Decentralized Receding Horizon Control of Cooperative Vehicles with Communication Delays

Hojjat A. Izadi

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

November 2009

Your file Votre référence
ISBN: 978-0-494-67324-9
Our file Notre référence
ISBN: 978-0-494-67324-9

ABSTRACT

Decentralized Receding Horizon Control of Cooperative

Vehicles with Communication Delays

Hojjat A. Izadi, Ph.D.

Concordia University, 2009

This thesis investigates the decentralized receding horizon control (DRHC) for a network of cooperative vehicles where each vehicle in the group plans its future trajectory over a finite prediction horizon time. The vehicles exchange their predicted paths with the neighbouring vehicles through a communication channel in order to maintain the cooperation objectives. In this framework, more frequent communication provides improved performance and stability properties. The main focus of this thesis is on situations where large inter-vehicle communication delays are present. Such large delays may occur due to fault conditions with the communication devices or limited communication bandwidth. Large communication delays can potentially lead to poor performance, unsafe behaviour and even instability for the existing DRHC methods.

The main objective of this thesis is to develop new DRHC methods that provide improved performance and stability properties in the presence of large communication delays. Fault conditions are defined and diagnosis algorithms are developed for situations with large communication delays. A fault tolerant DRHC architecture is then proposed which is capable of effectively using the delayed information. The main idea with the proposed approach is to estimate the path of the neighbouring faulty vehicles, when they

are unavailable due to large delays, by adding extra decision variables to the cost function. It is demonstrated that this approach can result in significant improvements in performance and stability. Furthermore, the concept of the tube DRHC is proposed to provide the safety of the fleet against collisions during faulty conditions. In this approach, a tube shaped trajectory is assumed in the region around the delayed trajectory of the faulty vehicle instead of a line shaped trajectory. The neighbouring vehicles calculate the tube and are not allowed to enter that region. Feasibility, stability, and performance of the proposed fault tolerant DRHC are also investigated. Finally, a bandwidth allocation algorithm is proposed in order to optimize the communication periods so that the overall teaming performance is optimized. Together, these results form a new and effective framework for decentralized receding horizon control with communication faults and large communication delays.

This thesis is dedicated to:

My parents,

for their patience and absolutely unconditional love.

Acknowledgements

It has been always pleasant to thank those who have contributed to my work. First I would like to thank my open-minded supervisor Dr Brandon W. Gordon, who has been always open to my new ideas leading to this thesis; without his useful comments, hints and support this thesis was impossible. Also, I would like to thank the committee members of my thesis, especially Dr Shahin Hashtrudi Zad and Dr Youmin Zhang whose useful comments helped me to improve the thesis quality. I should also thank Dr Camille A. Rabbath for helping me to promote my thesis and technical papers quality in the early stage of my PhD.

A special mention, I would like to acknowledge the Concordia University for giving me the opportunity to pursue my graduate studies in such a dynamic environment it possesses, and for granting me many valuable awards including Concordia Graduate Fellowship, International Students Tuition Fee Remission, Teaching Fellowship, Thesis Completion Award and many other awards and bursaries.

I would like also to thank the graduate advisor of Mechanical engineering department of Concordia University, Mrs Leslie Hossein and the international student office advisor, Mrs Jenny Calder, for their helpful advices.

I would like to express my grateful thanks to my wife, Mrs Houshin Nejati (Sara), for her absolute understanding of my situation during the final steps of my PhD.

I am thankful to my colleagues Ali Azimi, Farid Sharifi, Nandu Kishor Dwale, Reza Pedrami, Mostafa Mirzaie, Abbas Chamseddine and Sivaram Wijenddra, the people who

were the closest to my work to share all kinds of difficulties with a special warm friendship.

There are many friends who have indirectly helped me during my PhD; I would like to thank Arash Ashkan, Ehsan Kianmanesh Rad, Behbood Dehestani, Moein Mehrtash, Mehrdad Pakmehr, Ahmad Reza Momeni, Ehsan Barjasteh, Esmaeil Naderi, Masoud Pakdel, Pouria Rashidi, Saber Fallah, Eric Gogniat, Kamyar Gordnian, Kazem Khademi, Mohsen Kalantari and Noradin Fazli, who have helped me a lot in different ways during my PhD. I am so grateful to my teacher of guidance school, Mr Habib Javidi, who had a great impact on my life; I will never forget his favours.

The last, but off course not the least, my warmest appreciation goes to my family members including my parents, my sister, Elaheh, my brothers, Heshmat, Mohamad Hossein, Omid and their wife, my parents-in-law, my brothers-in-law, Ali and Amir Hossein, my sister-in-law, Ila; without their support and understanding I have not been able to complete such long journey.

I am sure many other people helped me during my PhD; I apologize for the missing names.

Montreal, Quebec, Canada,

November 21, 2009

Hojjat Izadi

Table of Contents

List of Figures

List of Tables

List of Symbols

A, B: Linear time-invariant matrices of dynamical system.

T: Prediction horizon.

δ : Execution horizon/Sampling time.

t_c: Computation time.

$\boldsymbol{x}$, $\boldsymbol{u}$: State and input vectors.

$\boldsymbol{z}$: Flat output vectors.

$\mathbb{X}^i$, $\mathbb{U}^i$: Set of admissible states and inputs of vehicle i.

$\mathbb{X}_f^i$: Terminal set of states of vehicle i.

$\mathbb{G}$: Interaction graph topology.

$\mathbb{V}$: Set of cooperative vehicles.

$\mathbb{E}$: Set of interconnected vehicles in the interaction graph.

N_v: Number of cooperative vehicles.

N_n^i: Number of neighbours of i^{th} vehicle.

N_l^i: Number of leaders of i^{th} vehicle.

N_f^i: Number of followers of i^{th} vehicle.

J_∞: Individual cost of each vehicle over the infinite horizon.

J_T: Individual cost of each vehicle over the finite prediction horizon T.

J^i: Decentralized cost of i^{th} vehicle.

$\mathcal{P}(t_k)$: RHC problem at time t_k.

$\mathcal{P}^i(t_k)$: Decentralized RHC of vehicle i at time t_k.

$\mathcal{P}_D^i(t_k)$: Delayed Decentralized RHC of vehicle i at time t_k.

d: Discrete Communication delay.

x^{-i}: Concatenated states of neighbours of i^{th} vehicle.

I^i: Decentralized performance index of i^{th} vehicle.

$x_{t_k}^{j,i}(t)$: State of the j^{th} vehicle at time t, predicted by the i^{th} vehicle at time t_k.

$x^{c,i}$: State vector of target of i^{th} vehicle.

P, Q, R, S: Matrix penalties.

List of Acronyms

RHC: Receding Horizon Control

DRHC: Decentralized Receding Horizon Control

MPC: Model Predictive control

LTI: Linear Time Invariant

RX: Receiver of Communication Device

TX: Transmitter of Communication Device

UAV: Unmanned Aerial Vehicle

CLF: Control Lyapunov Function

P, Q, R, S: Matrix penalties.

List of Acronyms

RHC: Receding Horizon Control

DRHC: Decentralized Receding Horizon Control

MPC: Model Predictive control

LTI: Linear Time Invariant

RX: Receiver of Communication Device

TX: Transmitter of Communication Device

UAV: Unmanned Aerial Vehicle

CLF: Control Lyapunov Function

Chapter 1. Introduction

The successful cooperative behaviour of biological systems, such as flocks of birds, has motivated engineers to develop and use cooperative vehicle systems instead of a single vehicle for complex missions. Cooperative control of multiple vehicle systems covers a wide range of applications including air traffic control, automated highway systems, search and rescue missions, mining robots, space exploration, satellite networks, security systems, and many others. In each case, using a team of cooperative vehicles is more efficient and reliable than using a single vehicle.

The control of cooperative vehicles is traditionally performed in a centralized manner, where a central controller communicates with each vehicle and coordinates their actions. However, using a central controller is difficult especially when the number of vehicles becomes large. The computation resource required rapidly grows with the number of group members. Furthermore, failure of the central decision maker leads to failure of the entire group. Due to these aforementioned problems, over the past several years there has been significant interest in developing the decentralized cooperative controllers [1-3]. In such systems, control decisions are made by individual vehicles or subgroups that require less computation than centralized implementations. The decentralized structure potentially results in increased autonomy of agents and affordable communication requirement. Decentralized control architectures are usually developed

by breaking the centralized control problem into local control problems of smaller size in order to reduce the computation requirements. Hence, the control action of every agent depends only on the local information, such as states and control inputs of neighbouring vehicles, to achieve the global objectives of the vehicle group. The local information is usually provided through inter-vehicle communication or onboard sensors. The main distinction between the decentralized approaches usually arises from the type of the information they need to communicate, the form of the communication topology, and the type of control strategy.

The decentralized implementation must provide and respect the primary objectives of cooperative control problem, these primary objectives are: 1) the total cost added due to cooperation must provide a greater increase in system effectiveness than the case of non-cooperating vehicles, 2) the performance lower bound of a cooperative system as communication degrades should never be worse than the performance of the same vehicles without cooperation. To achieve these primary goals of the cooperative control, the new potential approaches are under investigation. For example, with recent advances in distributed computation, there have been numerous attempts to use optimization-based control methods, such as receding horizon control (RHC) [4, 5] to decentralize the cooperative control problems. Although RHC is computationally expensive, it has some prominent capabilities which motivate researchers to develop the RHC based decentralized control architectures [1, 2, 3, and 4]. For instance, since the RHC generates the control action through minimizing a cost function, it is easy to provide cooperation among neighbouring subsystems in the cost function and/or in the constraints. Another capability of RHC that makes it an attractive control method from the industry

perspective is its prominent ability to handle the system's constraints and saturations while providing optimal or suboptimal control actions [3]. Further, the predictive nature of RHC makes it suitable for applications such as path planning and conflict resolution.

In this thesis, the hierarchical design for cooperative control problems has three main levels: *high-level*, *mid-level* and *low-level*. The *high-level* includes the tasks such as mission management, task assignment, timing/scheduling and has the most interaction with human supervisor. Some issues such as path planning, collision avoidance, obstacle avoidance, formation keeping, reconnaissance, and search algorithms are designed in the *mid-level*. The *low-level* (or vehicle-level) design discusses issues such as inner loop control, trajectory following, disturbance rejection and robust control. Sometimes the *mid-level* design can be combined partly with *low-level* design which provides consistency between these two levels. For instance, since the RHC has the capability for both path planning and inner loop control, these two tasks may be combined [6-8] which results in feasible paths. In this thesis, this type of combined design is utilized and the *high-level* design such as task assignment is not addressed, i.e., it is assumed that an efficient task assignment and mission management algorithm is available. Figure 1.1 shows the hierarchy of the cooperative team design where the *mid-level* and *low-level* designs are combined.

As seen in Figure 1.1 the information exchange are performed among the cooperative vehicles and also among *high-level*, *mid-level* and *low-level*. The *information structure* represents the access of each component in the team to the global or local information. Different scenarios can happen with different possibilities for *information structure* including unidirectional or bidirectional information flow, connectivity of

communication topology and fixed/varying communication topology. In this research it is assumed that the communication graph is not full, i.e., only some specific subgroups of vehicles can talk to each other and have access to each other's information.

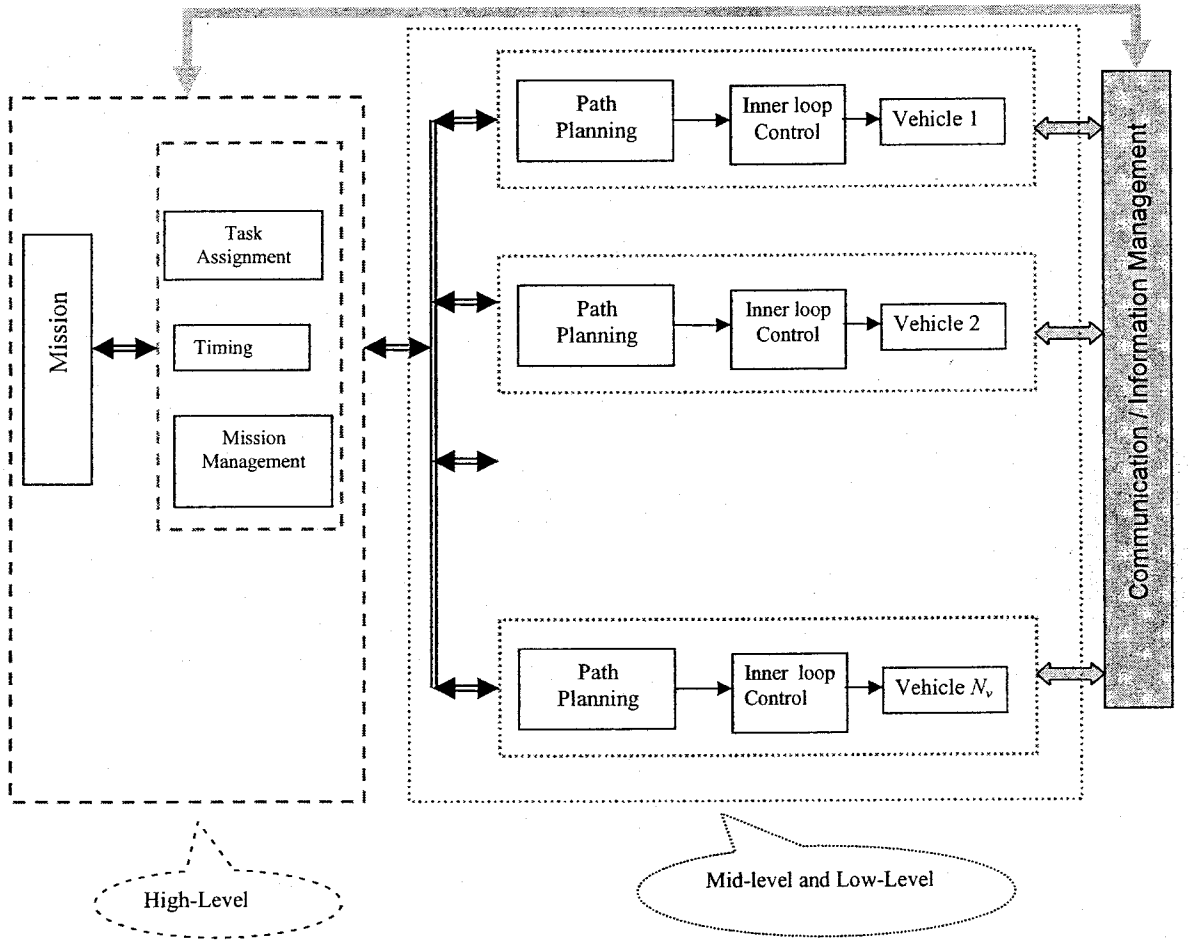

Figure 1.1: Hierarchy of a typical cooperative team design where the mid-level and low-level designs are combined.

1.1. Receding Horizon Control

With Receding Horizon Control (RHC), also known as Model Predictive Control (MPC), the control action is generated through the minimization of a cost function

subject to dynamical model and dynamical constraints of the process. The name "Receding Horizon Control" emerges from the fact that solving an infinite horizon optimal control problem is demanding and thus the horizon is receded to decrease the computation effort; whereas, the name "Model Predictive Control" comes from the fact that the RHC/MPC uses a mathematical model of the system to predict its future evolution by designing the control law.

Roughly, with RHC a cost function is minimized over a period of time called the *prediction horizon* denoted by T, to obtain the corresponding control trajectory over the *prediction horizon*. Meanwhile, only the first portion of the control trajectory over a smaller time called *execution horizon (time)* denoted by δ, is applied to the plant till the next sampling data is available. Repeating this procedure yields a close loop solution. Figure 1.2 shows the relation between *prediction horizon* T, and *execution horizon* δ (*sampling time*). Also, t_c denotes the computation time which is assumed to be negligible in most of the current research [9, 10] including this thesis. In situation where the computation time is not negligible and a zero computation time assumption is invalid two methods may be used: 1) *Retarded Actuation* [11] and 2) *On-the-Fly Computation* [12] where the input planned at previous time step (t_{k-1}) is used during $[t_k, t_k + t_c]$; the discrete sampling is denoted by t_k and the next sample time is given by $t_{k+1} = t_k + \delta$.

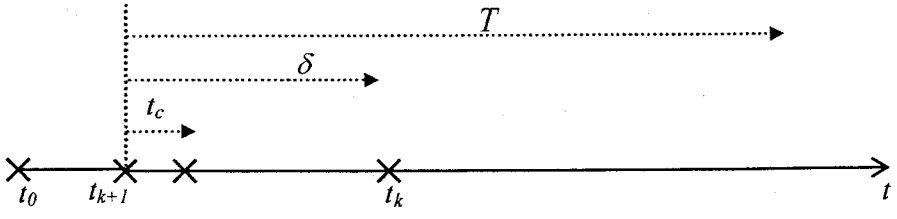

Figure 1.2: Timing in RHC.

Some of the main advantages of RHC are listed below:

1- *Constraint Handling*: one of the unique properties of the optimization based control methods such as RHC is the capability of handling the dynamical constraint, saturation and constraints resulting from the nature of the mission. The constrained nature of the most of the physical processes makes RHC an attractive control approach.

2- *Generality*: RHC is applicable to generic set of dynamics including linear and nonlinear. Also, RHC is one of the few methods which provide the feedback control design for nonlinear systems.

3- *Reconfigurability*: the outstanding flexibility and adaptive nature of RHC comes from the fact that it is capable to redefine the cost functions and modify the constraints as needed in an online fashion to follow the changes in the system, mission requirements and the environment.

4- *Optimality*: RHC provides the optimal or suboptimal solutions and allows applying control objectives through the cost function.

However, RHC has some shortcomings:

1- *Computational Demand*: the huge online computation effort of the classical RHC and other optimization based methods are a barrier for employing them for fast dynamics such as aerospace vehicles. Nevertheless, there have been several attempts to tackle this problem by modifying the cost function and terminal constraint that lead to reduced online computation burden. The advent of fast computers also facilitates the use of RHC for fast dynamics.

2- *Theoretical Demand*: in general, it is difficult to address the stability, feasibility and performance of RHC and often the stability conditions are too conservative and provide poor performance.

These drawbacks motivate new research to develop more efficient RHC architectures for benefiting the unique advantages of RHC.

1.2. Literature Review

The existing literatures on the decentralized receding horizon control (DRHC) cover a wide range of issues including stability, feasibility, implementation, performance analysis, robustness, applications, etc. Since most of the works on DRHC are extensions of classical RHC for single process, a short literature review on RHC is inevitable. In this section first some literatures on cooperative control are reviewed, and then the important contributions on the RHC are reviewed. Finally, the significance of RHC for control of cooperative vehicles is highlighted through reviewing the literatures on DRHC.

1.2.1. Cooperative Control

Research on cooperative control is carried out on three different levels of hierarchy (see Figure 1.1). For instance, at the *high-level*, one can point to works performed on task assignment [77, 96-98], search and classification [99, 100] and reconnaissance [101]. In the *mid-level*, very interesting works have been carried out in [6, 7, 78, 79, and 81] for path planning, obstacle avoidance and collision avoidance. Formation keeping, trajectory tracking, target tracking and safety [102] have also received good attention. For the purpose of this research, the literatures addressing path planning, formation control, swarming, flocking problems and consensus/agreement algorithms are reviewed.

Reviewing the papers based on the mentioned categories can reflect the *information* and *control structure* specifications as well as practical issues and different applications of the cooperative control.

1.2.1.1. Path planning

The path planning for cooperative vehicle systems has been investigated in several research works. Some of the main challenging issues in this field are collision avoidance, obstacle avoidance, highly obstructed complex environments, feasibility of the generated paths and trajectory planning in environments with non-convex obstacles. Different methods have been developed for the path planning of multiple vehicles.

In [78] a graph based method is developed where a sequence of vertices is assigned to discrete points in space, then edges are used to connect these vertices. A cost is assigned to each of the edges, and the graph is searched for the minimum cost. This method is well suited to complex environment with general form of obstacles.

Probabilistic roadmap planning (PRM) is another efficient method to compute collision-free paths for UAVs [81]. This method consists of two phases: a *building* and a *query* phase. The *building* phase is the construction of a graph called 'roadmap'. The nodes in the roadmap are collision-free configurations and the edges linking the nodes are collision-free paths. The *query* phase is finding a path between an initial and goal configurations by connecting these nodes to the road map and searching them for a sequence of edges linking the two nodes. This method was originally developed for holonomic robots in a static environment; however, it has been recently applied to non-holonomic robots with constrained kinematics and high degrees of freedom [82]. In a new extension of this method [83], a so-called coarse-roadmap is built during the

building phase and further refined in the *query* phase with focus on the area of interest, and it is customized to specific preferences such as maximum number of sharp turns.

The virtual potential fields and forces are another technique proposed by Bortoff [79] (see also [80]). In this method, a chain of masses connected to each other by springs and dampers represents a vehicle path. Obstacles to be avoided have repulsive force fields and the cohesion of the group produces an attracting force that shapes the path until equilibrium is reached. Bortoff concludes that the method is quite promising for uniform radar field [84]. Most recently, this method has found some attentions in collision avoidance of flocks and swarms [85, 86].

Another promising methodology for path planning is the optimization based methods [87, 55, 58]. In this approach a cost function consisting of a path length or time or fuel is minimized to generate an optimal path. The optimization problem is subject to initial and target state constraints and a model of the vehicle to ensure the feasibility of the paths. The main drawback of this method is that it needs a high computational power, and that the computation load increases dramatically with the scale of the problem. Hence, to reduce the computational time a finite horizon time is used instead of an infinite horizon time. This method is used in this research to generate the paths.

1.2.1.2. Formation control, Flocking and Swarming

Formation, *flocking* and *swarming* are three main scenarios studied in the cooperative control problems. In the formation control problem, the cooperation is characterized by forcing the entire group to move in a certain pre-specified shape such as triangle where the shape (formation) follows a given path (probably generated by any of methods of previous section). The formation shape must be preserved during the mission

while the entire group behaves like a rigid body. The formation control can be performed in three different methods: *leader-follower* [88], *virtual-leader* [89] and *leaderless* [90] formations. In the *leader-follower* the given trajectory is followed (or generated) by one (or more) of the vehicles (*leaders*) and the rest of the group (*followers*) align their velocity, position and heading angle with respect to their assigned leader. In the *virtual-leader* approach a virtual vehicle is considered which plays the role of the actual *leader*. In the *leaderless* approach each vehicle in the group considers itself as the *follower* of its neighbouring vehicle. A comprehensive review of the formation control problems can be found in [88]. In this thesis both *leaderless* and *leader-follower* formations are considered in the simulation examples.

According to the definition in [91] *flocking* is a form of collective behaviour of large number of interacting agents with a common group objective. In a more precise and mathematical definition [92] a group of mobile agents are called flock if all the agents attain the same velocity vector and the relative distances between agents are stabilized to some certain constant value. *Aggregation* and *cohesion* are two basic behaviours in flocking. The flocking is inspired by the collective behaviour of species such as simple bacteria colonies, flocks of birds, schools of fish, herds of mammals and swarm of insects. Such collective behaviours of biological species are observed to be helpful in achieving common teaming objectives such as avoiding predators, increasing the chance of finding food, etc. Huge numbers of agents is the main distinction between swarm and other cooperative scenarios, different applications of swarms are discussed in [93].

In general, the overall shape and size of group in formation control, flocking and swarming are the main interest; however, the key issue in formation control is the exact

inter-vehicle distances while in flocking and swarming the *aggregation* and *cohesion* of the group are the challenging issues.

1.2.1.3. Consensus/agreement algorithms

In the *consensus* algorithms the main problem is about having agreement on some quantities of interest. Such quantities might or might not be related to the motion of the individual agents [119]. For instance, in a formation control problem the agreement on the center of formation is required when the information flow is not bidirectional or communication structure is varying [94]. The *consensus* algorithms are very suitable tools to analyze the effect of information structure on the properties of mission. For instance, in [95], *graph Laplacians* are used to investigate the effect of communication topology on the formation stability for groups of agents with linear dynamics. It is shown that the algebraic graph theory is useful in modeling the communication network and providing a connection between the communication topology and the formation stability. The main issues in *consensus* research includes variety of assumptions on the network topology (being fixed or switching), presence or lack of communication, time-delays, connectivity of network and *direct* (unidirectional) or *indirect* (bidirectional) information structure. In general, the convergence analysis of consensus protocols on *direct* graphs is more challenging than the case of *indirect* graphs. This is partly due to the fact that the properties of graph topologies are mostly known for *direct* graphs.

1.2.2. Receding Horizon Control (RHC)

RHC was first introduced in the process control community. It has attracted the attention of many researchers due to its ability to handle the constraints on the states and

inputs in control problems [16-28]. But its huge online computation load did not first allow using RHC for fast dynamics such as aerospace application.

The first RHC concepts were derived from the optimal control theory where a cost function is optimized over an infinite horizon to generate the control action. With the optimal control problem using the infinite prediction horizon the stability is achieved easily but there is no analytical solution for most of cases except linear quadratic regulator (LQR); also, it often leads to solving two point boundary value problems (TPBVP). On the other hand, solving an optimization problem over an infinite time is computationally prohibitive that drives engineers to employ a reduced horizon scheme called receding horizon control (RHC).

In 1990, Bitmead et al. [62] showed that the reduced form of optimal control problem (RHC) does not guarantee closed-loop stability readily but the closed-loop stability can be achieved by a careful tuning of RHC parameters such as prediction horizon, and matrix penalties in the cost function along with including final equality ($\boldsymbol{x}(T)=0$) or inequality constraints ($\boldsymbol{x}(T)\in\mathbb{X}_f$) in the optimization problem. Bitmead et al. [62] proposed a stabilizing scheme for the case of linear systems without input constraint. In 1990, the similar results are published by Mayne and Michalska [20] where they showed that under some strong conditions such as imposing the final state equality constraint ($\boldsymbol{x}(T)=0$), the stability can be achieved for a class of nonlinear systems subject to input constraints (see also [22]). Although this work was a quick jump towards the nonlinear systems, this strong assumption ($\boldsymbol{x}(T)=0$) was restrictive computationally and theoretically since it needs infinite iterations to be satisfied. Also, it was required that the cost function be continuously differentiable which is a very strong assumption. Then,

in 1993 these assumptions were relaxed by Michalska et al. [31] where they proposed a dual mode RHC in which they employed a linear state feedback control in a neighbourhood of the origin by considering some terminal region (set) containing the origin; in this scheme the RHC is applied outside the terminal region.

In 1998, Chen and Allgower [10] modified the dual RHC approach by proposing a quasi-infinite RHC that relaxed the utilizing of linear state feedback inside the terminal region around the origin. In that, in this approach the RHC controller is used all the time, no matter whether the states are inside or outside of terminal region; also, the linear state feedback controller is only used for offline computation of terminal region and terminal matrix penalty. The terminal matrix penalty in this approach is obtained by solving the Lyapunov equation with feedback terminal controller under which the terminal region is an invariant set for the system. This approach has been the most useful approach among RHC architectures. In general, although all the approaches presented after then have tried to relax the final inequality constraint, they left the problem with another constraint and in some cases they need some restrictive assumptions. In this thesis, the quasi-infinite RHC is used.

Using the final inequality constraint instead of final equality constraint considerably reduced the computation load; but it was not efficient yet for very fast processes. Then, in 1998 a new approach was developed by Primbs et al. [65, 66] and De Nicolao et al. [64], and later in 1999 was extended by Jadbabaie et al. [67] in order to relax the final inequality constraint. This approach utilizes an appropriate Control Lyapunov Function (CLF) as the final cost to guarantee the stability. More precisely, it is proved that the CLF is an upper bound on the infinite cost function and can be found in some neighbourhood

of the origin. The method is successfully applied to the Caltech ducted fan which is known as one of the fast aerospace dynamics. To introduce a suitable choice of CLF in [68] Jadbabaie et al. employed the Linear Parameter Varying (LPV) method as a choice for CLF. However, it is not always possible to find a global CLF (the system may not be stabilizable). Then in 2001 by Jadbabaie et al. [69], the region of attraction of CLF based RHC and RHC is compared and it is shown that under some circumstances this region of attraction can be expanded to the region for infinite horizon controller by increasing the horizon length which is not attractive computationally.

Although the CLF based RHC methods are computationally attractive and the aforementioned efforts proposed significant theoretical background for RHC, they lack a sufficient stability guarantee for systems with input and state constraints; therefore, they were not interesting for industry applications because as mentioned previously, one of the main attractive features of RHC is its capability for handling the constraints. Hence, some new research has been started to include the state and input constraints into the optimization problem. For example, in [70] Jadbabaie et al. show that using Dini's theorem on the convergence of functions, the exponential stability is obtained for input constrained RHC with a general nonnegative terminal cost with sufficiently large horizons. It is shown that there is always a finite horizon for which the corresponding RHC is stabilizing without terminal cost. Also, in [71-73] the authors attempt to remove the final constraint while the state and input constraints are present.

Another promising approach is proposed by Scokaert et al. [74] for reducing the computation complexity of RHC by relaxing the optimality condition; the authors proposed a suboptimal RHC architecture where the feasibility of the solution implies the

stability. It was a great progress for reducing the online computation since reaching the global optimum is computationally expensive and sometimes it does not exist for non-convex optimization problems.

The attempts for reducing the online computation load of RHC have not been restricted to theoretical approaches. In 2000 by Milam et al. [11], a direct method for solving optimal control problems has been proposed based on the properties of flat outputs which reduce the online computation time by mapping the optimization problem to a smaller dimension. Taking the advantages of the new advances in the computation power, most recently it is used widely for fast dynamics. The method proposed in [11] was applied successfully to a vector thrust flight experiment in 2003 [29, 30], which is example of an aerospace system with fast dynamics. Further in [75], it is applied to the fast and nonlinear dynamics of vortex-coupled delta wing aircraft, simulations illustrates that the reasonable online computation expense allows employing the RHC for the fast aerospace dynamics subject to actuator saturation and state constraints.

Although it is tried to draw a general sketch for the history of RHC in this subsection, the readers are encouraged to see Mayne et al. [35] for a comprehensive review of RHC history, research, application and architectures.

1.2.3. Decentralized Receding Horizon Control (DRHC)

Interest in decentralized control dates back to 1970's when Wang and Davison [36] used the decentralized approach for the large scale systems and since then there has been a significant attempt to use this approach in a wide range of engineering applications including large scale systems [36-40], and cooperative control of dynamically coupled [41] and dynamically decoupled subsystems [42, 43]. In this thesis, the main focus is on

the decentralized control of networks of dynamically decoupled subsystems. The motivation for using decentralized approach for such networks of dynamically decoupled subsystems arises from the abundance of networks of independently actuated subsystems and also the necessity of avoiding centralized design for such large scale systems, due to their computational complexity [43]. Networks of vehicles in formation, production lines, units in a power plant, a network of security cameras at an airport [44], distributed paper machine control [45, 46] and mechanical actuators for deforming surface are just a few examples. More examples and applications to the decentralized control design are given in [47-55].

In the previous section some of the quite unique benefits of RHC are discussed. The motivation for using RHC in cooperative control problems arises from the mentioned benefits of RHC and the fact that it is easy to provide cooperation by RHC using the cost function. The optimality property of receding horizon control makes it a suitable tool to control the formation flight of a group of spacecraft where the main concern is to minimize the fuel consumption of the group [56, 57]. Also, the optimality property along with the constraint handling makes it a potential approach for feasible path planning; further, it is possible to combine the path planning and inner loop control design. As an example, a nonlinear decentralized model predictive control is proposed in [58] (see also [55]) for flying multiple autonomous helicopters in a complex three-dimensional dynamic environment. The proposed decentralized RHC in [58] provides a framework to solve optimal discrete control problem for the nonlinear systems under state constraint and input saturation. Also, the trajectory generation with operational constraints and stabilization of vehicle dynamics are combined by including a potential function

reflecting the state information of a possibly moving obstacle or other agents to the cost function. It is also shown that the computation load of this approach is small enough to be used for the real-time applications.

In [14, 15, 53, 59, 60, 164] a combination of different techniques is used to develop robust decentralized model predictive control architecture for path planning. For example, the constraint tightening technique is used to achieve robustness, it is proposed to modify the speed limit, turn rate and obstacle relative distance in order to guarantee the robust constraint satisfaction. Also, an invariant set is used around each vehicle to ensure safety. Further to account for non-convex coupling constraints arising from collision avoidance and obstacle avoidance constraints the Mixed Integer Linear Programming (MILP) technique is used. MILP is an implementation technique which breaks the non-convex constraint to some convex constraints. In this framework, a suitable cost function is selected to generate an intelligent trajectory around obstacles in the environment; the developed decentralized model predictive control architecture includes both *lower-level* issues such as inner loop control and trajectory planning in the *mid-level*. Each vehicle plans only for its own action, but feasibility of the sub-problems and collision avoidance between multiple aircrafts are guaranteed in a sequential, decentralized fashion, in which each aircraft takes into account the latest trajectory and loiter pattern of the other aircrafts. Also, UAVs communicate relevant plan data to ensure that decisions are consistent across the team. The efficiency of this approach is verified experimentally in [61] where two different test-beds with 4 and 8 UAVs are developed and tested.

As in classical RHC, most researchers have addressed stability and feasibility of DRHC by modifying the cost function and constraints. For example, a sophisticated work

is conducted by Keviczky et al. [1], where a DRHC is proposed to control a team of vehicles with decoupled discrete-time dynamics through breaking a centralized RHC architecture into distinct RHC controllers of smaller sizes. Each RHC controller is associated with a different vehicle and computes the local control inputs based only on the states of itself and of its neighbours. With such approach, the vehicles are coupled through a cost function and the required information from the neighbouring vehicles is provided through non-delayed communication or on-board sensor measurements. Each vehicle predicts its neighbours' behaviour from the dynamical model available and, based on this prediction, plans the trajectory of itself and its neighbours, but executes only its own trajectory. It is proved that if the mismatch between the predicted and actual trajectories of all neighbours is smaller than some value of the cost function related to the initial conditions then the global group stability is achieved [1]. In another work [2], Dunbar et al. proposed a distributed RHC for multiple vehicles with continuous-time dynamically decoupled subsystems whose state vectors are coupled through the cost function of an RHC control problem. Each vehicle solves an optimization problem and generates its own control action using an assumed control action of neighbouring vehicles. The key requirement for stability is that each control input does not significantly deviate from the previous one, which is used as the assumed control action by neighbouring vehicles. Throughout the thesis at each situation the related DRHC papers will be reviewed.

1.3. Thesis Objectives

The main objective of the thesis is to develop new decentralized RHC (DRHC) architectures for formation control and path planning of multiple vehicles in presence of large communication delays. Such large communication delays can result from communication failure and limited communication bandwidth. Communication failures leading to large communication delays are defined and some fault diagnosis algorithms are developed. A fault tolerant reconfigurable DRHC method is then developed which account explicitly for large communication delays. An approach for safety against collisions is proposed based on the tube DRHC concepts. Analysis of feasibility, stability, and performance of DRHC is also performed. Finally, an approach for optimal allocation of communication bandwidth for DRHC is proposed which improves the teaming performance in presence of limited communication bandwidth. The results together form a new framework for DRHC of cooperative vehicles with large communication delays.

1.4. Structure of the Thesis

Figure 1.3 shows the overall structure of the thesis. Chapter 2 presents the formulation of RHC and delay-free DRHC. The corresponding algorithms are also presented and simulations are used to illustrate the implementation issues. Then the overall problem statement is presented which includes a defined faulty condition involving the failure of a high performance communication device leading to large communication delays. Chapter 3 develops fault diagnosis algorithms for detecting communication failure in different cases including direct communication topology

(unidirectional communication), indirect communication topology (bidirectional communication), and cases where the transmitter and receiver communication devices are separate. In Chapter 4, a fault tolerant delayed DRHC architecture is proposed to explicitly account for large communication delays. A tube DRHC approach is proposed in Chapter 5 in order to provide fleet safety against possible collisions in faulty situations. Chapter 6 investigates the feasibility, stability, and performance of the proposed delayed DRHC approach. In Chapter 7, the case of limited communication bandwidth is considered and a bandwidth allocation algorithm is proposed in order to optimize the cooperative control performance. Chapter 8 presents conclusions and future work.

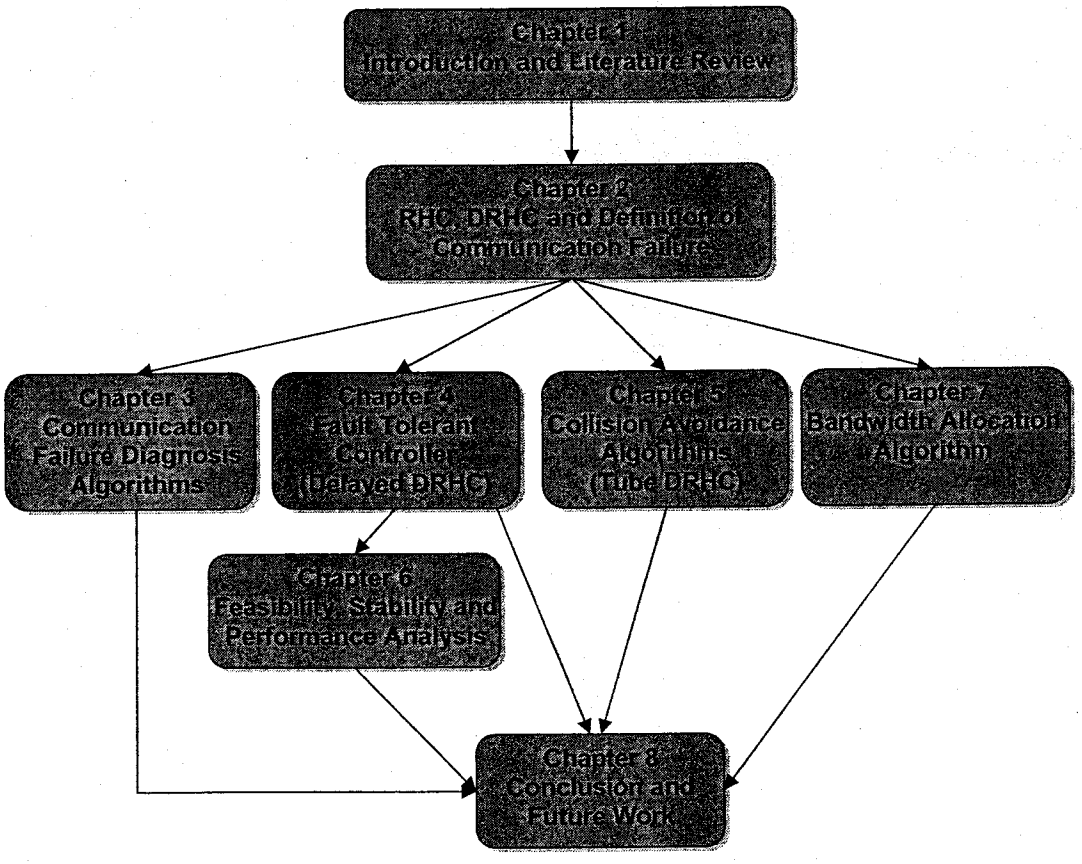

Figure 1.3: Structure of the thesis

1.5. Publications

The following publications have followed from the thesis contributions:

1) H. Izadi, B. W. Gordon, Y. Zhang, "Decentralized Receding Horizon Control for Multiple Vehicles subject to Communication Delay", *AIAA Journal of Guidance, Control and Dynamics,* vol. 32, no.6, 2009, pp. 1959-1965.

2) H. Izadi, B. W. Gordon, Y. Zhang, "Hierarchical Fault Diagnosis Algorithm for Decentralized Receding Horizon Control of Cooperative Multiple Vehicles subject to Communication Failure ", submitted to *Control Engineering Practice*.

3) H. Izadi, B. W. Gordon, Y. Zhang, "A Decentralized Receding Horizon Control in Presence of Large Communication Delays: Stability, Feasibility and Performance", Submitted to *Automatica.*

4) H. Izadi, B. W. Gordon, C. A. Rabbath, "Decentralized Receding Horizon Control for Multiple Vehicle Systems with Communication Bandwidth Allocation", submitted to *Optimal Control Applications and Methods*.

5) H. Izadi, B. W. Gordon, Y. Zhang, "Collision Avoidance for Cooperative Multiple Vehicle Systems in Presence of Large Communication Delays", Submitted to the *American Control Conference*, Baltimore, Maryland, USA, June 30 - July 2, 2010.

6) H. Izadi, B. W. Gordon, Y. Zhang, "Decentralized Communication Failure Diagnosis for Cooperative Multiple Vehicle Systems", Submitted to the *American Control Conference*, Baltimore, Maryland, USA, June 30 - July 2, 2010.

7) H. Izadi, B. W. Gordon, Y. Zhang, "Fault Tolerant Decentralized Receding Horizon Control for Cooperative Multiple Vehicle Systems", *Proceedings of the 2009 European Control Conference*, Budapest, Hungary, August 23-26, 2009.

8) H. Izadi, B. W. Gordon, Y. Zhang, "Decentralized Receding Horizon Control of Multiple Vehicles subject to Communication Failure", *Proceedings of the 2009 American Control Conference (ACC09)*, St Louis, Missouri, USA, June 10-12, 2009.

9) H. Izadi, B. W. Gordon, C. A. Rabbath, "Decentralized Receding Horizon Control Using Communication Bandwidth Allocation", *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico, December 9-11, 2008.

10) H. Izadi, B. W. Gordon, C. A. Rabbath, "Decentralized Control of Multiple Vehicles with Limited Communication Bandwidth", *the IEEE International conference on Systems, Man and Cybernetics (SMC)*, Singapore, October 12-15, 2008.

11) H. Izadi, B. W. Gordon, C. A. Rabbath, "Communication Bandwidth Allocation for Decentralized Receding Horizon", *the IEEE/ASME International conference on Advanced Intelligent Mechatronics (AIM)*, Xian, China, July 2-5 2008.

12) H. Izadi, B. W. Gordon, C. A. Rabbath, "Stability Improvement for Time Varying Decentralized Receding Horizon Control Systems", *the 13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics (MMAR)*, Szczecin, Poland, 27 - 30 August 2007.

13) H. Izadi, B. W. Gordon, C. A. Rabbath, "A Variable Communication Approach for Decentralized Receding Horizon Control of Multi-Vehicle Systems", *the 2007 American Control Conference*, New York City, USA, July 11-13, 2007.

14) H. Izadi, M. Pakmehr, B. W. Gordon and C. A. Rabbath, "A Receding Horizon Control Approach for Roll Control of Delta Wing Vortex-Coupled Dynamics", *the IEEE Aerospace conference*, March 3-10, 2007, Big Sky, MT, USA.

1.6. Notation and Terminology

In this thesis, the terms *group*, *team*, *fleet*, and *network* bear the same meaning; and so do the terms *vehicle*, *agent*, *team member*, and *subsystem*. Also, the text often talks from the local vehicle's perspective unless otherwise is emphasized.

Further, in general for scalars a lower-case *italic* notation is used (except T which is traditionally used for prediction horizon), e.g., x and y denote the components of position vector. For vectors the **bold** lower-case *italic* notation is used, e.g., $\boldsymbol{x}$ denotes the state vector. For matrices the UPPER-CASE *italic* letter is used, e.g., A is used for linear system dynamics. Also, the UPPER-CASE $\mathbb{BLACKBOARD\ BOLD}$ characters are used for sets; for instance: $\mathbb{R}$, $\mathbb{N}$, $\mathbb{X}$, $\mathbb{U}$, and $\mathbb{E}$. The specific notations corresponding to each part of the thesis are described in the first place where they appear.

Chapter 2. Background Materials and Problem Statement

This chapter presents the required background materials on the decentralized receding horizon control (DRHC) of the multiple vehicle systems. First, the formulation of quasi-infinite RHC [10] is presented, and then this framework is used to formulate the decentralized RHC. Quasi-infinite RHC architecture imposes the lowest online computation effort while satisfying the dynamical constraints. Also, it provides enough theoretical tools to analyze the stability and performance of the controller.

2.1. Receding Horizon Control Formulation

Consider the following general nominal model for the subsystems:

$$\dot{\boldsymbol{x}} = f\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector of the vehicle and $\boldsymbol{u} \in \mathbb{R}^m$ is the input vector satisfying the constraints:

$$\begin{aligned} \boldsymbol{u}(t) &\in \mathbb{U}; \quad \forall t \geq 0 \\ \boldsymbol{x}(t) &\in \mathbb{X}; \quad \forall t \geq 0 \end{aligned} \tag{2.2}$$

where $\mathbb{U}$ and $\mathbb{X}$ are the set of admissible inputs and states respectively.

Assumption 2.1: f is twice continuously differentiable and $f(0,0)=0$, which means the origin is the equilibrium of the system [10].

Assumption 2.2: $\mathbb{U}$ and $\mathbb{X}$ are compact and convex and contain the origin [10].

Assumption 2.3: All parameters of system in f are known and all states are either measurable or observable. The system is also controllable.

Assumption 2.4: the linear realization of the nonlinear dynamics (2.1) is introduced as follows:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \tag{2.3}$$

where $A = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}=0,\boldsymbol{u}=0)}$, $B = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}=0,\boldsymbol{u}=0)}$. Also, A is stabilizable [10].

Remark 2.1: the linearization is required for the off-line tuning of the matrix penalties in the cost function so that the stability is provided. In the online calculation the nominal model (2.1) is used.

In the optimal control a cost function is optimized over an infinite horizon to achieve the requirement for the stability, where the cost function is represented as follows:

$$J_\infty(\boldsymbol{x}(0), \boldsymbol{u}(0)) = \int_0^\infty q(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau \tag{2.4}$$

where $q(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))$ is assumed to be positive definite function of $\boldsymbol{x}$; it is usually selected as a quadratic cost function and reflects the performance specifications of the system and mission to minimize fuel, time, etc.

