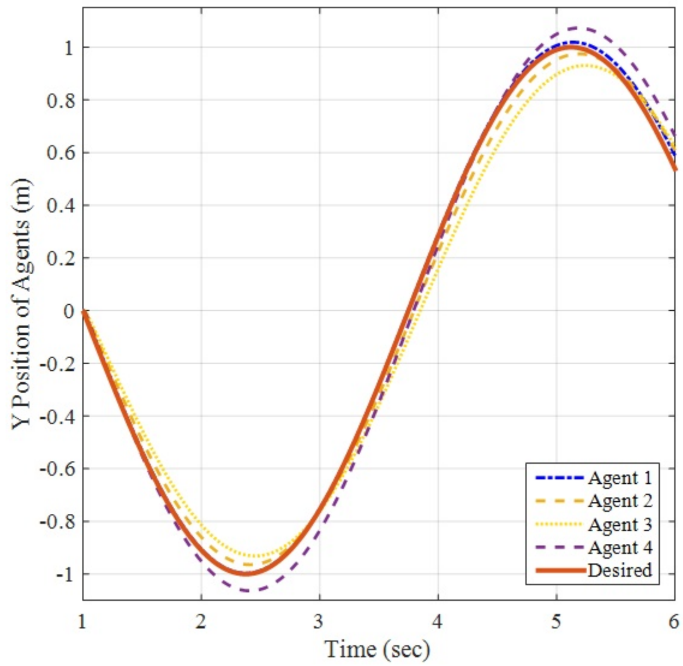

Figure 13: Agents movement in y direction.

and followers where capable of forming this shape from a basic layout, and keeping the formation while tracking the circle trajectory.

Chapter 6

6 Conclusion and Future Work

In this thesis we have presented different methods to form and maintain formation of multi–robot system by implementing effective control schemes on networks of interconnected agents.

In Chapter 3 we proposed an optimal control technique to address formation control for network of leaders with unrestricted motion, and followers acting based on a general consensus strategy. This algorithm could take predefined heterogeneous formation from an initial state to desired terminal state within an finite time. An insufficiency of this algorithm was the fact that forming the desired graph topology from a random agents layout was not considered. Moreover aside from dual classification of agents to leader and followers, individual differences or specified interests among agents were ignored in design.

Chapter 4 introduced a rather different approach to formation control based on differential games. i) flexibility to incorporate individual interests of each robot in the algorithm, ii) and ability to weight communication links in a way that accurately represent the hierarchy of inter-agent communications, were two main advantages of this technique. However this was achieved through an open-loop controller, built based on the open-loop information structure as discussed. Meanwhile, states feedback is an essential element of a practical control scenario and that renders strategies based on open-loop information structures impractical and insufficient.

In Chapter 5 we propose a method to address the lack of state feedback feature in finite horizon open-loop Nash differential game controller in chapter 4. By introducing a receding horizon methodology to this problem we can develop a receding horizon Nash formation control and provide state-feedback formation control for network of mobile robots. In addition, the proposed method is shown to be decentralized. In the simulation It was shown that both leaders, and followers where capable of forming triangle formation from a basic layout, and keeping the formation while tracking the circle trajectory.

Game theory has proven to be a powerful tool for controlling the networks with large number of agents [42]. However its advantage comes at a very high computation cost. In Chapter 4 we tried to reduce the computation load by choosing the open-loop information structure over the state–feedback information structure to solve the differential game. However by adding receding horizon method to provide state–feedback adds on to computation demand.

A natural line of thought is to implement this methods on networks with large number of participating agents. Questions about limitations, these methods will encounter can be subject to further investigation. One can argue that high computation cost poses a challenge especially on large scale formation scenarios.

In that regard, the proposed quasi-static optimal controller in Chapter 3 is relatively simple to implement and efficient. However, it is unable to form the desired network topology from indefinite agent layout. Moreover, agents' individual interest is not considered in this algorithm. These two features are provided through differential game approach proposed in Chapter 4, Finite horizon Nash formation control, and receding horizon Nash formation control in Chapter 5. The advantage of having individual agent's interest have been included, and the ability to form desired formation topology from arbitrary agent's layout, comes at a the high computation cost.

A suggestion for further research on this topic, is to implement a hybrid controller, that is, one can break a large graph into smaller partitions. If the new partitioning has at least one partition with more than one node, it is called *non-trivial equitable partition*. Abstracting the entire graph is possible through taking each partition as nodes of a new graph, and considering connections between agents of different cells as weighted edges of this new graph. Such a graph formed based on non-trivial equitable partitions is known as *Quotient* graph [24]. A hybrid controller can comprise a two level controller. Meaning, each partition is controlled separately using approach proposed in Chapter 3, through leaders and followers running a quasi-static optimal control. Then differential game based formation control is used to provide a high level controller on position of each partition in decentralized cooperative manner. Essentially receding horizon Nash differential game decides desired position for each partition with respect to other partitions to form a specific formation on a partition level. Meanwhile, quasi-static optimal control can be used to move members inside each partition to desired location and maintain agents together inside the partition.