Assumption 2.5: $q(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))$ is a positive definite function of $\boldsymbol{x}$, i.e., there exist a class κ function $\gamma_1(\cdot)$ such that $q(\boldsymbol{x}, \boldsymbol{u}) \geq \gamma_1(\| \boldsymbol{x} \|)$ [152].

To address the stability, the optimal cost function (2.4) is often selected as the Lyapunov candidate function; with the *infinite* horizon the optimal cost is guaranteed to be non-increasing and then the stability analysis is straightforward. However, in real time applications with fast dynamics solving the optimization problem over the *infinite* time is computationally prohibitive. Therefore, with RHC the infinite horizon is reduced

to a finite time called the *prediction horizon*, T and hence the cost function is presented as follows:

$$J_T(\boldsymbol{x}(t),\boldsymbol{u}(t)) = \int_t^{t+T} q(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau))d\tau + F(\boldsymbol{x}(t+T)) \tag{2.5}$$

The cost $F(\boldsymbol{x}(t+T))$ is called the *final cost* (or *terminal cost*) which is used to compensate for the removed tail of the infinite cost function (2.4) in order to provide the stability.

Definition 2.1: the set $\mathbb{X}_f = \{\boldsymbol{x} \in \mathbb{R}^n \mid F(\boldsymbol{x}) \le \rho, \rho \ge 0\}$ is called the *terminal set* and $\mathbb{X}_f \subset \mathbb{X}$. ρ is chosen so that there exist a feedback controller $\boldsymbol{u}=K\boldsymbol{x}$ called *terminal controller* under which $\mathbb{X}_f$ is positively invariant set for f, i.e., for all trajectories starting in $\mathbb{X}_f$ the remaining trajectory stay in $\mathbb{X}_f$ forever when the terminal controller $\boldsymbol{u}=K\boldsymbol{x}$ is applied.

Lemma 2.1: if $q(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau)) = \| \boldsymbol{x}(\tau) \|_Q^2 + \| \boldsymbol{u}(\tau) \|_R^2$ and $F(\boldsymbol{x}) = \| \boldsymbol{x} \|_P^2 = \boldsymbol{x}'P\boldsymbol{x}$, then P is the unique, positive definite and symmetric solution of the following Lyapunov equation:

$$\tilde{A}'P + P\tilde{A} = -\tilde{Q} \tag{2.6}$$

where $\tilde{A} = A + BK + aI$, $\tilde{Q} = Q + K'RK$, I is the identity matrix, also $a \in \mathbb{R}$ satisfies: $0 < a < -\lambda_{\max}(A+BK)$ where $\lambda_{\max}(A+BK)$ is the largest eigenvalue of $A+BK$. Further, there exists a positively invariant terminal set $\mathbb{X}_f$ such that $\forall \boldsymbol{x} \in \mathbb{X}_f$:

$$\frac{d(\| \boldsymbol{x} \|_P^2)}{dt} \le -\| \boldsymbol{x} \|_{Q+R'KR}^2 \tag{2.7}$$

Proof: see [10].

Lemma 2.1 provides a systematic method for choosing P.

The trajectory of control inputs over the prediction horizon computed at time t to be applied to plant is represented by $\boldsymbol{u}_t(\cdot)$:

$$\boldsymbol{u}_t(\cdot) = \{\boldsymbol{u}(s) \mid s \in [t, t+T]\} \tag{2.8}$$

The RHC problem $\mathcal{P}(t_k)$ is defined at time t_k where $t_{k+1} = t_k + \delta$ (and $t_0 = 0$) as follows:

Problem 2.1: RHC Problem $\mathcal{P}(t_k)$:

$$\underset{\boldsymbol{u}_{t_k}(\cdot)}{\text{Min}}\, J_T(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)) \tag{2.9}$$

subject to (during $t \in [t_k, t_k + T]$):

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)); \qquad \boldsymbol{x}(t_k) = \boldsymbol{x}_{actual}(t_k) \tag{2.10a}$$

$$\boldsymbol{u}(t) \in \mathbb{U}; \;\; \boldsymbol{x}(t) \in \mathbb{X} \tag{2.10b}$$

$$\boldsymbol{x}(t_k + T) \in \mathbb{X}_f \tag{2.10c}$$

where $\boldsymbol{u}_{t_k}(\cdot)$ denotes the optimal input trajectory for time interval $[t_k, t_k + T]$.

Remark 2.2: Constraint (2.10c) is added to guarantee the stability of the RHC according to [10]; in fact, by the means of the RHC the trajectory of system is driven to a neighbourhood of the origin and after that the *terminal controller* -a linear local feedback controller- stabilizes the system; hence, for the open loop system, control input is as follows:

$$\boldsymbol{u}(t) = \begin{cases} \boldsymbol{u}_{t_k}(t) & \boldsymbol{x} \in \mathbb{X} \\ K\boldsymbol{x}(t) & \boldsymbol{x} \in \mathbb{X}_f \end{cases} \tag{2.11}$$

Where, K is the gain of the *terminal controller*.

Remark 2.3: in the quasi-infinite RHC the *terminal controller* is used only in offline computations to tune terminal matrix penalty P for providing the stability of the RHC; but in the online computations only the RHC controller scheme is used inside and outside the *terminal set*.

To provide a closed loop solution only the first portion of the optimal solution is applied to the system during a period of time called *execution time*, δ until the next sampling is available, and again a new optimization problem is solved with the updated information in each time step.

The following algorithm is used for the online implementation of RHC problem $\mathcal{P}(t_k)$:

Algorithm 2.1: RHC:

1: k=0.

2: while $x(t_k) \neq 0$. //assuming the origin is the target point

3: Measure $x(t_k)$.

4: Solve the RHC problem $\mathcal{P}(t_k)$ and generate the control trajectory $u_{t_k}(\cdot)$.

5: Execute the control action during the time interval $[t_k, t_{k+1}]$.

6: k=k+1.

7: end

Repeating this algorithm yields a closed-loop feedback control law. Algorithm 2.1 is applicable to only the case of single vehicle. For the case of multiple cooperative vehicles a centralized or decentralized approach is needed.

2.1.1.Example: Vortex Coupled Roll Dynamics of Delta Wing Aircraft

In this section, to illustrate the implementation issues of RHC, it is used to control the roll dynamics of a delta wing aircraft augmenting vortex breakdown location. This dynamics is relatively difficult to control due to the highly nonlinear dynamics involved a variety of nonlinear behaviours such as bearing friction, saturations, zero dynamics, time delay and uncertainty in both model and parameters. The simulations show that RHC have prominent capabilities such as ability to handle the saturations, flexibility for trajectory generation and designing feedback control for nonlinear systems.

2.1.1.1. Dynamical Equations

A complete experimental study has been conducted to understand the flow physics and to obtain a mathematical model for roll dynamics of delta wing aircraft as described in [106], [107]. The mathematical model is a 4th order nonlinear dynamics formulated as follows [104]:

$$\begin{aligned}
\dot{x}_1(t) &= c\,x_2(t) \\
\dot{x}_2(t) &= -c\,x_1(t) - \varepsilon_d\,x_2(t) + x_4(t) + x_4(t-\tau_d) \\
\dot{x}_3(t) &= x_4(t) \\
\dot{x}_4(t) &= u(t)/I_w - C_l(\boldsymbol{x}_{vb}(t))\,q - f_c\,sgn(x_4(t))
\end{aligned} \tag{2.12}$$

$$\boldsymbol{x}_{vb}(t) = [x_{vbl}, x_{vbr}] \tag{2.13}$$

where C_l is the rolling moment coefficient, q is the dynamic air pressure, I_w is the moment of inertia, $u(t)$ is the control input (torque) [105], f_c is the bearing friction constant, ε_d is the damping constant, c is a positive constant and τ_d is the bounded state delay.

In the above dynamics, x_3 and x_4 are the state variables corresponding to the *bank angle* (Φ) and the *roll rate* respectively. The two first states are used to incorporate the vortex breakdown location into the roll dynamics. The rolling moment coefficient (C_l) is a 3rd order polynomial of vortex breakdown locations [105]:

$$C_l(x_{vbl}, x_{vbr}) = e_0 + e_1(x_{vbl} - x_{vbr}) + e_2({x_{vbl}}^2 - {x_{vbr}}^2) + e_3({x_{vbl}}^3 - {x_{vbr}}^3) \qquad (2.14)$$

where x_{vbl} and x_{vbr} represent the vortex breakdown locations for the left and right vortices, e_0, e_1, e_2 and e_3 are coefficients which are obtained from experimental data by nonlinear polynomial curve fitting. x_{vbl}, x_{vbr} and all coefficients are computed as indicated in [105]. By considering (2.14), dynamical equation (2.12) becomes highly nonlinear; it is desired to design a feedback control law for this nonlinear dynamics.

2.1.1.2. Flat Outputs

The dimension of the RHC problem can be reduced by means of so-called *flat outputs*. It is worth mentioning that the RHC problem can be solved with and without using the flat outputs; however, using the flat outputs reduces the online computation effort of optimization problem.

A flat system can be briefly described as follows: the dynamical system (2.1) is called a flat system if there exist outputs z [29] such that:

$$z = g(x, u) \qquad (2.15)$$

and the states and the control signal can be recovered from z and its derivatives; that is,

$$(x, u) = h(z, \dot{z}, ..., z^{(r)}) \qquad (2.16)$$

which means for a system to be flat, it is required that all the system's states and control inputs can be recovered from a finite number of flat outputs derivatives and without integration of the flat outputs; the interested reader is referred to [29] for more details.

If the dimension of z is smaller than that of x and u, then the optimization problem is mapped to a lower dimension. In delta wing example, the following flat outputs can be chosen:

$$\begin{aligned} z_1(t) &= x_1(t) \\ z_2(t) &= x_3(t) \end{aligned} \tag{2.17}$$

All states and control inputs can be recovered from these outputs and their derivatives as follows:

$$\begin{aligned} x_1(t) &= z_1(t) \\ x_2(t) &= \dot{z}_1(t) \\ x_3(t) &= z_2(t) \\ x_4(t) &= \dot{z}_2(t) \\ u(t) &= I_w[\ddot{z}_2(t) + C_l(\boldsymbol{x}_{wb}(t))q + f_c\, sgn(\dot{z}_2(t))] \end{aligned} \tag{2.18}$$

Also the cost function and all constraints have to be converted to flat outputs space then one has to solve an optimization problem with only two decision variables instead of five decision variables which leads to 2.5 times reduction in computation burden.

2.1.1.3. Simulations for Stabilizing Roll Control

The cost function (2.5) is selected as the following quadratic form:

$$q(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \boldsymbol{x}'Q\boldsymbol{x} + \boldsymbol{u}'R\boldsymbol{u} \tag{2.19}$$

$$F(\boldsymbol{x}(t)) = \boldsymbol{x}'P\boldsymbol{x} \tag{2.20}$$

where P, Q and R are the controller gains and have to be adjusted such that firstly they are positive definite for a unique global minimum point of cost function and secondly since they define the importance of the term they multiply by, they have to manage the distribution of error in states, control signal and final constraint on states. Matrix penalties Q and R are selected as identity matrix and P is obtained by solving the Lyapunov equation for linearized delta wing model [10]. Polynomial class function has

been used to parameterize the flat outputs as a function of time; the order of polynomials is three. A sequential quadratic programming (SQP) [108] method has been used to solve the optimal control problem. Briefly, SQP method is an optimization method in which a quadratic cost function is solved by means of any recursive method like the *steepest decent* (SD) method.

The saturations in our case are considered to be on bank angle and input as follows:

$$-90^\circ \le \phi \le 90^\circ \qquad (2.21)$$

$$-15 \le u \le 15 \qquad (2.22)$$

These inequality constraints will be checked in all break points for all polynomials. Furthermore, there are four equality constraints:

1- Continuity of polynomial functions in break points;

2- Continuity of 1st derivative in break points;

3- Continuity of 2nd derivative in break points;

4- Initial condition on states

Also, to investigate the robustness of the system against uncertainties, random term α has been added to the right hand side of nominal system in the simulations; that is:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\alpha} \qquad (2.23)$$

where $-0.2 \le \alpha_i \le 0.2$ in radian. The prediction horizon and execution horizon are respectively set to T=1 *sec* and $\delta = 0.1\,sec$. The response of the system to an initial condition of $\phi = 60^\circ$ is depicted in Figure 2.1; as seen after about 2 seconds states vanish.

Also, Figure 2.2 shows the history of the control input. As seen, the RHC controller can stabilize the roll dynamics properly.

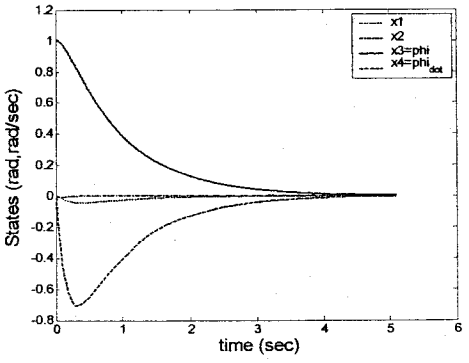

Figure 2.1: Time history of states due to an initial condition.

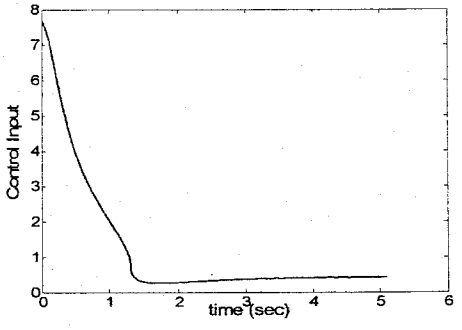

Figure 2.2: Control input time history.

2.1.1.4. Tracking Controller Design

In this section, the main purpose is to design a roll tracking controller by means of RHC. The following cost function has been used for tracking controller synthesis:

$$q(\boldsymbol{x}(t),\boldsymbol{u}(t)) = (\boldsymbol{x}-\boldsymbol{x}_{cmd})' Q(\boldsymbol{x}-\boldsymbol{x}_{cmd}) + \boldsymbol{u}' R\boldsymbol{u} \quad (2.24)$$

$$F(\boldsymbol{x}(t)) = (\boldsymbol{x}-\boldsymbol{x}_{cmd})' P(\boldsymbol{x}-\boldsymbol{x}_{cmd}) \quad (2.25)$$

where $\boldsymbol{x}_{cmd}$ is the command input, in our case this is set as a sinusoidal function. Figure 2.3 shows the time history of bank angle and the command trajectory. It shows that RHC controller perfectly follows the command trajectory.

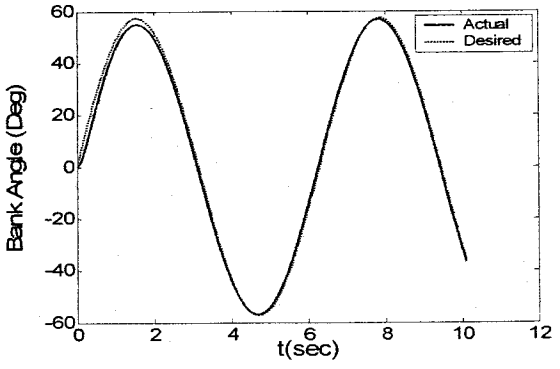

Figure 2.3: Actual and command bank angle time history.

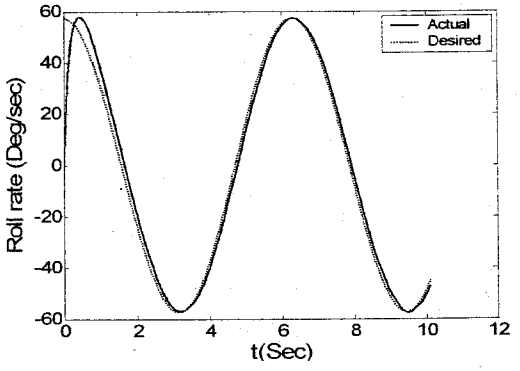

Figure 2.4: Actual and command roll rate time history.

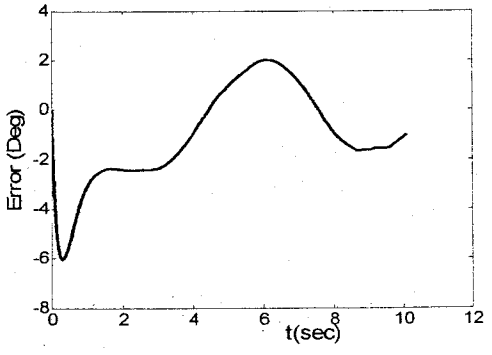

Figure 2.5: Roll rate error time history.

Also, in Figure 2.4, the actual and command trajectories of the roll rate have been depicted; it can be seen that contrary to initial mismatch between command and actual

roll rate, it follows the command roll rate properly. Figure 2.5 shows the error in roll rate. Further, Figure 2.6 shows the time history of input signal, comparing this control signal with control signals provided by other control methods [104], one can conclude that the control action generated by RHC is much smoother and without sharp oscillations which makes it possible to apply the control signal in real world applications.

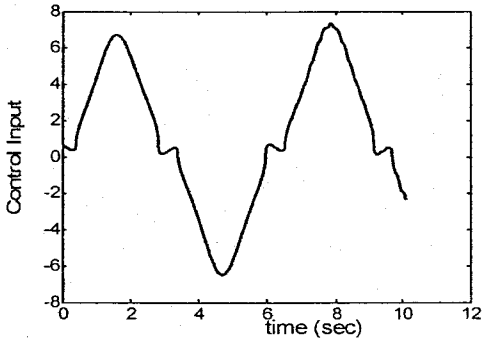

Figure 2.6: Control input time history

Together simulation results show that the RHC is capable of controlling the nonlinear dynamics of highly nonlinear, constrained and uncertain dynamics in both stabilizing and tracking cases. A Pentium IV computer with 2.5 GHz CPU with Matlab software is used to perform this simulation. It is seen that the overall computation time is reasonable and using faster optimization algorithms and programming languages such as C++, it is possible to apply the RHC method to real-time applications.

2.2. Cooperative Control

Consider a team of N_v cooperative vehicles with uncoupled dynamics. Each vehicle in the team is equipped with three main components: 1) measurement sensors, 2) communication device, and 3) computation resource. The measurement sensors of each vehicle measure its own states (it is assumed that the states are either measurable or

observable). The communication device is used to gather the information from the neighbouring vehicles and communicate with human operators. Using the computation resource, each vehicle solves a decentralized optimization problem at each sampling time based on its current measured states (from sensors) and the predicted plans of its neighbouring vehicles (provided through communication). Moreover, each vehicle has a dynamic model of its neighbouring vehicles available to calculate the neighbour's trajectory based on communicated neighbour's plans when required. It is also assumed that there are no sensor errors, actuator errors, model uncertainty, or communication noise. These assumptions allow one to focus on the core issue of the problem (communication delay); however, the proposed approaches in this thesis can be extended to the cases above by suitably modifying the proposed algorithms to account for these non-ideal effects.

In the cooperative control, one deals with a team of vehicles instead of one vehicle; then the *interaction* between vehicles must be considered. As some examples of such interactions one can point to the physical interactions such as collision avoidance, formation keeping and communication interaction.

2.2.1. Interaction Modelling

The interaction between cooperative vehicles is usually represented by an "*interaction graph*" which is described by two basic elements: *nodes* and *arcs*, where the *nodes* represent the subsystems/vehicles and an *arc* between two nodes denotes the existence of interaction between the two subsystems; the *interaction graph* is denoted by $\mathbb{G}(t)$ and represented as follows [1, 136]:

$$\mathbb{G}(t) = \{\mathbb{V}, \mathbb{E}(t)\} \tag{2.26}$$

where $\mathbb{V}$ is the set of nodes (vehicles) and $\mathbb{E}(t) \subseteq \mathbb{V} \times \mathbb{V}$ is the set of arcs (i, j) at time t, with $i, j \in \mathbb{V}$. This *interaction graph* enables one to represent all configurations of the subsystems.

Considering a set of N_v vehicles cooperating to perform a common mission, the i^{th} vehicle in the team is associated with the i^{th} node of the graph. If an arc (i, j) connecting the i^{th} and j^{th} node is present in $\mathbb{E}$, it means the i^{th} and j^{th} vehicles have an interaction; this relation is termed as *neighbourhood* for i^{th} and j^{th} vehicles.

In this thesis the main interactions arise from *control structure* and *information exchange structure*. In general it is assumed that the interaction graphs of these two interactions coincide unless otherwise is indicated. It means the vehicles will communicate if they are coupled in the cost function or constraints. It is usually assumed that the information exchange has a particular structure and is set by human operator prior to the mission. The "*interaction graph*" can be *direct* (unidirectional) or *indirect* (bidirectional).

2.2.1.1. Indirect Communication Graph Topology

The *indirect* communication graph is suitable to present a mutual relationship among vehicles, i.e., $(i, j) \in \mathbb{E}$ implies $(j, i) \in \mathbb{E}$ even though it does not appear in $\mathbb{E}$. The *indirect* communication graph topology is used when the inter-vehicle communication has a bidirectional nature. Let N_n^i denotes the number of neighbours of vehicle i when an *indirect* communication topology is used.

2.2.1.2. *Direct Communication Graph Topology*

Some extensions should be performed to the *indirect* graph topology to distinguish between *leader* and *follower*. Still if an arc (i, j) connecting the i^{th} node to the j^{th} node is present, it means that the i^{th} vehicle has a coupling term containing the j^{th} vehicle's states in its cost function and/or in its constraint (interaction), and hence communicate with j^{th} vehicle. This relationship is termed as *neighbourhood* for the i^{th} and j^{th} vehicles and it is said that:

- i^{th} and j^{th} vehicles are *neighbouring* vehicles and
- i^{th} vehicle is the *follower* of the j^{th} vehicle and
- j^{th} vehicle is the *leader* of the i^{th} vehicle.

The main distinction between *indirect* and *direct* graph topology is that with the *direct* interaction graph, $(i,j) \in \mathbb{E}$ does not imply necessarily $(j,i) \in \mathbb{E}$ unless it appears explicitly in $\mathbb{E}$. Also, with the *direct* interaction graph any vehicle *i* can be both *leader* and *follower* to neighbouring vehicle *j*. Further, any vehicle *i* can be *leader* to neighbouring vehicle *j* and *follower* to another vehicle. Using this flexible graph topology allows representing all types of interactions among subsystems. Also, let N_l^i and N_f^i denote the number of the *leaders* and *followers* of vehicle *i* respectively. This representation does not conflict with *indirect* graph formulation; it is a general form of *indirect* graph topology. The *direct* graph topology allows formulating the *leader-follower* formation problems.

2.2.2. DRHC Terminology

The notation presented in this section is the key issue to understand the analysis presented in this thesis. The possible state vectors are introduced as follows:

- $\boldsymbol{x}^{i}(t)$: the actual state vector of the i^{th} vehicle at time t.

- $\boldsymbol{x}_{t_k}^{i,j}(t)$: the state vector of the i^{th} vehicle at time t, predicted (estimated) by the j^{th} vehicle at time step t_k .

Then, the state of vehicle i predicted by itself at time t_k is represented by $\boldsymbol{x}_{t_k}^{i,i}(t)$. Further, the sequence of these states over the prediction horizon (during $[t_k, t_k+T]$) is called the state trajectory of vehicle i calculated by itself and is represented by $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$ which is defined by default on the interval $[t_k, t_k+T]$:

$$\begin{aligned}\boldsymbol{x}_{t_k}^{i,i}(\cdot) &= \left\{\boldsymbol{x}_{t_k}^{i,i}(t) \mid t \in [t_k, t_k+T]\right\} \\ \boldsymbol{u}_{t_k}^{i,i}(\cdot) &= \left\{\boldsymbol{u}_{t_k}^{i,i}(t) \mid t \in [t_k, t_k+T]\right\}\end{aligned} \tag{2.27}$$

However, if the trajectory is defined on an interval which is different from $[t_k, t_k+T]$ by other vehicles, then the beginning and end time are indicated as: $\boldsymbol{x}_{t_k}^{i}(t_b : t_e)$, i.e.,

$$\begin{aligned}\boldsymbol{x}_{t_k}^{i,j}(t_b : t_e) &= \left\{\boldsymbol{x}_{t_k}^{i,j}(t) \mid t \in [t_b, t_e],\ j \in \mathbb{V},\ (i,j) \in \mathbb{E}\right\} \\ \boldsymbol{u}_{t_k}^{i,j}(t_b : t_e) &= \left\{\boldsymbol{u}_{t_k}^{i,j}(t) \mid t \in [t_b, t_e],\ j \in \mathbb{V},\ (i,j) \in \mathbb{E}\right\}\end{aligned} \tag{2.28}$$

where $[t_b, t_e]$ is the interval on which the trajectory is defined. This notation is used in Chapter 4 where the missing trajectory of faulty vehicle is estimated by neighbours.

Then let the following represent the concatenated state and input trajectories of the neighbours of the i^{th} vehicle at time t_k :

$$\begin{aligned}\boldsymbol{x}_{t_k}^{-i}(\cdot) &= \left\{\boldsymbol{x}_{t_k}^{j,j}(\cdot) \mid j \in \mathbb{V},\ (i,j) \in \mathbb{E}\right\} \\ \boldsymbol{u}_{t_k}^{-i}(\cdot) &= \left\{\boldsymbol{u}_{t_k}^{j,j}(\cdot) \mid j \in \mathbb{V},\ (i,j) \in \mathbb{E}\right\}\end{aligned} \tag{2.29}$$

The following summarizes the notation presented in this section:

$$x^{owner,\ calculator}_{calculation\ time}(t_b, t_e) \tag{2.30}$$

where the parameter x belong to the first superscript (*owner*); the 2nd superscript (*calculator*) calculates that parameter at time mentioned in subscript (*calculation time*). t_b and t_e are the start and final time of the trajectory. For example, $x^{i,j}_{t_k}(t_b : t_e)$ is the state trajectory of vehicle i over the time interval $[t_b, t_e]$ which is computed by vehicle j at time t_k.

2.2.3. Delay-Free (Fault-Free[1]) DRHC Formulation

In some previous work [1, 161] a DRHC scheme is used where the vehicles need to exchange only their instant states and each vehicle predicts the trajectory of neighbouring vehicles to have an estimate of their future plan, the main disadvantage of that method is its high computation time and thus it is not suitable for very fast dynamics. However, for the scheme presented in this thesis the predicted trajectories are exchanged instead of being estimated thereby reducing the online computation time. Figure 2.7 shows the inter-vehicle communication between two neighbouring vehicles at time t_k for the delay-free condition. As seen the information exchange is not subject to communication delay.

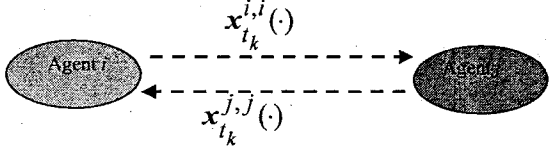

Figure 2.7: The inter-vehicle communication between two neighbours in the delay-free condition

[1] Since in this thesis in the fault-free conditions a delay-free DRHC scheme is used, both delay-free and fault-free terms refer to the same architecture. And so do the terms delayed-DRHC and fault tolerant DRHC.

However, due to computation delay at least a one-step delay should be considered as the computed trajectories are not available instantly; then the *information set* of the i^{th} vehicle for the case of delay-free DRHC is introduced as follows:

$$\Gamma^i(t_k) = \left\{ \boldsymbol{x}^i(t_k),\ \boldsymbol{x}^{-i}_{t_{k-1}}(\cdot) \right\} \tag{2.31}$$

where set $\Gamma^i(t_k)$ contains the updated information available to the i^{th} vehicle at time t_k and is referred to the *information set* in this thesis. This collects 1) the instant state vector of i^{th} vehicle and 2) the concatenated state trajectory of neighbours calculated at the previous time step ($\boldsymbol{x}^{-i}_{t_{k-1}}(\cdot)$).

For the particular case of formation control, the delay-free decentralized cost function for the i^{th} vehicle in the team at time t_k is defined as follows:

$$\begin{aligned} J^i(\Gamma^i(t_k)) = & \int_{t_k}^{t_k+T} \left(\left\| \boldsymbol{x}^{i,i}_{t_k}(t) - \boldsymbol{x}^{c,i} \right\|_Q^2 + \left\| \boldsymbol{u}^{i,i}_{t_k}(t) \right\|_R^2 \right) dt + \left\| \boldsymbol{x}^{i,i}_{t_k}(t_k+T) - \boldsymbol{x}^{c,i} \right\|_P^2 \\ & + \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_k+T} \left\| \boldsymbol{x}^{i,i}_{t_k}(t) - \boldsymbol{x}^{j,j}_{t_k}(t) - \boldsymbol{r}^{i,j} \right\|_S^2 dt \end{aligned} \tag{2.32}$$

where $\|\boldsymbol{x}\|_Q^2 = \boldsymbol{x}'Q\boldsymbol{x}$ and P, Q, and R are positive definite and symmetric matrices, S is positive semi-definite symmetric matrix, $\boldsymbol{x}^{c,i}$ is the state vector of target of vehicle i, and $\boldsymbol{r}^{i,j}$ is the vector of desired relative position between agents i and j.

Remark 2.4: The cost $\left\| \boldsymbol{x}^{i,i}_{t_k}(t) - \boldsymbol{x}^{j,j}_{t_k}(t) - \boldsymbol{r}^{i,j} \right\|_S^2$ is called the *coupling* or *cooperation cost*. This cost with positive semi-definite matrix penalty S allows incorporating the desired states with a desired degree of importance in the cooperation cost. In other literatures such as the work by Keviczky et al. [1], it is simply assumed that S=Q; but

practical and simulation trials suggest that different S and Q facilitates tuning the controller and providing a balance between global and local objectives as S represents the penalty for local cooperation objectives and Q represents the penalty for global objectives in the cost function.

2.2.4. Delay-Free DRHC Problem

Assume the dynamics of the homogeneous vehicles is represented by (2.1). Then, the delay-free decentralized receding horizon (DRHC) problem $\mathcal{P}^i(t_k)$ is then defined for the i^{th} vehicle at time t_k as follows:

Problem 2.2: Delay-Free DRHC Problem $\mathcal{P}^i(t_k)$ $(i \in \mathbb{V})$:

$$\underset{\{\boldsymbol{u}_{t_k}^{i,i}(\cdot),\, \boldsymbol{x}_{t_k}^{i,i}(\cdot)\}}{\text{Min}} \quad J^i(\Gamma^i(t_k)) \tag{2.33}$$

Subject to (for $t \in [t_k, t_k + T]$):

$$\begin{aligned} &\dot{\boldsymbol{x}}_{t_k}^{i,i}(t) = f(\boldsymbol{x}_{t_k}^{i,i}(t), \boldsymbol{u}_{t_k}^{i,i}(t)); \\ &\boldsymbol{x}_{t_k}^{i,i}(t_k) = \boldsymbol{x}^i(t_k); \end{aligned} \tag{2.34a}$$

$$\boldsymbol{x}_{t_k}^{i,i}(t) \in \mathbb{X}^i,\ \boldsymbol{u}_{t_k}^{i,i}(t) \in \mathbb{U}^i; \tag{2.34b}$$

$$\boldsymbol{x}_{t_k}^{i,i}(t_k + T) \in \mathbb{X}_f^i \tag{2.34c}$$

where $\mathbb{X}^i$, $\mathbb{U}^i$ and $\mathbb{X}_f^i$ denote the set of admissible states, inputs and final set (terminal region), respectively, for the i^{th} vehicle.

2.2.5. Delay-Free DRHC Algorithm

The following algorithm is presented for the on-line implementation of the delay-free DRHC problem $\mathcal{P}^i(t_k)$. The algorithm is formulated for the i^{th} vehicle as follows:

Algorithm 2.2: Delay-Free DRHC (online)

1: Let *k=0,* measure $\boldsymbol{x}^i(t_k)$ and GOTO step 3.

2: Receive the trajectory $\boldsymbol{x}_{t_{k-1}}^{j,j}(\cdot)$ from neighbours j (where $(i,j) \in \mathbb{E}$), measure $\boldsymbol{x}^i(t_k)$ and update the information set $\Gamma^i(t_k)$ from Eq. (2.31).

3: Solve $\mathcal{P}^i(t_k)$ and generate the control action $\boldsymbol{u}_{t_k}^{i,i}(\cdot)$ and the state trajectory $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$.

4: Send the trajectory $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$ to the neighbouring vehicles.

5: Execute the control action for the individual vehicle *i* during $[t_k, t_{k+1}]$:

$$\boldsymbol{u}^i(t) = \boldsymbol{u}_{t_k}^{i,i}(t);\ t \in [t_k, t_{k+1}] \tag{2.35}$$

6: *k=k+1.* GOTO step 2.

Algorithm 2.2 is repeated until the assigned targets are reached. The targets are assumed to be known and assigned to each agent *a priori* by a task assignment algorithm.

Algorithm 2.2 is a relevant algorithm for DRHC implementation and is used extensively in other literatures [42, 109] with some small changes.

2.2.6. Example: Formation Control of Unmanned Vehicles

A leaderless formation of a fleet of 6 unmanned vehicles with the following 3DOF dynamics is considered [110]:

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -x_2 + u_1\\ \dot{x}_3 &= x_4\\ \dot{x}_4 &= -x_4 + u_2\end{aligned} \tag{2.36}$$

where $\boldsymbol{x} = [x, \dot{x}, y, \dot{y}]$ and $\boldsymbol{u} = [u_1, u_2]$ are the state and input vectors respectively, also, x and y are the components of position vector. The inputs are saturated at: $0 \le u_1 \le 10$ and $0 \le u_2 \le 10$ (m/sec^2). Further: $\sqrt{\dot{x}^2 + \dot{y}^2} \le 10\ m/\sec$. These values are used for the modeling of all team members.

For the simulations CORA (Control Optimization and Resource Allocation) library developed in CIS (Control and Information Systems) laboratory of Concordia University is used. CORA is an object oriented library based on the Microsoft C++ environment and uses the SNOPT optimization package [112] to solve the RHC and other optimization problems. Also, for generating the trajectories CORA uses the B-Spline type of basis functions. One of the main functions added to CORA by user for formation problems is Set_Formation() that gets executed before starting any optimization to calculate the geometry of formation and defining the graph topology. In this function all vehicles are assigned manually to one specific node of a formation, such as triangular formation, and during the mission the vehicles are supposed to participate in formation cohesion. The formation in this chapter is *leaderless*, i.e., at any time each vehicle aligns itself with neighbouring vehicles; in the other word, each vehicle assumes itself as the *follower* of all its neighbours. The *indirect* communication graph topology is selected as follows:

$$\begin{aligned}\mathbb{V} &= \{1,2,3,4,5,6\}\\ \mathbb{E} &= \{(1,2),(2,3),(2,4),(4,5),(6,3)\}\end{aligned} \tag{2.37}$$

The following section describes the formation geometry calculation used in Set_Formation().

2.2.6.1. Formation Geometry

Assume vehicle F (Follower) tries to align itself with its neighbour L (Leader):

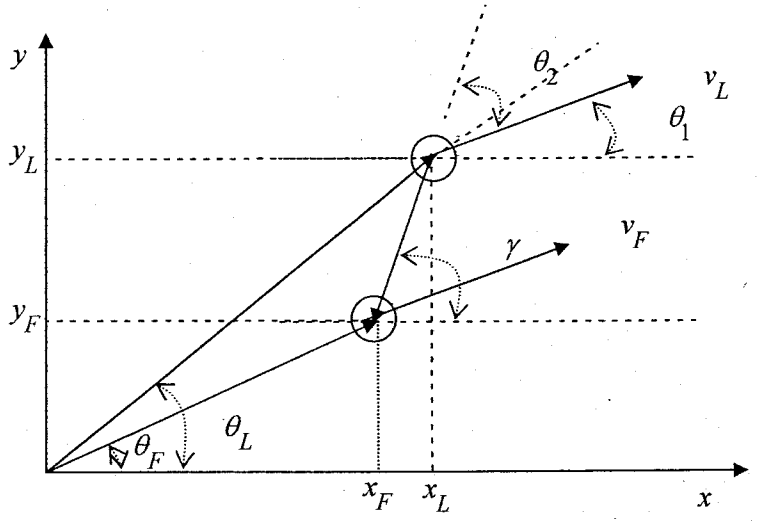

Figure 2.8: Geometry of formation for each pair of vehicles: Follower (F), Leader (L).

For the formation it is required that at any time the angle between the velocity vector and relative position vector (θ_2) remain constant to a predefined value, and velocity vector of Follower be the same as Leader, then:

$$\begin{aligned} u_F &= u_L \\ v_F &= v_L \\ x_F &= x_L - d\cos(\gamma) \\ y_F &= y_L - d\sin(\gamma) \end{aligned} \tag{2.38}$$

where, u and v are the component of velocity vector and:

$$\begin{aligned} \gamma &= \theta_1 + \theta_2 \\ \theta_1 &= \text{atan}(\frac{v_l}{u_l}) \end{aligned} \tag{2.39}$$

where θ_2 is the desired angle between the velocity vector of vehicle L and relative position vector and to be defined by user, the formation is kept by regulating this angle. If $\theta_2 = 60^\circ$ then an equilateral triangle formation is resulted. Other formulations for formation along with the stability discussions are carried out by Dunbar et al. [145], Gu et al [147] and Wesselowski et al [146].

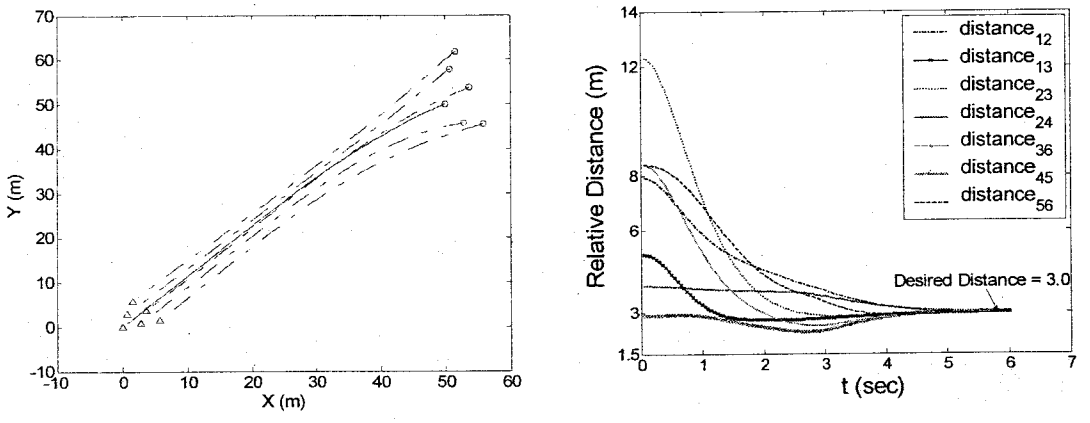

Figure 2.9: Triangular formation of a fleet of 6 unmanned rotorcrafts.

A triangular formation of a fleet of 6 vehicles and the corresponding relative distances are shown in Figure 2.9 where the initial positions of vehicles are perturbed. It is seen that the distances are stabilized to the desired value.

2.3. Problem Statement: Communication Failure

Recent research in the field of cooperative vehicle systems has increasingly considered practical implementation issues in addition to the theoretical issues. Issues such as communication requirements, and model uncertainties have thus received significant attention, see for example: [132, 136, 150, 153-156, and 158]. In this thesis,

the decentralized receding horizon control (DRHC) of multiple cooperative vehicles where the inter-vehicle communication is subject to delay is considered. The neighbouring vehicles exchange their predicted trajectory at each sample time to maintain the cooperation objectives. Such large communication delays can lead to poor performance and even instability. The communication delay can result from two main sources: 1) the communication failure and 2) limited communication bandwidth of communication channel.

Some examples of communication failures leading to large communication delays for the team of cooperative vehicles can be found in [128, 132, and 133]. In [128], the wireless communication packet loss/delay is considered. Also, in [132], the communication failure in the formation flight of multiple UAVs leads to break in the communicated messages that force the fleet to redefine the communication graph.

It is assumed that in the fault-free condition, the vehicles communicate with each other through a high performance communication channel. Then, the high performance communication devices (transmitter-receiver), installed on each vehicle, enable vehicles to communicate through the high performance channel with neighbouring vehicles with a very small delay, typically smaller than the sampling time. Then the communication failure involves the failure of the high performance communication device and hence the faulty condition is defined as follows:

Faulty Condition: The high performance communication device installed on one vehicle fails, which does not allow this vehicle to send/receive the information to/from the neighbouring vehicles; the vehicle whose communication device fails is called the faulty vehicle.

The rest of the thesis will answer the questions regarding the detection of such failures and designing new controllers which are capable of handling such failure while still maintain cooperation among vehicles and the safety of the fleet in faulty condition is provided.

Limited communication bandwidth can also give rise to communication delay and lead to poor performance and instability; then, an appropriate communication bandwidth allocation algorithm is required which allows each vehicle to distribute the available communication bandwidth to its neighbouring vehicles so that the overall teaming performance is optimized.

2.3.1. Hierarchical Fault Diagnosis and Fault Tolerant Algorithm

Since, the fault detection, fault tolerant controller and team safety need different computation and communication requirements a hierarchical approach is used for managing the fleet in the faulty conditions. The proposed hierarchy has two main layers:

1) *Higher layer* which accounts for coordination among neighbouring vehicles for fault diagnosis.

2) *Lower layer* which accounts for control of local vehicles. This layer includes fault tolerant delayed DRHC and collision avoidance algorithm. The *lower layer* also uses the decision of the *higher level* to adjust the control action.

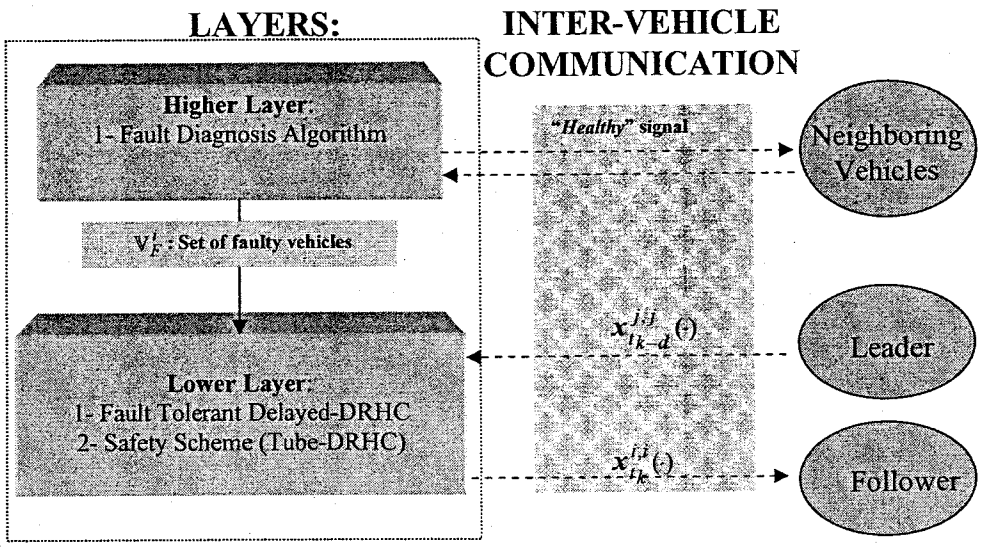

Figure 2.10: Hierarchy of the fault handling scheme and the corresponding communication requirement for each layer.

Each layer needs different communication, information flow and computation requirements. A sketch of the hierarchy is shown in Figure 2.10. The fault diagnosis algorithm needs more cooperation among the vehicles and is placed in the higher layer of the hierarchy. The fault tolerant controller deals with the local control actions and hence is placed in the lower layer. Also, the safety scheme is placed in the lower layer of the hierarchy due to its close relation with the fault tolerant controller. The information flow and communication requirements are also shown (and described throughout the thesis); the dashed arrows show the inter-vehicle communication.

2.4. Summary

In this Chapter first the quasi-infinite RHC architecture is formulated and applied to a complex nonlinear dynamics in order to demonstrate the implementation issues of RHC and also illustrate the capabilities of RHC to handle the highly nonlinear and constrained

dynamics. Then, the *direct* and *indirect* graph topology is introduced to formulate the interaction among the cooperative vehicles. Also, a decentralized receding horizon controller (DRHC) along with the implementation algorithms is formulated. Finally the main problem of the thesis which is the failure of high performance communication devices is explained; a hierarchical approach is used in this thesis to handle the communication failure.

Chapter 3. Communication Failure Diagnosis

The failure diagnosis algorithm is the main part of any fault tolerant architecture. For the cooperative decentralized multiple vehicle systems, the decentralized nature of the problem and the requirement that each vehicle in the group must independently detect the communication failures result in a challenging diagnosis problem.

A few research works explicitly address the communication failure detection for multiple vehicles. A very closely related work is presented in [132] where it is desired to manage the communication failures in formation flight of multiple UAVs; it is assumed that the communication failure leads to information flow blockage to and from faulty vehicle. Hence, in [132] to keep all aircrafts informed about all operational members in the group, an extra broadcasting communication channel is used. If after some specific time one aircraft has not sent its "*alive*" signal through the backup communication channel, that aircraft is considered lost. Whenever an aircraft is lost the formation must be reconfigured to a predetermined allowable formation, available to all vehicles, and the vehicles use an appropriate manoeuvre to reconfigure to the new formation. In another related work [128], two faults for formation flight of UAVs are considered: 1) GPS sensor failure and 2) wireless communication packet losses. To detect the GPS sensor failure a state/output observer is used which monitors the behaviour of a UAV. The output of the observer is compared with the GPS data, and if the difference is larger than some threshold then a GPS fault is identified. Furthermore, in [128] to detect the

communication packet loss/delay, the faults are identified by numbering the packets sequentially and the number of the packet is also transmitted; a mismatch between the expected packet number and the received packet number implies the occurrence of a failure (packet loss). Once the packet loss/delay occurs, the previous available trajectory of the UAV is extrapolated to predict the future reference trajectory.

The failure diagnosis algorithm for each vehicle $i \in \mathbb{V}$ includes: 1) monitoring and detecting the faulty situation, 2) identifying the faulty vehicle in the team (whether $i \in \mathbb{V}$ is faulty or its neighbours, and what neighbour is faulty).

The proposed diagnosis algorithms in this thesis are based on the notion that communication can lead to break or large delay in the exchanged messages. Depending on the communication topology and the devices employed, different algorithms may be required to diagnose failure and identify the faulty vehicles in the group. Three fault diagnosis algorithms are developed for cases with communication that is bidirectional, unidirectional, and employed with separate transmitter and receiver units. For each case, the necessary conditions are derived for the communication graph topology under which failure is detectable. Using probability analysis the reliability of the proposed algorithms is also investigated.

3.1. Communication system

Figure 3.1 shows a general schematic of a communication system. Every communication system has three main units [129]: *transmitter (TX)*, *receiver (RX)* and *communication channel*. *Transmitter* is a device, installed on the source vehicle, which converts the messages to the suitable signals such as electrical or electromagnetic signals;

also, using an antenna the *transmitter* propagates this signal through the communication channel. A *receiver* device has an operation inverse to the operation of the *transmitter* [129]; the *receiver* is installed on the destination vehicle; radio and telephone are two examples of receivers. A *channel* is a medium used to carry the information from *transmitter* to *receiver* [129]; it can be a wire, a band of frequencies, light or whatever that can carry the signals.

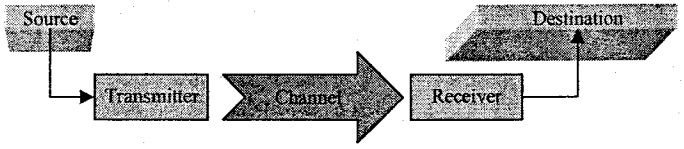

Figure 3.1: Schematic of communication system

In this thesis it is assumed that each vehicle in the team receives/sends the information to/from neighbours. Hence, each vehicle needs both *transmitter* and *receiver* units. The communication system may have both these parties together in one unit similar to *duplex communication systems*, or two separate devices may be used for TX and RX. In this chapter both cases are addressed.

More precisely, the communication failure in this thesis is referred to the situations where the *transmitter-receiver* devices do not work due to any reason. If they are embedded in one single unit then the failure of each one implies the failure of both. The neighbouring vehicles cooperate to detect such failures.

3.2. Failure Detection Scheme

To monitor the status of the communication devices a "*Healthy*" signal is introduced which is communicated between each pair of neighbouring vehicles at each time step.

The "*Healthy*" signal does not impose considerable communication load but allows providing more coordination among vehicles for communication failure detection. Also, the terms *small communication delays* τ_S and *large communication delays* τ_L are defined; let δ denotes the sampling time then $\tau_S < \delta$ and $\tau_L \geq \delta$. The sampling time is chosen as the threshold between the small and large communication delays as most of the decentralized control schemes for cooperative multiple vehicles require the information from neighbouring vehicles before any sampling time. It means if the communicated messages are subject to large delays a communication failure is concluded from the control perspective and a reconfigurable fault tolerant controller which relies on the delayed information should be employed. This is the reason why in this thesis the small communication delays are considered as the delay-free cases.

In the fault-free situation, at each time step every vehicle in the team receives/sends the "*Healthy*" signal from/to their neighbours without delay (or a small delay as less than sampling time or any other thresholds). Figure 3.2 shows the inter-vehicle communication between two neighbouring vehicles and the information exchanged at time t for the fault-free (delay-free) condition. As seen the exchanged messages are not subject to delay.

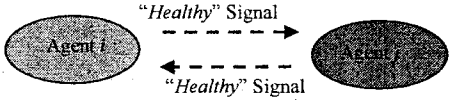

Figure 3.2: The inter-vehicle communications between two neighbours in fault-free condition

The proposed communication failure scheme is based on the fact that the communication failure results in break/delay in the communicated messages and hence if the communication delay of received messages is larger than sampling time, which is the

limit between *small communication delays* and *large communication delays*, the occurrence of communication failure is concluded (See Figure 3.3 and compare with Figure 3.2). Both faulty vehicle and its neighbours can use this sign to detect the failure. However, depending on the type of communication topology this idea needs to be expanded to find which vehicle is faulty in the team.

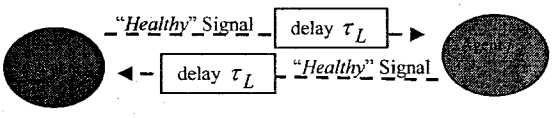

Figure 3.3: The inter-vehicle communication between faulty vehicle *i* and healthy neighbour *j*.

3.2.1. Failure Detection with Indirect Communication Graph Topology

When the communication topology is *indirect* the vehicles are forced to maintain a bidirectional communication structure.

Assume at some time the vehicle $i \in \mathbb{V}$ does not hear from its neighbours. The question is: how vehicle *i* determines whether the break in the messages is due to failure in its own communication device or that of its neighbours. The approach presented here to answer this question requires that each vehicle in the group to have at least two neighbours, i.e., $N_n^i \geq 2 \; ; i \in \mathbb{V}$ and works based on the following rules:

1- *If vehicle i hears from all neighbours* ($N_n^i \geq 2$) *then it concludes that neither its communication device nor those of neighbours is faulty.*

2- *If vehicle i does not hear after a certain time* ($\tau_L \geq \delta$) *from all its neighbours it concludes that its communication device is faulty and does not allow it to communicate with neighbours.*

3- *Accordingly, once the vehicle i hears from at least one neighbour without delay it concludes that its communication device is not faulty, and the communication device of its neighbour(s) is faulty which does not allow vehicle i to hear from that (them).*

In this way the failure is detected and the faulty vehicle is identified by all neighbours. This fault detection algorithm is summarized in Figure 3.4.

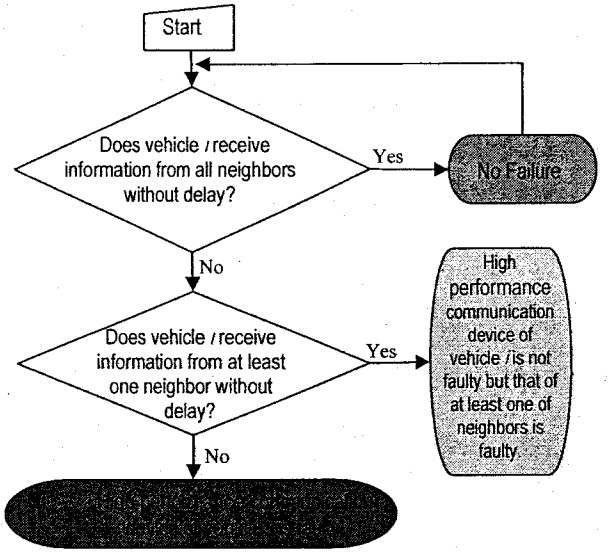

Figure 3.4: Fault detection algorithm for vehicle *i* with *indirect* communication topology

3.2.2. Failure Detection with Direct Communication Graph Topology

The main difference between the *indirect* and *direct* communication topology is that with the direct graph topology the communication flow may not be necessarily bidirectional between two neighbouring vehicles; this implies there may exist some team members which do not receive any information from neighbours as they are only leaders. On the other hand, the proposed fault detection algorithm presented in Section 3.2.1 for *indirect* graph topology requires that each vehicle in the team receives the information from at least two team members. Thus, the algorithm in Section 3.2.1 fails for the case of

direct communication graph where there may exist some vehicles which are only *leader* and do not receive information from other team members. Thus the condition $N_n^i \geq 2 \; ; i \in \mathbb{V}$ for *indirect* graph is changed to $N_l^i \geq 2 \; ; i \in \mathbb{V}$ for *direct* graph. The rest of the job follows the same idea and rules as Section 3.2.1 for indirect graph. This fault detection algorithm is summarized in Figure 3.6.

In the fault-free condition all the followers receive the "*Healthy*" signal from their leaders with no delay (or a small delay as less than sampling time). If the communication delay of received "*Healthy*" signal is larger than sampling time (execution horizon), which is the limit between *small communication delays* and *large communication delays*, the occurrence of communication failure is concluded (See Figure 3.5 and compare with Figure 3.2).

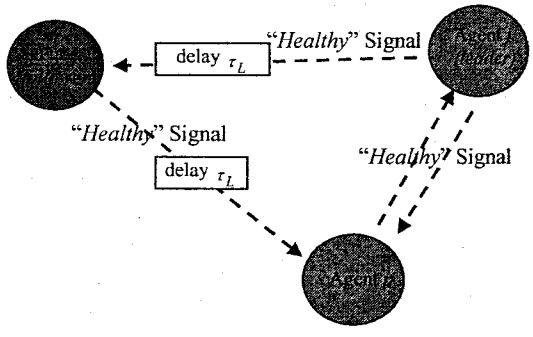

Figure 3.5. The inter-vehicle communication between faulty (agent *i*) and healthy (agents *p* and *j*) vehicles using *direct* communication topology.

3.2.3. Separate Receiver (RX) and Transmitter (TX) Devices

So far, it is assumed that the transmission and receiving the information is performed through the same unit for each vehicle and the communication unit failure implies the failure of both transmitter and receiver. However, if the communication unit has separate

receiver (RX) and transmitter (TX) devices [129] (or channels) then the failure may happen to only one of them; this case needs more complex fault detection algorithms.

For instance, if the TX of one vehicle is faulty and the RX is healthy then the faulty vehicle can receive the information from neighbours but it is not able to send the information to neighbours; in this situation the faulty vehicle does not know whether the neighbouring vehicles can receive the information or not if the presented algorithms of previous sections are used. Then a suitable algorithm is required to monitor and detect the failure in RX and TX separately.

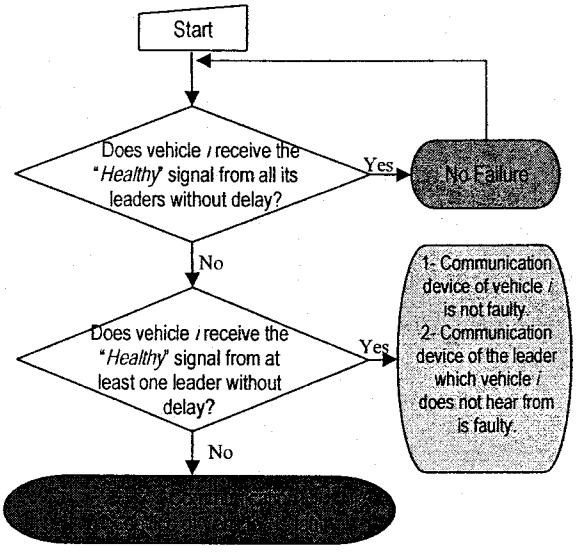

Figure 3.6. Fault detection algorithm for vehicle *i* with *direct* communication topology

3.2.3.1. RX Fault Detection

If the RX of one vehicle is faulty then depending on whether the communication topology is *direct* or *indirect* any of the algorithms presented in Sections 3.2.1 and 3.2.2 can be used to detect this failure. In this case, the "*Healthy*" signal is used to detect the possible failures in RX.

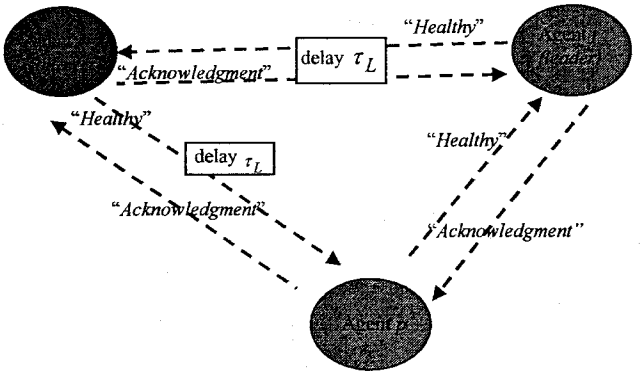

Figure 3.7. The inter-vehicle communication between faulty (agent *i*) and healthy (agents *p* and *j*) vehicles when the transmitter TX and receiver RX communication devices are separate.

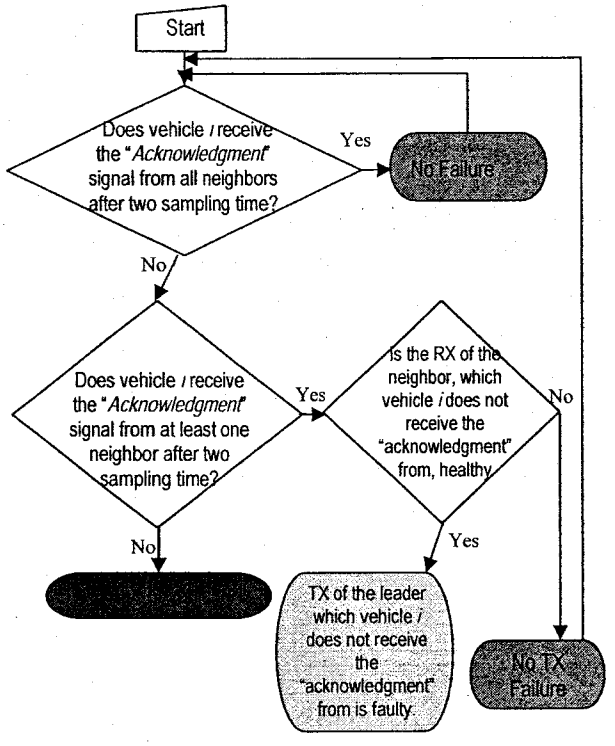

Figure 3.8. TX fault detection algorithm for vehicle *i* with *direct* or *indirect* communication topology

3.2.3.2. TX Fault Detection

To detect the failure an "*Acknowledgment*" signal is considered which imposes the neighbouring vehicles to acknowledge the receipt of the "*Healthy*" signal, see Figure 3.7.

Any vehicle $j \in \mathbb{V}$ receiving the "*Healthy*" signal from neighbour $i \in \mathbb{V}$ will acknowledge the receipt of the "*Healthy*" to the sender $i \in \mathbb{V}$ by sending the "*Acknowledgment*" signal. If any vehicle $i \in \mathbb{V}$ does not receive the "*Acknowledgement*" signal after two sampling time from at least one of its neighbours, then a failure in TX is concluded. Also, it is required that $N_f^i \geq 2 \; ; i \in \mathbb{V}$. This way each vehicle can detect the possible failures in its TX. Figure 3.8 shows the algorithm for detecting the TX failure.

3.3. Reliability against Simultaneous Failures

The presented fault detection algorithms in this chapter for the communication failure may fail if simultaneous failure happens and the communication graph topology is not well-connected. The reliability of the proposed algorithms is investigated through probability analysis. The proposed fault detection algorithms of Sections 3.2.1 and 3.2.2 work properly if the communication device of at least one neighbour is healthy. The first question is: is it possible that the communication devices of all neighbours of vehicle *i* fail simultaneously which leads to false positive failure conclusion for vehicle *i* about its communication device? The answer is yes; if the probability of failure for one communication device is *p*, the probability for simultaneous failure of communication device of all neighbours for *indirect* communication topology is calculated as follows:

$$Probability\ of\ Simultaneous\ Failures = p^{N_n^i} \qquad (3.1)$$

Then the reliability is:

$$Reliability = 1 - p^{N_n^i} \qquad (3.2)$$

and likewise for *direct* communication topology:

$$Reliability = 1 - p^{N_l^i} \tag{3.3}$$

For the case where TX and RX are separate, since the TX fault detection is based on sending the "*Healthy*" signal and receiving the confirmation through the "*Acknowledgment*" signal then the Byzantine fault [130, 131] may occur. The Byzantine fault is first referenced in a network of computers where the receipt of any message to any destination computer is confirmed through sending back a confirmation signal. However, the confirmation signal needs another confirmation signal from the recipient. This leads to an inconclusive sequence of events and is referred to as the Byzantine fault and for distributed networked a Byzantine fault tolerance algorithm may be required [131]. Then the reliability of TX fault detection against Byzantine fault is calculated as follows:

$$Reliability = 1 - \left[p^{N_l^i} + p^{N_f^i} \right] \tag{3.4}$$

The reliability decreases dramatically as the number of neighbours increases. For example if the probability of failure of one vehicle's communication device is p=10%=0.1 and vehicle i has $N_n^i = 2$ neighbours, then the probability for simultaneous failure of communication devices of both neighbours of vehicle i is 0.1^2=0.01 and the reliability of algorithm is: 1.0-0.01 = 0.99 or 99%. This suggests that for a more reliable fault detection algorithm more vehicles should communicate and exchange the "*Healthy*" and "*Acknowledgment*" signals; in fact, more cooperation leads to more reliability and the desired reliability of the algorithms can be achieved by appropriate setting the communication graph topology.

3.4. Summary

In this chapter three algorithms are presented to detect the communication failure and the faulty vehicle in the team for the case of *direct* communication graph, *indirect* communication graph and cases where there exist separate devices for receiving and transmitting the information. The proposed algorithms require that each vehicle in the team exchange the information with at least two other neighbours.

The reliability of the proposed algorithms is also discussed. A formula is suggested for calculating the reliability for each algorithm. It is concluded that more reliability against the simultaneous failures can be achieved by strongly connecting the *communication graph topology*.

Since the strong connection of the *communication graph topology* is only required for fault detection purpose, it suggest that using two different communication graph topologies for failure detection in *higher-layer* of hierarchy of Figure 2.10 and fault tolerant controller in *lower-layer* of that hierarchy can optimize the communication load over the network. Therefore, a hierarchical approach for the fault detection algorithm and the fault tolerant controller is preferred.

Chapter 4. Fault Tolerant Controller

Once the failure is detected and the faulty vehicle is identified in the team a suitable fault tolerant scheme is required to address the communication failure. In Section 2.3 it is mentioned that all the vehicles have access to a high performance communication channel, and that the faulty condition involves the failure of the high performance communication devices by which the communication through the high performance communication channel is possible. If no line of communication exists between the faulty vehicle and its neighbours, then there would be no remedy except forcing the faulty vehicle to leave the group, because in a communication based approach the major means of cooperation is communication; once the communication is broken there will be no cooperation among the vehicles. Therefore, it is assumed that each vehicle has access to a "high performance communication channel" and a backup "low performance communication channel"; both communication channels possess constant communication bandwidth. In the fault-free condition the high performance communication channel is used which leads to *small communication delays,* typically smaller than the sampling time; once the high communication device becomes faulty the high performance communication channel can no longer be used, therefore in the faulty conditions the low performance communication channel is used by suitable backup communication devices installed on the vehicles; the low performance communication devices can maintain the

communication between faulty vehicle and neighbours; however, the communicated messages are subject to the *large communication delays*.

In general, it is assumed that the communication failure leads to large communication delays due to any reason such as failure in the communication devices and communication channels, inappropriate weather conditions, high communication loads, etc. Hence, an appropriate reconfigurable fault tolerant DRHC controller is required to address such large communication delays, then such controller is called delayed DRHC or fault tolerant DRHC in this thesis.

In this chapter, it is assumed that the failure applies to both receiver and transmitter which means any communication failure causes information blockage to and from faulty vehicle. This is a general case and the formulation for other cases (e.g., TX failure or RX failure) can be derived from that. Also, the formulation is consistent with both *direct* and *indirect* communication topologies.

4.1. Fault Tolerant DRHC

In the faulty condition, all the vehicles involving in the failure (the faulty vehicle and those which have a faulty neighbour) will construct the set of faulty neighbours, which is denoted by $\mathbb{V}_F^i$, the set of faulty neighbours of vehicle *i*. The vehicles which have a faulty neighbour assign the faulty neighbour to this set, and the faulty vehicle assigns all of its neighbours to this set, even though they are not faulty, because the faulty vehicle receives the information from healthy neighbours with a large delay.

Since in the faulty conditions the vehicles receive the faulty neighbour's trajectory with a delay, for the tail of the cost function there is no trajectory of neighbours to set the

formation. More precisely, if it is assumed that at time t_k the vehicle i receives the information from neighbour j with time-delay τ, then the trajectory of neighbour j is available to vehicle i, only for the interval $[t_k - \tau, t_k + T - \tau]$; meanwhile, according to the cost function of Eq. (2.32) vehicle i needs the trajectory of neighbour j during the entire segment $[t_k, t_k + T]$. Hence for the portion $[t_k + T - \tau, t_k + T]$ the trajectory of vehicle j is not available due to the delay (see Figure 4.1). If the time delay is small then this lack of information is not critical, but in presence of the large communication delays, the tail of cost function during $[t_k + T - \tau, t_k + T]$ becomes large and as shown in the example section it can lead to poor performance and even instability (see also [135, 136]). One remedy to this problem is proposed here by estimating the tail of the cost function via including extra decision variables in the cost function.

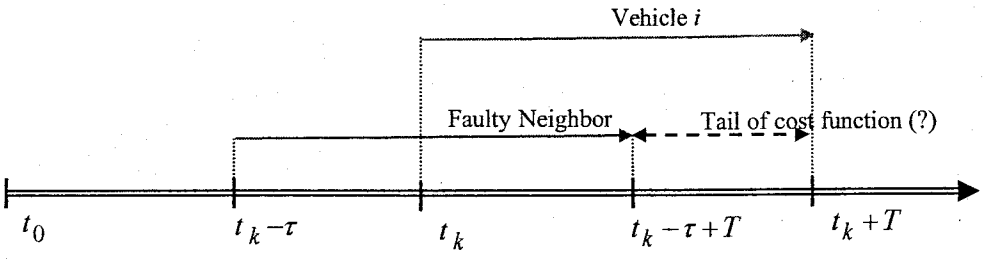

Figure 4.1: The tail of the cost function in the presence of communication delay

Although recently there has been a significant interest to study the communication delay in the networks of systems [137-144], a relatively small amount of existing work has investigated the communication delay in the DRHC framework. Few research works have proposed a systematic estimation process for the tail of the trajectories during $[t_k + T - \tau, t_k + T]$ to compensate for large communication delays. For instance, in [121, 122, and 123], which address quite the same problem, no prediction or estimation for the

trajectory of neighbouring vehicles is performed and it is assumed that the neighbouring vehicles remain at the last delayed states broadcasted by them. Such assumptions may yield poor performance for large communication delays since the constant state vector is not a good estimation of a trajectory of states in general. Similar issues are also investigated in [124] where for the hardware implementation of a robust decentralized model predictive control (DMPC) to wheeled robots, both computation and communication delays are considered. To account explicitly for the time delays a model of the neighbouring vehicle is used to estimate its state vector when required. The uncertainty arising from this estimation is taken into account by accommodating that into the effective disturbance force used for constraint tightening.

The main idea with the proposed fault tolerant DRHC approach in this thesis is that it estimates the trajectory of the neighbouring vehicles for the tail of the prediction horizon which would otherwise not be available due to the communication delay. The tail of the cost function is estimated by adding extra decision variables in the cost function.

Remark 4.1: The proposed fault tolerant algorithms proposed in this thesis can handle the multiple simultaneous failures (without further extensions); however this case happens rarely and single failure is the main subject of this thesis.

4.1.1. Delayed Cost Function

It is assumed that in the faulty conditions the communication delay is larger than the sampling time, i.e., $\tau > \delta$, see Figure 4.2. Further, assume $(d-1)\delta \le \tau \le d\delta$ where $d \in \mathbb{N}$; hence, in this thesis d resembles the (discrete) communication time delay. This is used instead of τ in most of the cases to provide synchronization between the communication delay and RHC sampling time (see Figure 4.3).

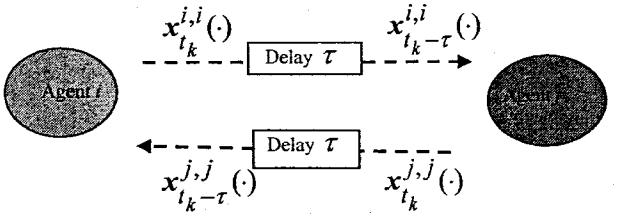

Figure 4.2: The inter-vehicle communication between two faulty vehicles

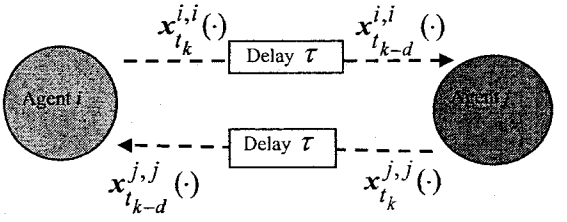

Figure 4.3: Synchronization of communication delay with RHC timing

As mentioned previously, due to computation delay at least a one-step delay should be considered as the computed trajectories are not available instantly even if an infinite communication bandwidth is used; then the following assumption should be made on d:

Assumption 4.1: always $d \geq 1$ or $d \in \mathbb{N}_1$. Also, delay d is time invariant.

As mentioned, in the faulty conditions the vehicles receive the delayed information from faulty neighbour and the non-delayed information from the fault-free neighbours; consequently, the *information set* is updated according to the following general form (compare with Eq. (2.31)):

$$\Gamma^i(t_k) = \left\{ x^i(t_k),\ x^{-i}_{t_{k-d}}(\cdot) \right\} \tag{4.1}$$

where $d=1$ for healthy neighbours and $d>1$ for faulty neighbours:

$$\begin{cases} d=1 & (i,j) \in \mathbb{E} \quad \& \quad j \notin \mathbb{V}_F^i \\ d>1 & (i,j) \in \mathbb{E} \quad \& \quad j \in \mathbb{V}_F^i \end{cases} \tag{4.2}$$

The *information set* $\Gamma^i(t_k)$ represents updated information available to the i^{th} vehicle at time t_k. It implies that at time t_k each vehicle i has access to its own delay-free information and delayed information from neighbours. The later includes the delay–free (small delay) information of healthy neighbours and the delayed information (large delay) of its faulty neighbours.

The delay dependent cost function for the faulty conditions (large communication delay) can now be presented as follows for the i^{th} vehicle in the team at time t_k:

$$J_D^i(\Gamma^i(t_k)) = \int_{t_k}^{t_k+T} \left(\left\| x_{t_k}^{i,i}(t) - x^{c,i} \right\|_Q^2 + \left\| u_{t_k}^{i,i}(t) \right\|_R^2 \right) dt + \left\| x_{t_k}^{i,i}(t_k+T) - x^{c,i} \right\|_P^2 + \quad (4.3a)$$

$$\sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_k}^{t_{k-d}+T} \left\| x_{t_k}^{i,i}(t) - x_{t_{k-d}}^{j,j}(t) - r^{i,j} \right\|_S^2 dt + \int_{t_{k-d}+T}^{t_k+T} \left\| x_{t_k}^{i,i}(t) - x_{t_k}^{j,i}(t) - r^{i,j} \right\|_S^2 dt \right] \quad (4.3b)$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k-d}+T}^{t_k+T} \left(\left\| x_{t_k}^{j,i}(t) - x^{c,j} \right\|_Q^2 + \left\| u_{t_k}^{j,i}(t) \right\|_R^2 \right) dt + \left\| x_{t_k}^{j,i}(t_k+T) - x^{c,j} \right\|_P^2 \right] \quad (4.3c)$$

where, the subscript "D" stands for Delay. The delay dependent decentralized cost function J_D^i of each vehicle i includes two main parts:

1) The first part is associated with the cost of the local vehicle i and therefore uses the delay-free information. It is used to compute the trajectory of the local vehicle i over the time interval $[t_k, t_k+T]$ (see Eq. (4.3a)).

2) The second part (Eq. (4.3b) and Eq. (4.3c)) is associated with the cost of the neighbours and then uses the information subject to delay (only one step delay for healthy and larger delay for faulty vehicles). It is used to compute the cooperation cost over the time interval $[t_k, t_k+T]$ (see Eq. (4.3b). For the tail of the cost function during

$[t_{k-d}+T, t_k+T]$ an estimation of the faulty neighbour's trajectory is required; hence, the cost of Eq. (4.3c) is added to incorporate some decision variables for this portion.

The main distinction between the proposed delay-dependent cost function (4.3) and the delay-free cost function (2.32) is the inclusion of two terms: the term (4.3c) and the second term in (4.3b), which together present a systematic way for estimating the tail of faulty neighbour's trajectory where the information is not available due to communication delay (see Figure 4.1). In other research [121-124], the tail of the faulty neighbour's trajectory is estimated by setting $\boldsymbol{u}=\mathbf{0}$ or extrapolating the previous control input. These ordinary estimations may suffice for small communication delays but it is shown by simulations that for large communication delays they yield poor performance and can lead to instability while the estimation by minimizing the proposed cost function (4.3) is more efficient as it imitates the calculation procedure of the faulty neighbour.

4.1.2. Fault Tolerant Delayed DRHC Problem

The delayed DRHC problem $\mathcal{P}_D^i(t_k)$ for the faulty conditions is defined below at time t_k for any i^{th} vehicle which involves in the fault (either it is faulty or its neighbour is faulty). The outputs of this decentralized optimization problem are:

1- The input trajectory of the local vehicle i over the prediction horizon, i.e., $\boldsymbol{u}_{t_k}^{i,i}(\cdot)$,

2- The state trajectory of the local vehicle i over the prediction horizon, i.e., $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$,

3- The estimated tail of faulty neighbour's trajectory which is not available due to communication delay, i.e., $\boldsymbol{x}_{t_k}^{-i}(t_{k-d}+T : t_k+T)$ (see Figure 4.1).

Problem 4.1: Fault Tolerant Delayed DRHC Problem $\mathscr{P}_D^i(t_k)$:

$$\underset{\{\boldsymbol{u}_{t_k}^{i,i}(\cdot),\ \boldsymbol{x}_{t_k}^{i,i}(\cdot),\ \boldsymbol{x}_{t_k}^{-i}(t_{k-d}+T:t_k+T)\}}{\text{Min}} \quad J_D^i(\Gamma^i(t_k)) \tag{4.4}$$

subject to:

- for $t \in [t_k, t_k+T]$:

$$\dot{\boldsymbol{x}}_{t_k}^{i,i}(t) = f(\boldsymbol{x}_{t_k}^{i,i}(t), \boldsymbol{u}_{t_k}^{i,i}(t)); \qquad \boldsymbol{x}_{t_k}^{i,i}(t_k) = \boldsymbol{x}^i(t_k) \tag{4.5a}$$

$$\boldsymbol{x}_{t_k}^{i,i}(t) \in \mathbb{X}^i,\ \boldsymbol{u}_{t_k}^{i,i}(t) \in \mathbb{U}^i \tag{4.5b}$$

- for $t \in [t_{k-d}+T, t_k+T]$ *and* $(i,j) \in \mathbb{E}$

$$\dot{\boldsymbol{x}}_{t_k}^{j,i}(t) = f(\boldsymbol{x}_{t_k}^{j,i}(t), \boldsymbol{u}_{t_k}^{j,i}(t));\ \ \boldsymbol{x}_{t_k}^{j,i}(t_{k-d}+T) = \boldsymbol{x}_{t_{k-d}}^{j,j}(t_{k-d}+T) \tag{4.5c}$$

$$\boldsymbol{x}_{t_k}^{j,i}(t) \in \mathbb{X}^j,\ \boldsymbol{u}_{t_k}^{j,i}(t) \in \mathbb{U}^j \tag{4.5d}$$

$$\begin{aligned} &\boldsymbol{x}_{t_k}^{i,i}(t_k+t) \in \mathbb{X}_f^i \\ &\boldsymbol{x}_{t_k}^{j,i}(t_k+t) \in \mathbb{X}_f^j;\quad (i,j) \in \mathbb{E} \end{aligned} \tag{4.5e}$$

In (4.4), J_D^i is calculated from (4.3). Constraints (4.5a) and (4.5b) are the same as (2.34a) and (2.34b) for the delay-free DRHC problem $\mathscr{P}^i(t_k)$ and correspond to the trajectory for calculating the cost (4.3a). Constraints (4.5c) and (4.5d) correspond to the cost function (4.3c). Constraint (4.5e) is the terminal constraint and the same as (2.34c) for $\mathscr{P}^i(t_k)$.

The problem $\mathscr{P}_D^i(t_k)$ can be used in the delay-free conditions too; hence it can be used always by all the vehicles; thus, in $\mathscr{P}_D^i(t_k)$ it is perfectly valid to choose $i \in \mathbb{V}$.

4.1.3. Fault Tolerant Delayed DRHC Algorithm

The following algorithm is presented for the on-line implementation of the delayed DRHC problem $\mathcal{P}_D^i(t_k)$. The algorithm is formulated for the i^{th} vehicle; in fact, all vehicles run this algorithm during the mission simultaneously:

Algorithm 4.1: Delayed DRHC (online)

1: $k \leftarrow 0$, and GOTO step 3.

2: Receive $\boldsymbol{x}_{t_{k-d}}^{j,j}(\cdot)$ from leaders where $(i,j) \in \mathbb{E}$.

3: Measure $x^i(t_k)$ and update the *information vector* of Eq. (4.1).

4: Solve $\mathcal{P}_D^i(t_k)$.

5: Send the state trajectory $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$ to followers where $(j,i) \in \mathbb{X}$.

6: Execute the control action for the local vehicle i during $[t_k, t_{k+1}]$.

7: $k \leftarrow k+1$. Goto step 2.

This algorithm is a modified version of Algorithm 2.2 and handles the large communication delays for faulty conditions; by choosing d =1 this algorithm reduces to Algorithm 2.2.

4.2. Simulation Results

The application of the proposed delayed DRHC algorithm to cases where the subsystems are described by either a linear or nonlinear dynamics is investigated through simulations.

4.2.1. Example 1: Vehicles with LTI Dynamics

In this section, the proposed *Algorithm 4.1* is tested on the formation problem of a fleet of unmanned vehicles with linear dynamics. It is assumed that the communication graph is *indirect* and hence a *leaderless* formation can be handled. The *indirect* communication graph topology is set as follows:

$$\begin{aligned} \mathbb{V} &= \{1,2,3\} \\ \mathbb{E} &= \{(1,2),(1,3),(2,3)\} \end{aligned} \tag{4.6}$$

The dynamics of each subsystem is described by the following 2DOF dynamics:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_2 + u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -x_4 + u_2 \end{aligned} \tag{4.7}$$

where x_1 and x_2 denote the components of position vector in x-y coordinate and x_3 and x_4 are their corresponding velocity components. The input vector is given by $\boldsymbol{u} = [u_1, u_2]$.

4.2.1.1. Investigating the Effect of Tail Cost

In the first simulation example, it is desired to examine the effect of the tail cost added to the cost function. The simulation was run for two cases:

1) Using the cost function *without the tail cost*. In this case the control input is set to $\boldsymbol{u}$=**0** for the tail of the cost function (4.3c). The extra decision variables for the tail cost estimation are not included in the optimization *Problem 4.1*.

2) Using the cost function *with the tail cost*. In this case the tail of the cost function (4.3c) is estimated using the extra decision variables in the optimization *Problem 4.1*.

The matrix penalties in the cost function are chosen as follows:

$$Q=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \quad R=\begin{bmatrix}1&0\\0&1\end{bmatrix} \quad P=\begin{bmatrix}0.72&0&0&0\\0&0.5&0&0\\0&0&0.72&0\\0&0&0&0.5\end{bmatrix} \quad S=\begin{bmatrix}2&0&0&0\\0&1&0&0\\0&0&2&0\\0&0&0&1\end{bmatrix} \tag{4.8}$$

Final penalty matrix P is calculated from the Lyapunov equation [10]. The optimization horizon and the execution horizon are given by $T = 3.0$ sec and $\delta = 0.1$ sec, respectively. In all cases no disturbances, sensor noise, or model uncertainty are considered in the simulations in order to focus on the effect of the communication delay.

A triangular leaderless formation of three vehicles is first considered. The vehicles start around the origin and are asked to follow a moving target initiating at point (25, 25) and ending at point (45, 0) in xy-plane while moving in triangle formation. The moving target is used to ensure that the optimization problem updates the trajectories at each time step so that the delayed trajectories become different from the updated ones. Figure 4.4 shows the snapshot of the formation for the two mentioned cases. The red-dotted triangles are the formation for the case 1 (without tail cost) and the green solid triangles, are the formation for case 2 (with the tail cost). It can be seen easily that the formation corresponding to the case 2 is more analogous to an equilateral triangle when comparing with formation corresponding to case 1. To measure the deviation from the desired equilateral triangle formation, the decentralized formation error is calculated as the performance index, as follows:

$$E^i(t)=\sum_{j|(i,j)\in\mathbb{E}}\left\|\boldsymbol{x}^i(t)-\boldsymbol{x}^j(t)-\boldsymbol{r}^{i,j}(t)\right\|_S^2 \tag{4.9}$$

This error is depicted in Figure 4.5 (right); as seen the error corresponding to the case 2 is much less than the case 1 and this implies the advantage of using the tail cost for improving the performance.

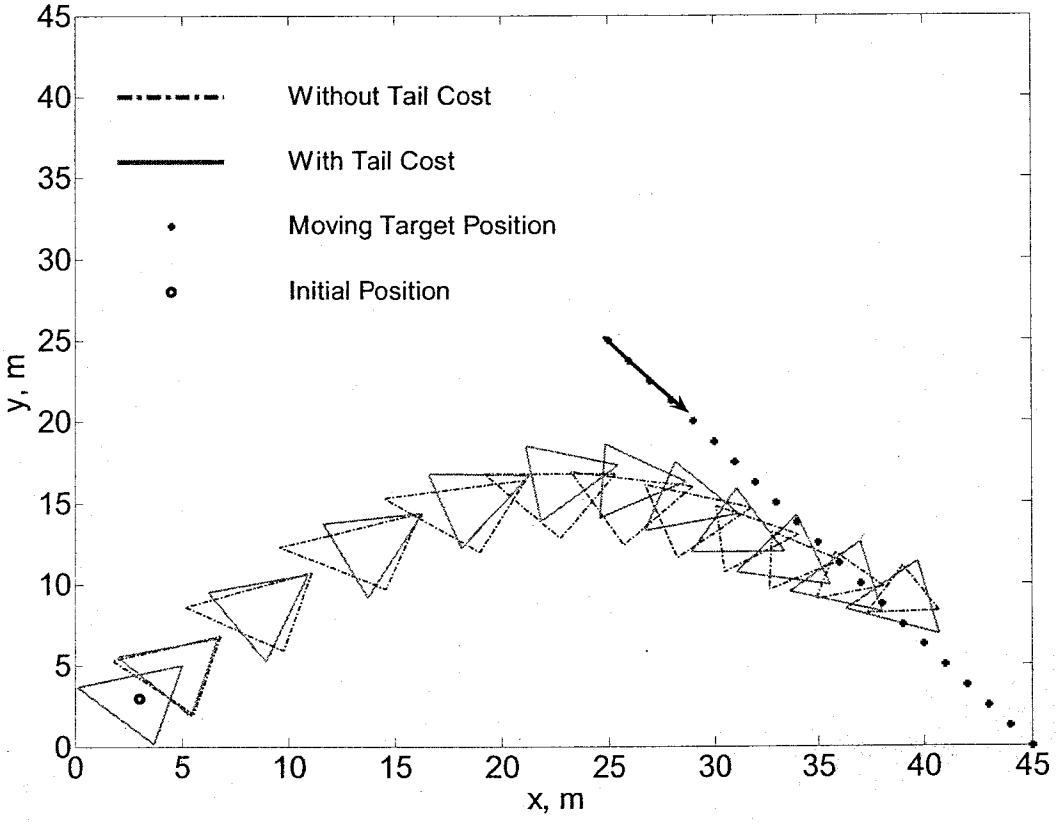

Figure 4.4: Snapshot of trajectory for three vehicles in triangular leaderless formations

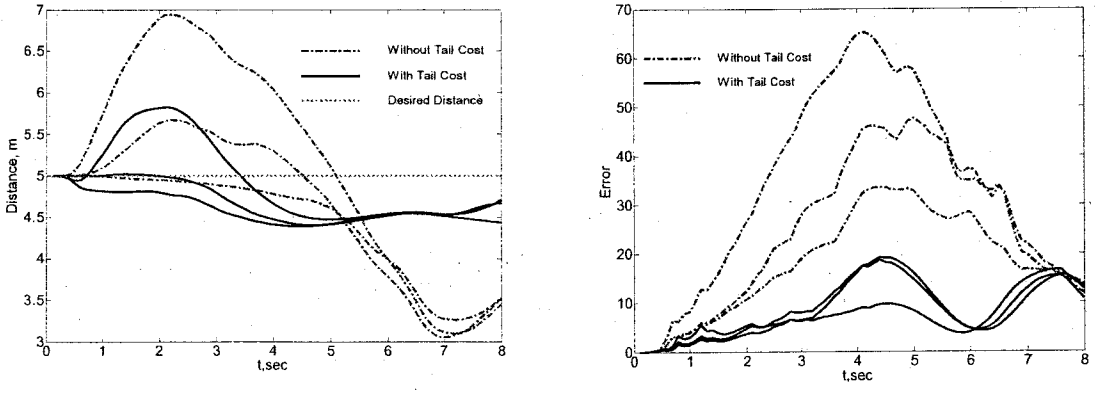

Figure 4.5: Distances between each pair of vehicles (left) and formation error (right)

The simulation was repeated for several cases with different communication delays and the results are gathered in Figure 4.6 which illustrates the average and maximum of the formation error (4.9) with each point representing a single simulation. It can be seen from Figure 4.6 that using the tail of the cost function yields a smaller error and in some cases it can reduce the error by 150%.

It can also enhance the stability of the formation; for this particular example, it is seen that if the communication delay is increased to around d=30 time steps (or $\tau = 3.0$ sec), the formation becomes unstable when using the cost function without the tail cost. However, it is still stable with the proposed cost function including the tail cost. This result is consistent with that of [145, 147] where a final cost is added to the cost function for formation stability, although they didn't consider communication delays. The overall trend of the graphs in Figure 4.6 shows that the error increases with delay. The small downward fluctuations are probably due to the time delay related nonlinearities and imperfect numerical optimization.

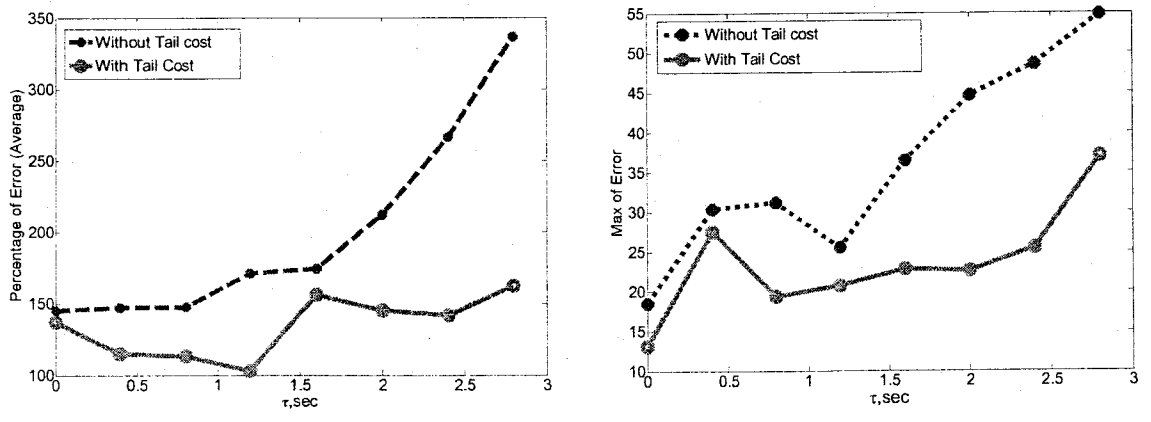

Figure 4.6: Percentage of average (left) and maximum (right) error versus communication delay for a triangle leaderless formulation of three vehicles.

4.2.2. Example 2: Application to Formation Control of Miniature Hovercrafts

In this section, the new approach is tested on the formation problem of a fleet of miniature hovercrafts with a nonlinear dynamics. The *leader-follower* triangular formation of 6 vehicles is considered. It is assumed that the communication graph is *direct* which allows performing a *leader-follower* formation; the *direct* communication graph topology is set as follows:

$$\begin{aligned} \mathbb{V} &= \{1,2,3,4,5,6\} \\ \mathbb{E} &= \{(1,2),(2,1),(3,1),(4,2),(5,4),(6,3)\} \end{aligned} \tag{4.10}$$

The matrix penalties in the cost function are chosen as $Q=I$, $R=I$, $P=2.2I$ and $S=20I$, where I is the identity matrix. Final penalty matrix P is approximated from the approach proposed in [10]. The optimization horizon and the execution horizon (sampling time) are given as $T = 2.0$ sec and $\delta = 0.2$ sec.

4.2.2.1. Hovercraft Dynamics and Modeling

The hovercraft configuration is illustrated in Figure 4.7. The three degree of freedom (DOF) motion of each hovercraft is controlled using two DC motor propeller actuators that are computer controlled through wireless radio communication links. The position of the hovercraft is measured using a 4 camera overhead vision system with a sampling rate of 26Hz. The velocity and acceleration are estimated from the position values.