Another reasonable thread for future work is to remove the connectivity assumption form our problem description. In both methods the graph connectivity was deliberately taken for granted. A systematic approach needs to be in placed around notion of graph connectivity throughout the formation control. This becomes even more interesting when we want topology of our large scale graph to evolve in order to better adapt with the existing constraints of our problem, while maintaining graph connectivity.

These two threads, expandability of a control scheme to larger networks and creating desirable characteristics for our network and altering them promptly to incorporate our design constraints are two proposed ideas for further research in this field.

References

[1] J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *Proceedings. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.*, vol. 2, 2001, pp. 631–636 vol.2.

[2] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, Apr. 2006.

[3] M. Mesbahi, "Dynamic Extension of the Theory of Graphs," pp. 0–5, 2002.

[4] D. Gu, "A Differential Game Approach to Formation Control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 85–93, Jan. 2008.

[5] J. Engwerda, *LQ Dynamic Optimization and Differential Games.* John Wiley & Sons, 2005.

[6] H. Weimerskirch, J. Martin, Y. Clerquin, P. Alexandre, and S. Jiraskova, "Energy saving in flight formation," *Nature*, vol. 413, no. 6857, pp. 697–698, 2001.

[7] A. Pant, P. Seiler, and K. Hedrick, "Mesh stability of look-ahead interconnected systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 403–407, 2002.

[8] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[9] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.

[10] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems*, vol. 20, no. 6, pp. 34–44, 2000.

[11] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: Application to formation flight," in *43rd IEEE Conference on Decision and Control, 2004. CDC.*, vol. 3. IEEE, 2004, pp. 2453–2459.

[12] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[13] C. R. McInnes, "Autonomous ring formation for a planar constellation of satellites," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, pp. 1215–1217, 1995.

[14] P. Ogren, M. Egerstedt, and X. Hu, "A control lyapunov function approach to multi-agent coordination," in *Proceedings of the 40th IEEE Conference on Decision and Control, 2001.*, vol. 2. IEEE, 2001, pp. 1150–1155.

[15] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *Control Systems, IEEE*, vol. 22, no. 3, pp. 52–67, 2002.

[16] a. Jadbabaie and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[17] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, Aug 1995.

[18] R. Olfati and S. Richard, "Agreement Problems," *IEEE Transactions on Automatic Control*, no. December, 2003.

[19] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," vol. 49, no. 4, pp. 622–629, 2004.

[20] J. R. Marden, G. Arslan, and J. S. Shamma, "Cooperative control and potential games." *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 39, no. 6, pp. 1393–407, Dec. 2009.

[21] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205–2213, Oct. 2009.

[22] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic, "Distributed Seeking of Nash Equilibria With Applications to Mobile Sensor Networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 904–919, Apr. 2012.

[23] B. Bollobás, *Modern Graph Theory.* Springer Science & Business Media, 1998, vol. 184.

[24] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks.* Princeton University Press, 2010.

[25] T. Basar, G. J. Olsder, G. Clsder, T. Basar, T. Baser, and G. J. Olsder, *Dynamic noncooperative game theory.* SIAM, 1995, vol. 200.

[26] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation and Control.* CRC Press, 1975.

[27] J. Engwerda, *LQ dynamic optimization and differential games.* John Wiley & Sons, 2005.

[28] D. S. Naidu, *Optimal Control Systems.* CRC press, 2002, vol. 2.

[29] R. Diestel, *Graph theory.* Springer Heidelberg, 2005.

[30] J. A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," vol. 49, no. 9, pp. 1465–1476, 2004.

[31] M. Ji and M. Egerstedt, "Connectedness Preserving Distributed Coordination," pp. 93–98, 2005.

[32] S. Martínez, F. Bullo, S. Member, J. Cortés, S. Member, and E. Frazzoli, "On Synchronous Robotic Networks Part II : Time Complexity of Rendezvous and Deployment Algorithms," vol. 52, no. 12, pp. 2214–2226, 2007.

[33] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networkspart i: Models, tasks, and complexity," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2199–2213, 2007.

[34] R. Olfati-Saber and R. Murray, "Consensus Problems in Networks of Agents With Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[35] R. Olfati-Saber, “Flocking with obstacle avoidance,” Tech. Rep., 2003.

[36] M. Ji, “Graph-based control of networked systems,” Ph.D. dissertation, Georgia Institute of Technology, 2007.

[37] a. Muhammad and M. Egerstedt, “Leader-based multi-agent coordination: controllability and optimal control,” *Proc. of 2006 American Control Conference*, p. 6 pp., 2006.

[38] C. D. Godsil, G. Royle, and C. Godsil, *Algebraic graph theory.* Springer New York, 2001, vol. 207.

[39] J. Shinar and V. J. Glizer, “Application of receding horizon control strategy to pursuit-evasion problems,” *Optimal control applications & methods*, vol. 16, no. 2, pp. 127–141.

[40] J. Cruz, M. Simaan, A. Gacic, and B. Letelliier, “Game-theoretic modeling and control of a military air operation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 4, pp. 1393–1405, 2001.

[41] N. Biggs, *Algebraic Graph Theory.* Cambridge University Press, 1993.

[42] M. Huang, R. P. Malhamé, P. E. Caines *et al.*, “Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle,” *Communications in Information & Systems*, vol. 6, no. 3, pp. 221–252, 2006.