The equations for each hovercraft in the body frame (X_B, Y_B) are presented by Aguiar, Cremean, & Hespanha [148] as follows (see Figure 4.7):

$$\begin{aligned} M(\dot{u} - vr) &= c_1 u + F_r + F_l \\ M(\dot{v} + ur) &= c_2 v \\ J\dot{r} &= c_3 r + \frac{l}{2}(F_r - F_l) \end{aligned} \tag{4.11}$$

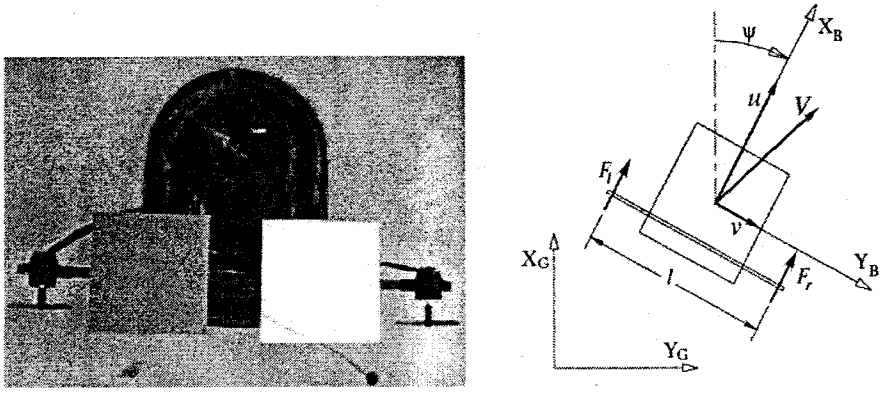

Figure 4.7: Miniature hovercraft (left) and schematic model (right).

where u and v represent the components of forward and side velocity in X_B and Y_B directions, respectively. The variable r represents the yaw rate. The mass is represented by M and the moment of inertia by J. The parameters c_1 and c_2 represent the coefficients of viscous friction in the X_B and the Y_B directions, respectively. The rotational coefficient of viscous friction is represented by c_3. Also, F_r and F_l are the inputs to the system and saturated at: $-10 \le F_r \le 10$ and $-10 \le F_l \le 10$ (N).

In order to minimize the number of parameters to be identified in the model, the equations are arranged as follows:

$$\begin{aligned}
\dot{u} &= c_1 u + vr + a_1 F_r + a_2 F_l \\
\dot{v} &= c_2 v - ur \\
\dot{r} &= c_3 r + a_3 F_r + a_4 F_l \\
\dot{\psi} &= r \\
\dot{x} &= u\cos\psi - v\sin\psi \\
\dot{y} &= u\sin\psi + v\cos\psi
\end{aligned} \qquad (4.12)$$

where the last three equations give the relationship between *(u, v, r)* and the coordinates in the inertial reference frame (X_G, Y_G) (Figure 4.7). Also, ψ denotes the yaw angle, x and y are the components of position vector in frame (X_G, Y_G) .

The parameter values for the hovercraft model were identified experimentally using a vision-based measurement setup and by performing a least-squares curve fit to experimental data; the results are listed in Table 1.

Table 4.1: Parameters of hovercraft

a_1	a_2	a_3	a_4	c_1	c_2	c_3
1.40	*0.45*	*14.9*	*-2.47*	*-0.25*	*-0.34*	*-9.11*

These values are used for modeling of all the team members.

4.2.2.2. The effect of tail cost

In the first simulation example, it is desired to examine the effect of the tail cost added to the cost function. The simulation was run for two cases:

1) Using the cost function *without the tail cost*. In this case the control input is set to $\boldsymbol{u}=\mathbf{0}$ for the tail of the cost function (4.3c). The extra decision variables for tail cost estimation are not included in the optimization Problem *4.1*.
2) Using the cost function *with the tail cost*. In this case the tail of the cost function (4.3c) is estimated using the extra decision variables in the optimization Problem *4.1*.

Figure 4.8 shows the trajectory and snapshot of the trajectories for the case where no fault tolerant algorithm is used.

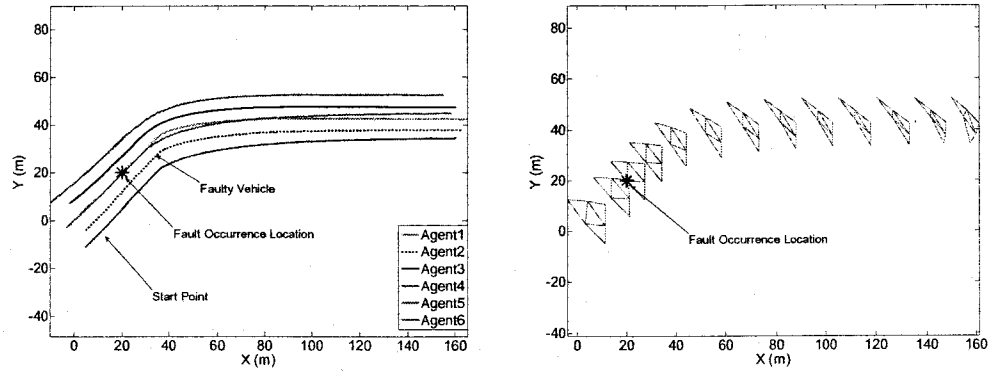

Figure 4.8: Trajectory (left) and snapshot (right) of six vehicles in triangular leader-follower formation when no tolerant algorithm is used.

This simulation was repeated for many cases with different communication delays and the results are gathered in Figure 4.9 in terms of average and maximum of formation error of Eq. (4.9). As seen using the tail of the cost function yields a smaller error and in some cases it can reduce the error by 255%. For this particular example if the communication delay is increased to around d=10 time step (or $\tau = d \cdot \delta = 2.0$ sec) the formation becomes unstable when using the cost function without tail cost; however, it is still stable with the proposed cost function with tail cost. This result is consistent with that of [145] and [147] where a final cost is added to the cost function for the formation stability, although they didn't consider the communication delay.

It is also seen in simulations that in faulty conditions although adding the final cost can lead to more precise estimation and stable formation, the vehicles may still get too close to each other and collide. For example, in Figure 4.10 the minimum distance between each pair of neighbouring vehicles for a set of simulations is depicted versus communication delay. The desired distance between each pair of neighbouring vehicles is

7.07 m. As seen even for the case 2 where the tail cost estimation is used the vehicles may get too close to each other.

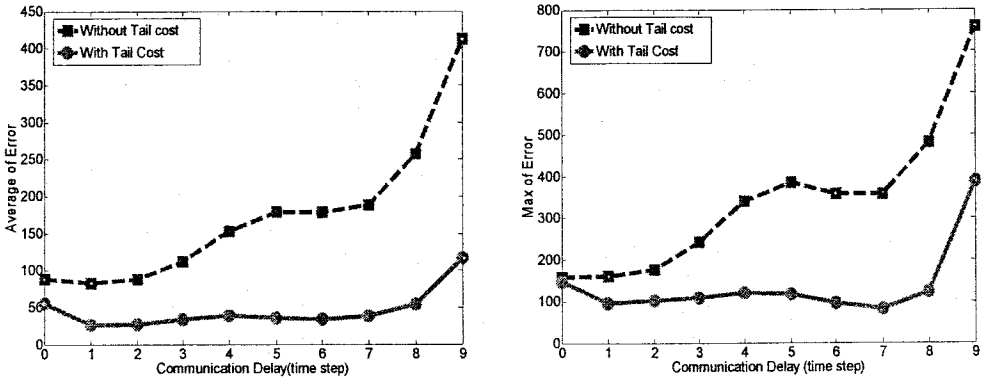

Figure 4.9: Average (left) and maximum error (right) versus communication delay.

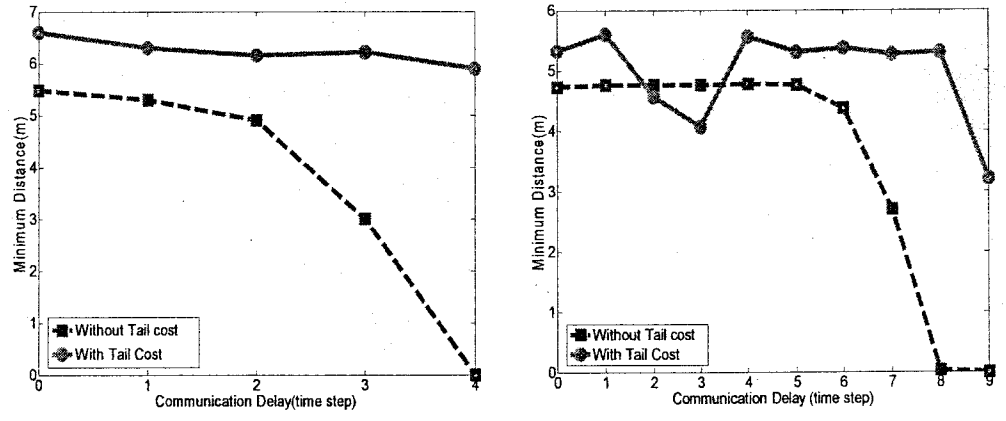

Figure 4.10: Minimum distance between vehicles during a communication failure. Left: T=1sec, Right: T=2 sec.

4.3. Summary

In this chapter a reconfigurable fault tolerant delay dependent DRHC architecture is proposed. The key feature of the proposed fault tolerant algorithm is that the tail of the neighbour's trajectory, where the trajectories are not available due to the large

communication delays, is estimated by adding extra decision variables in the DRHC problem. Simulations illustrate that the proposed delayed DRHC can reconfigure effectively in the presence of the communication failures leading to large communication delays. It is also demonstrated that using a prediction for the tail of the trajectories can lead to better overall performance and stability.

Chapter 5. Tube DRHC Approach for Collision Avoidance

The collision avoidance constraint can not be included in the optimization problem of DRHC because of its non-convex nature as it can lead to multi-optima. Hence, the collision avoidance can not be guaranteed with the DRHC framework. Instead, in practice the desired distance in formation problems should be chosen large enough to ensure the collision avoidance. However, in the faulty conditions due to the large communication delays the lack of updated information on the trajectory of neighbours can lead to collisions if the desired relative distances do not account for the delays. This is the reason why in some simulations of Chapter 5 in faulty conditions, although adding the tail cost can lead to more precise estimation and a stable formation, the vehicles may still get too close to each other and collide, see for example Figure 4.5 (left) and Figure 4.10.

Furthermore, for non-formation problems such as air traffic control [170-172], road traffic control [173, 174], mobile robots [175, 176] and cooperative UAVs [177, 162], the classical DRHC is complained about its weakness in handling the non-convex constraints arising from collision avoidance problem [177, 162].

In this chapter to address the collision avoidance problem, the Tube-DRHC approach is proposed. Roughly speaking, in this approach, the neighbours of the faulty vehicle consider a tube shape trajectory set around the trajectory of the faulty vehicle instead of a

line shape trajectory. This will put the faulty vehicle in a safe zone (tube) where the neighbours of the faulty vehicles are not allowed to enter.

The radius of the tube is a function of the communication time-delay d, and manoeuvrability ($\mathbb{U}^i$). Then, if a constraint is imposed on the manoeuvrability of the faulty vehicle, then the reachable set (tube) of the faulty vehicle can be computed by neighbouring vehicles using the available, albeit delayed, information from faulty vehicle. The manoeuvrability of faulty vehicle is restricted by imposing an input constraint in its optimization problem such that at any time instant the computed inputs do not deviate too far from the previous one.

The concept of the tube MPC (or tube RHC) in existing works [126, 157] is normally used to calculate a robust bound on the states due to model uncertainty, but in this thesis the approach is used to calculate bounds on the uncertainties arising from the large communication delays.

In this chapter, the collision avoidance is addressed for both formation problems and non-formation scenarios such as air traffic control problems.

5.1. Reachable Set and Tube Formulation

5.1.1. Reachable Set Formulation

The reachable set of vehicle i at time t is formulated as:

$$\Lambda^i(t, \boldsymbol{x}_0^i, \mathbb{U}^i) = \left\{ \boldsymbol{x}^i(t) \mid \dot{\boldsymbol{x}}^i(s) = f(\boldsymbol{x}^i(s), \boldsymbol{u}^i(s)),\ \boldsymbol{x}^i(t_0) = \boldsymbol{x}_0^i, \boldsymbol{u}^i \in \mathbb{U}^i, s \in [t_0, t] \right\} \quad (5.1)$$

Figure 5.1 shows a graphical sketch of the reachable set of 3 neighbouring vehicles at time t:

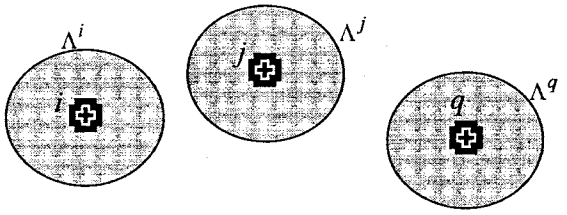

Figure 5.1: The reachable set for three vehicles at some specific time.

It is evident that if the reachable sets of neighbouring vehicles have no intersection ($\Lambda^i \cap \Lambda^j \cap \Lambda^q = \varnothing$) then no collision will happen at time t.

In general, in this thesis it is assumed that the reachable sets are convex; it is not a restrictive assumption as even for most of the nonlinear systems a convex reachable set can be computed, and non-convex reachable sets can be bounded by a convex set.

5.1.2. Tube Formulation

In order to take advantage of the predictive nature of RHC and predict the possible collisions during the prediction horizon, all reachable sets should be computed over the prediction horizon and connected together which results in the *tube* concept; in fact, *tube* is formed by connecting the reachable sets over the prediction horizon. Figure 5.2 shows how by connecting the reachable sets a *tube* is formed. From Figure 5.2 it is clear that if the tube of neighbouring vehicles have intersection (i.e., $\mathbb{H}^i \cap \mathbb{H}^j \cap \mathbb{H}^q \neq \varnothing$) then a collision is possible. In general sufficient condition for collision avoidance is that: $\mathbb{H}^1 \cap \mathbb{H}^2 \cap \mathbb{H}^3 \cap \ldots\ldots \cap \mathbb{H}^{N_v} = \varnothing$. This *tube* analysis allows predicting the possible collisions and hence, changing the plan to avoid the collisions.

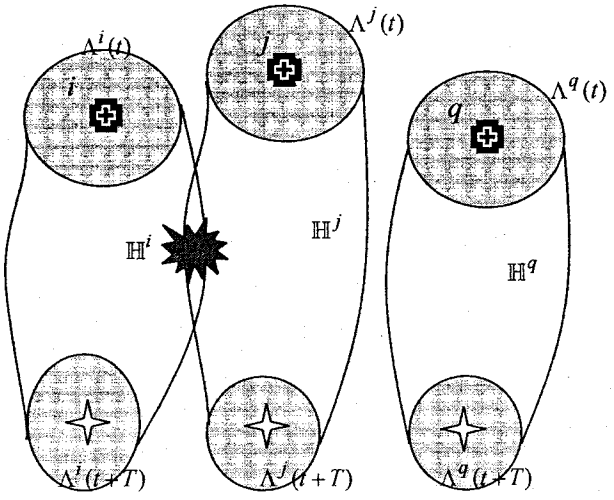

Figure 5.2: The reachable sets of three vehicles over the prediction horizon. The tube is formed by connecting the reachable set over the prediction horizon.

Definition 5.1: the absolute value of vector is shown by $|\cdot|$ and is defined as follows:

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \end{bmatrix} \Rightarrow |v| = \begin{bmatrix} |v_1| \\ |v_2| \\ \cdots \\ |v_n| \end{bmatrix} \tag{5.2}$$

The state vector of each vehicle contains two types of state variables: 1) the states which involve in physical collision such as position components and are denoted by vector ξ, and 2) The rest of states such as velocity, and are denoted by vector υ; hence: $x = [\xi, \upsilon]'$. Tube, in this thesis, is referred to as an extraction of the reachable set which includes only the position states ξ. Figure 5.3 shows the tube $\mathbb{H}$ around a nominal trajectory $\xi(\cdot, \boldsymbol{u}_0)$. Then, the tube $\mathbb{H}$ is formulated as follows (see also [152]):

$$\mathbb{H}(t_k) = \left\{ (t, \xi) \in [t_k, t_k + T] \times \mathbb{R}^p \mid \left| \xi(t,u) - \xi(t, u_{t_k}) \right| < \alpha(t) \right\} \tag{5.3}$$

where $\alpha(t) \in \mathbb{R}^p$ is the radius of tube at time t; also, p is the dimension of ξ (for a 2-D motion p=2). Some methods are presented for calculating this tube for vehicles obeying linear or nonlinear dynamics.

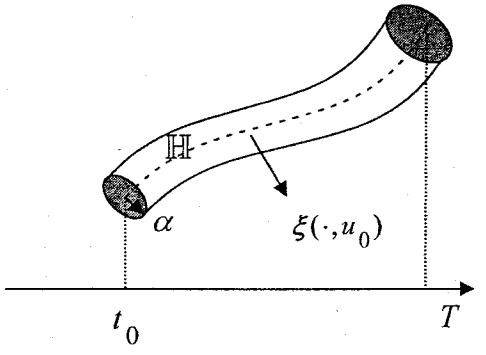

Figure 5.3: A tube around a nominal trajectory.

5.2. Tube Calculation Algorithms

5.2.1. Tube Calculation for LTI Systems

Different methods may be used to approximate the reachable set of dynamical systems; in this section an LMI based approach is used.

5.2.1.1. LMI Approximation of Reachable Set

Consider the following nominal dynamics of each vehicle:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \;\; ; \;\; \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \qquad (5.4)$$

Then the following lemma presents a method for approximation of the reachable set of LTI systems:

Lemma 5.1: for LTI system (5.4) if:

$$\mathbb{U}^i(t)=\left\{\boldsymbol{u}\,|\,\int_{t_0}^{t}\boldsymbol{u}'\boldsymbol{u}\,dt\le\beta\right\} \tag{5.5}$$

Then the reachable set $\Lambda^i(t,\boldsymbol{x}_0,\mathbb{U}^i)$ is bounded by the ellipsoid Υ centered at $\boldsymbol{x}_0$:

$$\Upsilon=\left\{\boldsymbol{x}\,|\,(\boldsymbol{x}-\boldsymbol{x}_0)'M(\boldsymbol{x}-\boldsymbol{x}_0)\le\beta\right\} \tag{5.6}$$

where M is symmetric and diagonal solution of the following LMI:

$$M>0 \qquad \begin{bmatrix}A'M+MA & MB\\ B'M & -I\end{bmatrix}\le 0 \tag{5.7}$$

Also if (5.4) is controllable then $M=W^{-1}$ where W is the *controllability Gramian*.

Proof: see [169], page 78.

5.2.1.2. *Tube around the Delayed Trajectory*

The following theorem presents a method for calculating the tube around the delayed trajectories from neighbours and *Lemma 5.1*:

Theorem 5.1: Assume at time t_k the d step delayed trajectory of neighbour j, i.e., $\boldsymbol{x}^{j,j}_{t_{k-d}}(\cdot)$ is available to vehicle i. If:

$$\int_{t_{k-d}}^{t_k}\left[\boldsymbol{u}^{j,j}_{t_k}(t)-\boldsymbol{u}^{j,j}_{t_{k-d}}(t)\right]'\left[\boldsymbol{u}^{j,j}_{t_k}(t)-\boldsymbol{u}^{j,j}_{t_{k-d}}(t)\right]dt\le\beta^j(d)\ ,$$

then the trajectory of vehicle j at time t_k belongs to the tube around the delayed trajectory of neighbour j calculated by vehicle i as:

$$\mathbb{H}^{j,i}_{t_k}=\left\{(t,\xi)\in[t_k,t_{k-d}+T]\times\mathbb{R}^p\ \middle|\ \left|\xi(t)-\xi^{j,j}_{t_{k-d}}(t)\right|<\alpha^{j,i}(t)\right\} \tag{5.8}$$

where:

$$\alpha^{j,i}(t)=\left[\sqrt{\frac{\beta^{j}(d)}{m_{11}}},\sqrt{\frac{\beta^{j}(d)}{m_{22}}},.....,\sqrt{\frac{\beta^{j}(d)}{m_{pp}}}\right]' \qquad (5.9)$$

and where $M=diag(m_{11},m_{22},.....m_{nn})$, $m_{11},m_{22},.....m_{nn}\in\mathbb{R}$ is a solution of the LMI (5.7).

Proof: assume at time t_{k-d} vehicle j uses the input trajectory $\boldsymbol{u}^{j,j}_{t_{k-d}}(\cdot)$ which yields the state trajectory $\boldsymbol{x}^{j,j}_{t_{k-d}}(\cdot)$; then:

$$\boldsymbol{x}^{j,j}_{t_{k-d}}(t)=\varphi(t,t_{k-d})\boldsymbol{x}^{j}(t_{k-d})+\int_{t_{k-d}}^{t}\varphi(t,s)B\boldsymbol{u}^{j,j}_{t_{k-d}}(s)ds \quad ; \quad t\in[t_{k-d},t_{k-d}+T]$$

where φ is the state transition matrix. But if vehicle j uses the input trajectory $\boldsymbol{u}^{j,j}_{t_{k-d}}(\cdot)+\Delta\boldsymbol{u}$ -which is assumed by neighbours of j- then the trajectory is different and calculated as follows:

$$\hat{\boldsymbol{x}}^{j,j}_{t_{k-d}}(t)=\varphi(t,t_{k-d})\boldsymbol{x}^{j}(t_{k-d})+\int_{t_{k-d}}^{t}\varphi(t,s)B[\boldsymbol{u}^{j,j}_{t_{k-d}}(s)+\Delta\boldsymbol{u}]ds \qquad (5.10)$$

Then the difference between these two trajectories is:

$$\Delta\boldsymbol{x}^{j,j}_{t_{k-d}}(t)=\hat{\boldsymbol{x}}^{j,j}_{t_{k-d}}(t)-\boldsymbol{x}^{j,j}_{t_{k-d}}(t)=\int_{t_{k-d}}^{t}\varphi(t,s)\,B\,\Delta\boldsymbol{u}\,ds \qquad (5.11)$$

Considering (5.11) $\Delta\boldsymbol{x}^{j,j}_{t_{k-d}}(t)$ is the solution of the following LTI:

$$\Delta\dot{\boldsymbol{x}}^{j}=A\Delta\boldsymbol{x}^{j}+B\Delta\boldsymbol{u}^{j} \;\; ; \;\; \Delta\boldsymbol{x}^{j}(t_{k-d})=0 \qquad (5.12)$$

Using *Lemma 5.1* and considering:

$$\int_{t_{k-d}}^{t_k} \left[\Delta \boldsymbol{u}^j(t)\right]' \left[\Delta \boldsymbol{u}^j(t)\right] dt \le \beta^j(d)$$

then the reachable set of LTI system (5.12) is bounded by the ellipsoid:

$$\Upsilon^j(t) = \left\{ \boldsymbol{x} \,|\, [\Delta \boldsymbol{x}^j_{t_{k-d}}(t)]' M [\Delta \boldsymbol{x}^j_{t_{k-d}}(t)] \le \beta^j(d) \right\} \qquad t \in [t_{k-d}, t_{k-d}+T] \tag{5.13}$$

Substituting (5.11) into (5.13) yields:

$$\Upsilon^j(t) = \{ \boldsymbol{x} \,|\, [\hat{\boldsymbol{x}}^{j,j}_{t_{k-d}}(t) - \boldsymbol{x}^{j,j}_{t_{k-d}}(t)]' M [\hat{\boldsymbol{x}}^{j,j}_{t_{k-d}}(t) - \boldsymbol{x}^{j,j}_{t_{k-d}}(t)] \le \beta^j(d) \} \qquad t \in [t_{k-d}, t_{k-d}+T] \tag{5.14}$$

Using the ellipsoid formula, the radius of the ellipsoid (5.14) for vehicle $j \in \mathbb{V}$ in each direction is calculated as follows:

$$\boldsymbol{r}_{\Upsilon^j} = \left[\sqrt{\frac{\beta^j}{m_{11}}}, \sqrt{\frac{\beta^j}{m_{22}}}, \ldots\ldots, \sqrt{\frac{\beta^j}{m_{nn}}} \right]' \tag{5.15}$$

Since the ellipsoid (5.14) over $t \in [t_{k-d}, t_{k-d}+T]$ is equivalent to the tube (5.8); then, abstracting the tube $\mathbb{H}^{j,i}$ from the ellipsoid Υ^j is straightforward. For each component of ξ (p^{th} component) one should find the corresponding component in $\boldsymbol{r}_{\Upsilon^j}$.

Lemma 5.2: Assume in the DRHC problem the control input varies as follows:

$$\left| \boldsymbol{u}^{j,j}_{t_k}(t) - \boldsymbol{u}^{j,j}_{t_{k-1}}(t) \right| \le \mu^j \quad t \in [t_k, t_{k-1}+T] \quad ; \quad k \in \mathbb{N} \tag{5.16}$$

where $\mu^j \ge 0$ is the *manoeuvrability parameter* of vehicle j. Also, the communicated trajectories between vehicles i and j are subject to d step delay. Then β^j for $j \in \mathbb{V}$ is calculated as follows:

$$\beta^j(d) = d^3 \delta[\mu^j]'[\mu^j] \tag{5.17}$$

Proof:

$$\Delta \boldsymbol{u} = [\boldsymbol{u}_{t_k}^{j,j} - \boldsymbol{u}_{t_{k-d}}^{j,j}] = [(\boldsymbol{u}_{t_k}^{j,j} - \boldsymbol{u}_{t_{k-1}}^{j,j}) + (\boldsymbol{u}_{t_{k-1}}^{j,j} - \boldsymbol{u}_{t_{k-2}}^{j,j}) + ... + (\boldsymbol{u}_{t_{k-d+1}}^{j,j} - \boldsymbol{u}_{t_{k-d}}^{j,j})]$$
$$\leq [\mu^j + \mu^j + ... + \mu^j] = d.\mu^j$$

Then:

$$\int_{t_{k-d}}^{t_k} \Delta \boldsymbol{u}' \Delta \boldsymbol{u}\, dt = \int_{t_{k-d}}^{t_k} [d.\mu^j]'[d.\mu^j] dt$$
$$\leq d^2 \mu^{j'} \mu^j \int_{t_{k-d}}^{t_k} dt = d^2 \mu^{j'} \mu^j (t_{k-d} - t_k) = d^3 \delta [\mu^j]'[\mu^j]$$
$$\Rightarrow \beta^j(d) = d^3 \delta [\mu^j]'[\mu^j]$$

The procedure presented in this section for tube calculation is summarized in the following algorithm. Assuming the vehicle *i* calculates the tube around the trajectory of neighbour *j*:

Algorithm 5.1: Tube Calculator for LTI Subsystems:

Given the manoeuvrability $\mu^j \geq 0$ of neighbour *j*, delay *d*, and matrix M from LMI (5.7):

1: Calculate $\beta^j(d) = d^3 \delta [\mu^j]'[\mu^j]$.

2: Calculate the tube radius $\alpha^j = \left[\sqrt{\frac{\beta^j}{m_{11}}}, \sqrt{\frac{\beta^j}{m_{22}}}, ..., \sqrt{\frac{\beta^j}{m_{nn}}} \right]'$ around the delayed trajectory of neighbour *j*.

Calculation of tube radius does not impose any on-line computation time as it can be computed off-line. Because the only parameter which may not be known prior to mission is *d* which can be decomposed from formula of $\beta^j(d) = d^3 \delta [\mu^j]'[\mu^j]$ and multiplied by the computed bound when determined online.

5.2.2. Tube-DRHC for Nonlinear Systems

For a general class of dynamics the analytical bound on the states can be found, but the available analytical bounds are usually too conservative and not in the desired format for tube analysis (often a bound on the norm of states is available). Hence, a numerical bound should be calculated instead, for any specific form of dynamics. The following problem represents a method for calculating the tube for a general class of nonlinear systems:

Problem 5.1: Consider the dynamics of each vehicle is described by (2.1), with nominal trajectory $\boldsymbol{x}(\cdot, \boldsymbol{u}_0)$. Then calculate the tube $\mathbb{H}$ on the interval $[t_0, T]$ so that:

$$|\boldsymbol{u}(t) - \boldsymbol{u}_0(t)| \leq \beta \quad ; t \in [t_0, T] \tag{5.18}$$

Solution: The nominal solution of differential Eq. (2.1) on the interval $[t_0, T]$ is calculated as:

$$\boldsymbol{x}(t, u_0) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} f(\boldsymbol{x}(s, \boldsymbol{u}_0), \boldsymbol{u}_0(s))ds \tag{5.19}$$

And any other trajectory $\boldsymbol{y}(\cdot, \boldsymbol{u})$ of the system on the interval $[t_0, T]$ is calculated as follows:

$$\boldsymbol{y}(t, u) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} f(\boldsymbol{y}(s, u), \boldsymbol{u}(s))ds \tag{5.20}$$

Subtracting (5.19) from (5.20) yields:

$$\boldsymbol{y}(t, u) - \boldsymbol{x}(t, u_0) = \int_{t_0}^{t} \left[f(\boldsymbol{y}(s, \boldsymbol{u}), \boldsymbol{u}(s)) - f(\boldsymbol{x}(s, \boldsymbol{u}_0), \boldsymbol{u}_0(s)) \right] ds \tag{5.21}$$

Then, using (5.21) in the tube formula (5.3) yields:

$$\mathbb{H} = \left\{(t,y) \in [t_0,T] \times \mathbb{R}^n \mid \left|y(t,u) - x(t,u_0)\right| < \alpha(t)\right\} \quad \Rightarrow$$

$$\mathbb{H} = \left\{(t,y) \in [t_0,T] \times \mathbb{R}^n \mid \left|\int_{t_0}^{t} f(y(s,u),u(s))ds - \int_{t_0}^{t} f(x(s,u_0),u_0(s))ds\right| < \alpha(t) \;\&\; \left|u - u_0\right| < \beta\right\} \Rightarrow$$

$$\mathbb{H} = \left\{(t,y) \in [t_0,T] \times \mathbb{R}^n \mid \left|\int_{t_0}^{t} \left[f(y(s,u),u(s)) - f(x(s,u_0),u_0(s))\right]ds\right| < \alpha(t) \;\&\; \left|u - u_0\right| < \beta\right\}$$

Finding $\alpha(t)$ determines the tube $\mathbb{H}$. $\alpha(t)$ is the upper bound on $\left|\int_{t_0}^{t} \left[f(y(s,u),u(s)) - f(x(s,u_0),u_0(s))\right]ds\right|$ and then is calculated by solving the following maximization problem:

$$\alpha(t) = \underset{\{\alpha(t)\}}{\text{Max}} \left[\left|\int_{t_0}^{t} \left[f(y(s,u),u(s)) - f(x(s,u_0),u_0(s))\right]ds\right| \right] \tag{5.22}$$

subject to:

$$\begin{aligned} &\dot{y} = f(y,u); \qquad y(0,u) = x(0,u_0) \\ &y(t,u) \in \mathbb{X} \\ &u(t) \in \mathbb{U} \\ &\left|u(t) - u_0(t)\right| \le \beta \\ &t \in [t_0,T] \end{aligned}$$

In the optimization problem of equation (5.22) a nominal trajectory $x(\cdot,u_0)$ is required, in other words the optimal value depends on any nominal trajectory. On the other hand, at any time step DRHC generates a new trajectory which is served as nominal trajectory in (5.22). Hence, (5.22) should be modified to be independent of any nominal trajectory and be applicable for a general $x(\cdot,u_0)$. This way, the tube can be computed offline and used for online applications. Hence, $x(\cdot,u_0)$ is considered as another decision variable in the optimization problem as follows:

$$\alpha(t) = \underset{\{\alpha(t),\, x(\cdot,u_0)\}}{\text{Max}} \left[\left| \int_{t_0}^{t} \left[f(y(s,u),u(s)) - f(x(s,u_0),u_0(s)) \right] ds \right| \right] \qquad (5.23)$$

subject to:

$$\begin{aligned}
&\dot{y} = f(y,u); && y(0,u) = x(0,u_0) \\
&\dot{x} = f(x,u_0); && x(0,u_0) \in \mathbb{X} \\
&y(t,u) \in \mathbb{X} \\
&x(t,u_0) \in \mathbb{X} \\
&u(t)\ \&\ u_0(t) \in \mathbb{U} \\
&|u(t) - u_0(t)| \le \beta \\
&t \in [t_0, T]
\end{aligned}$$

The following translates this algorithm to the DRHC notation. The results of Problem 5.1 are used to calculate the tube $\mathbb{H}$ around the trajectory of each neighbouring vehicle experiencing the fault:

Problem 5.2: Assume that the control input for faulty vehicle j is bounded as follows:

$$\left| u_{t_k}^{j,j}(t) - u_{t_{k-1}}^{j,j}(t) \right| \le \mu \quad ; t \in [t_k, t_{k-1} + T] \qquad (5.24)$$

where μ is the bound on input variation. Then, if at time t_k vehicle i receives the information from faulty neighbour j with d steps time delay, i.e., $x_{t_{k-d}}^{j,j}(\cdot)$, then calculate the tube around the trajectory of vehicle j at time t_k.

Solution: the results of Problem 5.1 is applicable by finding α and β. To find β after d step delay the input constraint (5.24) can be used sequentially as follows (the superscript j is dropped temporarily):

$$
\begin{aligned}
&-\mu \le u_{t_k}(t) - u_{t_{k-1}}(t) \le \mu \quad ; \quad t \in [t_k, t_{k-1}+T] \\
&-\mu \le u_{t_{k-1}}(t) - u_{t_{k-2}}(t) \le \mu \quad ; \quad t \in [t_{k-1}, t_{k-2}+T] \\
&\quad . \qquad\qquad\qquad\qquad . \\
&\quad . \qquad\qquad\qquad\qquad . \\
&\quad . \qquad\qquad\qquad\qquad . \\
&-\mu \le u_{t_{k-d+1}}(t) - u_{t_{k-d}}(t) \le \mu \quad ; t \in [t_{k-d+1}, t_{k-d}+T] \\
&---------- \\
&-d.\mu \le u_{t_k}(t) - u_{t_{k-d}}(t) \le d.\mu \quad ; \quad t \in [t_k, t_{k-d}+T]
\end{aligned}
\tag{5.25}
$$

Hence,

$$
\beta = d.\mu \; ; \quad t \in [t_k, t_{k-d}+T] \tag{5.26}
$$

Then the tube H^j around the trajectory of vehicle j is presented as follows:

$$
\mathbb{H}^j = \left\{ (t, \boldsymbol{y}) \in [t_k, t_{k-d}+T] \times \mathbb{R}^n \;\middle|\; \left| \int_{t_0}^{t} \left[f(\boldsymbol{y}(s,\boldsymbol{u}), \boldsymbol{u}(s)) - f(\boldsymbol{x}^{j,j}_{t_{k-d}}(s), \boldsymbol{u}^{j,j}_{t_{k-d}}(s)) \right] ds \right| < \alpha^j(t) \;\&\; \left| \boldsymbol{u} - \boldsymbol{u}^{j,j}_{t_{k-d}} \right| < d.\mu \right\} \tag{5.27}
$$

where $\alpha^j(t)$, the radius of tube $\mathbb{H}^j$, is calculated from the following algorithm:

Algorithm 5.2: Tube Calculator (off-line)

1: **For** $t = 0 : \Delta t : T - d\delta$ % (choose Δt as small as appropriate)

2: Solve the following maximization problem:

$$
\alpha^j(t) = \max_{\{\alpha^j(t),\, \boldsymbol{x}(\cdot,\boldsymbol{u}_0)\}} \left[\left| \int_{t_0}^{t} \left[f(\boldsymbol{y}(s,\boldsymbol{u}), \boldsymbol{u}(s)) - f(\boldsymbol{x}(s,\boldsymbol{u}_0), \boldsymbol{u}_0(s)) \right] ds \right| \right]
$$

subject to :

$$
\begin{aligned}
&\dot{\boldsymbol{y}} = f(\boldsymbol{y}, \boldsymbol{u}); \qquad \boldsymbol{y}(0,\boldsymbol{u}) = \boldsymbol{x}(0,\boldsymbol{u}_0) \\
&\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}_0); \qquad \boldsymbol{x}(0,\boldsymbol{u}_0) \in \mathbb{X} \\
&\boldsymbol{y}(t,\boldsymbol{u}) \in \mathbb{X} \\
&\boldsymbol{x}(t,\boldsymbol{u}_0) \in \mathbb{X} \\
&\boldsymbol{u}(t) \;\&\; \boldsymbol{u}_0(t) \in \mathbb{U} \\
&\left| \boldsymbol{u}(t) - \boldsymbol{u}_0(t) \right| \le d.\mu
\end{aligned}
$$

3: Save $\alpha^j(t)$ vs. t to be used for on-line tube calculation.

4: **end**

The output of this algorithm is the trajectory of vector $\alpha^j(t)$ over the time interval $[t_0, T-d\delta]$ and will be used in the online Algorithm 5.3.

Remark 5.1: Calculation of $\alpha^j(t)$ does not impose any on-line computation time as it can be computed off-line. For example, Figure 5.4 shows the computed bound for different time delays, $T = 5$ sec, and the dynamics described in Section 4.2.2.

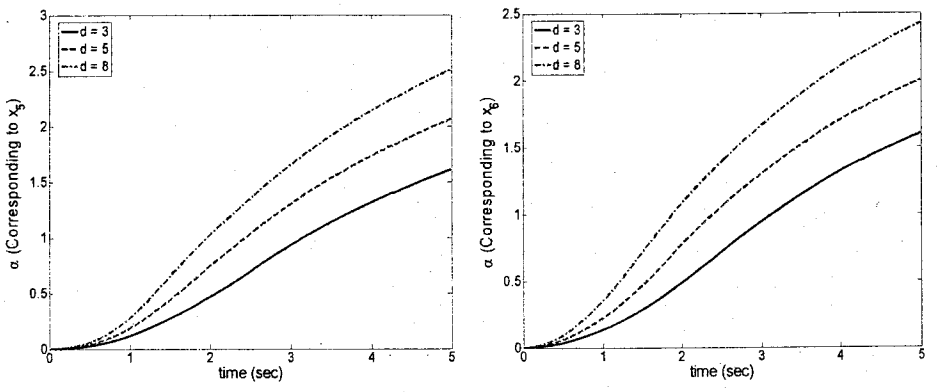

Figure 5.4: Bound on the states over time.

These graphs can be given to the DRHC controller as some tabulated data and be used in the faulty conditions; in this manner no online computation is imposed.

5.3. Collision Avoidance in Formation Problems

For the formation problems the collision avoidance is provided by adding the tube radius, as an extra distance, to the desired relative distance between healthy and faulty vehicles (see Figure 5.5). The following subsections explain how the non-convexity avoidance and formation keeping in this framework is addressed.

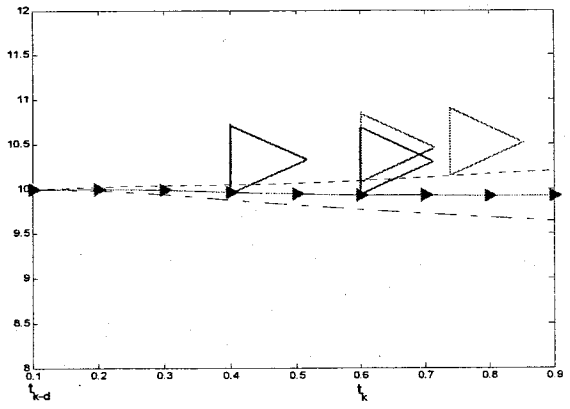

Figure 5.5: Safety Guarantee Using Tube-DRHC in Formation Problems

5.3.1. Non-Convexity Avoidance

In the tube around the trajectory of vehicle j, (i.e., $\mathbb{H}^j$) at any time there is a set of possible states for faulty neighbour instead of a single state. Based on the tube-DRHC idea, for a safe trajectory (no collision with neighbours), in the cost function of Eq. (4.3b), $\boldsymbol{x}^{j,j}_{t_{k-d}}(t)$ must be chosen from the boundary of the tube $\mathbb{H}^j$ and not the trajectory $\boldsymbol{x}^{j,j}_{t_{k-d}}(t)$; however, this can lead to non-convexity of the optimization problem due to non-convex nature of tube. Thus, in order to avoid the non-convexity, in the cost function of Eq. (4.3b), $\boldsymbol{x}^{j,j}_{t_{k-d}}(t)$ is not modified (or replaced by tube $\mathbb{H}^j$), instead the desired relative position $\boldsymbol{r}^{i,j}(t)$ will be modified as follows:

$$\boldsymbol{r}^{i,j}(t) \leftarrow \boldsymbol{r}^{i,j}(t) + sign(\boldsymbol{r}^{i,j}(t)) \cdot \Delta\boldsymbol{r}^{i,j}(t) \qquad (5.28)$$

In fact, the margin $\Delta\boldsymbol{r}^{i,j}(t) \geq \mathbf{0}$ is added to the desired distance to ensure the safety. Since $\boldsymbol{r}^{i,j}(t)$ is the relative position vector, $\Delta\boldsymbol{r}^{i,j}(t)$ is multiplied by diagonal matrix $sign(\boldsymbol{r}^{i,j}(t))$ to make sure that change happens in magnitude of $\boldsymbol{r}^{i,j}(t)$ and hence the

second term does not decrease the magnitude of $\boldsymbol{r}^{i,j}(t)$. The margin $\Delta\boldsymbol{r}^{i,j}(t)$ is the radius of the tube at any time t; hence:

$$\Delta\boldsymbol{r}^{i,j}(t) = \alpha^{j}(t) \tag{5.29}$$

where $\alpha^{j}(t)$ is the tube radius at time t and is calculated from algorithms presented in Section 0.

5.3.2. Preserving Formation Shape in Faulty Conditions

Using a tube instead of a trajectory in the formation leads also to the concept of tight and loose formations; because in faulty conditions the position of the faulty vehicle is assumed to be a closed set (like a sphere), rather than a single point (Figure 5.6 (middle)). In a loose formation the vehicles will keep a larger distance than desired from the faulty vehicle. However, if only the distances to the faulty vehicle become larger than others, the desired formation shape will be ruined; hence, in this case all the vehicles in the team will keep the larger relative distance to preserve the formation shape, no matter if they involve in the fault or not (Figure 5.6 (right)). In general, it is desired to have a *tight* formation if possible.

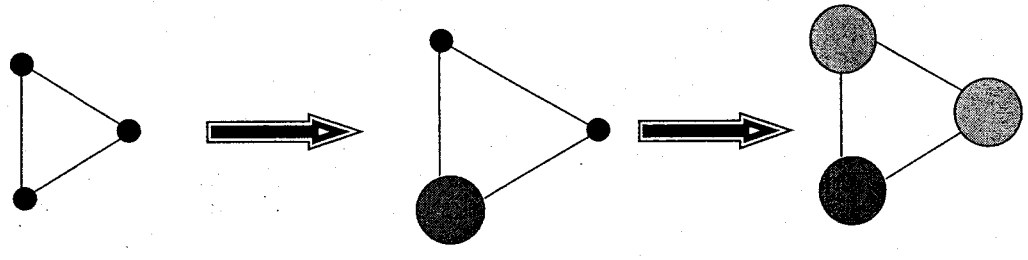

Figure 5.6: Preserving the Formation Shape, Tight and Loose Formations.

5.3.3. Fault Tolerant Tube-DRHC Problem

The fault tolerant Tube-DRHC problem $\mathcal{P}_D^i(t_k)$ for the faulty conditions is defined below at time t_k for any i^{th} vehicle which involves in the fault (either faults with itself or its neighbours).

Problem 5.3: Tube-DRHC Problem $\mathcal{P}_D^i(t_k)$:

$$\underset{\{\boldsymbol{u}_{t_k}^{i,i}(\cdot),\ \boldsymbol{x}_{t_k}^{i,i}(\cdot),\ \boldsymbol{x}_{t_k}^{-i}(t_{k-d}+T:t_k+T)\}}{\text{Min}} J_D^i(\Gamma^i(t_k)) \tag{5.30}$$

subject to:

- *for* $t \in [t_k, t_k+T]$:

$$\dot{\boldsymbol{x}}_{t_k}^{i,i}(t) = f(\boldsymbol{x}_{t_k}^{i,i}(t), \boldsymbol{u}_{t_k}^{i,i}(t))\ ;\ \boldsymbol{x}_{t_k}^{i,i}(t_k) = \boldsymbol{x}^i(t_k) \tag{5.31a}$$

$$\boldsymbol{x}_{t_k}^{i,i}(t) \in \mathbb{X}^i,\quad \boldsymbol{u}_{t_k}^{i,i}(t) \in \mathbb{U}^i \tag{5.31b}$$

- *for* $t \in [t_{k-d}+T, t_k+T]$ *and* $(i,j) \in \mathbb{E}$:

$$\dot{\boldsymbol{x}}_{t_k}^{j,i}(t) = f(\boldsymbol{x}_{t_k}^{j,i}(t), \boldsymbol{u}_{t_k}^{j,i}(t))\ ;\ \boldsymbol{x}_{t_k}^{j,i}(t_{k-d}+T) = \boldsymbol{x}_{t_{k-d}}^{j,j}(t_{k-d}+T) \tag{5.31c}$$

$$\boldsymbol{x}_{t_k}^{j,i}(t) \in \mathbb{X}^j,\quad \boldsymbol{u}_{t_k}^{j,i}(t) \in \mathbb{U}^j \tag{5.31d}$$

$$\begin{aligned} &\boldsymbol{x}_{t_k}^{i,i}(t_k+T) \in \mathbb{X}_f^i \\ &\boldsymbol{x}_{t_k}^{j,i}(t_k+T) \in \mathbb{X}_f^j\ ;\ (i,j) \in \mathbb{E} \end{aligned} \tag{5.31e}$$

- *for* $t \in [t_k, t_{k-1}+T]$:

$$\left| \boldsymbol{u}_{t_k}^{i,i}(t) - \boldsymbol{u}_{t_{k-1}}^{i,i}(t) \right| \le \mu \tag{5.31f}$$

In Eq. (5.30) J_D^i is calculated from Eq. (4.3). Constraint of Eq. (5.31f) is imposed for safety guarantee purpose. This constraint restricts the manoeuvrability of the vehicles

involving in the fault in the faulty conditions and allows the neighbouring vehicles to calculate the reachable set and tube.

5.3.4. Fault Tolerant Tube-DRHC Algorithm

The following algorithm is presented for the on-line implementation of the proposed fault tolerant Tube-DRHC problem $\mathcal{P}_D^i(t_k)$. The algorithm is formulated for the i^{th} vehicle; in fact, all vehicles run this algorithm during the mission simultaneously:

Algorithm 5.3: Fault Tolerant Tube-DRHC (online)

1: $k \leftarrow 0$, and GOTO step 4.

2: Receive $\boldsymbol{x}_{t_{k-d}}^{j,j}(\cdot)$ from leaders where $(i,j) \in \mathbb{E}$.

3: Take $\Delta r^{i,j} = \alpha^j(t)$ and update $r^{i,j}(t) \leftarrow r^{i,j}(t) + sign(r^{i,j}(t)).\Delta r^{i,j}(t)$ and update $r^{i,j}$ in the cost function (4.3).

4: Measure $\boldsymbol{x}^i(t_k)$ and update the *information set* of Eq. (4.1).

5: Solve $\mathcal{P}_D^i(t_k)$ of *Problem 5.3*.

6: Send the state trajectory $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$ to followers where $(j,i) \in \mathbb{E}$.

7: Execute the control action for individual vehicle *i* during $[t_k, t_{k+1}]$.

8: $k \leftarrow k+1$. Goto step 2.

This algorithm is a modified version of Algorithm 2.2 and handles the large communication delays for faulty conditions. It also performs the safety guarantee by executing the step 3 and updating the tube around the trajectory of faulty vehicle(s).

5.3.5. Example 1: Tube-DRHC for LTI Systems

In the following simulation the effectiveness of the proposed tube-DRHC for avoiding the collision in formation problems is investigated. This case involves the triangular formation control of six vehicles with linear dynamics (4.7). The *direct* communication graph topology is set as follows:

$$\begin{aligned}\mathbb{V} &= \{1,2,3,4,5,6\} \\ \mathbb{E} &= \{(1,2),(1,3),(2,3),(2,4),(3,6),(4,5),(5,6)\}\end{aligned} \tag{5.32}$$

The neighbour assignment is performed manually prior to the mission by selecting the two or more vehicles that are closest in the desired formation. The results are shown in Figure 5.7 and Figure 5.8. In this case, two set of way points are considered to be visited by the fleet. At first the fleet is not faulty but after 5 sec (around point (70,60)) vehicle 2 (whose trajectory is dotted) becomes faulty which leads to a d=8 time step delay in the messages communicated to and from vehicle 2. As shown in Figure 5.7, due to the result of using the tube-DRHC approach the distances between vehicles increase and the formation is expanded for safety upon fault occurrence. The distances between each pair of neighbouring vehicles are shown in Figure 5.8 for two cases: 1) faulty without any fault tolerant algorithm (*Algorithm 2.2*) and 2) faulty with the proposed fault tolerant *Algorithm 5.3*. It is desired that vehicles keep a 7m distance from neighbours. As seen from Figure 5.8 (right) in the case of *Algorithm 2.2*, the vehicles get too close to each other and may collide. However, the reconfigurable *Algorithm 5.3* offers a larger distance (Figure 5.8, left) and the formation is safe as a consequence of using the tube-DRHC approach.

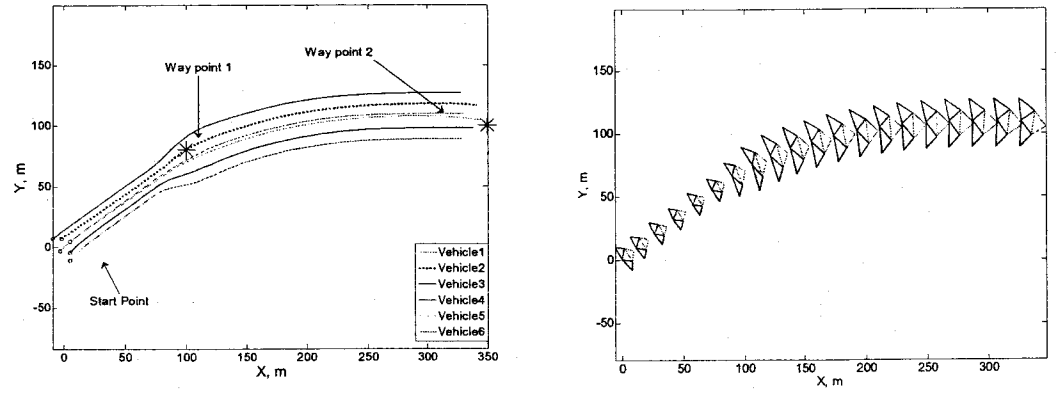

Figure 5.7: Trajectory (left) and snapshot (right) of a six vehicle triangle configuration experiencing a communication failure: the formation expands upon fault occurrence.

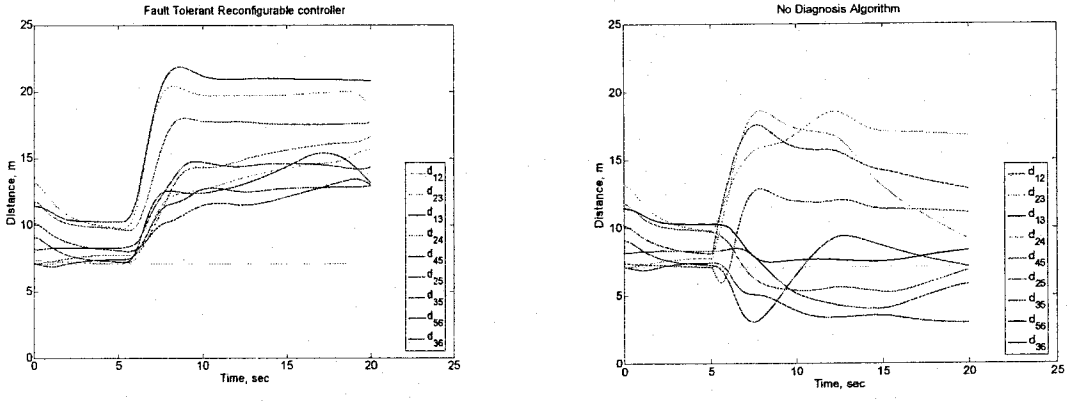

Figure 5.8: Distances between each pair of vehicles in a six vehicle triangle configuration experiencing a communication failure (at t=5s): Algorithm 5.3 (left) and Algorithm 2.2 (right).

The time history of control input is depicted in Figure 5.9; as seen the control input is bounded and does not fluctuate. Figure 5.10 shows the time history of velocity vector for all the vehicles. These results imply that the controller is feasible and can be implemented in practice.

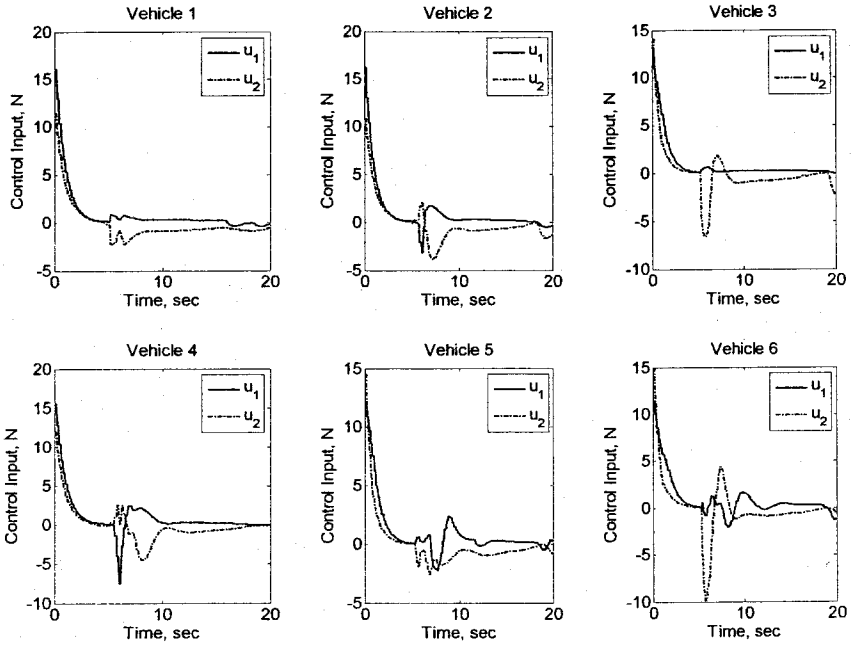

Figure 5.9: Vehicle control inputs

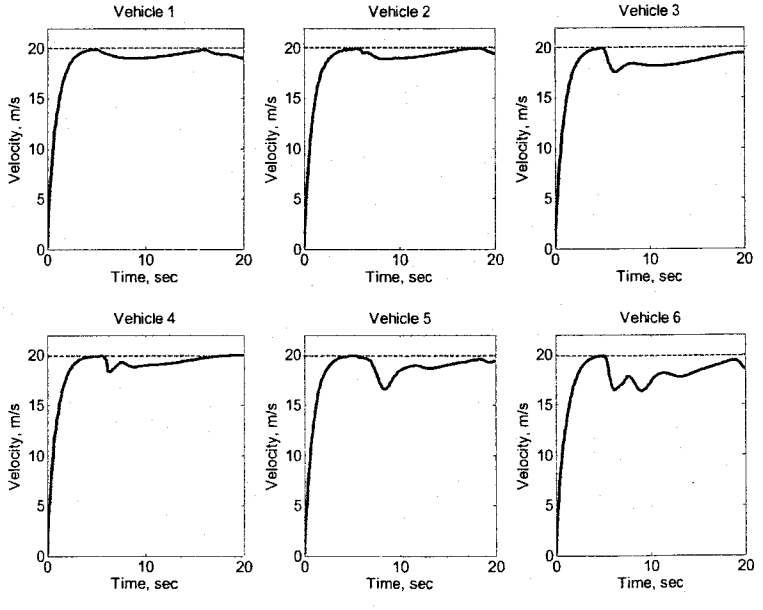

Figure 5.10: Vehicle velocities and constraints

5.3.6. Example 2: Tube-DRHC for Nonlinear Systems

In this section the proposed Algorithm 4.1 which implements tube DRHC approach is employed for formation control of six vehicles with nonlinear dynamics (4.12). The results are shown in Figure 5.11 through Figure 5.13. In this case, two sets of way points are considered to be visited by the fleet. At the beginning the fleet is not faulty but after 3 sec (around point (22,22)) the vehicle 2 (whose trajectory is dotted) becomes faulty, this leads to d=7 time step delay in the communicated messages to and from vehicle 2. As seen from Figure 5.11 the vehicles start to keep a larger distance and the formation is expanded for safety upon fault occurrence. Figure 5.12 shows the same scenario when no detection algorithm is used (Algorithm 2.2).

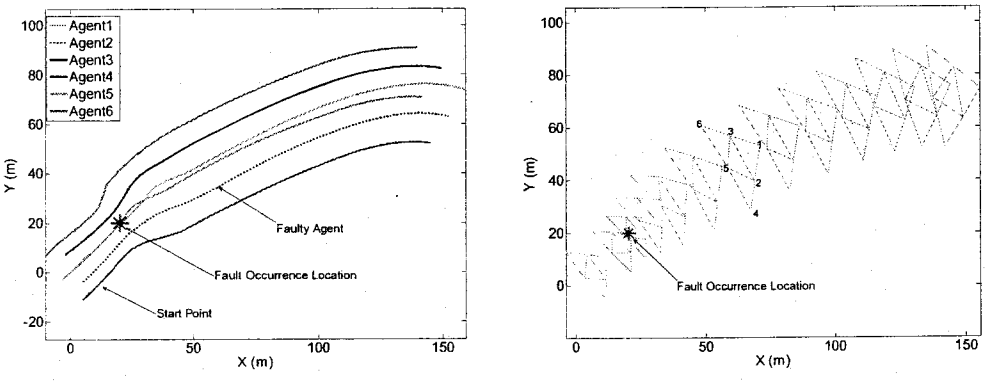

Figure 5.11: Trajectory (left) and formation snapshot (right) of six vehicles in triangular formations when the reconfigurable fault tolerant controller is used.

The distances between each pair of neighbouring vehicles are shown in Figure 5.13 for two cases: 1) faulty without any fault tolerant algorithm (Algorithm 2.2) and 2) faulty with proposed fault tolerant algorithm (Algorithm 4.1). It is desired that vehicles keep a 7.07m distance from neighbours. As seen from Figure 5.13 (right) in the case of Algorithm 2.2, vehicles get too close to each other and may collide. However, the

reconfigurable Algorithm 4.1 offers a loose but safe formation as the consequence of using tube RHC for safety.

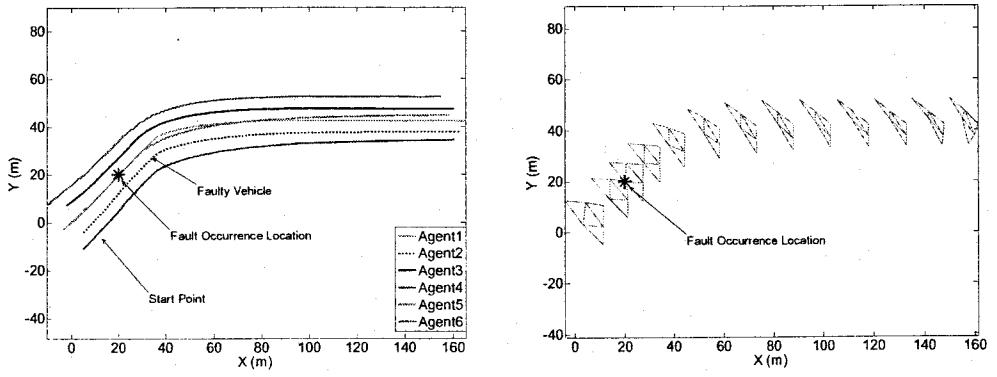

Figure 5.12: Trajectory (left) and formation snapshot (right) of six vehicles in triangular formations when no fault tolerant algorithm is used.

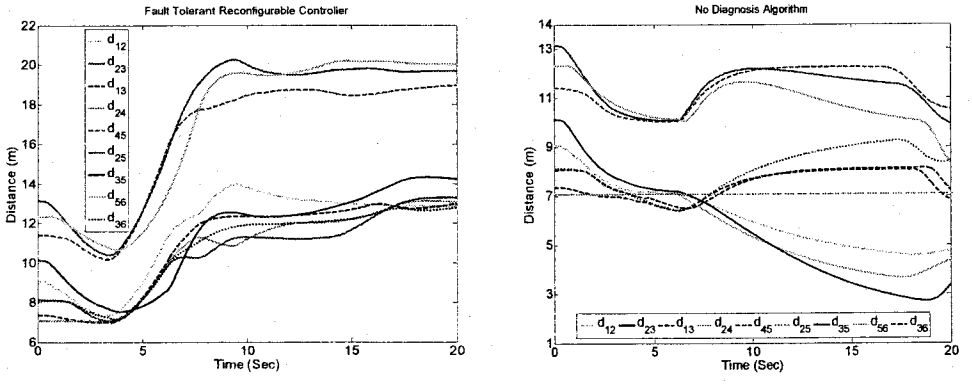

Figure 5.13: Distances between each pair of vehicles for algorithm 4 (left) and without detection (right).

5.4. General Collision Avoidance Problems

In this section using the general approach of Tube-DRHC the collision avoidance and conflict resolution for non-formation problems is addressed. The new demands in decentralized collision avoidance and conflict resolution span a wide range of application such as air traffic control [170-172], road traffic control [173, 174], mobile robots [175,

176] and cooperative UAVs [177, 162]. Many approaches have been used to design a safe trajectory planner for such applications. Among them model predictive control has found a considerable attention while it is complained constantly about its weakness in handling the non-convex constraints arising from collision avoidance problem. To tackle this problem, in [162] a hybrid rule-based extension of the decentralized receding horizon control (DRHC) is proposed to avoid possible collisions. Also, in [177] a mixed integer linear programming (MLIP) approach is utilized to handle the non-convex collision avoidance constraint by decentralized model predictive control architecture. In [175] the safety is provided by seeking new manoeuvres such that all conflicts are avoided. Most recently, in [178], using the concept of invariant sets a set of emergency manoeuvres is computed to avoid collisions whenever the feasibility is lost.

Using RHC for developing the collision avoidance and conflict resolution algorithms is motivated by three main property of RHC: its *predictive* nature allows predicting the possible collisions, its unique advantage for *handling the constraints* helps avoiding the predicted collisions by imposing some constraints on the inputs. Also, it is easy to provide *cooperation* through the cost function among the vehicles to avoid collisions. In this section, all these three advantages are utilized to develop a novel DRHC approach which guarantees the collision avoidance in presence of large communication delays.

In Section 5.3 a tube-based DRHC is proposed to provide formation safety under communication failures where the desired relative distances in the cost function are set to be larger than the radii of the reachable sets; although it has shown great efficiency to provide safe formation in simulations, in general it is not applicable to non-formation problems; also, it may not guarantee the collision avoidance as the local minimum of the

cooperation cost is computationally difficult to be achieved; further, the overshoots in transient response may lead to collisions. The approach presented in this section for collision avoidance in a decentralized framework is based on a simple idea: "*I restrict my manoeuvrability then you compute my reachable set and avoid that*". Technically speaking, a tube is assumed around the delayed trajectory of neighbouring vehicles; since the neighbouring vehicles may not stay on the delayed paths the radius of the tube is non-zero; as mentioned previously the tube radius depends on the delay and the manoeuvrability during the delay time. Then the neighbouring vehicles are assumed to avoid each others tube. Also, the desired relative distance is chosen to be larger than the radii of the reachable sets of neighbouring vehicles and hence respect the previous results in Section 5.3. Although the main problem in formation control is to force the team members to move in a pre-specified shape, for applications such as air traffic control the formation formulation can be used by relaxing this requirement, i.e., the vehicles may be forced to get different relative positions from their neighbouring vehicles and these relative positions may vary depending on the communication delay and manoeuvrability.

The *tube* analysis allows each vehicle to predict the possible collisions and hence, change the plan to avoid the collisions; the collision avoidance policy is based on setting the admissible input set $\mathbb{U}^i$ so that the tubes do not intersect. Then at each time step each vehicle $i \in \mathbb{V}$:

1- *Calculates the neighbour's tube from delayed information by assuming limited manoeuvrability for neighbours.*

2- *Sets* $\mathbb{U}^i$ *(manoeuvrability) so that its tube does not intersect with neighbour's tube.*

Based on this idea, an algorithm is proposed by which each vehicle determines its manoeuvrability so that its tube does not intersect with neighbour's tube and hence the collision is avoided.

Each vehicle i uses the formula $\beta^j(d) = d^3\delta[\mu^j]'[\mu^j]$ in (5.17) where $(i,j) \in \mathbb{E}$ to calculate the bound on the inputs of its neighbour j as it sees a d step delay from j; in fact, this is the bound calculated by vehicle i then the second superscript is added in order to indicate that vehicle i calculates this bound for neighbour j, i.e., $\beta^{j,i}(d) = d^3\delta[\mu^j]'[\mu^j]$.

However, the vehicle i has access to its updated trajectory with only one step delay and then set d=1 which yields $\beta^{i,i} = \delta[\mu^i]'[\mu^i]$ to be used by vehicle i for calculation of its own tube. Then the tube of vehicle j calculated at time t_k by neighbour i from the delayed trajectory $\xi^{j,j}_{t_{k-d}}$ is denoted by $\mathbb{H}^{j,i}_{t_k}$ in (5.8). Also the tube of vehicle i calculated at time t_k by itself from one step delayed trajectory $\xi^{i,i}_{t_{k-1}}$ is denoted by $\mathbb{H}^{i,i}_{t_k}$ and calculated from (5.8) by setting d=1 and hence using $\beta^{i,i} = \delta[\mu^i]'[\mu^i]$. Then, each vehicle $i \in \mathbb{V}$ chooses its manoeuvrability μ^i so that the following collision avoidance condition holds:

$$\mathbb{H}^{i,i} \cap \mathbb{H}^{i,j} = \varnothing \quad i,j \in \mathbb{V} \ \ \& \ (i,j) \in \mathbb{E} \qquad (5.33)$$

In fact, each vehicle i calculates the tube around the delayed trajectory of each neighbour and then define its manoeuvrability μ^i so that there is no intersection between its tube and the tube of neighbouring vehicles. Thanks to the capability of DRHC it is

easy to enforce the manoeuvrability condition (5.16) in the optimization problem via input constraints in (5.31b). Hence the input set in DRHC is updated as follows:

$$\mathbb{U}^i(t_k)=\left\{\boldsymbol{u}(t)\,\middle|\,\left|\boldsymbol{u}_{t_k}^{j,j}(t)-\boldsymbol{u}_{t_{k-1}}^{j,j}(t)\right|\le\mu^j,\ t\in[t_k,t_{k-1}+T]\right\} \quad (5.34)$$

If the graph topology is well-connected then the decentralized condition (5.33) can imply condition $\mathbb{H}^1\cap\mathbb{H}^2\cap\mathbb{H}^3\cap.....\cap\mathbb{H}^{N_v}=\varnothing$ and hence collision avoidance satisfaction.

5.4.1. Formation Setting

The desired relative distance $\boldsymbol{r}^{i,j}$ for all state components in ξ is well-defined as follows:

$$\boldsymbol{r}^{i,j}(t)=\boldsymbol{\alpha}^i(t)+\boldsymbol{\alpha}^j(t) \qquad (i,j)\in\mathbb{E} \qquad (5.35)$$

If the cooperation cost reaches its minimum, i.e., zero, then this condition also guarantees the collision avoidance.

5.4.2. Collision Avoidance DRHC Algorithm

Initial feasibility Assumption: assume at time t_k for $\forall i\in\mathbb{V}$ there exist μ_0^i so that a feasible solution satisfying collision avoidance condition (5.33) exists.

The following algorithm is presented for the on-line implementation of DRHC with the proposed collision avoidance scheme, $\forall i\in\mathbb{V}$:

Algorithm 5.4: *DRHC with Collision Avoidance*

1- Let k=0, and GOTO step 4.
2- Receive the trajectory $\boldsymbol{x}_{t_{k-d}}^{j,j}(\cdot)$; $(i,j)\in\mathbb{E}$ (wit6h appropriate d).
3- Calculate admissible $\mathbb{U}^i$:
   a. Choose $\mu^j=\mu_0^j$ and calculate $\mathbb{H}^j$ for $(i,j)\in\mathbb{E}$.

b. Choose $\mu^i = \mu_0^i$ and calculate $\mathbb{H}^i$. If $\mathbb{H}^i \cap \mathbb{H}^j \neq \varnothing; (i,j) \in \mathbb{E}$ then reduce μ^i and redo the step.

c. Compute the admissible input set $\mathbb{U}^i$ from (5.34).

4- Measure $x^i(t_k)$ and calculate $\Gamma^i(t_k)$.

5- Solve $\mathcal{P}^i(t_k)$ and generate: $u_{t_k}^{i,i}(\cdot)$ and $x_{t_k}^{i,i}(\cdot)$.

6- Send the trajectory $x_{t_k}^{i,i}(\cdot)$ to the neighbouring vehicles.

7- Execute the control action for individual vehicle *i* over the time interval $[t_k, t_{k+1}]$.

8- *k=k+1*. Goto step 2.

The only difference between this algorithm and *Algorithm 5.3* is step 3 which provides collision avoidance.

5.4.3. Simulation Results

Collision avoidance of a fleet of unmanned vehicles with double integrator dynamics and velocity damping in the 2D plane is considered, where $\xi \in \mathbb{R}^2$, $v \in \mathbb{R}^2$, $u \in \mathbb{R}^2$ and $\dot{\xi} = v$, $\dot{v} = -v + [1,\ 0.4]u$. This dynamics is marginally stable; hence, to take advantage of the Controllability *Gramian* for stable systems an inner loop feedback controller is first designed to stabilize the dynamics by placing the poles at [-0.0513; -0.0513; -1.9487; -1.9487] using the feedback controller: K= [0.1, 0, 1.0, 0; 0, 0.25, 0.0, 2.5].

5.4.3.1. Off-line Calculations

Then the solution of the LMI (5.7) is $M = W^{-1}$ = diag(12.0, 120.0, 75.0, 750.0). To verify this ellipsoid bound the corresponding ellipsoid along with 100 random simulations for four different β (see (5.5) are depicted in Figure 5.14. As seen the ellipsoid is non conservative bound on the reachable set.

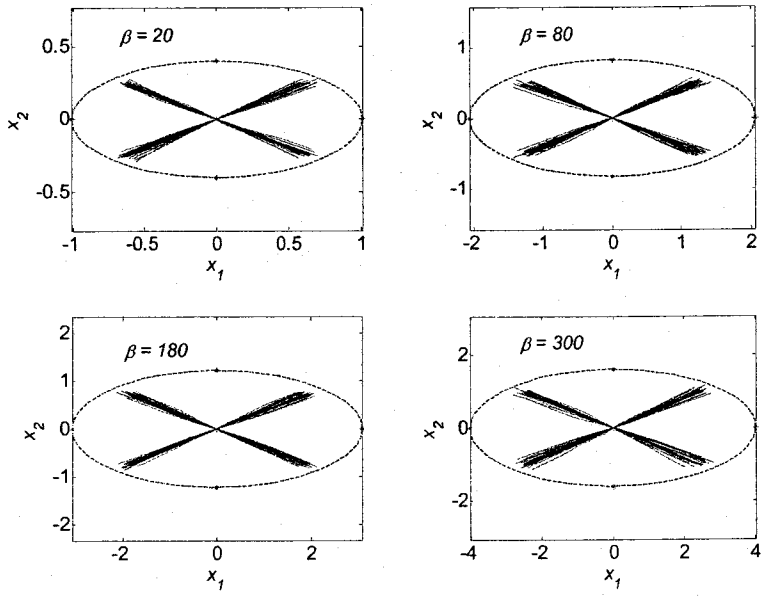

Figure 5.14: The reachable sets corresponding to positions for different manoeuvrability.

5.4.3.2. Simulation of On-line Scenarios

The prediction horizon and execution horizon (sampling time) of DRHC is set to T=1 sec and $\delta = 0.2$ sec, respectively. Also, it is assumed that the communicated messages are subject to a delay of d=3 steps (or $\tau = d.\delta = 0.6\text{sec}$). Further the initial manoeuvrability parameter is set to $\mu_0^i = 40I$. To test the proposed collision avoidance algorithm, a scenario (see Figure 5.15) is considered where the vehicles start from some random positions (circles) and they have to visit some targets (cross product sign). The target positions are assigned so that the vehicles potentially collide. This scenario can imitate the air traffic control scenario near the airports. For example, the Figure 5.15-Left shows the case where no collision avoidance constraint is used; in fact, the vehicles do not cooperate and hence two of the vehicles collide according to the corresponding distance time history in Figure 5.16-Left. For this scenario the Figure 5.15-Right shows the effect

of the proposed collision avoidance algorithm. As seen from the corresponding distance profile of Figure 5.16-Right the proposed algorithm can predict the possible collisions and avoid them.

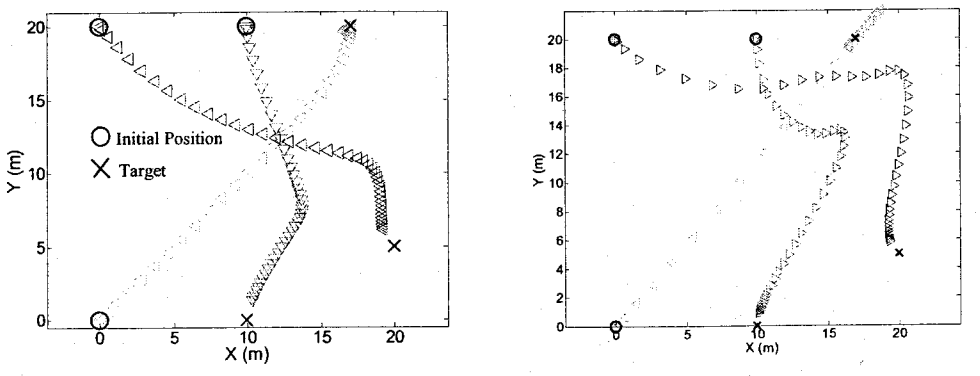

Figure 5.15: The snapshot of the trajectories of three vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm.

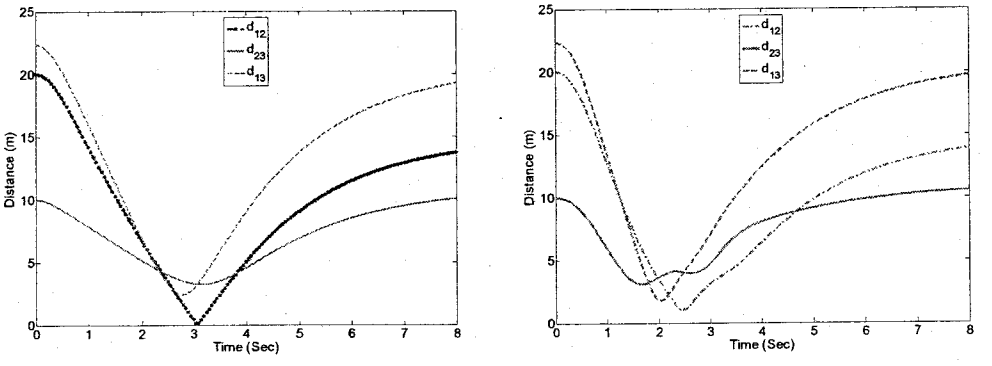

Figure 5.16: The distance between each pair of vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm.

To consider a more complex scenario 3 vehicles are added to the mission and the snapshot of the trajectories for both cases, when no collision avoidance algorithm is used and when the proposed collision avoidance algorithm is used, is shown in Figure 5.17. Also the corresponding distance profile is shown in Figure 5.18. For this case, if the

collision avoidance algorithm is not used two pair of vehicles will collide according to Figure 5.18-Left, but the proposed collision avoidance algorithm is able to resolve the conflicts.

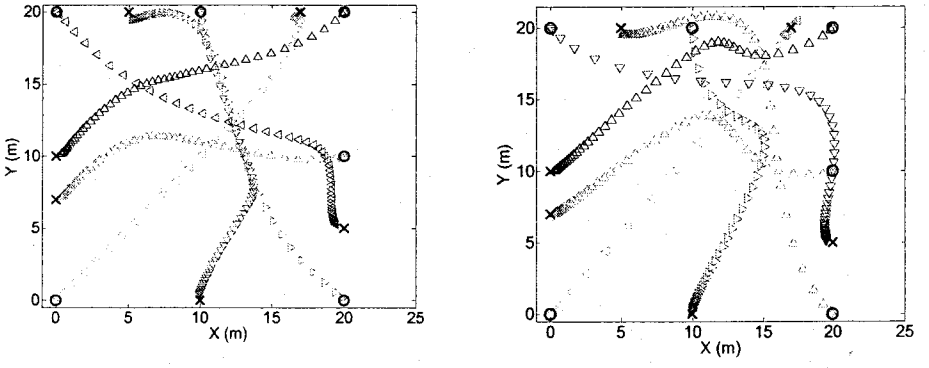

Figure 5.17: The snapshot of the trajectories of six vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm.

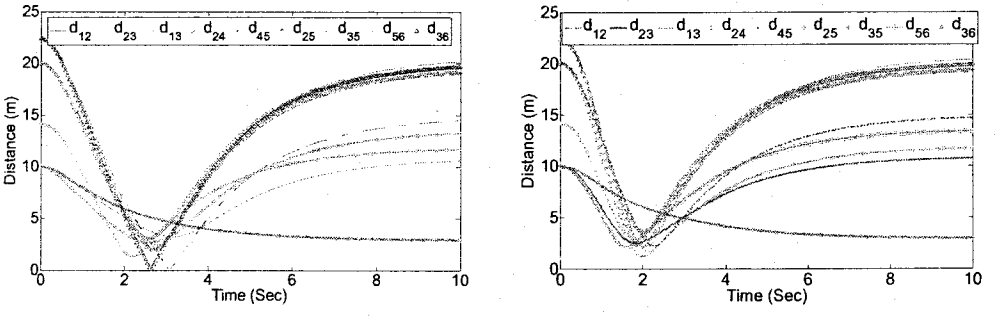

Figure 5.18: The distance between each pair of vehicles: Left: no collision avoidance algorithm, Right: collision avoidance algorithm.

5.5. Summary

This chapter involves the development of a tube-DRHC approach to provide safety against possible collisions in presence of communication failures giving rise to large communication delays. The main idea is to consider a tube shape trajectory around the

trajectory of faulty vehicle instead of a single trajectory. Using the state transition matrix and LMI approach a method for calculating such tubes for subsystems with linear dynamics is presented which allows offline calculation of tube and hence, does not impose any on-line computation. Also, a computational algorithm for calculation of such tube for a general class of dynamics is presented which does not need any online computation. Both formation problem and non-formation scenarios are considered; non-formation scenarios can address a general class of applications including road or air traffic control. Then two algorithms are proposed for implementing the Tube-DRHC architecture. Simulations illustrate that the proposed approach can lead to a safe formation in presence of communication failures.

Chapter 6. Stability and Feasibility Analysis

In this chapter the feasibility, stability and performance analysis of the delayed DRHC architecture proposed in Chapter 4 are discussed. Even for the classical RHC it is not straightforward to provide the stability and feasibility; as it is discussed in Chapter 1 the stability of RHC is provided by fine tuning of *prediction horizon*, *execution horizon*, *terminal set* and *terminal matrix penalty* which may lead to very conservative and poor performance. For DRHC it is even more difficult to provide the stability and feasibility due to interactions among neighbouring vehicles and unavailability of updated neighbour's plans. In most of the cases an asymptotic stability is not guaranteed and a general stability argument is studied, for instance see [117]. In some cases a restrictive condition should be imposed; for example, in [151] the asymptotic stability is achieved by imposing the final equality constraint $x(t+T)=0$ which is computationally prohibitive.

Similar to RHC the stability and performance of DRHC is often improved by careful formulation of the cost function and constraint [1, 2, and 3]. For example, Keviczky et al. in [1] and [151], proposed to control a team of vehicles with decoupled discrete-time dynamics by breaking down a centralized RHC architecture into a set of distinct RHC controllers of smaller sizes; with such approach, the vehicles are coupled through the cost function. In [151], each vehicle predicts its neighbour's behaviour from the dynamical model available; and based on such predictions every vehicle plans its future behaviour. Stability analysis shows that a smaller mismatch between the predicted and the actual

trajectories of all the neighbours can lead to improved closed-loop stability. Furthermore, in [2] and [117], Dunbar et al. proposed a distributed RHC for multivehicle systems where the continuous-time dynamically decoupled subsystems have their state vectors coupled through the cost function of an RHC control problem. Each vehicle solves an optimization problem and generates its own control action using the available information of neighbouring vehicles, which is possibly delayed. In [117], the key requirement to ensure multi-vehicle systems stability is that each vehicle's control input, at each time step, does not significantly deviate from its control action applied at the previous time step. Also, in [3, 13], authors a decentralized robust safe but knowledgeable (RSBK) model predictive control algorithm [14] is developed. The RSBK MPC uses the constraint tightening technique to achieve robustness and lower online computational burden. Using local knowledge, it is shown that each vehicle always has a solution for the DRHC problem guaranteeing robust feasibility for the entire fleet in presence of disturbances acting on the vehicle dynamics. The algorithm is extended in [15] to account for the communication and computation delays in presence of bounded disturbances acting on the vehicle dynamics.

One of the few works on the stability of delayed DRHC is conducted by Parisini et al. [121] where a receding horizon approach is utilized for the distributed control of cooperative agents with delayed information exchange and linear dynamics. In such framework, to incorporate the communication delay, the control law is broken down in two components: one due to feedback from local states and the other based on the delayed information gathered from neighbouring vehicles. But the cost function is not delayed itself. Using the analytical solution of finite RHC, the global stability of the

system is provided and a rigorous performance analysis is established. Also, in [122] to derive a relationship between the stability of the team of cooperating agents with the coupling matrix penalties chosen in the local cost functions, the results of [121] are extended by developing a set of bounding expressions for the linear control law. It is shown that the stability of the overall team of cooperating agents can be guaranteed by appropriate selection of matrix penalties in the cost function. The authors extend their approach to the case of agents with nonlinear dynamics in [123] by exploiting the input-to-state stability (ISS) argument.

The stability analysis of this chapter is based on the *quasi-infinite*-RHC formulation [10] where the key issue is to tune the *terminal cost* so that the closed-loop DRHC bear the property of an infinite horizon controller. In this approach the states are driven to a neighbourhood of the origin where it is a positively invariant set under a feedback *terminal controller* for linearized system. In fact, the terminal set is the region of attraction of the linearized system under the feedback *terminal controller*. Hence, the size of the *terminal region* depends on the degree of nonlinearity of the system; and since for linear stable systems a globally stabilizing feedback *terminal controller* can be found this restriction can be relaxed.

6.1. Performance Analysis

The simulation results in Chapter 4 show the promising performance of the proposed Delayed DRHC over the classical DRHC algorithms for handling the delays. Although it is not straightforward to prove that the proposed controller always provides better

performance than previous ones, two important issues should be pointed out about the performance results:

1- By predicting the tail of the trajectories, the proposed Delayed DRHC controller can imitate the process which neighbouring vehicles are using to predict their trajectories; and hence, obtain similar results and behaviour as neighbours.

2- The proposed controller allows providing a bound on the performance while other methods do not. In fact, by imposing some limitation on the manoeuvrability of each vehicle, which is possible by constraint handling property of DRHC, depending on the communication delay, the desired bound on the performance can be achieved. For instance, assume the manoeuvrability of each vehicle is restricted so that $\forall j \in \mathbb{V}$:

$$\begin{aligned}
&\left| x_{t_k}^{j,j}(t) - x_{t_{k-1}}^{j,j}(t) \right| \le \mu && t \in [t_k, t_{k-1} + T] \\
&\left| x_{t_k}^{j,j}(t) - x_{t_{k-1}}^{j,j}(t_{k-1} + T) \right| \le \mu t && t \in (t_{k-1} + T, t_k + T] \\
&\left| x_{t_k}^{j,i}(t) - x_{t_{k-d}}^{j,j}(t_{k-d} + T) \right| \le \mu t && t \in (t_{k-d} + T, t_k + T]
\end{aligned} \tag{6.1}$$

where μ is a vector with appropriate length and called the *manoeuvrability* vector. Then the actual cooperation cost is measured as the performance index:

$$I^i(t_k) = \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_k+T} \left\| x_{t_k}^{i,i} - x_{t_k}^{j,j} \right\|_S^2 dt \tag{6.2}$$

Also, assume that each optimization problem $\mathcal{P}_D^i(t_k)$ reaches its global minimum which results in zero predicted cooperation cost; hence:

$$
\begin{aligned}
I^i(t_k) &= \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k-d}+T} \left\| x_{t_k}^{i,i} - x_{t_k}^{j,j} \right\|_S^2 dt + \int_{t_{k-d}+T}^{t_k+T} \left\| x_{t_k}^{i,i} - x_{t_k}^{j,j} \right\|_S^2 dt = \\
&\sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k-d}+T} \left\| x_{t_k}^{i,i} - x_{t_{k-d}}^{j,j} + x_{t_{k-d}}^{j,j} - x_{t_k}^{j,j} \right\|_S^2 dt + \\
&\sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_k+T} \left\| x_{t_k}^{i,i} - x_{t_k}^{j,i} + x_{t_k}^{j,i} - x_{t_k}^{j,j} \right\|_S^2 dt \\
&\leq \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k-d}+T} \left(\left\| x_{t_k}^{i,i} - x_{t_{k-d}}^{j,j} \right\|_S^2 + \left\| x_{t_{k-d}}^{j,j} - x_{t_k}^{j,j} \right\|_S^2 \right) dt \\
&\quad + \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_k+T} \left(\left\| x_{t_k}^{i,i} - x_{t_k}^{j,i} \right\|_S^2 + \left\| x_{t_k}^{j,i} - x_{t_k}^{j,j} \right\|_S^2 \right) dt \\
&= \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k-d}+T} \left(0 + \left\| x_{t_{k-d}}^{j,j} - x_{t_k}^{j,j} \right\|_S^2 \right) dt + \int_{t_{k-d}+T}^{t_k+T} \left(0 + \left\| x_{t_k}^{j,i} - x_{t_k}^{j,j} \right\|_S^2 \right) dt \\
&\leq \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k-d}+T} \left(\left\| x_{t_{k-d}}^{j,j} - x_{t_k}^{j,j} \right\|_S^2 \right) dt + \\
&\sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_k+T} \left(\left\| x_{t_k}^{j,i} - x_{t_{k-d}}^{j,j}(t_{k-d}+T) \right\|_S^2 + \left\| x_{t_{k-d}}^{j,j}(t_{k-d}+T) - x_{t_k}^{j,j} \right\|_S^2 \right) dt
\end{aligned}
\tag{6.3}
$$

Then using the bound (6.1) sequentially yields:

$$
\begin{aligned}
&\left| x_{t_k}^{j,j}(t) - x_{t_{k-d}}^{j,j}(t) \right| \leq d\mu \qquad t \in [t_k, t_{k-d}+T] \\
&\left| x_{t_k}^{j,j}(t) - x_{t_{k-d}}^{j,j}(t_k+T) \right| \leq \mu t + (d-1)\delta \times \mu \leq d\delta\mu \quad t \in (t_{k-d}+T, t_k+T] \\
&\left| x_{t_k}^{j,i}(t) - x_{t_{k-d}}^{j,j}(t_{k-d}+T) \right| \leq \mu t \leq d\delta\mu \qquad t \in (t_{k-d}+T, t_k+T]
\end{aligned}
\tag{6.4}
$$

Using (6.4) in (6.3) yields:

$$
\begin{aligned}
I^i(t_k) \le & \int_{t_k}^{t_{k-d}+T} N_n^i \left(d^2 \lambda_{\max}(S) \times \mu' \mu \right) dt + \\
& N_n^i \int_{t_{k-d}+T}^{t_k+T} \left(d^2 \delta^2 \lambda_{\max}(S) \times \mu' \mu + d^2 \delta^2 \lambda_{\max}(S) \times \mu' \mu \right) dt \qquad (6.5) \\
& = N_n^i (d^2 \lambda_{\max}(S) \times \mu' \mu) \times (T - d\delta) + 2 N_n^i (d^2 \delta^2 \lambda_{\max}(S) \times \mu' \mu) \times (d\delta) \\
& = N_n^i d^2 \lambda_{\max}(S) \mu' \mu (T - d\delta) + 2 N_n^i d^3 \delta^3 \lambda_{\max}(S) \mu' \mu
\end{aligned}
$$

Inequality (6.5) implies that the bound on the performance (cooperation cost) is a nonlinear function of communication delay d, the manoeuvrability μ , sampling time δ , prediction horizon T, and the maximum eigenvalue of S. then by appropriately setting these parameters for the proposed controller the desired bound on the performance can be achieved.

6.2. Feasibility Analysis

A solution set $(\boldsymbol{x}^i(t), \boldsymbol{u}^i(t), \boldsymbol{x}^{-i}(t), \boldsymbol{u}^{-i}(t))$ to the optimization problem $\mathcal{P}_D^i(t_k)$ is called feasible if it satisfies constraints (4.5a) through (4.5e) and (5.31f) in finite time. The following theorem addresses the feasibility of the developed controller.

Theorem 6.1(Feasibility): Assume the initial optimization problems $\mathcal{P}_D^i(t_0)$, $\mathcal{P}_D^i(t_1)$, $\ldots \mathcal{P}_D^i(t_d)$ are feasible for $\forall i \in \mathbb{V}$. Further, assume the sets of admissible states and inputs contain the origin. Then all future subsequent DRHC problems are feasible.

Proof: It is sufficient to prove that if $\mathcal{P}_D^i(t_k)$, $\mathcal{P}_D^i(t_{k-1})$, $\ldots \mathcal{P}_D^i(t_{k-d})$ are feasible then there will be a feasible solution to the problem $\mathcal{P}_D^i(t_{k+1})$ for $k \in \mathbb{N}$ and $i \in \mathbb{V}$. The feasible solution can be constructed from the solution of $\mathcal{P}_D^i(t_k)$ and *terminal controllers* K^i and

K^j where $(i,j) \in \mathbb{E}$. The terminal sets $\mathbb{X}_f^i$ and $\mathbb{X}_f^j$ are invariant sets under the terminal controllers K^i and K^j respectively.

For simplicity consider the case of two vehicles: (i=1, j=2). Consider the optimization problem $\mathcal{P}_D^1(t_k)$. Since, $u_{t_k}^{2,1}(t_{k+1})$ and $u_{t_k}^{2,2}(t_{k+1})$ are calculated from two different optimization problems, $\mathcal{P}_D^1(t_k)$ and $\mathcal{P}_D^2(t_k)$ respectively, they are not necessarily the same even if there are no uncertainties in the model. This means the following shifted optimization solutions may not be feasible:

$$\begin{aligned} u^1_{t_{k+1}}(t) &= \begin{cases} u^{1,1}_{t_k}(t) & t \in [t_{k+1}, t_k + T] \\ K^1 \bar{x}^{1,1}_{t_k}(t) & t \in [t_k + T, t_{k+1} + T] \end{cases} \\ u^2_{t_{k+1}}(t) &= \begin{cases} u^{2,1}_{t_k}(t) & t \in [t_{k+1-d}, t_k + T] \\ K^2 \bar{x}^{2,1}_{t_k}(t) & t \in [t_k + T, t_{k+1} + T] \end{cases} \end{aligned} \tag{6.6}$$

where, the parameter with bar denotes those corresponding to the *terminal controller*. However, since the communication is performed among vehicles it is possible to construct the following feasible shifted control trajectory by exchanging the predicted trajectory of vehicles:

$$u^{1,f}_{t_{k+1}}(t) = \begin{cases} u^{1,1}_{t_k}(t) & t \in [t_{k+1}, t_k + T] \\ K^1 \bar{x}^{1,1}_{t_k}(t) & t \in (t_k + T, t_{k+1} + T] \end{cases} \tag{6.7a}$$

$$u^{2,f}_{t_{k+1}}(t) = \begin{cases} u^{2,2}_{t_{k+1-d}}(t) & t \in [t_{k+1-d}, t_{k-d+1} + T] \\ K^2 \bar{x}^{2,1}_{t_k}(t) & t \in (t_{k+1-d} + T, t_{k+1} + T] \end{cases} \tag{6.7b}$$

where the superscript "f" denotes the feasible trajectories, e.g., $\boldsymbol{u}^{i,f}_{t_{k+1}}(t)$ is the feasible input of i^{th} vehicle at time t constructed at time t_{k+1} (from the optimized solution at time t_k).

To prove the feasibility of the candidate inputs (6.7), it is shown that the corresponding trajectories satisfy (4.5a) through (4.5e) and (5.31f) at time t_{k+1}.

Dynamics constraints (4.5a) *and* (4.5c): Since the dynamics of the vehicles are not coupled the trajectory corresponding to (6.7a) satisfies (4.5a) during $[t_{k+1}, t_k+T]$. During $(t_k+T, t_{k+1}+T]$ since the feedback input $\bar{\boldsymbol{u}}^{1,1}_{t_k} = K^1 \bar{\boldsymbol{x}}^{1,1}_{t_k}(t) \in \mathbb{U}^1$ is applied to vehicle 1, the trajectories satisfy (4.5a). Further, since it is assumed that there is no uncertainty in the dynamics, the initial condition in (4.5a) is satisfied.

Since the dynamics of the vehicles are not coupled the trajectory corresponding to (6.7b) satisfies (4.5c) during $[t_{k+1-d}, t_{k+1-d}+T]$. During $(t_{k+1-d}+T, t_{k+1}+T]$ since the feedback control $\bar{\boldsymbol{u}}^{2,2}_{t_k} = K^2 \bar{\boldsymbol{x}}^{2,1}_{t_k}(t)$ is applied to vehicle 2, the trajectories satisfy (4.5c). Further, the initial condition in (4.5c) is satisfied as the delayed trajectory $\boldsymbol{u}^{2,2}_{t_{k+1-d}}(t)$ is used which yields: $\boldsymbol{x}^{2,1}_{t_{k+1}}(t_{k+1-d}+T) = \boldsymbol{x}^{2,2}_{t_{k+1-d}}(t_{k+1-d}+T)$.

Saturation Constraints (4.5*b*) *and* (4.5*d*):

1) *Constraint* (4.5*b*): Since it is assumed that $\mathscr{P}^1_D(t_k)$ is feasible, then the trajectories generated by (6.7a) satisfy (4.5*b*) during $[t_{k+1}, t_k+T]$. Also, during $(t_k+T, t_{k+1}+T]$ the control input $\bar{\boldsymbol{u}}^{1,1}_{t_k} = K^1 \bar{\boldsymbol{x}}^{1,1}_{t_k}(t) \in \mathbb{U}^1$ is applied to vehicle 1 and since $\boldsymbol{x}^{1,1}_{t_k}(t_k+T) \in \mathbb{X}^1_f$ and

$\mathbb{X}_f^1$ is an invariant set under the control action $\bar{\boldsymbol{u}}_{t_k}^{1,1} = K^1 \bar{\boldsymbol{x}}_{t_k}^{1,1}(t)$, then for all times beyond the prediction horizon ($t \geq t_k + T$) the trajectory remains in $\mathbb{X}_f^1$, i.e., $\bar{\boldsymbol{x}}_{t_k}^{1,1}(t) \in \mathbb{X}_f^1$; on the other hand $\mathbb{X}_f^1 \subseteq \mathbb{X}^1$ which implies $\bar{\boldsymbol{x}}_{t_k}^{1,1}(t) \in \mathbb{X}^1$ for $t \in (t_k + T, t_{k+1} + T]$.

2) *Constraint* (4.5*d*): Likewise since it is assumed that $\mathscr{P}_D^2(t_{k-d+1})$ is feasible, then the trajectories generated by (6.7b) satisfy (4.5*d*) during $[t_{k+1-d}, t_{k-d+1} + T]$. Also, according to (6.7) during $(t_{k+1-d} + T, t_{k+1} + T]$ the control input $\bar{\boldsymbol{u}}_{t_k}^{2,1} = K^2 \bar{\boldsymbol{x}}_{t_k}^{2,1}(t) \in \mathbb{U}^2$ is applied to vehicle 2 and since $\boldsymbol{x}_{t_k}^{2,1}(t_k + T) \in \mathbb{X}_f^2$ and $\mathbb{X}_f^2$ is an invariant set under the control action $\bar{\boldsymbol{u}}_{t_k}^{2,1} = K^2 \bar{\boldsymbol{x}}_{t_k}^{2,1}(t)$, hence, for all times beyond the prediction horizon ($t \geq t_{k+1-d} + T$) the trajectory remains in $\mathbb{X}_f^2$; on the other hand $\mathbb{X}_f^2 \subseteq \mathbb{X}^2$ which implies $\bar{\boldsymbol{x}}_{t_k}^{2,1}(t) \in \mathbb{X}^2$ for $t \in (t_k + T, t_{k+1} + T]$.

Final constraints (4.5e): During $(t_k + T, t_{k+1} + T]$ the control input $\bar{\boldsymbol{u}}_{t_k}^{1,1} = K^1 \bar{\boldsymbol{x}}_{t_k}^{1,1}(t) \in \mathbb{U}^1$ is applied to vehicle 1 and since $\boldsymbol{x}_{t_k}^{1,1}(t_k + T) \in \mathbb{X}_f^1$ and $\mathbb{X}_f^1$ is an invariant set under control action $\bar{\boldsymbol{u}}_{t_k}^{1,1} = K^1 \bar{\boldsymbol{x}}_{t_k}^{1,1}(t)$, hence, for all times beyond the prediction horizon ($t \geq t_k + T$) the trajectory remains in $\mathbb{X}_f^1$ hence $\bar{\boldsymbol{x}}_{t_k}^{1,1}(t_{k+1} + T) \in \mathbb{X}_f^1$. A similar statement can be used to prove that $\bar{\boldsymbol{x}}_{t_k}^{2,1}(t_{k+1} + T) \in \mathbb{X}_f^2$.

Manoeuvrability constraint (5.31f): for satisfaction of this constraint it is enough to prove that $\left| \boldsymbol{u}_{t_k}^{1,1} - \boldsymbol{u}_{t_{k+1}}^{1,f} \right| \leq \mu$ for $\forall t \in [t_{k+1}, t_k + T]$; since $\left| \boldsymbol{u}_{t_k}^{1,1} - \boldsymbol{u}_{t_{k+1}}^{1,f} \right| = \mathbf{0}$ during this time and it

is assumed in general that $\mu \geq \mathbf{0}$ then input (6.7a) satisfies constraint (5.31f), (In general, it is assumed that the radius of $\mathbb{U}^1$ is much larger than μ).

Hence, the shifted input (6.7) satisfies constraints (4.5a) through (4.5e) and (5.31f) which implies the feasibility at time t_{k+1}. Then, since at time t_0 the problem $\mathcal{P}_D^1(t_0)$ and $\mathcal{P}_D^2(t_0)$ are feasible by induction it is concluded that all the future $\mathcal{P}_D^1(t_k)$ and $\mathcal{P}_D^2(t_k)$ are feasible.

The same analysis can be carried out for the general case with N_v vehicles for any $i \in \mathbb{V}$ and $(i,j) \in \mathbb{E}$ with the following shifted trajectory:

$$\boldsymbol{u}_{t_{k+1}}^{i,f}(t) = \begin{cases} \boldsymbol{u}_{t_k}^{i,i}(t) & t \in [t_{k+1}, t_k + T] \\ K^i \bar{\boldsymbol{x}}_{t_k}^{i,i}(t) & t \in (t_k + T, t_{k+1} + T] \end{cases} \tag{6.8a}$$

$$\boldsymbol{u}_{t_{k+1}}^{j,f}(t) = \begin{cases} \boldsymbol{u}_{t_{k-d+1}}^{j,j}(t) & t \in [t_{k-d+1}, t_{k-d+1} + T] \\ K^j \bar{\boldsymbol{x}}_{t_k}^{j,i}(t) & t \in (t_{k-d+1} + T, t_{k+1} + T] \end{cases} \tag{6.8b}$$

6.3. Stability Analysis

In this section, the constructed feasible solution (6.8) is used to provide the sufficient stability condition of the entire group.

Assumption 6.1: the linear realization of the nonlinear dynamics (2.1) is introduced as follows:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \tag{6.9}$$

where $A = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}=0, \boldsymbol{u}=0)}$, $B = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}=0, \boldsymbol{u}=0)}$. Also, assume A is stabilizable.

Lemma 6.1: Under *Assumption 6.1*, there exists a closed and convex *terminal set* $\mathbb{X}_f^i$ and *terminal feedback controller* K^i for each vehicle $i \in \mathbb{V}$ so that:

a) P^i is the unique, positive definite and symmetric solution of the following Lyapunov equation:

$$\tilde{A}^T P^i + P^i \tilde{A} = -\tilde{Q} \tag{6.10}$$

where $\tilde{A} = A + BK^i + aI$, $\tilde{Q} = Q + K^{i'} R K^i + (N_l^i + N_f^i)S$, I is the identity matrix and a is a positive value satisfying: $0 < a < -\lambda_{\max}(A + BK^i)$ and $\lambda_{\max}(A + BK^i)$ is the largest eigenvalue of $A + BK^i$.

b) $\forall x^i \in \mathbb{X}_f^i$:

$$\frac{d\left\|x^i\right\|_{P^i}^2}{dt} \leq -\left\|x^i\right\|_{Q+K^{i'}RK^i+(N_l^i+N_f^i)S}^2 \tag{6.11}$$

where N_l^i and N_f^i are the number of *leaders* and *followers* of vehicle *i* respectively. Also, P^i is the terminal matrix penalty of vehicle *i*. Equation (6.11) implies that $\mathbb{X}_f^i$ is an invariant set for vehicle *i* under the feedback terminal controller K^i.

Proof:

a) It is well known that the Lyapunov equation (6.10) has a unique, positive-definite and symmetric solution P^i if $\tilde{A}$ is stable (Hurwitz) and $\tilde{Q}$ is positive-definite. Since $Q > 0$, $R > 0$ and $S \geq 0$ then $\tilde{Q} > 0$ and it is assumed that K^i is a feedback controller in $\mathbb{X}_f^i$ and considering

$a < -\lambda_{\max}(A + BK^i)$, it follows that in $\mathbb{X}_f^i$ a unique positive-definite and symmetric P^i exists. The existence of K^i follows from the fact that A is stabilizable in $\mathbb{X}_f^i$.

b) By choosing $Q \leftarrow Q + (N_l^i + N_f^i)S$ the proof is the same as in [10].

Theorem 6.2 (Convergence): Assume that the communication graph is connected, i.e., each team member has at least one follower or leader which implies: $N_l^i + N_f^i \geq 1$. Then under the feasible solution of optimization problem $\mathcal{P}_D^i(t_k)$ we have:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \leq -\beta_1^i + \beta_2^i + \beta_3^i \tag{6.12}$$

where superscript * denotes the optimal value of the parameter and:

$$\begin{aligned}\beta_1^i = & \int_{t_k}^{t_{k+1}} \left(\left\| \underset{t_k}{x^{i,i}}(t) \right\|_Q^2 + \left\| \underset{t_k}{u^{i,i}}(t) \right\|_R^2 \right) dt \\ & + \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k-d}+T}^{t_{k-d+1}+T} \left(\left\| \underset{t_k}{x^{j,i}}(t) \right\|_Q^2 + \left\| \underset{t_k}{u^{j,i}}(t) \right\|_R^2 \right) dt + \int_{t_k}^{t_{k+1}} \left\| \underset{t_k}{x^{i,i}}(t) - \underset{t_{k-d}}{x^{j,j}}(t) \right\|_S^2 dt \right]\end{aligned} \tag{6.13}$$

and:

$$\beta_2^i = \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \underset{t_k}{u^{j,i}}(t) - \underset{t_{k+1}}{\bar{u}^{j,i}}(t) \right\|_R^2 + \left\| \underset{t_k}{x^{j,i}}(t) - \underset{t_{k+1}}{\bar{x}^{j,i}}(t) \right\|_{S+Q}^2 \right) dt \right] \tag{6.14}$$

and:

$$\beta_3^i = \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1}}^{t_{k-d}+T} \left\| \underset{t_{k-d}}{x^{j,j}}(t) - \underset{t_{k+1-d}}{x^{j,j}}(t) \right\|_S^2 dt + \int_{t_{k-d}+T}^{t_{k+1-d}+T} \left\| \underset{t_k}{x^{j,i}}(t) - \underset{t_{k+1-d}}{x^{j,j}}(t) \right\|_S^2 dt \right] \tag{6.15}$$

Proof: Consider the following optimal delayed cost functions for vehicle i at time instant t_k:

$$J_D^{i*}(\Gamma^i(t_k)) = \int_{t_k}^{t_k+T}\left(\left\|\boldsymbol{x}_{t_k}^{i,i}(t)\right\|_Q^2 + \left\|\boldsymbol{u}_{t_k}^{i,i}(t)\right\|_R^2\right)dt + \left\|\boldsymbol{x}_{t_k}^{i,i}(t_k+T)\right\|_{P^i}^2$$
$$+ \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_k+T}\left(\left\|\boldsymbol{x}_{t_k}^{j,i}(t)\right\|_Q^2 + \left\|\boldsymbol{u}_{t_k}^{j,i}(t)\right\|_R^2\right)dt + \left\|\boldsymbol{x}_{t_k}^{j,i}(t_k+T)\right\|_{P^j}^2 \tag{6.16}$$
$$+ \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k-d}+T}\left\|\boldsymbol{x}_{t_k}^{i,i}(t) - \boldsymbol{x}_{t_{k-d}}^{j,j}(t)\right\|_S^2 dt + \int_{t_{k-d}+T}^{t_k+T}\left\|\boldsymbol{x}_{t_k}^{i,i}(t) - \boldsymbol{x}_{t_k}^{j,i}(t)\right\|_S^2 dt$$

The feasible solution (6.8) is used to calculate the cost function at time instant t_{k+1} which is not optimal necessarily:

$$J_D^i(\Gamma^i(t_{k+1})) = \int_{t_{k+1}}^{t_{k+1}+T}\left(\left\|\boldsymbol{x}_{t_{k+1}}^{i,f}(t)\right\|_Q^2 + \left\|\boldsymbol{u}_{t_{k+1}}^{i,f}(t)\right\|_R^2\right)dt + \left\|\boldsymbol{x}_{t_k}^{i,f}(t_{k+1}+T)\right\|_{P^i}^2$$
$$+ \sum_{j|(i,j)\in\mathbb{E}}\left[\int_{t_{k+1-d}+T}^{t_{k+1}+T}\left(\left\|\boldsymbol{x}_{t_{k+1}}^{j,f}(t)\right\|_Q^2 + \left\|\boldsymbol{u}_{t_{k+1}}^{j,f}(t)\right\|_R^2\right)dt + \left\|\boldsymbol{x}_{t_k}^{j,f}(t_{k+1}+T)\right\|_{P^j}^2\right]$$
$$+ \sum_{j|(i,j)\in\mathbb{E}}\left[\int_{t_{k+1}}^{t_{k+1-d}+T}\left\|\boldsymbol{x}_{t_{k+1}}^{i,f}(t) - \boldsymbol{x}_{t_{k+1-d}}^{j,j}(t)\right\|_S^2 dt + \int_{t_{k+1-d}+T}^{t_{k+1}+T}\left\|\boldsymbol{x}_{t_{k+1}}^{i,f}(t) - \boldsymbol{x}_{t_{k+1}}^{j,f}(t)\right\|_S^2 dt\right]$$
$$= \int_{t_{k+1}}^{t_k+T}\left(\left\|\boldsymbol{x}_{t_k}^{i,i}(t)\right\|_Q^2 + \left\|\boldsymbol{u}_{t_k}^{i,i}(t)\right\|_R^2\right)dt + \int_{t_k+T}^{t_{k+1}+T}\left(\left\|\bar{\boldsymbol{x}}_{t_k}^{i,i}(t)\right\|_Q^2 + \left\|\bar{\boldsymbol{u}}_{t_k}^{i,i}(t)\right\|_R^2\right)dt + \left\|\bar{\boldsymbol{x}}_{t_k}^{i,i}(t_{k+1}+T)\right\|_{P^i}^2 \tag{6.17}$$
$$+ \sum_{j|(i,j)\in\mathbb{E}}\left[\int_{t_{k+1-d}+T}^{t_{k+1}+T}\left(\left\|\bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t)\right\|_Q^2 + \left\|\bar{\boldsymbol{u}}_{t_{k+1}}^{j,i}(t)\right\|_R^2\right)dt + \left\|\bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t_{k+1}+T)\right\|_{P^j}^2\right]$$
$$+ \sum_{j|(i,j)\in\mathbb{E}}\left[\int_{t_{k+1}}^{t_{k+1-d}+T}\left\|\boldsymbol{x}_{t_k}^{i,i}(t) - \boldsymbol{x}_{t_{k+1-d}}^{j,j}(t)\right\|_S^2 dt + \int_{t_{k+1-d}+T}^{t_k+T}\left\|\boldsymbol{x}_{t_k}^{i,i}(t) - \bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t)\right\|_S^2 dt\right.$$
$$\left. + \int_{t_k+T}^{t_{k+1}+T}\left\|\bar{\boldsymbol{x}}_{t_k}^{i,i}(t) - \bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t)\right\|_S^2 dt\right]$$

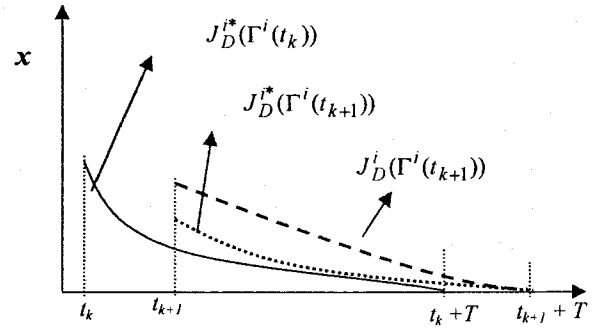

Figure 6.1: Schematic representation of the trajectories used to compute the three used cost functions in analyzing the stability.

Figure 6.1 shows schematically the three trajectories used to calculate the cost functions (6.16) and (6.17). Following the fact that some portions of (6.16) and (6.17) overlap and using the optimality property yield:

$$J_D^{i*}(\Gamma^i(t_{k+1}))-J_D^{i*}(\Gamma^i(t_k))\le J_D^{i}(\Gamma^i(t_{k+1}))-J_D^{i*}(\Gamma^i(t_k))= \tag{6.18a}$$

$$-\int_{t_k}^{t_{k+1}}\left(\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)\right\|_Q^2+\left\|\underset{t_k}{\boldsymbol{u}^{i,i}}(t)\right\|_R^2\right)dt-\sum_{j|(i,j)\in\mathbb{E}}\int_{t_{k-d+1}+T}^{t_{k-d+1}+T}\left(\left\|\underset{t_k}{\boldsymbol{x}^{j,i}}(t)\right\|_Q^2+\left\|\underset{t_k}{\boldsymbol{u}^{j,i}}(t)\right\|_R^2\right)dt$$
$$-\sum_{j|(i,j)\in\mathbb{E}}\int_{t_k}^{t_{k+1}}\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)-\underset{t_{k-d}}{\boldsymbol{x}^{j,j}}(t)\right\|_S^2 dt \tag{6.18b}$$

$$\int_{t_k+T}^{t_{k+1}+T}\left(\left\|\underset{t_k}{\bar{\boldsymbol{x}}^{i,i}}(t)\right\|_Q^2+\left\|\underset{t_k}{\bar{\boldsymbol{u}}^{i,i}}(t)\right\|_R^2\right)dt+\sum_{j|(i,j)\in\mathbb{E}}\int_{t_k+T}^{t_{k+1}+T}\left(\left\|\underset{t_{k+1}}{\bar{\boldsymbol{x}}^{j,i}}(t)\right\|_Q^2+\left\|\underset{t_{k+1}}{\bar{\boldsymbol{u}}^{j,i}}(t)\right\|_R^2\right)dt$$
$$+\sum_{j|(i,j)\in\mathbb{E}}\int_{t_k+T}^{t_{k+1}+T}\left\|\underset{t_k}{\bar{\boldsymbol{x}}^{i,i}}(t)-\underset{t_{k+1}}{\bar{\boldsymbol{x}}^{j,i}}(t)\right\|_S^2 dt \tag{6.18c}$$

$$+\left\|\underset{t_k}{\bar{\boldsymbol{x}}^{i,i}}(t_{k+1}+T)\right\|_{P^i}^2+\sum_{j|(i,j)\in\mathbb{E}}\left\|\underset{t_{k+1}}{\bar{\boldsymbol{x}}^{j,i}}(t_{k+1}+T)\right\|_{P^j}^2$$
$$-\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t_k+T)\right\|_{P^i}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\underset{t_k}{\boldsymbol{x}^{j,i}}(t_k+T)\right\|_{P^j}^2 \tag{6.18d}$$

$$
\begin{aligned}
&\int_{t_{k+1}}^{t_k+T}\left(\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)\right\|_Q^2+\left\|\underset{t_k}{\boldsymbol{u}^{i,i}}(t)\right\|_R^2\right)dt+\sum_{j|(i,j)\in\mathbb{E}}\int_{t_{k+1-d}+T}^{t_k+T}\left(\left\|\underset{t_{k+1}}{\bar{\boldsymbol{x}}^{j,i}}(t)\right\|_Q^2+\left\|\underset{t_{k+1}}{\bar{\boldsymbol{u}}^{j,i}}(t)\right\|_R^2\right)dt\\
&+\sum_{j|(i,j)\in\mathbb{E}}\left[\int_{t_{k+1}}^{t_{k+1-d}+T}\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)-\underset{t_{k+1-d}}{\boldsymbol{x}^{j,j}}(t)\right\|_S^2dt+\int_{t_{k+1-d}+T}^{t_k+T}\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)-\underset{t_{k+1}}{\bar{\boldsymbol{x}}^{j,i}}(t)\right\|_S^2dt\right]\\
&-\int_{t_{k+1}}^{t_k+T}\left(\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)\right\|_Q^2+\left\|\underset{t_k}{\boldsymbol{u}^{i,i}}(t)\right\|_R^2\right)dt-\sum_{j|(i,j)\in\mathbb{E}}\int_{t_{k+1-d}+T}^{t_k+T}\left(\left\|\underset{t_k}{\boldsymbol{x}^{j,i}}(t)\right\|_Q^2+\left\|\underset{t_k}{\boldsymbol{u}^{j,i}}(t)\right\|_R^2\right)dt\\
&-\sum_{j|(i,j)\in\mathbb{E}}\int_{t_{k+1}}^{t_{k-d}+T}\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)-\underset{t_{k-d}}{\boldsymbol{x}^{j,j}}(t)\right\|_S^2dt-\int_{t_{k-d}+T}^{t_k+T}\left\|\underset{t_k}{\boldsymbol{x}^{i,i}}(t)-\underset{t_k}{\boldsymbol{x}^{j,i}}(t)\right\|_S^2dt
\end{aligned}
\tag{6.18e}
$$

From (6.11) for vehicle *i*, it is concluded that:

$$
\frac{d\left\|\bar{\boldsymbol{x}}^i\right\|_{P^i}^2}{dt}\le-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i+(N_l^i+N_f^i)S}^2\le-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i+N_l^iS}^2 \tag{6.19}
$$

Likewise, for all leaders *j* of vehicle *i*:

$$
\frac{d\left\|\bar{\boldsymbol{x}}^j\right\|_{P^j}^2}{dt}\le-\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j+(N_l^j+N_f^j)S}^2\le-\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j+S}^2 \tag{6.20}
$$

Since it is assumed in general that $N_l^j+N_f^j\ge 1$ then (6.20) holds. Then the summation of (6.19) and (6.20) (for all leaders) yields:

$$
\begin{aligned}
&\frac{d\left\|\bar{\boldsymbol{x}}^i\right\|_{P^i}^2}{dt}+\sum_{j|(i,j)\in\mathbb{E}}\frac{d\left\|\bar{\boldsymbol{x}}^j\right\|_{P^j}^2}{dt}\le-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i+N_l^iS}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j+S}^2=\\
&-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i}^2-\left\|\bar{\boldsymbol{x}}^i\right\|_{N_l^iS}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_S^2=\\
&-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j}^2-N_l^i\left\|\bar{\boldsymbol{x}}^i\right\|_S^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_S^2=\\
&-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j}^2-\sum_{j|(i,j)\in\mathbb{E}}\left(\left\|\bar{\boldsymbol{x}}^i\right\|_S^2+\left\|\bar{\boldsymbol{x}}^j\right\|_S^2\right)\le\\
&-\left\|\bar{\boldsymbol{x}}^i\right\|_{Q+K^{i'}RK^i}^2-\sum_{j|(i,j)\in\mathbb{E}}\left\|\bar{\boldsymbol{x}}^j\right\|_{Q+K^{j'}RK^j}^2-\sum_{j|(i,j)\in\mathbb{E}}\left(\left\|\bar{\boldsymbol{x}}^i-\bar{\boldsymbol{x}}^j\right\|_S^2\right)
\end{aligned}
\tag{6.21}
$$

The last inequality follows from the fact that $\|a\|_s^2 + \|b\|_s^2 \geq \|a-b\|_s^2$. Then integrating both side of inequality (6.21) yields:

$$\int_{t_k+T}^{t_{k+1}+T} \left(\frac{d\left\|\bar{x}^i\right\|_{P^i}^2}{dt} + \sum_{j|(i,j)\in\mathbb{E}} \frac{d\left\|\bar{x}^j\right\|_{P^j}^2}{dt} \right) dt \leq$$
$$-\int_{t_k+T}^{t_{k+1}+T} \left\|\bar{x}^i\right\|_{Q+K^{i'}RK^i}^2 dt - \int_{t_k+T}^{t_{k+1}+T} \sum_{j|(i,j)\in\mathbb{E}} \left\|\bar{x}^j\right\|_{Q+K^{j'}RK^j}^2 dt - \int_{t_k+T}^{t_{k+1}+T} \sum_{j|(i,j)\in\mathbb{E}} \left(\left\|\bar{x}^i - \bar{x}^j\right\|_S^2 \right) dt \tag{6.22}$$

Then, since for $\forall \bar{x}^i \in \mathbb{X}_f^i$, then $\bar{u}^i = K^i \bar{x}^i$ and hence $\left\|\bar{x}^i\right\|_{Q+K^{i'}RK^i}^2 = \left\|\bar{x}^i\right\|_Q^2 + \left\|\bar{u}^i\right\|_R^2$. Therefore:

$$\left\|\bar{x}^i(t_{k+1}+T)\right\|_{P^i}^2 + \sum_{j|(i,j)\in\mathbb{E}} \left\|\bar{x}^j(t_{k+1}+T)\right\|_{P^j}^2 - \left\|\bar{x}^i(t_k+T)\right\|_{P^i}^2 - \sum_{j|(i,j)\in\mathbb{E}} \left\|\bar{x}^j(t_k+T)\right\|_{P^j}^2 \leq$$
$$-\int_{t_k+T}^{t_{k+1}+T} \left(\left\|\bar{x}^i\right\|_Q^2 + \left\|\bar{u}^i\right\|_R^2 \right) dt - \int_{t_k+T}^{t_{k+1}+T} \sum_{j|(i,j)\in\mathbb{E}} \left(\left\|\bar{x}^j\right\|_Q^2 + \left\|\bar{u}^j\right\|_R^2 \right) dt - \int_{t_k+T}^{t_{k+1}+T} \sum_{j|(i,j)\in\mathbb{E}} \left(\left\|\bar{x}^i - \bar{x}^j\right\|_S^2 \right) dt \tag{6.23}$$

And this means the summation of terms (6.18c) and (6.18d) is non-positive and can be removed from inequality (6.18) which results in:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \leq \tag{6.24a}$$

$$-\int_{t_k}^{t_{k+1}} \left(\left\| x_{t_k}^{i,i}(t) \right\|_Q^2 + \left\| u_{t_k}^{i,i}(t) \right\|_R^2 \right) dt - \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_{k-d+1}+T} \left(\left\| x_{t_k}^{j,i}(t) \right\|_Q^2 + \left\| u_{t_k}^{j,i}(t) \right\|_R^2 \right) dt$$
$$- \sum_{j|(i,j)\in\mathbb{E}} \int_{t_k}^{t_{k+1}} \left\| x_{t_k}^{i,i}(t) - x_{t_{k-d}}^{j,j}(t) \right\|_S^2 dt \tag{6.24b}$$

$$\sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \bar{x}_{t_{k+1}}^{j,i}(t) \right\|_Q^2 + \left\| \bar{u}_{t_{k+1}}^{j,i}(t) \right\|_R^2 \right) dt$$
$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1}}^{t_{k+1-d}+T} \left\| x_{t_k}^{i,i}(t) - x_{t_{k+1-d}}^{j,j}(t) \right\|_S^2 dt + \int_{t_{k+1-d}+T}^{t_k+T} \left\| x_{t_k}^{i,i}(t) - \bar{x}_{t_{k+1}}^{j,i}(t) \right\|_S^2 dt \right] \tag{6.24c}$$

$$- \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \underset{t_k}{x^{j,i}}(t) \right\|_Q^2 + \left\| \underset{t_k}{u^{j,i}}(t) \right\|_R^2 \right) dt$$

$$- \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k+1}}^{t_{k-d}+T} \left\| \underset{t_k}{x^{i,i}}(t) - \underset{t_{k-d}}{x^{j,j}}(t) \right\|_S^2 dt - \int_{t_{k-d}+T}^{t_k+T} \left\| \underset{t_k}{x^{i,i}}(t) - \underset{t_k}{x^{j,i}}(t) \right\|_S^2 dt \tag{6.24d}$$

Using the fact that $\| a \|_S^2 - \| b \|_S^2 \le \| a - b \|_S^2$ yields:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \le \tag{6.25a}$$

$$- \int_{t_k}^{t_{k+1}} \left(\left\| \underset{t_k}{x^{i,i}}(t) \right\|_Q^2 + \left\| \underset{t_k}{u^{i,i}}(t) \right\|_R^2 \right) dt$$

$$- \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k-d}+T}^{t_{k-d+1}+T} \left(\left\| \underset{t_k}{x^{j,i}}(t) \right\|_Q^2 + \left\| \underset{t_k}{u^{j,i}}(t) \right\|_R^2 \right) dt + \int_{t_k}^{t_{k+1}} \left\| \underset{t_k}{x^{i,i}}(t) - \underset{t_{k-d}}{x^{j,j}}(t) \right\|_S^2 dt \right] \tag{6.25b}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \underset{t_k}{u^{j,i}}(t) - \underset{t_{k+1}}{\bar{u}^{j,i}}(t) \right\|_R^2 + \left\| \underset{t_k}{x^{j,i}}(t) - \underset{t_{k+1}}{\bar{x}^{j,i}}(t) \right\|_{S+Q}^2 \right) dt \right] \tag{6.25c}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1}}^{t_{k-d}+T} \left\| \underset{t_{k-d}}{x^{j,j}}(t) - \underset{t_{k+1-d}}{x^{j,j}}(t) \right\|_S^2 dt + \int_{t_{k-d}+T}^{t_{k+1-d}+T} \left\| \underset{t_k}{x^{j,i}}(t) - \underset{t_{k+1-d}}{x^{j,j}}(t) \right\|_S^2 dt \right] \tag{6.25d}$$

Or simply:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \le -\beta_1^i + \beta_2^i + \beta_3^i \tag{6.26}$$

where β_1^i = - terms in (6.25b), β_2^i = terms in (6.25c), β_3^i = terms in (6.25d), are class-$\mathcal{K}$ functions (positive definite).

Theorem 6.3. There exist functions $\gamma_1(\cdot)$, $\theta_1(\cdot)$ and $\rho_1(\cdot) \in \mathcal{K}_\infty$ and the positive constant c_1 such that for $\forall i \in \mathbb{V}$:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \le \left[-\gamma_1 \left(\| x^i(t_k) \| \right) + \theta_1 \left(\| x^i(t_k) \| \right) \right] + \sum_{(i,j)\in\mathbb{E}} \rho_1 \left(2 \| x_{t_{k-d}}^{j,j}(\cdot) \| \right) + c_1 \tag{6.27}$$

Proof: From Theorem *6.2* we have:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \le -\left\| \boldsymbol{x}_{t_k}^{i,i}(t_k) \right\|_Q^2 \tag{6.28a}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \boldsymbol{u}_{t_k}^{j,i}(t) - \bar{\boldsymbol{u}}_{t_{k+1}}^{j,i}(t) \right\|_R^2 + \left\| \boldsymbol{x}_{t_k}^{j,i}(t) - \bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t) \right\|_{S+Q}^2 \right) dt \right] \tag{6.28b}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\lambda_{\max}(S) \mu' \mu (T - d\delta + \delta) \right]$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k-d}+T}^{t_{k+1-d}+T} \left\| \boldsymbol{x}_{t_k}^{j,i}(t) - \boldsymbol{x}_{t_{k-d}}^{j,j}(t_{k-d}+T) + \boldsymbol{x}_{t_{k-d}}^{j,j}(t_{k-d}+T) - \boldsymbol{x}_{t_{k+1-d}}^{j,j}(t) \right\|_S^2 dt \right] \tag{6.28c}$$

Then considering (6.4):

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \le -\left\| \boldsymbol{x}^i(t_k) \right\|_Q^2 \tag{6.29a}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \boldsymbol{u}_{t_k}^{j,i}(t) - \bar{\boldsymbol{u}}_{t_{k+1}}^{j,i}(t) \right\|_R^2 + \left\| \boldsymbol{x}_{t_k}^{j,i}(t) - \bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t) \right\|_{S+Q}^2 \right) dt \right] \tag{6.29b}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\lambda_{\max}(S) \mu' \mu (T - d\delta + \delta) \right]$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k-d}+T}^{t_{k+1-d}+T} \left(\left\| \boldsymbol{x}_{t_k}^{j,i}(t) \right\|_S^2 + \left\| \boldsymbol{x}_{t_{k-d}}^{j,j}(t_{k-d}+T) \right\|_S^2 \right) dt + (\delta^3 \lambda_{\max}(S) \times \mu' \mu) \right] \tag{6.29c}$$

Hence:

$$J_D^{i*}(\Gamma^i(t_{k+1})) - J_D^{i*}(\Gamma^i(t_k)) \le -\left\| \boldsymbol{x}^i(t_k) \right\|_Q^2 + \delta \left\| \boldsymbol{x}_{t_{k-d}}^{j,j}(t_{k-d}+T) \right\|_S^2 \tag{6.30a}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \boldsymbol{u}_{t_k}^{j,i}(t) - \bar{\boldsymbol{u}}_{t_{k+1}}^{j,i}(t) \right\|_R^2 + \left\| \boldsymbol{x}_{t_k}^{j,i}(t) - \bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t) \right\|_{S+Q}^2 \right) dt \right] \tag{6.30b}$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_{k+1-d}+T} \left\| \boldsymbol{x}_{t_k}^{j,i}(t) \right\|_S^2 dt$$

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\lambda_{\max}(S) \mu' \mu (T - d\delta + \delta) \right] + \sum_{j|(i,j)\in\mathbb{E}} \left[(\delta^3 \lambda_{\max}(S) \times \mu' \mu) \right] \tag{6.30c}$$

The terms in (6.30b) are increasing functions of the initial condition of optimization problem $\mathcal{P}_D^i(t_k)$ and vanish if the initial conditions of optimization problem $\mathcal{P}_D^i(t_k)$ vanish; then there exist functions $\theta_1(\cdot)$ and $\theta_2(\cdot) \in \mathcal{K}_\infty$ such that:

$$+ \sum_{j|(i,j)\in\mathbb{E}} \left[\int_{t_{k+1-d}+T}^{t_k+T} \left(\left\| \boldsymbol{u}_{t_k}^{j,i}(t) - \bar{\boldsymbol{u}}_{t_{k+1}}^{j,i}(t) \right\|_R^2 + \left\| \boldsymbol{x}_{t_k}^{j,i}(t) - \bar{\boldsymbol{x}}_{t_{k+1}}^{j,i}(t) \right\|_{S+Q}^2 \right) dt \right]$$
$$+ \sum_{j|(i,j)\in\mathbb{E}} \int_{t_{k-d}+T}^{t_{k+1-d}+T} \left\| \boldsymbol{x}_{t_k}^{j,i}(t) \right\|_S^2 dt \le \theta_1 \left(\| \boldsymbol{x}^i(t_k) \| \right) + \sum_{j|(i,j)\in\mathbb{E}} \theta_2 \left(\| \boldsymbol{x}_{t_{k-d}}^{j,j}(\cdot) \| \right) \tag{6.31}$$

Substituting into (6.30) yields (6.27), where c_1 = terms in (6.30c), $\gamma_1(s) = \lambda_{\max}(Q) \| s \|^2$ and $\rho_1(s) = \lambda_{\max}(S) \| s \|^2 \delta + \theta_2(s)$.

Definition 6.1: A system is called input-to-state practically stable (ISpS), if there exist $\beta(\cdot,t) \in \mathcal{KL}$, $\gamma(\cdot) \in \mathcal{K}_\infty$ and positive constant c such that:

$$\| \boldsymbol{x}(t) \| \le \beta(\| \boldsymbol{x}_0 \|, t) + \gamma(\| \boldsymbol{u}(t) \|_\infty) + c \tag{6.32}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u}(\cdot) \in \mathbb{R}^m$.

Lemma 6.2: A system is input-to-state practically stable (ISpS), if there exist an ISpS Lyapunov function $V(\boldsymbol{x}, \pi(\cdot)) : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}$ with $\alpha_1(\cdot)$, $\alpha_2(\cdot)$, $\gamma(\cdot)$ and $\sigma(\cdot) \in \mathcal{K}_\infty$, $\rho(\cdot) \in \mathcal{K}$ and a positive constant c such that:

$$\alpha_1(\| \boldsymbol{x} \|) \le V(\boldsymbol{x}, \pi(\cdot)) \le \alpha_2(\| \boldsymbol{x} \|) + \sigma(\| \pi(\cdot) \|) \tag{6.33}$$

and:

$$V(\boldsymbol{x}(k+1), \pi_{k+1}(\cdot)) - V(\boldsymbol{x}(k), \pi_k(\cdot)) \le -\gamma(\| \boldsymbol{x} \|) + \rho(\| \pi_k(\cdot) \|) + c \tag{6.34}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\pi(\cdot) \in \mathbb{R}^m$.

Proof: See [179].

Remark 6.1: If $\sigma(\cdot)=0$ and c=0, then Lemma 6.2 reduces to the sufficient condition for the input-to-state stability (ISS) argument.

The following theorem summarizes the results to cast the stability problem into the conditions of Lemma 6.2.

Theorem 6.4 (Stability): All of the team members are input-to-state practically stable under the solution of the optimization problems $\mathscr{P}_D^i(t_k)$ for $\forall i \in \mathbb{V}$ if there exist a function $\gamma(\cdot) \in \mathscr{K}_\infty$ such that:

$$-\gamma_1(s)+\theta_1(s) \le -\gamma(s) \tag{6.35}$$

where $\gamma_1(s)=\lambda_{\max}(Q)\|s\|^2$ and $\theta_1(\cdot)$ is associated with bound in (6.31).

Proof: Let $\tilde{\boldsymbol{x}}(t_k)=[\boldsymbol{x}^1(t_k),\boldsymbol{x}^2(t_k),.....,\boldsymbol{x}^{Nv}(t_k)]'$ and $\tilde{\boldsymbol{u}}(t_k)=[\boldsymbol{u}^1(t_k),\boldsymbol{u}^2(t_k),.....,\boldsymbol{u}^{Nv}(t_k)]'$ then:

$$J_\Sigma(\tilde{\boldsymbol{x}}(t_k),\tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot),\tilde{\boldsymbol{u}}_{t_k}(\cdot))=\sum_{i\in\mathbb{V}} J_D^i(\Gamma^i(t_{k+1})) \tag{6.36}$$

Then:

$$J_\Sigma^*(\tilde{\boldsymbol{x}}(t_k),\tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot))=\sum_{i\in\mathbb{V}} J_D^{i^*}(\Gamma^i(t_{k+1})) \tag{6.37}$$

Then, the sum of optimal value function $J_\Sigma^*(\tilde{\boldsymbol{x}}(t_k),\tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot))$ can be considered as an ISpS Lyapunov function for the overall system of multiple vehicles; because first: since $\boldsymbol{x}_{t_k}^{i,i}(t_k)=\boldsymbol{x}^i(t_k)$

$$J_\Sigma^*(\tilde{\boldsymbol{x}}(t_k),\tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot)) \ge \sum_{i\in\mathbb{V}} \left\|\boldsymbol{x}^i(t_k)\right\|_Q^2 \tag{6.38}$$

Then considering the fact that $\|\boldsymbol{x}^i(t_k)\|_Q^2 \ge \lambda_{\min}(Q)\|\boldsymbol{x}^i(t_k)\|^2$ then there exist $\alpha_1(\cdot)\in\mathscr{K}_\infty$ such that: $\alpha_1(\|\tilde{\boldsymbol{x}}(t_k)\|) \le \sum_{i\in\mathbb{V}} \left\|\boldsymbol{x}^i(t_k)\right\|_Q^2$. Also, considering $J_\Sigma^*(\mathbf{0},\mathbf{0})=0$ and

the fact that $J^*_{\sum}(\tilde{\boldsymbol{x}}(t_k), \tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot)) > 0$ for $\forall \tilde{\boldsymbol{x}}(t_k) \neq \mathbf{0}$ and $\forall \tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot) \neq \mathbf{0}$, it can be concluded that there exist $\alpha_2(\cdot)$, and $\sigma(\cdot) \in \mathcal{K}_\infty$ so that:

$$J^*_{\sum}(\tilde{\boldsymbol{x}}(t_k), \tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot)) \leq \alpha_2(\tilde{\boldsymbol{x}}(t_k)) + \sigma(\tilde{\boldsymbol{x}}_{t_{k-d}}(\cdot)) \tag{6.39}$$

Then the first condition of Lemma 6.2 is satisfied. Second, considering the condition (6.35) and Theorem *6.3*, the second condition of Lemma 6.2 is straightforward. Hence, according to Lemma 6.2 all of the vehicles are input-to-state practically stable.

Remark 6.2: Existence of $\gamma(\cdot) \in \mathcal{K}_\infty$ in Theorem *6.4* requires that the function $\theta_1(s)$ be sufficiently small; then considering (6.31) it follows that if at the terminal set $\mathbb{X}^i_f$, the solution of DRHC optimization problem $\mathcal{P}^i_D(t_k)$ for $\forall i \in \mathbb{V}$ is close enough to the solution of terminal controller then a small $\theta_1(s)$ can result. On the other hand, since $\gamma_1(s) = \lambda_{\max}(Q) \| s \|^2$ then if $\theta_1(s) \leq \lambda_{\max}(Q) \| s \|^2$ the condition in Theorem *6.4* is satisfied.

6.4. Summary

In this Chapter, it is shown that using the proposed delayed DRHC of Chapter 4 always a lower bound on the performance can be guaranteed. Also, under quite mild conditions always there exists a feasible solution to the optimization problems. Further, employing the input-to-state practical stability (ISpS) argument, sufficient stability condition for the proposed delayed DRHC is derived.

Chapter 7. Bandwidth Allocation Algorithm

Technically speaking, the large communication delays happen due to limited communication bandwidth of the low performance communication channel employed in the faulty conditions. Hence, in this chapter, it is desired to design a bandwidth allocation algorithm which enables each vehicle to distribute its available communication bandwidth to the neighbouring vehicles (followers) so that the teaming performance is optimized. Then the proposed bandwidth allocation algorithm is integrated with the proposed delayed DRHC architecture of Chapter 4.

According to the discussion on the performance analysis of Chapter 6 (equation (6.3)), it is mentioned that the mismatch between the neighbour's delayed trajectories and the actual trajectories is the main source of poor performance. Hence, the key idea with the proposed bandwidth allocation algorithm is that at each time step, each vehicle measures the mismatches between its delayed and updated trajectories; this mismatch causes an error in the solution of neighbouring vehicle's optimization problem. Then, based on minimizing a bound on the error in the cost function of all neighbouring vehicles, each vehicle allocates its available communication bandwidth to its neighbouring vehicles. In fact, the available communication bandwidth is employed to reduce the effect of mismatch which leads to poor performance.

The resource allocation problems are normally performed in a centralized manner when the global information is available to a central decision maker. Due to the decentralized nature of the problem in this thesis the proposed bandwidth allocation

algorithm must be decentralized and rely on the local information which makes the problem more challenging.

Few works in the context of DRHC have addressed the bandwidth allocation algorithm for DRHC. In [120], the problem of optimal formation control under limited communication capacity is considered for two-vehicle formations. They demonstrate that, in the case of noise-free communication, bit-limited exchanges can reduce the performance of the fleet by as much as 20% when compared to the case of unlimited communication capacity. However, the work in [120] does not investigate dynamic bandwidth allocation. Also, another work in [168] studies the effect of limited communication bandwidth on the control of multiple miniature robots; a resource allocation algorithm is proposed to dynamically assign the available communication bandwidth; the team control in [168] is accomplished in a centralized fashion, contrary to the present work.

In this chapter by focusing on the systems with linear dynamics an analytical bound on the cost function due to mismatch between the actual and the delayed trajectories is calculated. Then a bandwidth allocation algorithm is developed which works based on minimizing the bound on the cost function.

7.1. The Bandwidth Allocation Scheme

For bandwidth allocation purpose, it is desired to find the relationship between the DRHC performance and the communication bandwidth, and then allocate the bandwidth to each neighbour so that the overall teaming performance is optimized.

The proposed bandwidth allocation algorithm works based on measuring the current mismatches between the delayed and the updated trajectories and improving the future undesirable behaviour caused due to current mismatches; Figure 7.1 shows a schematic sketch of the approach.

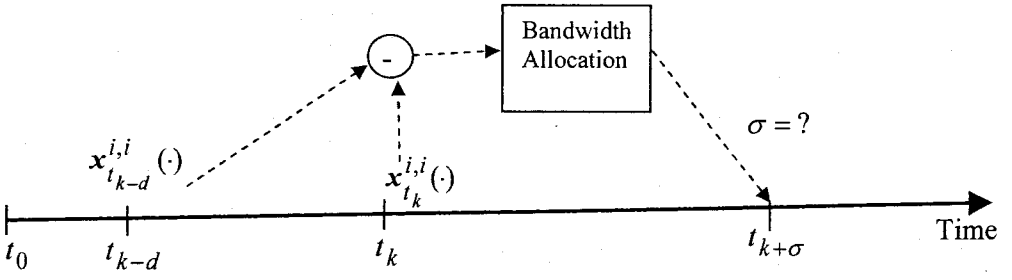

Figure 7.1: Performance enhancement using a Feed-forward loop.

As seen in Figure 7.1, the mismatch is measured using the available trajectories at time t_k; any mismatch implies an error in future behaviour; hence, the mismatch is used by a bandwidth allocation algorithm to reduce the effect of such error. Assuming the perfect optimization, availability of feasible solutions at all time, no communication noise, no model uncertainty, the mismatch between the delayed and the actual trajectories is the only source of poor performance. Hence, the main idea here is to allocate the available bandwidth based on such mismatches.

From now on, the index i is used for the vehicle allocating its communication bandwidth to its neighbours; also the index j is used for neighbours to which the vehicle i allocates the bandwidth for transmitting the information.

A more detailed graphical sketch of this approach is shown in Figure 7.2 (considering the only neighbour of vehicle j is vehicle i). As seen from Figure 7.2 at time t_k, the vehicle i computes and uses the updated trajectory $x^{i,i}_{t_k}(\cdot)$ while neighbour j does

not have access to this trajectory due to communication delay and uses the delayed trajectory of i, i.e., $x^{i,i}_{t_{k-d}}(\cdot)$ where $d \in \mathbb{R}$. The mismatch between these two trajectories is denoted by $\varepsilon^{i,j}_{t_k}(\cdot) = x^{i,i}_{t_k}(\cdot) - x^{i,i}_{t_{k-d}}(\cdot)$. The mismatch $\varepsilon^{i,j}_{t_k}(\cdot)$ causes an error in input of the optimization problem $\mathcal{P}^j_D(t_k)$ and consequently will lead to an error in desired solution of $\mathcal{P}^j_D(t_k)$, the error in solution of $\mathcal{P}^j_D(t_k)$ is denoted by $\Delta x^{j,i}_{t_k}(\cdot)$: the error in solution of optimization problem $\mathcal{P}^j_D(t_k)$ due to mismatch between delayed and updated trajectory of neighbour i at time t_k.

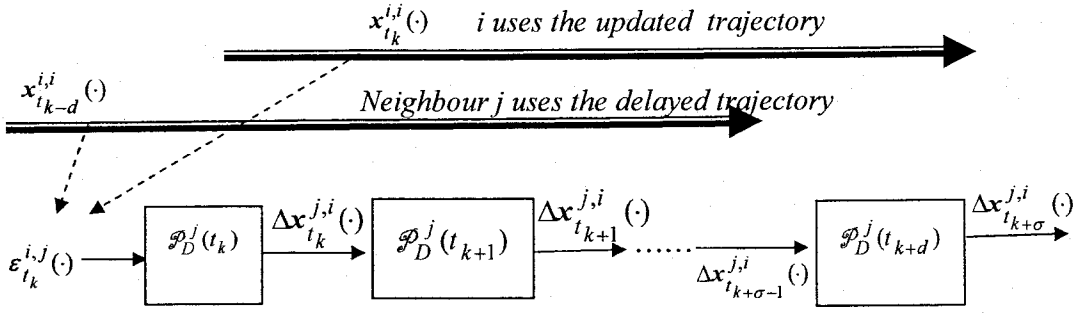

Figure 7.2: Error propagation due to mismatch between delayed and updated trajectory of vehicle i, as seen by vehicle j.

Likewise $\Delta x^{j,i}_{t_k}(\cdot)$ is served as an error in the input of the next step optimization problem $\mathcal{P}^j_D(t_{k+1})$ which leads to error $\Delta x^{j,i}_{t_{k+1}}(\cdot)$; this error can propagate to the future optimization problems. Note that finally $\Delta x^{j,i}_{t_{k+\sigma}}(\cdot)$ must be calculated for all neighbours of vehicle i where $\sigma \in \mathbb{R}$.

In Figure 7.2 only the neighbour j is shown for problem presentation purpose. The vehicle i must calculate $\Delta x^{j,i}_{t_{k+\sigma}}(\cdot)$ for all neighbours j (where $(j,i) \in \mathbb{E}$). Allocate the

available bandwidth to neighbours bases on $\Delta x^{j,i}_{t_{k+\sigma}}(\cdot)$ from each neighbour j so that the effect of $\Delta x^{j,i}_{t_{k+\sigma}}(\cdot)$ on the cost function is minimized. Then this approach is presented in the following steps:

1) Calculate $\Delta x^{j,i}_{t_k}(\cdot)$, the effect of $\varepsilon^{i,j}_{t_k}(\cdot)$ on the solution of $\mathcal{P}^j_D(t_k)$, (see *Lemma 7.1*).

2) Calculate $\Delta x^{j,i}_{t_k}(\cdot)$, the effect of error in solution of $\mathcal{P}^j_D(t_k)$, on the solution of all subsequent optimization problems: $\mathcal{P}^j_D(t_{k+1})$, $\mathcal{P}^j_D(t_{k+2})$,..., $\mathcal{P}^j_D(t_{k+\sigma})$ for any arbitrary σ (i.e., calculate $\Delta x^{j,i}_{t_{k+1}}(\cdot)$, $\Delta x^{j,i}_{t_{k+2}}(\cdot)$,..... $\Delta x^{j,i}_{t_{k+\sigma}}(\cdot)$), (see *Lemma 7.2*).

3) Calculate $\Delta x^{j,i}_{t_{k+\sigma}}(\cdot)$, the effect of the error in the solution of $\mathcal{P}^j_D(t_{k+\sigma})$, on the cost function $J^j(\Gamma^j(t_{k+\sigma}))$, this effect is denoted by $\Delta J^{j,i}$, (see *Lemma 7.3* and *Theorem 7.1*).

4) Do the previous steps for all the neighbours of i, and calculate the overall mismatches as follows:

$$\Delta J^{\cdot,i} = \sum_{j|(j,i)\in\mathbb{E}} \Delta J^{j,i}$$

5) Minimize $\Delta J^{\cdot,i}$ subject to bandwidth allocation constraint to find the optimum allocated bandwidth to each neighbour j, (see *Problem 7.1*)

The following subsections and lemmas are presented to accomplish each step.

7.1.1. The Bound on the Cost Function

The following lemma is presented to calculate the errors due to mismatch $\varepsilon_{t_k}^{i,j}(\cdot)$ between the updated and delayed trajectories, for a class of LTI systems using the analytical solution of DRHC problem.

Lemma 7.1: Consider the following LTI dynamics for the j^{th} vehicle:

$$\dot{x}^j(t) = Ax^j(t) + Bu^j(t);\ x^j(t_0) = x^j(0) \qquad (7.1)$$

with the following non-delayed cost function of j^{th} vehicle at any time t_k:

$$J^j(\Gamma^j(t_k)) = \int_{t_k}^{t_k+T} \left(\left\| x_{t_k}^{j,j}(t) \right\|_Q^2 + \left\| u_{t_k}^{j,j}(t) \right\|_R^2 + \sum_{i|(j,i)\in\mathbb{E}} \left\| x_{t_k}^{j,j}(t) - x_{t_k}^{i,i}(t) \right\|_S^2 \right) dt + \left\| x_{t_k}^{j,j}(t_k+T) \right\|_P^2 \qquad (7.2)$$

Also, assume the neighbours of j at time t_k calculate the update trajectory $x_{t_k}^{i,i}(\cdot)$ where $(j,i)\in\mathbb{E}$. Whereas vehicle j uses the delayed trajectory $x_{t_{k-d}}^{i,i}(\cdot)$ of neighbour i; hence, there is a mismatch $\varepsilon_{t_k}^{i,j}(\cdot) = x_{t_k}^{i,i}(\cdot) - x_{t_{k-d}}^{i,i}(\cdot)$. Then the error in the solution of $\mathcal{P}_D^j(t_k)$ due to mismatch $\varepsilon_{t_k}^{i,j}(\cdot)$ in the updated and the delayed trajectory of neighbour i, denoted by $\Delta x_{t_k}^{j,i}(t)$, is calculated as follows:

$$\Delta x_{t_k}^{j,i}(t) = \int_{t_k}^{t} e^{(t-s)(-0.5BR^{-1}B')} \times 2S\varepsilon_{t_k}^{i,j} ds \ ;\ t\in[t_k, t_k+T] \qquad (7.3)$$

Proof: Since the analytical solution of the delayed DRHC problem $\mathcal{P}_D^j(t_k)$ is not available, then the solution of the $\mathcal{P}_D^j(t_k)$ is approximated by the solution of the non-delayed DRHC problem $\mathcal{P}^j(t_k)$. To find the effect of mismatch $\varepsilon_{t_k}^{i,j}(\cdot)$ on the solution of

$\mathcal{P}_D^j(t_k)$, the analytical solution of $\mathcal{P}_D^j(t_k)$ is sought using *Hamiltonian* equation as follows [167]:

$$H = \left\| x_{t_k}^{j,j} \right\|_Q^2 + \left\| u_{t_k}^{j,j} \right\|_R^2 + \sum_{i|(j,i)\in\mathbb{E}} \left\| x_{t_k}^{j,j}(t) - x_{t_k}^{i,i}(t) \right\|_S^2 + q'(Ax_{t_k}^{j,j} + Bu_{t_k}^{j,j}) \tag{7.4}$$

where q defines the vector of co-states. The necessary conditions for the optimality are [167]:

$$\begin{aligned} &1)\nabla_{u_{t_k}^{j,j}} H = 0 \Rightarrow u_{t_k}^{j,j}(t) = -(R+R')^{-1}B'q = -\frac{1}{2}R^{-1}B'q \\ &2)\dot{q} = -\nabla_{x_{t_k}^{j,j}} H \end{aligned} \tag{7.5}$$

Combining this with system dynamics yields:

$$\begin{bmatrix} \dot{x}_{t_k}^{j,j} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} A & -\frac{1}{2}BR^{-1}B' \\ -2(Q+N_l^j S) & -A' \end{bmatrix} \begin{bmatrix} x_{t_k}^{j,j} \\ q \end{bmatrix} + \sum_{i|(j,i)\in E} \begin{bmatrix} \bar{0} \\ 2Sx_{t_k}^{i,i} \end{bmatrix} \tag{7.6}$$

with boundary conditions:

$$\begin{aligned} &x_{t_k}^{j,j}(t_k) = x^j(t_k) \\ &q(t_k + T) = 2Px_{t_k}^{j,j}(t_k + T) \end{aligned} \tag{7.7}$$

Define:

$$\begin{aligned} z^j &= \begin{bmatrix} x_{t_k}^{j,j} \\ q \end{bmatrix} \\ y^i &= \begin{bmatrix} \bar{0} \\ 2Sx_{t_k}^{i,i} \end{bmatrix} \\ \bar{A} &= \begin{bmatrix} A & -0.5BR^{-1}B' \\ -2(Q+N_l^j S) & -A' \end{bmatrix} \end{aligned} \tag{7.8}$$

Then using state transition matrix, the solution of (7.6) is:

$$z^j(t) = \varphi(t,t_k)z^j(t_k) + \sum_{i|(j,i)\in\mathbb{E}} \int_{t_k}^{t} \varphi(t,s)y^i(s)ds \tag{7.9}$$

Now assume there is a mismatch in broadcasted information and hence leads to error in $y^i(t)$. The effect of such error in $y^i(t)$ is investigated using the perturbation theory as follows:

$$\begin{aligned} &y^i(t) \leftarrow y^i(t) + \Delta y^i(t) \Rightarrow \\ &z^j(t) + \Delta z^j(t) = \varphi(t,t_k)z^j(t_k) + \sum_{i|(j,i)\in\mathbb{E}} \int_{t_k}^{t} \varphi(t,s)(y^i(s) + \Delta y^i(t))ds \Rightarrow \\ &\Delta z^j(t) = \sum_{i|(j,i)\in\mathbb{E}} \int_{t_k}^{t} \varphi(t,s)\Delta y^i(t)ds \end{aligned} \tag{7.10}$$

The effect of each neighbour can be decoupled as follows: assume $\Delta z^{j,i}$ is the effect of error in broadcasted information from neighbour i to vehicle j; then:

$$\Delta z^{j,i}(t) = \int_{t_k}^{t} \varphi(t,s)\Delta y^i(t)ds \tag{7.11}$$

Then form (7.8):

$$\begin{aligned} \Delta z^{j,i}(t) &= \int_{t_k}^{t} \varphi(t,s)\Delta y^i(t)ds = \int_{t_k}^{t} e^{(t-s)\bar{A}}\Delta y^i(t)ds \\ &= \int_{t_k}^{t} e^{(t-s)\begin{bmatrix} A & -0.5BR^{-1}B^{'} \\ -2(Q+N_I^jS) & -A^{'} \end{bmatrix}} \times \begin{bmatrix} \bar{0} \\ 2S\varepsilon_{t_k}^{i,j} \end{bmatrix} ds \\ &= \int_{t_k}^{t} \begin{bmatrix} e^{(t-s)(-0.5BR^{-1}B^{'})} \times 2S\varepsilon_{t_k}^{i,j} \\ e^{(t-s)(-A^{'})} \times 2S\varepsilon_{t_k}^{i,j} \end{bmatrix} ds \end{aligned} \tag{7.12}$$

Then using (7.8) for decomposing $\Delta z^{j,i}(t)$ yields:

$$\Delta x_{t_k}^{j,i}(t) = \int_{t_k}^{t} e^{(t-s)(-0.5BR^{-1}B^{'})} \times 2S\varepsilon_{t_k}^{i,j} ds \tag{7.13}$$

And this completes the proof.

Lemma 7.2: Consider the effect of $\Delta \boldsymbol{x}_{t_k}^{j,i}(\cdot)$ (error in solution of $\mathcal{P}_D^j(t_k)$) on the solution of the next optimization problem $\mathcal{P}_D^j(t_{k+1})$ is denoted by $\Delta \boldsymbol{x}_{t_{k+1}}^{j,i}(\cdot)$; and the effect of $\Delta \boldsymbol{x}_{t_{k+1}}^{j,i}(\cdot)$ on $\mathcal{P}_D^j(t_{k+2})$ is denoted by $\Delta \boldsymbol{x}_{t_{k+2}}^{j,i}(\cdot)$ and so on. Also, assume that the sampling time δ is small enough so that during $[t_k, t_k+\delta]$, $\Delta \boldsymbol{x}_{t_k}^{j,i}(\cdot)$ can be approximated by a constant vector $\bar{\boldsymbol{\varepsilon}}^{i,j}(t_k)$, i.e., $\boldsymbol{\varepsilon}_{t_k}^{j,i}(\cdot) \approx \bar{\boldsymbol{\varepsilon}}^{i,j}(t_k)$. Then the error in the solution of $\mathcal{P}_D^j(t_{k+\sigma})$ denoted by $\Delta \boldsymbol{x}_{t_{k+\sigma}}^{j,i}(\cdot)$ and $\Delta \boldsymbol{u}_{t_{k+\sigma}}^{j,i}(\cdot)$ is calculated as follows:

$$\Delta \boldsymbol{x}_{t_{k+\sigma}}^{j,i}(t) = \left(e^{(t-t_k-\delta)A} \times M_2 + e^{(t-t_k-\delta)(-0.5BR^{-1}B')} \times M_4 \right) 2S\bar{\boldsymbol{\varepsilon}}^{i,j}(t_k); \quad t \in [t_{k+\sigma}, t_{k+\sigma}+T] \tag{7.14}$$

And:

$$\Delta \boldsymbol{u}_{t_{k+\sigma}}^{j,i}(t) = -R^{-1}B' \left[e^{(t-t_k-\delta)[-2(Q+N_I^j S)]} \times M_2 + e^{(t-t_k-\delta)(-A')} \times M_4 \right] S\bar{\boldsymbol{\varepsilon}}^{i,j}(t_k); \quad t \in [t_{k+\sigma}, t_{k+\sigma}+T] \tag{7.15}$$

where M is a partitioned matrix of matrixes with $(n \times n)$ dimension: M_1, M_2, M_3 and M_4; (n is the size of state vector of each vehicle) and is calculated as follows:

$$\begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix} = M = \left(\bar{A}\right)^{-1} \left(e^{\delta \bar{A}} - I \right) \tag{7.16}$$

Proof: Again (7.9) is used as follows:

$$z_{t_{k+1}}^j(t) = \varphi(t, t_{k+1}) z^j(t_{k+1}) + \sum_{i|(j,i)\in\mathbb{E}} \int_{t_{k+1}}^{t} \varphi(t,s) \boldsymbol{y}_{t_{k+1}}^i(s) ds \tag{7.17}$$

Since, (7.3) represents the error at time t_{k+1} which is the initial condition of optimization problem $\mathcal{P}_D^j(t_{k+1})$, this error in the initial condition is studied using the perturbation in the initial condition as follows:

$$\begin{aligned}
&z^j(t_{k+1}) \leftarrow z^j(t_{k+1}) + \Delta z^j(t_{k+1}) \Rightarrow \\
&z^j_{t_{k+1}}(t) + \Delta z^j_{t_{k+1}}(t) = \varphi(t, t_{k+1})(z^j(t_{k+1}) + \Delta z^j(t_{k+1})) + \sum_{i|(j,i)\in\mathbb{E}} \int_{t_{k+1}}^{t} \varphi(t,s) y^i_{t_{k+1}}(s) ds \\
&\Delta z^j_{t_{k+1}}(t) = \varphi(t, t_{k+1}) \Delta z^j(t_{k+1})
\end{aligned} \tag{7.18}$$

From (7.11):

$$\Delta z^j_{t_{k+1}}(t) = \varphi(t, t_{k+1}) \Delta z^j(t_{k+1}) = e^{(t-t_{k+1})\bar{A}} \times \int_{t_k}^{t} e^{(t-s)\bar{A}} \times \Delta y^i(t) ds \tag{7.19}$$

Then:

$$\Delta z^j_{t_{k+1}}(t) = \varphi(t, t_{k+1}) \Delta z^j(t_{k+1}) \tag{7.20}$$

Using $t = t_{k+2}$ yields:

$$\Delta z^j_{t_{k+1}}(t_{k+2}) = \varphi(t_{k+2}, t_{k+1}) \Delta z^j(t_{k+1}) \tag{7.21}$$

Analogously for subsequent optimization problems:

$$\begin{aligned}
&\Delta z^j_{t_{k+2}}(t_{k+3}) = \varphi(t_{k+3}, t_{k+2}) \Delta z^j(t_{k+2}) = \varphi(t_{k+3}, t_{k+2}) \varphi(t_{k+2}, t_{k+1}) \Delta z^j(t_{k+1}) \\
&\Delta z^j_{t_{k+3}}(t_{k+4}) = \varphi(t_{k+4}, t_{k+3}) \varphi(t_{k+3}, t_{k+2}) \varphi(t_{k+2}, t_{k+1}) \Delta z^j(t_{k+1}) \\
&\cdot \\
&\cdot \\
&\Delta z^j_{t_{k+\sigma-1}}(t_{k+\sigma}) = \prod_{n=2}^{\sigma} \varphi(t_{k+n}, t_{k+n-1}) \Delta z^j(t_{k+1})
\end{aligned} \tag{7.22}$$

If again (7.18) and (7.22) are used together, the error of $\mathcal{P}_D^j(t_{k+\sigma})$ is calculated as follows:

$$\Delta z^j_{t_{k+\sigma}}(t) = \varphi(t, t_{k+\sigma}) \Delta z^j(t_{k+\sigma}) = \varphi(t, t_{k+\sigma}) \prod_{n=2}^{\sigma} \varphi(t_{k+n}, t_{k+n-1}) \Delta z^j(t_{k+1}) \tag{7.23}$$

Substituting for the state transition matrix, the above error can be simplified. The state transition matrix is calculated as follows:

$$\varphi(t,t_0)=e^{(t-t_0)\bar{A}} \tag{7.24}$$

which has the following property:

$$\varphi(t_{k+1},t_k)=e^{(t_{k+1}-t_k)\bar{A}}=e^{\delta\bar{A}} \quad ; \quad \forall k\in\mathbb{N} \tag{7.25}$$

Using (7.25) in (7.23) yields:

$$\Delta z^j_{t_{k+\sigma}}(t)=e^{(t-t_{k+\sigma}+(\sigma-1)\delta)\bar{A}}\,\Delta z^j(t_{k+1})=e^{(t-t_k-\delta)\bar{A}}\,\Delta z^j(t_{k+1})\ ;\ t\in[t_{k+\sigma},t_{k+\sigma}+T] \tag{7.26}$$

The simplification in (7.26) is done using the fact that $t_{k+\sigma}=t_k+\sigma\delta$.

In (7.26), $\Delta z^j(t_{k+1})$ is the error in initial condition of $\mathcal{P}^j_D(t_{k+1})$; since it is desired to find the effect of mismatch of information of neighbour *i*, then $\Delta z^j(t_{k+1})=\Delta z^{j,i}(t_{k+1})$ where $\Delta z^{j,i}(t_{k+1})$ is calculated from (7.11); a superscript *i* is also added to $\Delta z^j_{t_{k+\sigma}}(t)$ to clarify that this error is due to mismatch from neighbour *i* as follows:

$$\Delta z^{j,i}_{t_{k+\sigma}}(t)=e^{(t-t_k-\delta)\bar{A}}\times\int_{t_k}^{t_{k+1}}e^{(t_{k+1}-s)\bar{A}}\Delta y^i(s)ds\ ; \qquad t\in[t_{k+\sigma},t_{k+\sigma}+T] \tag{7.27}$$

In (7.27) the integration interval is $[t_k,t_{k+1}]$; hence if the sampling time $\delta=t_{k+1}-t_k$ is chosen small enough, it can be assumed that $\Delta y^i(t)$ is constant during this interval, i.e., $\Delta y^i(t)\approx\overline{\Delta y}^i$; $t\in[t_k,t_{k+1}]$; then:

$$\Delta z^{j,i}_{t_{k+\sigma}}(t)=e^{(t-t_k-\delta)\bar{A}}\times\left(\bar{A}\right)^{-1}\left(e^{\delta\bar{A}}-I\right)\overline{\Delta y}^i; \qquad t\in[t_{k+\sigma},t_{k+\sigma}+T] \tag{7.28}$$

Denote:

$$M=\left(\bar{A}\right)^{-1}\left(e^{\delta\bar{A}}-I\right)=\begin{bmatrix}M_1 & M_2\\ M_3 & M_4\end{bmatrix} \tag{7.29}$$

where M is a partitioned matrix with $(n\times n)$ matrix entries: M_1, M_2, M_3 and M_4; also n is the size of state vector of each vehicle. Then, from (7.8) since the first entry of Δy^i is zero:

$$\begin{bmatrix}M_1 & M_2\\ M_3 & M_4\end{bmatrix}\overline{\Delta y}^i=\begin{bmatrix}M_1 & M_2\\ M_3 & M_4\end{bmatrix}\begin{bmatrix}0\\ 2S\bar{\varepsilon}^{i,j}\end{bmatrix}=\begin{bmatrix}2M_2S\bar{\varepsilon}^{i,j}\\ 2M_4S\bar{\varepsilon}^{i,j}\end{bmatrix} \tag{7.30}$$

Then from (7.28) and (7.30):

$$\begin{bmatrix}\Delta x^{j,i}_{t_{k+\sigma}}(t)\\ \Delta q(t)\end{bmatrix}=e^{(t-t_k-\delta)\bar{A}}\begin{bmatrix}2M_2S\bar{\varepsilon}^{i,j}(t_k)\\ 2M_4S\bar{\varepsilon}^{i,j}(t_k)\end{bmatrix}=$$

$$e^{(t-t_k-\delta)\begin{bmatrix}A & -0.5BR^{-1}B'\\ -2(Q+N_l^jS) & -A'\end{bmatrix}}\times\begin{bmatrix}2M_2S\bar{\varepsilon}^{i,j}(t_k)\\ 2M_4S\bar{\varepsilon}^{i,j}(t_k)\end{bmatrix} \tag{7.31}$$

$$=\begin{bmatrix}e^{(t-t_k-\delta)A}\times 2M_2S\bar{\varepsilon}^{i,j}(t_k)+e^{(t-t_k-\delta)(-0.5BR^{-1}B')}\times 2M_4S\bar{\varepsilon}^{i,j}(t_k)\\ e^{(t-t_k-\delta)(-2(Q+N_l^jS))\times 2M_2S\bar{\varepsilon}^{i,j}(t_k)+e^{(t-t_k-\delta)(-A')}}\times 2M_4S\bar{\varepsilon}^{i,j}(t_k)\end{bmatrix}$$

$$;\qquad t\in[t_{k+\sigma},t_{k+\sigma}+T]$$

Then, separating the solution for q and $\Delta x^{j,i}_{t_{k+\sigma}}(t)$ in (7.31) yields:

$$\Delta x^{j,i}_{t_{k+\sigma}}(t)=\left(e^{(t-t_k-\delta)A}\times M_2+e^{(t-t_k-\delta)(-0.5BR^{-1}B')}\times M_4\right)2S\bar{\varepsilon}^{i,j}(t_k);\qquad t\in[t_{k+\sigma},t_{k+\sigma}+T] \tag{7.32}$$

and substituting for q in (7.5) yields:

$$\Delta u^{j,i}_{t_{k+\sigma}}(t)=-R^{-1}B'\left[e^{(t-t_k-\delta)[-2(Q+N_l^jS)]}\times M_2+e^{(t-t_k-\delta)(-A')}\times M_4\right]S\bar{\varepsilon}^{i,j}(t_k);\qquad t\in[t_{k+\sigma},t_{k+\sigma}+T] \tag{7.33}$$

where M_2 and M_4 are partitioned matrices built from A, B, Q, R and S and calculated from (7.29).

Equations (7.32) and (7.33) represent the effect of mismatch $\varepsilon_{t_k}^{i,j}(\cdot)$ on the solution of the j^{th} vehicle at time $t_{k+\sigma}$.

Remark 7.1: One approximation to $\bar{\varepsilon}^{i,j}(t_k)$ for interval $[t_k, t_{k+1}]$ can be:

$$\begin{aligned} \varepsilon_{t_k}^{i,j}(t) &= x_{t_k}^{i,i}(t) - x_{t_{k-d}}^{i,i}(t) \\ \bar{\varepsilon}^{i,j}(t_k) &= \frac{\varepsilon_{t_k}^{i,j}(t_{k+1}) - \varepsilon_{t_k}^{i,j}(t_k)}{2} \end{aligned} \tag{7.34}$$

To find how the mismatch can affect the cost function (performance index) the following lemma is presented.

Lemma 7.3: consider the desired trajectories $x_{t_k}^{j,j}(\cdot)$ and $u_{t_k}^{j,j}(\cdot)$; also, consider the cost function (7.2). Assume there is an error in the mentioned desired trajectories, i.e., $x_{t_k}^{j,j}(\cdot) \leftarrow x_{t_k}^{j,j}(\cdot) + \Delta x_{t_k}^{j,i}(\cdot)$ and $u_{t_k}^{j,j}(\cdot) \leftarrow u_{t_k}^{j,j}(\cdot) + \Delta u_{t_k}^{j,i}(\cdot)$, due to mismatch between delayed and actual trajectory of neighbour j, theses errors lead to an overhead cost $\Delta J^{j,i}$ to the non-delayed cost function J^j. Then, the extra cost added to non-delayed cost function (7.2) due to these errors is bounded as follows:

$$\Delta J^{j,i} \leq \int_{t_k}^{t_k+T} \left(\left\| \Delta x_{t_k}^{j,i}(t) \right\|_Q^2 + \left\| \Delta u_{t_k}^{j,i}(t) \right\|_R^2 + N_l^j \left\| \Delta x_{t_k}^{j,i}(t) \right\|_S^2 \right) dt + \left\| \Delta x_{t_k}^{j,i}(t_k + T) \right\|_P^2 \tag{7.35}$$

where N_l^j is the number of leaders of vehicle j, $\Delta J^{j,i}$ is the deviation from the desired non-delayed cost J^j and should be minimized.

Proof: Using perturbation theory yields:

$$\left.\begin{aligned} &\boldsymbol{x}_{t_k}^{j,j}(\cdot) \leftarrow \boldsymbol{x}_{t_k}^{j,j}(\cdot) + \Delta\boldsymbol{x}_{t_k}^{j,i}(\cdot) \\ &\boldsymbol{u}_{t_k}^{j,j}(\cdot) \leftarrow \boldsymbol{u}_{t_k}^{j,j}(\cdot) + \Delta\boldsymbol{u}_{t_k}^{j,i}(\cdot) \\ &J^j \leftarrow J^j + \Delta J^j \end{aligned}\right\} \Rightarrow$$

$$J^j + \Delta J^{j,i} = \int_{t_k}^{t_k+T} \left(\left\| \boldsymbol{x}_{t_k}^{j,j}(t) + \Delta\boldsymbol{x}_{t_k}^{j,i}(t) \right\|_Q^2 + \left\| \boldsymbol{u}_{t_k}^{j,j}(t) + \Delta\boldsymbol{u}_{t_k}^{j,i}(t) \right\|_R^2 + \right.$$

$$\left. \sum_{i|(j,i)\in\mathbb{E}} \left\| \boldsymbol{x}_{t_k}^{j,j}(t) + \Delta\boldsymbol{x}_{t_k}^{j,i}(t) - \boldsymbol{x}_{t_k}^{i,i}(t) \right\|_S^2 \right) dt + \left\| \boldsymbol{x}_{t_k}^{j,j}(t_k+T) + \Delta\boldsymbol{x}_{t_k}^{j,i}(t_k+T) \right\|_P^2$$

Using the homogeneity axiom of vector norms and applying $\|a+b\|_Q^2 \le \|a\|_Q^2 + \|b\|_Q^2$ leads to (7.35).

The following theorem integrates the results of *Lemma 7.1*, *Lemma 7.2* and *Lemma 7.3* to complete the mentioned steps for bandwidth allocation problem.

Theorem 7.1: Assume that vehicle i at time t_k calculated the updated trajectory $\boldsymbol{x}_{t_k}^{i,i}(\cdot)$ while follower j have access to delayed trajectory $\boldsymbol{x}_{t_{k-d}}^{i,i}(\cdot)$. Hence, there is a mismatch $\boldsymbol{\varepsilon}_{t_k}^{i,j}(\cdot) = \boldsymbol{x}_{t_k}^{i,i}(\cdot) - \boldsymbol{x}_{t_{k-d}}^{i,i}(\cdot)$ between updated and delayed trajectories of vehicle i. The effect of this mismatch in the cost function $J^j(\Gamma^j(t_{k+\sigma}))$ where $\sigma \in \mathbb{R}$ of vehicle j at future time $t_{k+\sigma}$ is the extra cost calculated by:

$$\Delta J^{j,i} \le \int_{t_{k+\sigma}}^{t_{k+\sigma}+T} \left(\left\| \Delta\boldsymbol{x}_{t_{k+\sigma}}^{j,i}(t) \right\|_Q^2 + \left\| \Delta\boldsymbol{u}_{t_{k+\sigma}}^{j,i}(t) \right\|_R^2 + N_l^j \left\| \Delta\boldsymbol{x}_{t_{k+\sigma}}^{j,i}(t) \right\|_S^2 \right) dt + \left\| \Delta\boldsymbol{x}_{t_{k+\sigma}}^{j,i}(t_k+T) \right\|_P^2 \quad (7.36)$$

where $\Delta\boldsymbol{x}_{t_{k+\sigma}}^{j,i}(t)$ and $\Delta\boldsymbol{u}_{t_{k+\sigma}}^{j,i}(t)$ are calculated from (7.32) and (7.33).

Proof: using *Lemma 7.3* and shifting forward the calculation time in (7.35) by σ step yields (7.36); then using *Lemma 7.1* and *Lemma 7.2*, the proof is straightforward.

The extra cost $\Delta J^{j,i}$ is imposed by errors $\Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma}}(t)$ and $\Delta \boldsymbol{u}^{j,i}_{t_{k+\sigma}}(t)$ which can be due to any source. For example, imperfect optimization, communication noise, model uncertainty which lead to mismatch between actual and delayed communicated trajectories in previous time steps can cause errors $\Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma}}(t)$ and $\Delta \boldsymbol{u}^{j,i}_{t_{k+\sigma}}(t)$. More precisely, $\Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma}}(t)$ and $\Delta \boldsymbol{u}^{j,i}_{t_{k+\sigma}}(t)$ are functions of 1- communication delay (σ) 2- mismatch between actual and delayed communicated information from neighbours of vehicle *i*.

7.1.2. Bandwidth Allocation Formulation

The general overview of the proposed communication bandwidth algorithm is explained in Section 7.1.1; this section will add more detail to that and connect the previous analysis with the communication bandwidth equations and DRHC algorithm.

Consider a network of vehicles, where the communication channel of each vehicle is used to communicate with neighbouring vehicles. Hence, in such situation, the following communication constraint must be satisfied by each vehicle *i* when communicating with neighbours [180]:

$$\sum_{j|(j,i)\in\mathbb{E}} \frac{K_{ij}}{\tau_{ij}} \leq B \qquad (7.37)$$

where, τ_{ij} is the delay for transmitting the information from i to j. K_{ij} (bytes/sample) is the size of packet sent to j by i and B (bytes/sec) is the available bandwidth of the communication channel after counting for other communications such as operator commands.

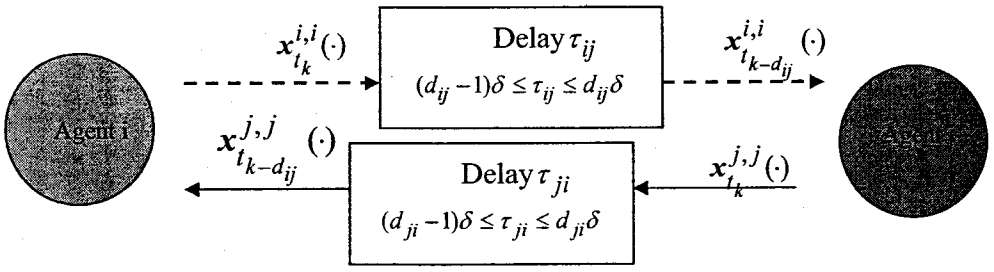

Figure 7.3: Communication delays between neighbouring agents

In the previous sections to avid making the notation too complicated, the short notation d and τ are used for time-delay; however, the nature of bandwidth allocation algorithm makes the communication delay time-varying and different for each pair of vehicles. Then in this section the algorithms will be updated with the new notation.

Assume the amount of allocated bandwidth to follower j by vehicle i is denoted by B_{ij} and the maximum available communication bandwidth is used; hence:

$$\sum_{j|(j,i)\in\mathbb{E}} B_{ij} = B \qquad (7.38)$$

then from (7.37):

$$\tau_{ij} = \frac{K_{ij}}{B_{ij}} \quad ; (j,i) \in \mathbb{E} \qquad (7.39)$$

7.1.2.1. Equal Bandwidth Allocation

If the communication bandwidth allocation algorithm is not used then an equal bandwidth allocation strategy may be employed which leads to equal communication delay for neighbouring vehicles; hence, considering (7.37) the vehicle i allocates equal

bandwidth to each neighbour j ; then the communication bandwidth allocated to the j^{th} neighbour of i and the equivalent communication delay is calculated as follows:

$$\begin{aligned} B_{ij} &= \frac{B}{N_f^i} \quad ; \quad j \in \mathbb{V} \,\&\, (j,i) \in \mathbb{E} \\ \tau_{ij} &= N_f^i \frac{K_{ij}}{B} \quad ; \quad j \in \mathbb{V} \;\&\; (j,i) \in \mathbb{E} \end{aligned} \tag{7.40}$$

where N_f^i is the number of followers of i^{th} vehicle in a *direct* communication graph (N_f^i is changed to N_n^i for an *indirect* communication graph).

7.1.2.2. Variable Bandwidth Allocation

The proposed bandwidth allocation approach is based on minimizing the bound on the cost function (7.36). In this approach the mismatch of the previous time steps is measured and then based on the mismatch, the bound (7.36) on the future cost function is approximated and is minimized to find the optimal communication bandwidth for each vehicle. The notation in bound (7.36) needs to be improved as $\sigma = \sigma_{ij}$, for bandwidth allocation purpose:

$$\Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma_{ij}}}(t) = \left(e^{(t-t_k-\delta)A} \times M_2 + e^{(t-t_k-\delta)(-0.5BR^{-1}B')} \times M_4 \right) 2S\bar{\varepsilon}^{i,j}(t_k); \quad t \in [t_{k+\sigma_{ij}}, t_{k+\sigma_{ij}} + T] \tag{7.41}$$

$$\Delta \boldsymbol{u}^{j,i}_{t_{k+\sigma_{ij}}}(t) = -R^{-1}B' \left[e^{(t-t_k-\delta)[-2(Q+N_f^j S)]} \times M_2 + e^{(t-t_k-\delta)(-A')} \times M_4 \right] S\bar{\varepsilon}^{i,j}(t_k); \quad t \in [t_{k+\sigma_{ij}}, t_{k+\sigma_{ij}} + T] \tag{7.42}$$

Then the bandwidth allocation algorithm is the solution of the following minimization problem:

Consider the vehicle i has the available bandwidth B. Then, a solution to the bandwidth allocation problem solved by vehicle $i \in \mathbb{V}$ is the solution of the following minimization problem:

Problem 7.1: Bandwidth Allocation

$$\begin{aligned}\underset{\{\sigma_{ij}|(j,i)\in\mathbb{E}\}}{\text{Min}} \sum_{j|(j,i)\in\mathbb{E}} \int_{t_{k+\sigma_{ij}}}^{t_{k+\sigma_{ij}}+T} \left(\left\| \Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma_{ij}}}(t) \right\|^2_{Q+N_I^j S} + \left\| \Delta \boldsymbol{u}^{j,i}_{t_{k+\sigma_{ij}}}(t) \right\|^2_{R} \right) dt \\ + \left\| \Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma_{ij}}}(t_{k+\sigma_{ij}}+T) \right\|^2_{P} \end{aligned} \tag{7.43}$$

$$\text{subject to}: \sum_{j|(j,i)\in\mathbb{E}} \frac{K_{ij}}{\delta\sigma_{ij}} \le B$$

where $\Delta \boldsymbol{x}^{j,i}_{t_{k+\sigma_{ij}}}(t)$ and $\Delta \boldsymbol{u}^{j,i}_{t_{k+\sigma_{ij}}}(t)$ are calculated from (7.41) and (7.42) respectively. It is worth mentioning that the integration interval in (7.43) is consistent with time interval on which (7.41) and (7.42) are available.

In the next section, this bandwidth allocation problem is integrated with the delayed DRHC $\mathcal{P}_D^i(t_k)$.

7.2. Delayed DRHC with Bandwidth Allocation

The following algorithm is a modification to the delayed DRHC *Algorithm 4.1* where the bandwidth allocation algorithm is integrated with the delayed DRHC $\mathcal{P}_D^i(t_k)$:

Algorithm 7.1: Delayed DRHC with Bandwidth Allocation

1: **Initialization:**

a. $k \leftarrow 0$,

b. measure $x^i(t_k)$,

c. Calculate the equal bandwidth allocation for initialization from (7.40) as

$$B_{ij} = \frac{B}{N_n^i} \quad ; \quad j \mid (j,i) \in \mathbb{E}.$$

d. GOTO step 5.

2: **Receive** $x_{t_{k-d}}^{j,j}(\cdot)$ **from leaders where** $(i,j) \in \mathbb{E}$.

3: **Bandwidth Allocation:**

a. Find the delay d_{ij} where the delayed trajectory that the follower *j* is using at the current time t_k is $x_{t_{k-d_{ij}}}^{j,i}(\cdot)$.

b. Measure $\overline{\Delta\varepsilon}^{i,j}(t_k)$; $(j,i) \in E$ from (7.34) by setting $d = d_{ij}$ from previous step.

c. Solve the minimization Problem 7.1 to find the optimal communication time delays $\sigma = \sigma_{ij}$ where $(j,i) \in \mathbb{E}$.

d. Calculate the allocated bandwidth to each follower from $B_{ij} = \frac{K_{ij}}{\delta\sigma_{ij}}$; $(j,i) \in \mathbb{E}$ according to (7.39).

4: **Measure** $x^i(t_k)$ **and update the *information set* from (4.1).**

5: **Solve** $\mathcal{P}_D^i(t_k)$.

6: **Send the state trajectory** $x_{t_k}^{i,i}(\cdot)$ **to followers utilizing the allocated bandwidth** B_{ij} **(of step 1 or step 3) where** $(j,i) \in \mathbb{E}$.

7: **Execute the control action for individual vehicle *i* during** $[t_k, t_{k+1}]$.

8: $k \leftarrow k+1$. **Goto step 2.**

This algorithm is a modified version of *Algorithm 4.1*, while at each iteration step 3, is executed to allocate the communication bandwidth efficiently. The following describes the changes to each step:

Step 1: since at the beginning, there is no history of communication with followers, an equal bandwidth allocation strategy is employed.

Step 2: the vehicle i receives the most available trajectory of leader computed by them. Also, the time these trajectories are calculated are sent to measure the communication delay.

Step 3.a: it is determined that at the current time the followers use which of the previously computed and communicated trajectories. It doesn't need any communication; each vehicle i has a buffer that stores the history of time and corresponding delay of the previously communicated information to each neighbour. For example, assume $\delta = 0.5\,\text{sec}$; then at $t_5 = 2.5$ sec the vehicle i has the history of the previous communication times $\mathbb{T} = \{0.5, 1.0, 1.5, 2.0\}$ with follower j with the following delays respectively: $\mathbb{D} = \{4,1,2,3\}$; then it follows that $d_{ij} = 2$, because $1.5 + 2 \times \delta = 2.5 = t_5 = \textit{current time}$. Also, in the case of two or more answers for this procedure the most updated information will be used and the previous information is removed to free the buffer for new information.

Step 3.b: this step calculates the mismatch between the updated trajectory, which local vehicle i is using, and the delayed trajectory, which follower j is using.

Step 3.c: based on the calculated mismatch of step 3.b, the bound on the cost function is minimized to find the optimum future delays for each neighbour.

Step 3.d: the bandwidth is allocated to followers to be used for communication in step 6.

Steps 4, 5, 6, 7 and 8 are the same as the corresponding steps in the *Algorithm 4.1*.

It is worth mentioning that the above algorithm does not intend to increase the capacity of the communication facility. Rather, it tries to balance the available communication in the sense that it distributes the available resources efficiently based on the needs of the neighbouring vehicles.

7.3. Simulation Results

A leaderless formation of a fleet of three unmanned vehicles with the following 3DOF dynamics is considered [110]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_2 + u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -x_4 + u_2 \end{aligned} \tag{7.44}$$

where $\boldsymbol{x} = [x, \dot{x}, y, \dot{y}]$ and $\boldsymbol{u} = [u_1, u_2]$ are the state and input vectors respectively, also, x and y are the components of position vector. The inputs are saturated at: $0 \le u_1 \le 10$ and $0 \le u_2 \le 10$ (N). Also: $\sqrt{\dot{x}^2 + \dot{y}^2} \le 10\ m/\sec$. These values are used for the modeling of all team members.

The actual trajectories for three vehicles in a triangular formation and the corresponding distance profile are shown in Figure 7.4. The vehicles are supposed to fly in a triangular formation and visit some predefined waypoints. In this formation, it is desired that moving vehicles keep a relative distance of 3m while flying in a triangular formation. As seen from Figure 7.4 the vehicles reach the desired distances after some time.

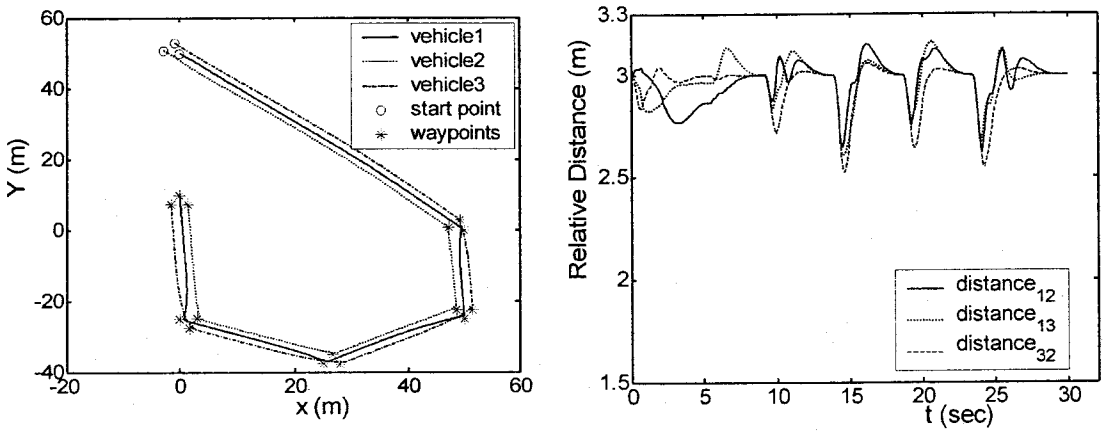

Figure 7.4: Triangular formation of a fleet of unmanned rotorcrafts visiting waypoints.

7.3.1. Coupling Cost (Performance) & Communication Delay versus Mismatch:

Figure 7.5 shows the maximum coupling cost (summation of errors in desired relative distance) versus mismatch (7.34) for different simulations. Each point corresponds to a simulation where different parameters of the system such as the initial conditions are changed to span a wide range of errors and mismatches, while the overall scenario of Figure 7.4 remains the same for all simulations. As seen from Figure 7.5 the maximum coupling cost will increase with the mismatch.

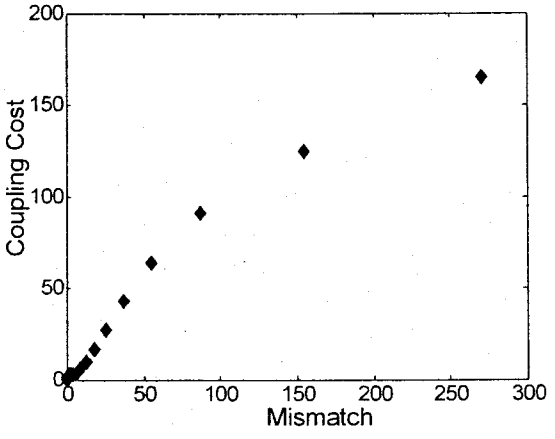

Figure 7.5: Coupling cost vs. Mismatch

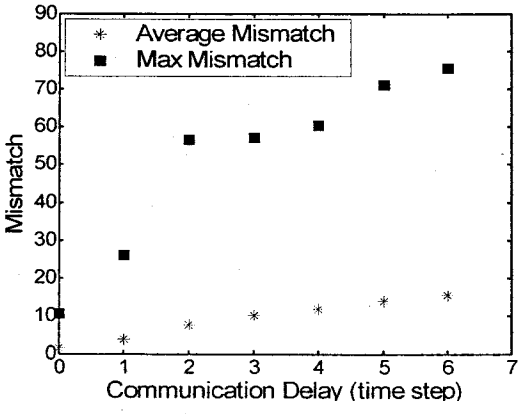

Figure 7.6: Mismatch vs. Communication Time Delays

As another case study, the effect of communication delay on mismatch (7.34) is investigated. Figure 7.6 shows the average and maximum mismatch for 7 different simulations versus communication delay. The simulations differ only in communication delays. As seen from Figure 7.6, the overall mismatch will increase with the communication delay. Consequently, one can conclude that communication delay can have an adverse effect on the stability and performance, which is intuitively expected.

7.3.2. Example: Bandwidth Allocation

In the following simulations it is assumed that the same size for all communicated messages, i.e., $K_{12} = K_{13} = ... = K_{21} = K_{23} = ... = K = 1000\ bytes / sample$. And the communication bandwidth is $B = 500 bytes / sec$.

The simulation results for two different cases are depicted in Figure 7.7 and Figure 7.8. In Figure 7.7, the delayed cost function of the second vehicle is plotted versus time for two different cases (the cost for other vehicles follows the same pattern).

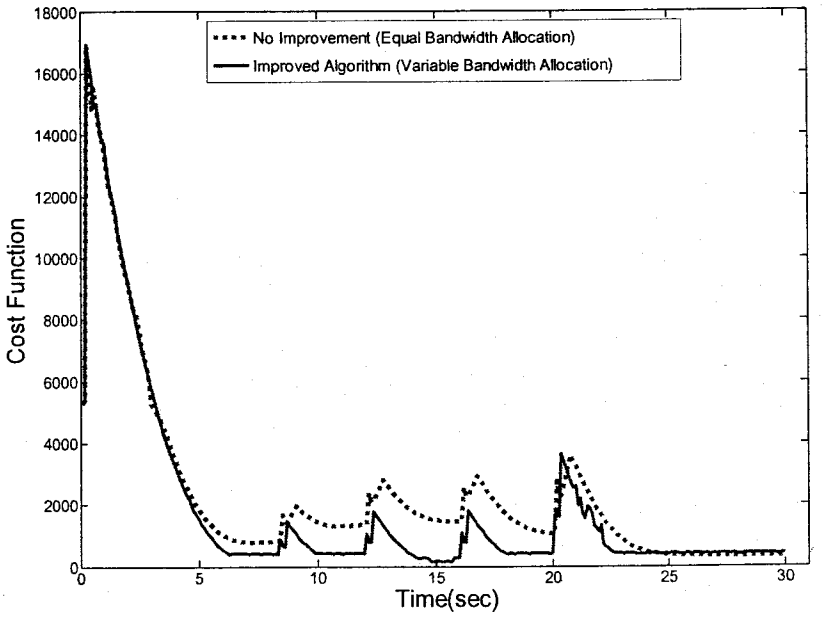

Figure 7.7: Cooperation cost history corresponding to improved *Algorithm 7.1* vs. *Algorithm 4.1*.

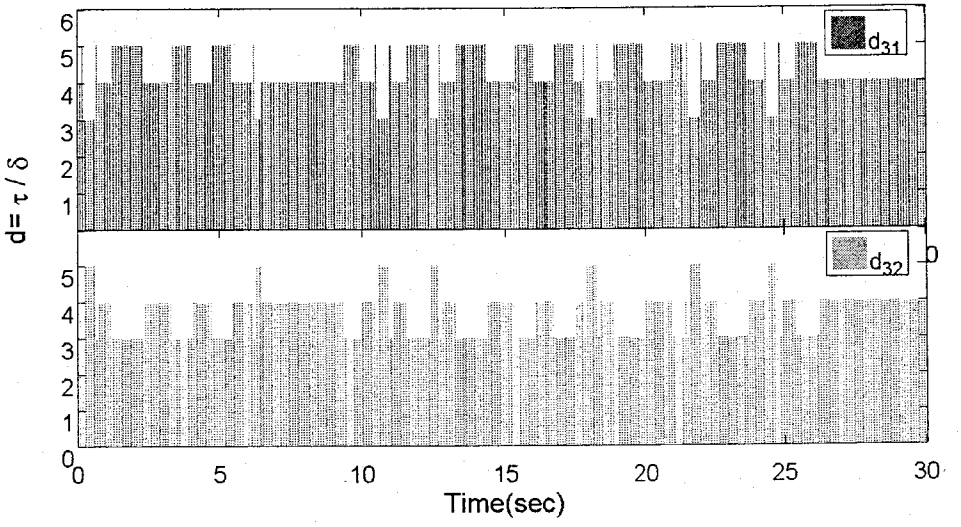

Figure 7.8: Delay variation in the channel of the third vehicle using improved Algorithm 7.1.

First, an equal bandwidth allocation strategy is utilized based on (7.40) using *Algorithm 4.1*; the average cost for this case is 1736.6 as seen from Figure 7.8. Second, proposed Algorithm 7.1 allocates bandwidth so that the average cost is reduced to 1537.

The bandwidth allocation leads to varying communication delays for each vehicle as seen in Figure 7.8, where the time history of the delay allocation (due to bandwidth allocation) is plotted for the communication channel of the third vehicle. The communication delay is denoted by τ_{ij} and $(d_{ij}-1)\delta \le \tau_{ij} \le d_{ij}\delta$ where $d_{ij} \in N$. As seen from Figure 7.8 whenever there is no critical situation both the neighbouring vehicles 1 and 2 are assigned the same communication delay, namely $d_{31} = d_{32} = 4$. However, in the case of one agent in critical situation where the communication delay of one vehicle is reduced to $d=3$, the penalty is that the communication delay corresponding to the other neighbour will increase to $d=5$ to satisfy the communication constraint (7.37).

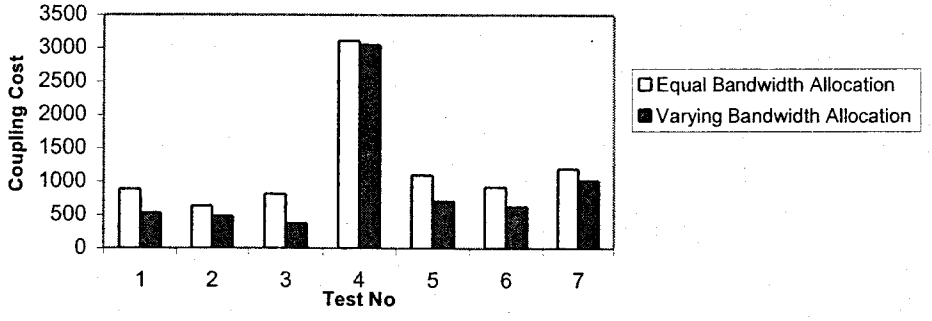

Figure 7.9: Summation of cooperation cost for the entire fleet corresponding to improved Algorithm 7.1 (Varying Bandwidth Allocation) vs. Algorithm 4.1 (Equal Bandwidth Allocation).

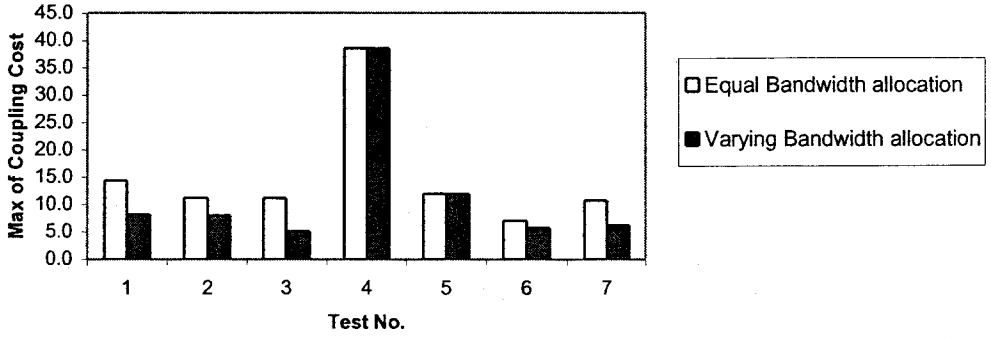

Figure 7.10: Maximum of cooperation cost of the entire fleet corresponding to improved *Algorithm 7.1* (Varying Bandwidth Allocation) vs. *Algorithm 4.1* (Equal Bandwidth Allocation).

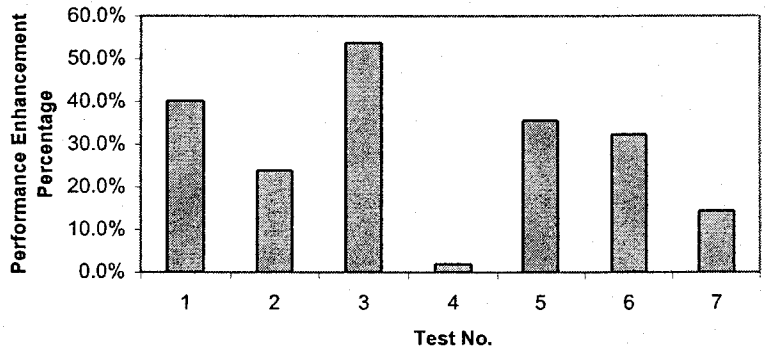

Figure 7.11: Percentage of improvement in performance using improved *Algorithm 7.1* (Varying Bandwidth Allocation) vs. *Algorithm 4.1* (Equal Bandwidth Allocation).

This simulation has been done for several cases to see the effectiveness of the approach; the results are shown in Figure 7.9, Figure 7.10 and Figure 7.11. As seen the approach can lead to more than 50% improvement in the performance.

7.4. Summary

A new delayed DRHC algorithm with bandwidth allocation capability is proposed. The bandwidth allocation algorithm works based on measuring the mismatches between the delayed and the updated trajectories and then the vehicle exposed to larger mismatch

(misunderstanding) is allocated more bandwidth. To find a relation between the delayed-DRHC and mismatches between the delayed and updated trajectories, the analytical solution of the delayed-DRHC is approximated by the solution of the non-delayed DRHC since the analytical solution of the delayed DRHC is not straightforward to calculate; contrary to this, the simulation results imply that this is an appropriate approximation and can lead to improvement in performance for DRHC. However, to further improve the effectiveness of the proposed approach the exact analytical solution of the delayed DRHC is required. The proposed method is decentralized, and does not impose significant online computation and communication loads.

Chapter 8. Conclusions and Future Work

This chapter summarizes the thesis contributions and presents future directions for the research.

8.1. Conclusions

A new framework has been developed in this thesis for decentralized receding horizon control (DRHC) of cooperative multiple vehicle systems with large communication delays. The main thesis contributions are as follows:

1) In Chapter 3, three new fault diagnosis algorithms are proposed for communication failures which lead to large delays. The research suggests that a hierarchical approach which isolates the required information for fault diagnosis and fault tolerant control can lead to more efficient algorithms with optimal communication over the network.
2) In Chapter 4, a new fault tolerant delayed DRHC approach is proposed that explicitly accounts for large communication delays. The main idea with this approach is to estimate the path of the neighbouring faulty vehicles, when they are unavailable due to large delays, by adding extra decision variables to the cost function. This enables the delayed DRHC system to use available computational resources to improve the stability and performance.

3) A new delayed DRHC approach is proposed in Chapter 5 using tube DRHC concept to provide safety of the fleet against collisions during faulty conditions. With this approach, a tube shaped trajectory is assumed around the trajectory of the faulty vehicle and the non-faulty vehicles are restricted from entering the unsafe region. Collision avoidance for two cases are addressed:
    i. Formation control problems: The safety is provided by adding the tube radius to the desired relative position.
    ii. General collision avoidance problems: Each vehicle calculates the tube around the neighbour's delayed trajectory and sets its manoeuvrability parameters such that its tube does not intersect with the neighbour's tube.
4) Feasibility, stability, and performance results are developed in Chapter 6 for the delayed DRHC approach proposed in Chapter 4. It is demonstrated that:
    i. With the proposed framework, it is possible to guarantee a lower bound on the performance.
    ii. Always there exists a feasible solution to the proposed delayed DRHC problem if it is initially feasible.
    iii. A sufficient stability condition based on the input-to-state practical stability (ISpS) argument for the proposed delayed DRHC algorithm is derived.

5) A new bandwidth allocation algorithm is proposed in order to optimize the communication periods to achieve improved control performance. The proposed algorithm adjusts the communication periods subject to communication bandwidth constraints. Simulations indicate a significant increase in cooperation performance can be achieved with this approach.

The thesis contributions are summarized in the hierarchal diagram presented in Figure 8.1. The results together provide a new framework for DRHC of cooperative vehicle systems with communication faults, large communication delays and limited communication bandwidth.

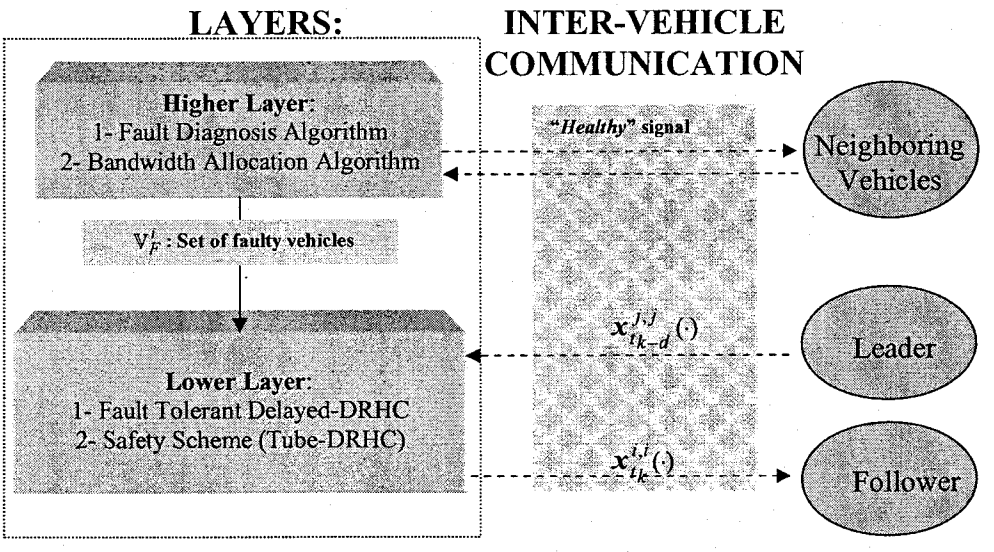

Figure 8.1: Hierarchy of the proposed approach for handling the communication delay.

8.2. Future Work

The following problems are suggested for future research:

1) Feasibility analysis of the proposed collision avoidance algorithm in Chapter 5 is required. The condition under which there always exists a feasible tube for each

vehicle in the team is required. Algorithms for resolving dead-lock situations are also required.

2) The bandwidth allocation algorithm presented in this thesis addresses only the case where the subsystems have linear dynamics. This approach could be generalized to subsystems with nonlinear dynamics by developing an efficient approach for calculating the bound on the cost function for this situation.
3) Experimental application of the proposed methods can be performed to investigate implementation issues and determined how well the proposed approaches scale for existing computation and communication hardware.
4) The effect of non-ideal situations including model uncertainty, communication noise, imperfect optimization, etc. on the proposed algorithms should be investigated.

Bibliography

[1] T. Keviczky, F. Borrelli and G. J. Balas, "Stability analysis of decentralized receding horizon control for the decoupled systems", In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, Dec. 12-15, 2005, pp. 1689-1694.

[2] W. B. Dunbar and R. M. Murray, "Receding horizon control of multi-vehicle formations: a distributed implementation", In *Proceedings of the 43rd IEEE Conference on Decision and Control,* Bahamas, Dec. 14-17, 1995-2002.

[3] A. Richards and J. How, "A decentralized algorithm for robust constrained model predictive control", In *Proceedings of the 45th American Control Conference*, 2004.

[4] M.C De Gennaro and A. Jadbabaie, "Formation control for a cooperative multi-agent system using decentralized navigation functions", In *Proceedings of American Control Conference,* 2006, June 14-16, 2006, pp. 1346 – 1351.

[5] R. Franz, M. Milam, and J. Hauser, "Applied receding horizon control of the Caltech ducted fan", In *Proceedings of the American Control Conference*, Anchorage, 2002.

[6] T. Schouwenaars, J. P. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees", In *Proceedings of the IEEE American Control Conference*, 2004, pp. 5576 - 5581.

[7] T. Schouwenaars, B. De Moor, E. Feron, and J. P. How, "Mixed integer linear programming for multi-vehicle path planning", In *Proceedings of the European Control*

Conference, European Union Control Association, Porto, Portugal, September, 2001, pp. 2603-2608.

[8] J. Bellingham, A. Richards, and J. P. How, "Receding horizon control of autonomous aerial vehicles", In *Proceedings of the IEEE American Control Conference*, May 2002, pp. 3741-3746.

[9] A. Jadbabaie, "*Receding horizon control of nonlinear systems: A control Lyapunov function approach*", PhD Thesis, California Institute of Technology, Pasadena, CA, 2000.

[10] H. Chen and F. Allgower, "A quasi infinitive horizon nonlinear model predictive control scheme with guaranteed stability", *Automatica*, Vol. 34, No. 10, 1998, pp. 1205-1217.

[11] M. B. Milam, K. Mushambi and R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems", In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, pp. 845–851, 2000.

[12] D. Henriksson, J. Akesson, "*Flexible implementation of Model Predictive Control using sub-optimal solutions*", Technical Report, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 2004.

[13] Y. Kuwata, A. Richards, T. Schouwenaars and J. How, "Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance", In *Proceedings of the 2006 American Control Conference,* Minneapolis, Minnesota, USA, June 14-16, 2006.

[14] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust constrained receding horizon control for trajectory planning", In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2005.

[15] A. Richards and J. How, "Implementation of robust decentralized model predictive control", In *Proceedings of AIAA Guidance, Navigation and Control Conference,* 2005.

[16] C. R. Cutler, B. L. Ramaker, "Dynamic matrix control – A computer control algorithm", *Proceedings of Joint Automatic Control Conference*, San Francisco, CA, 1980.

[17] A. Ashutosh, J. Patwardhan, B. Rawlings, and T. F. Edgar, "Model predictive control of nonlinear processes in the presence of constraints", *Annual AIChE Meeting*, Washington, D.C., November 1988.

[18] J. B. Rawlings and V. Manousiouthakis, "Comparing model-predictive control to the min-max approach for robust control", *Annual AIChE Meeting,* Los Angeles, California, November 1991.

[19] R. Bindlish and J. B. Rawlings, "Model predictive control of a prototypical copolymerization process", *Annual AIChE Meeting*, Dallas, Texas, November 1999.

[20] D. Q. Mayne, and H. Michalska, "Receding horizon control of nonlinear systems", *IEEE Transactions on Automatic control*, vol. 35, No. 7, July 1990. pp. 814-824.

[21] D. Q. Mayne, and H. Michalska, "Robust receding horizon control", In *Proceedings of the 30th Conference on Decision and Control*, Brighton, England, Dec. 1991.

[22] S. Keerthi and E. Gilbert, "Optimal infinite-horizon feedback laws for general class of constrained discrete-time systems: stability and moving horizon approximations", *Journal of Optimization Theory and applications*, 1988, pp. 265-293.

[23] W. H. Kwon and A. E. Pearson, "A modified quadratic cost problem and feedback stabilization of a linear system", *IEEE Transactions on Automatic Control*, vol. AC-22, 1977.

[24] W. H. Kwon, A. N. Bruckstein and T. Kailath, "Stabilizing state feedback design via the moving horizon method", *International Journal of Control,* vol. 37, 1983.

[25] C. V. Rao and J. B. Rawlings, "Optimization strategies for linear model predictive control", In *Proceedings of the 1998 IFAC DYCOPS Symposium*, Corfu, Greece, *1998*.

[26] J. W. Eaton, J. B. Rawlings, and T. F. Edgar, "Model-predictive control and sensitivity analysis for constrained nonlinear processes", In *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, Oxford, 1989, pages 129-135.

[27] J. W. Eaton and J. B. Rawlings, "Model predictive control of chemical processes", In *Proceedings of the 1991 American Control Conference*, 1991, pages 1790-1795.

[28] S. A. Middlebrooks and J. B. Rawlings, "Model predictive control of Si-Ge thin film chemical vapour deposition", *IEEE Transactions on Semiconductor Manufacturing*, Vol. 20, No. 2, pp 114-125, May 2007.

[29] M. B. Milam, "*Real-time optimal trajectory generation for constrained dynamical systems*", PhD Thesis, Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, 2003.

[30] M. B. Milam, R. Franz, and J. E. Hauser and R. M. Murray, "Receding horizon control of a vectored thrust flight experiment", In *IEE Proceedings on Control Theory and Applications*, Vol. 152, No.3., May 2005, pp. 340-348.

[31] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems", In *IEEE Transactions on Automatic Control*, Vol. 38, No. 11, November 1993, pp. 1623–1633.

[32] R. H. Findeisen, "*Suboptimal nonlinear model predictive control*", M.Sc. thesis, University of Wisconsin- Madison, 1997.

[33] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering", *International Journal of Robust and Nonlinear control*, Vol. 16, Feb 2006, pp. 333–351.

[34] P. O. M. Scokaert and D. Q. Mayne, "Min–max feedback model predictive control for constrained linear systems", *IEEE Transactions on Automatic Control*, Vol. 43, No. 8, 1998, pp. 1136–1142.

[35] D. Q. Mayne, J. B. Rawlings, C. V. Rao and P. O. Scokaert., "Constrained model predictive control: Stability and optimality", *Automatica*, Vol. 36, 2000, pp. 789-814.

[36] S. Wang and E. J. Davison, "On the stabilization of decentralized control systems", *IEEE Transaction on Automatic Control*, Vol. 18, No. 5, 1973, pp. 473-478.

[37] X. G. Yan, J. Lam, H. S. Li and I. M. Chen, "Decentralized Control of Nonlinear Large –Scale Systems Using Dynamic Output Feedback", *Journal of Optimization Theory and Applications,* Vol. 104, No. 2, 2000.

[38] M. K. Sundareshan, M. K., and Elbanna, R. M., "Qualitative analysis and decentralized controller synthesis for a class of large-scale systems with symmetrically Interconnected subsystems", *Automatica*, Vol. 27, 1991, pp. 383–388.

[39] F. Nardi, N. Hovakimyan, A. J. Calise, " Decentralized control of large-scale systems using single hidden layer neural networks", In *Proceedings of American Control Conference*, Vol. 4, 25-27 June 2001, pp. 3122–3127.

[40] Y. H. Chen, "Decentralized robust output and estimated state feedback controls for of large-scale systems", *International Journal of Control*, Vol. 46, 1987, pp. 1979–1992.

[41] W. B. Dunbar, "Distributed receding horizon control of dynamically coupled nonlinear systems ", *IEEE Transactions on Automatic Control*, Vol. 52, No. 7, 2007, pp. 1249-1263.

[42] W. B. Dunbar, "*Distributed receding horizon control of multi-agent systems*", PhD Thesis, California Institute of Technology, Pasadena, California, 2004.

[43] T. Keviczky, "*Decentralized receding horizon control of large scale dynamically decoupled systems*", PhD Dissertation, University of Minnesota, 2005.

[44] S. Fleck, S. Lanwer, W. Straber, "A smart camera approach to real-time tracking", In *Proceedings of 13th European Conference on Signal Processing*, Antalya, Turkey, Sep. 2005.

[45] J. U. Backstrom, C. Gheorghe, G. E. Stewart, and R. N. Vyse, "Constrained model predictive control for cross directional multi-array processes", *Pulp Paper Canada*, vol. 102, no. 5, May 2001, pp. T128–T131.

[46] F. Borrelli, T. Keviczky and G. E. Stewart, "Decentralized constrained optimal control approach to distributed paper machine control", In *Proceedings of 44th IEEE Conference on Decision and Control, and European Control Conference*, Seville, Spain, December 12-15, 2005.

[47] A. Bicchi and L. Pallottino, "On optimal cooperative conflict resolution for air traffic management systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, Dec. 2000, pp. 221-231.

[48] M. Egerstedt and C. F. Martin, "Conflict resolution for autonomous vehicles: A case study in hierarchical control design", *International Journal of Hybrid Systems*, vol. 2, no. 3, Sept. 2002, pp. 221-234.

[49] A. P. Aguiar and J. Hespanha, "Logic-based switching control for trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," In *Proceedings of American Control Conference*, 2004.

[50] J. H. van Schuppen, "Decentralized supervisory control with information structures," In *Proceedings of the International Workshop on Discrete Event Systems* (WODES98), IEE Press, London, 1998, pp. 36-41.

[51] P. Ogren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multi-agent coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, Oct. 2002, pp. 847-851.

[52] A. Benveniste, E. Fabre, and S. Haar, "Markov nets: probabilistic models for distributed and concurrent systems", *IEEE Transactions on Automatic Control*, vol. 48, no. 11, Nov. 2003, pp. 1936-1950.

[53] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs", In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 14-17, 2004.

[54] J. Yan and R. R. Bitmead, "Coordinated control and information architecture," In *Proceedings of the. 42nd IEEE Conference on Decision and Control*, vol. 4, Dec. 2003, pp. 3919-3923.

[55] D. H. Shim, H. J. Kim, S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots in Dynamic Environments", In *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 4, 2003, pp. 3621-3626.

[56] M. Tillerson, G. Inalhan and J. P. How, "Coordination and control of distributed spacecraft systems using convex optimization techniques", *International Journal of Robust and Nonlinear Control*, vol. 12, 2002, pp. 207–242.

[57] E Bacconi, E. Mosca and A. Casavola, "Formation flying control of a pair of nano-satellites based on switching predictive control", In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, USA, December 2003.

[58] D. Shim, H. Kim, and S. Sastry, "*Decentralized reflective model predictive control of multiple flying robots in dynamic environment*", Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Tech. Rep., 2003.

[59] T. Schouwenaars, J. P. How, and E. Feron, "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees", In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, 16 - 19 August 2004.

[60] Y. Kuwata and J. P. How, "Robust cooperative decentralized trajectory optimization using receding horizon MILP", In *Proceedings of the 2007 American Control Conference*, New York City, USA, July 11-13, 2007.

[61] E. King, Y. Kuwata, M. Alighanbari, L. Bertuccelli, and J. P. How, "Coordination and control experiments on a multi-vehicle testbed", In Proceeding *of the 2004 American Control Conference*, Boston, Massachusetts June 30 - July 2, 2004.

[62] R. R. Bitmead, M. Gevers and V. Wertz, "*Adaptive optimal control-the thinking man's GPC*", Printice-Hall, New York, 1990.

[63] J. B. Rawlings and K. R. Muske, "The stability of constraint receding horizon control", *IEEE Transactions on Automatic Control*, vol. AC-38, 1993, pp. 1512-1516.

[64] G. D. Nicolao, L. Magni and R. Scattolini, "Stabilizing receding horizon control of nonlinear time varying systems", *IEEE Transactions on Automatic Control*, vol. 43, July 1998, pp. 1030-1036.

[65] J. A. Primbs, V. Nevistic, and John C. Doyle. "On control Lyapunov functions and receding horizon control: Part I background", *IEEE Transactions on Automatic Control,* 1998.

[66] J. A. Primbs, V. NevistiC, and John C. Doyle. "On control Lyapunov functions and receding horizon control: part II a receding horizon extension of pointwise min-norm controllers", *IEEE Transactions on Automatic Control,* 1998.

[67] A. Jadbabaie, J. Yu, and J. Hauser, "Stabilizing receding horizon control of nonlinear systems: A control Lyapunov function approach", In *Proceedings of the American Control Conference*, 1999.

[68] A. Jadbabaie, J. Yu, and J. Hauser, "Receding horizon control of the caltech ducted fan: a control Lyapunov function approach", In *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala, Coast-Island of Hawaii, USA, Aug, 1999.

[69] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding horizon control of nonlinear systems", *IEEE Transactions on Automatic Control*, vol. 50, No. 5, 2001, pp. 674-678.

[70] A. Jadbabaie, J. Yu, and J. Hauser, "On the stability of receding horizon control with general terminal cost", *IEEE Transactions on Automatic Control*, vol. 46, No. 5, May 2001, pp. 776-783.

[71] D. Limon, T. Alamo, F. Salas, and E. F. Camacho, "On the stability of constrained MPC without terminal constraint", *IEEE Transactions on Automatic Control*, Vol. 51, No. 5, May 2006, pp 832-836.

[72] D. Limon, T. Alamo, and E. F. Camacho, "Stable constrained MPC without terminal constraint", In *Proceedings of the American Control Conference*, Vol. 6, June 2003, pp 4893- 4898.

[73] B. Hu and A. Linnemann, "Toward infinite-horizon optimality in nonlinear model predictive control", *IEEE Transactions on Automatic Control*, vol. 47, no. 4, Apr. 2002, pp. 679–682.

[74] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)", *IEEE Transactions on Automatic Control*, Vol. 44, No. 3, March 1999, pp. 648-654.

[75] H. Izadi, M. Pakmehr, B. W. Gordon and C. A. Rabbath, "A receding horizon control approach for roll control of delta wing vortex-coupled dynamics", In *Proceedings of the IEEE Aerospace conference*, March 3-10, 2007, Big Sky, MT, USA.

[76] P. R. Chandler, "Cooperative Control of a Team of UAVs for Tactical Missions", *AIAA 1st Intelligent Systems Technical Conference*, Chicago, Illinois, 20 - 22 September 2004.

[77] D. Zhang, G. Xie, Junzhi Yu, Long Wang, "Adaptive task assignment for multiple mobile robots via swarm intelligence approach", *Robotics and Autonomous Systems*, Vol. 55, Issue 7, July 2007, pp 572-588.

[78] D. T. Wooden, "*Graph-based Path Planning for Mobile Robots*", PhD Thesis, Georgia Institute of Technology, 2006.

[79] S. A. Bortoff, "Path-*planning for unmanned air vehicles*", Right-Patterson Air Force Base, Ohio, August 1999.

[80] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *International Journal of Robotics Research*, Vol. 5, Issue 1, pp 90–98, 1986.

[81] L. E. Kavraki, P. Svestka, J. C. Latombe, M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 1996, pp. 566-580.

[82] M. Overmars and P. Svestka, "A probabilistic learning approach to motion planning," *Algorithmic Foundations of Robotics, the 1994 Workshop on Algorithmic Foundations of Robotics*, Edited by Goldberg, Halperin, Latombe, and Wilson, 1995.

[83] G. Song, S. Miller, and N. Amato, "Customizing PRM roadmaps at query time", In *Proceedings of IEEE International Conference on Robotics and Automation*, 2001.

[84] M. C. Lechliter, "*Decentralized Control for UAV Path Planning and Task Allocation*", MS Thesis, Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, 2004.

[85] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, Part I: Fixed Topology", In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, Dec. 2003.

[86] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, Part II: Dynamic Topology", In *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, December 2003.

[87] Y. Kuwata and J. P. How "Stable trajectory design for highly constrained environments using receding horizon control", In *Proceeding of the 2004 American Control Conference,* Boston, Massachusetts June 30 - July 2, 2004.

[88] Y. Chen, and Z. Wang, "Formation control: a review and a new consideration", In *Proceedings of the IEEE Conference on Intelligent Robots and Systems,* Aug. 2005, pp 3181- 3186.

[89] Xi. Xiaorui, E. H. Abed, "Formation control with virtual leaders and reduced communications", In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, Spain, 2005, pp. 1854- 1860.

[90] W. Ren, Y. Chen, "Leaderless formation control for multiple autonomous vehicles", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, August 2006,

[91] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory", *IEEE Transactions on Automatic Control*, Vol. 51, No. 3, March 2006.

[92] N. Moshtagh, A. Jadbabaie "Distributed geodesic control laws for flocking of nonholonomic agents", *IEEE Transactions of Automatic Control*, Vol. 52, No. 4, April 2007, pp. 681-686.

[93] J. Penders, "Robot Swarming Applications", downloaded from: http://www.cs.unimaas.nl/jaap60/papers/B31_penders.pdf.

[94] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multi vehicle cooperative control: Collective group behavior through local interaction", *IEEE Control Systems Magazine*, Vol. 27, No. 2, April 2007, pp.71–82.

[95] J. Alexander Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations", *IEEE Transactions on Automatic Control,* Vol. 49, No. 9, Sep 2004, pp. 1465- 1476.

[96] R.A. Murphey, "An approximate algorithm for a weapon target assignment stochastic program", in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems,* Edited by: P.M. Pardalos, Ed., Boston: Kluwer Academic Publishers, pp. 1-2. 1999.

[97] J. Bellingham, M. Tillerson, A. Richards, and J. How, "Multi-task allocation and path planning for cooperative UAVs", in *Cooperative Control: Models, Applications and Algorithms,* Edited by: S. Butenko, R. Murphey, and P.M. Pardalos, Kluwer Academic Publishers, 2003, pages 2341.

[98] R.W. Beard, T.W. McLain, M. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles", *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 6, Dec. 2002, pp. 911-922.

[99] M. M. Polycarpou, Y. Yang, and K. Passino, "A cooperative search framework for distributed agents", In *Proceedings of the IEEE ISIC 2001,* pp 16, 2001.

[100] D.R. Jacques, "Search, classification and attack decisions for cooperative wide area search munitions", In *Cooperative Control: Models, Applications and Algorithms,* Edited by: S. Butenko, R. Murphey, and P. M. Pardalos, Kluwer, Academic Publishers, 2003, pp 7593.

[101] J.C. Leachtenauer and R.G. Driggers, "*Surveillance and Reconnaissance Imaging Systems - Modelling and Performance Prediction*", Massachusetts: Artech House, 2001.

[102] S. Rathinam and R. Sengupta, "A safe flight algorithm for unmanned aerial vehicles" In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, USA, March 2004.

[103] P. E. Hill, W. Murray and M. H. Wright, "*Practical optimization*", Academic Press, 1981, page 65-66.

[104] M. Pakmehr, B.W. Gordon and C.A. Rabbath, "Control Oriented Modeling and Identification of Delta Wing Vortex-Coupled Roll Dynamics", In *Proceedings of the American Control Conference 2005,* Portland, Oregon, June 8-10, 2005. pp. 1521-1526.

[105] M. Pakmehr, B. W. Gordon, C. A. Rabbath, "Robust Adaptive Tracking Control of Delta Wing Vortex-Coupled Roll Dynamics subject to State Delay and SMA Actuator Dynamics", In *Proceedings of the IEEE international conference on Mechatronics and Automation*, Niagara Falls, Canada, July 2005. pp. 980-986.

[106] X. Z. Huang, "*Non-Linear Indicial Response and Internal State-Space (NIRISS) Representation and Its Application on Delta Wing Configurations*", RTO Technical Report, RTO-TR-047.

[107]] X. Z. Huang, H. Y. Lou, E. S. Hanff, 2002, "Non-Linear Indicial Response and Internal State-Space Representation for Free-to-Roll Trajectory Prediction of a 65° Delta Wing at High Incidence", *AIAA Paper* 2002-4713.

[108] M. J. D. Powell, "Introduction to Constrained Optimization", *Numerical Methods for Constrained Optimization*, Edited by: P. E. Gill, W. Murray, Academic Press, 1974.

[109] Eduardo Camponogara, Dong Jia, Bruce H. Krogh, and Sarosh Talukdar, "Distributed Model Predictive Control", *IEEE Control Systems Magazine*, Feb 2002, pp 44-52.

[110] V. Gavrilets, B. Mettler, E. Feron, "*Dynamic Model for a Miniature Aerobatic Helicopter*", MIT Internal Report.

[111] S. A. Dyer and J. S. Dyer, "Cubic-Spline Interpolation: Part 1", *IEEE Instrumentation & Measurement Magazine*, March 2001, pp.44-46.

[112] P. E. Gill, W. Murray and M. A. Saunders, "*User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*".

[113] T. Keviczky, F. Borrelli and G. J. Balas, "A Study on Decentralized Receding Horizon Control for Decoupled Systems", In *Proceeding of the 2004 American Control Conference*, Boston, Massachusetts June 30 - July 2, 2004.

[114] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, Feb. 2002.

[115] H. Izadi, B. W. Gordon, C. A. Rabbath, "Stability Improvement for Time Varying Decentralized Receding Horizon Control Systems", In *Proceedings of 13th IEEE IFAC International Conference on Methods and Models in Automation and Robotics (MMAR)*, Szczecin, Poland, 27 - 30 August 2007.

[116] H. Izadi, B. W. Gordon, "Decentralized Receding Horizon Control of Multiple Rotorcrafts Using Variable Communication Approach", In *Proceedings of the International Conference on System Identification and Control Problems (SICPRO 08)*, Moscow, Russia, Jan. 2008.

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

[118] J. A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations", *IEEE Transactions on Automatic Control*, VOL. 49, NO. 9, Sep. 2004, pp. 1465-1476.

[119] R. Olfati-Saber and Richard M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays", *IEEE Transactions on Automatic Control*, VOL. 49, NO. 9, Sep. 2004, pp. 1520-1533.

[120] L. Shi, C. Ko, Z. Jin, D. Gayme, V. Gupta, S. Waydo and R. M. Murray, "Decentralized Control across Bit-Limited Communication Channels: An Example", In *Proceedings of the American Control Conference*, Portland, OR, USA June 2005.

[121] E. Franco, T. Parisini and M. M. Polycarpou, "Cooperative Control of Discrete-Time Agents with Delayed Information Exchange: a Receding-Horizon Approach", In *Proceedings of 43rd IEEE conference on Decision and Control*, Atlantis, Paradise Island, Bhamas, Dec. 2004.

[122] E. Franco, T. Parisini and M. M. Polycarpou, "Stable Receding-Horizon Cooperative Control of a Class of Distributed Agents", In *Proceedings of the American Control Conference*, Portland, OR, USA, June 2005.

[123] E. Franco, T. Parisini and M. M. Polycarpou, "Cooperative Control of Distributed Agents with Nonlinear dynamics and Delayed Information Exchange: a Stabilizing Receding-Horizon Approach", In *Proceedings of the 44rd IEEE conference on Decision and Control and the European Control Conference*, Seville, Spain, Dec. 2005.

[124] Richards, A. G. and How, J. P., "Implementation of Robust Decentralized Model Predictive Control," *AIAA Guidance, Navigation and Control Conference*, AIAA, August 2005.

[125] Venkat, A. N., Rawlings, J. B., and Wright, S. J., "Stability and optimality of distributed model predictive control," In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, 2005, pp. 6680-6685.

[126] Alvarado, I., Limon, D., Alamo, T., and Camacho, E. F., "Output Feedback Robust Tube Based MPC for Tracking of Piece-Wise Constant References," In *Proceedings of the 46th IEEE Conference on Decision and Control,* 2007, pp. 2175-2180.

[127] Trodden, P., and Richards, A., "Robust Distributed Model Predictive Control Using Tubes," In *Proceedings of the American Control Conference*, 2006, pp 2034-2039.

[128] Zhang, X., Xu, R., Kwan, C., Haynes, L., Yang, Y., and Polycarpou, M. M., "Fault Tolerant Formation Flight Control of UAVs", *International Journal of Vehicle Autonomous Systems*, Vol. 2, No.3/4, 2004, pp. 217-235.

[129] C. E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, (July and October), 1948.

[130] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem", *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, July 1982, pp: 382–401.

[131] C. Miguel and, B. Liskov, "Practical Byzantine fault tolerance", the *3rd symposium on the Operating Systems Design and Implementation*, New Orleans, USA, February 1999.

[132] Innocenti, M., and Pollini, L., "Management of Communication Failures in Formation Flight", *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 1, 2004, pp. 19-35.

[133] Cheng, L., Wang, Y., and Zhu, Q., "A Communication-Based Multi-Robot Rigid Formation Control System: Design and Analysis," *International Journal of Modeling, Identification and Control,* Vol. 1, No. 1, 2006, pp. 13-22.

[134] Richards, A. G., and How, J. P., "Robust Distributed Model Predictive Control," *International Journal of Control*, 2007, Vol. 80, pp. 1517-1531.

[135] Garagic, D., "On Delay-Dependent Stability of a Swarm of Networked Autonomous Vehicles under Communication Constraints," In *Proceedings of 2005 IEEE Swarm Intelligence Symposium*, 2005, pp. 289-293.

[136] Olfati-Saber, R., and Murray, R. M., "Consensus Problems in Networks of Agents with Switching Topology and Time Delays," *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, 2004, pp.1520-533.

[137] Rum, F. and B. W. Gordon, "Distributed Simulation of Differential-Algebraic Systems using Optimal Sliding Mode Control", *IEEE Systems, Man, and Cybernetics*: Part C, March 2007, issue 2, v 37, p 222-233.

[138] R. Brockett, "Stabilization of motor networks", *the 34th IEEE Conference on Decision and Control*, 1995, pp. 1484–1488.

[139] W.S. Wong and R.W. Brockett, "Systems with finite communication bandwidth constraints II: Stabilization with limited information feedback", *IEEE Transactions on Automatic Control*, vol. 44, pp. 1049 – 1053, 1999.

[140] D. Hristu and K. Morgansen, "Limited communication control", *Systems & Control Letters*, vol. 37, pp. 193-205, 1999.

[141] D. Hristu-Varsakelis, "Feedback control systems as users of a shared network: communication sequences that guarantee stability", IEEE *Conference on Decision and Control*, vol. 4, 2001, pp. 3631-3636.

[142] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control", *Automatica*, vol. 40, pp. 491-500, 2004.

[143] X. Lin, M. Johansson, H. Hind, S. Boyd. and A. Goldsmith, "Joint optimization of communication rates and linear systems", *IEEE Transactions on Automatic Control*, vol. 48, 2003, pp. 148-153.

[144] J.P. Richard, "Time-delay systems: an overview of some recent advances and open problems", *Automatica*, vol. 39, 2003, pp. 1667-1694.

[145] Dunbar, W., and Murray, R., "Model Predictive Control of Coordinated Multi-Vehicle Formations", *Proceedings of the 41st IEEE Conference on Decision and Control,* Vol. 4, 2002, pp. 4631-4636.

[146] Wesselowski, K., and Fierro, R., "A Dual-Mode Model Predictive Controller for Robot Formations," In *Proceedings Of the 42nd IEEE Conference on Decision and Control*, Vol. 4, 2003, pp. 3615-3620.

[147] Gu D., and Yang, E., "A Suboptimal Model Predictive Formation Control", In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1295- 1300.

[148] Aguiar A. P., Cremean L., Hespanha J. P., "Position Tracking for a Nonlinear Underactuated Hovercraft: Controller Design and Experimental Results", In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, pp. 3858–3863.

[149] Izadi, H., Gordon, B. W., & Rabbath, C. A., "Decentralized Receding Horizon Control Using Communication Bandwidth Allocation", In *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec 9-11.

[150] Izadi, H., Gordon, B. W., & Zhang, Y. M., "Reconfigurable Decentralized Receding Horizon Control of Multiple Vehicles subject to Communication Failure", *Proceedings of the 2009 American Control Conference.*

[151] Keviczky, T., Borrelli, F., & Balas, G. J. "Decentralized Receding Horizon Control for Large Scale Dynamically Decoupled Systems", *Automatica,* 42(12), pp 2105-2115.

[152] Khalil, H. K. "*Nonlinear Systems*", 2nd ed. New Jersey: Prentice Hall Inc. 1996.

[153] Langson, W., Chryssochoos, I., Rakovic, S. V., & Mayne, D. Q., "Robust Model Predictive Control Using Tubes", *Automatica,* 2004, 40(1), pp 125-133.

[154] Mehra, R. K., Boskovic, J. D., & Li, S.-M., "Autonomous formation flying of multiple UCAVs under communication failure" In *IEEE Position, Location and Navigation Symposium*, San Diego, USA, March 13-16 2000, 371-378.

[155] Ousingsawat, J., & Campbell, M. E. "Optimal Cooperative Reconnaissance Using Multiple Vehicles", *AIAA Journal of Guidance, Control and Dynamics,* 2007, 30(1), 122-132.

[156] Tino, R., Terra, M. H., & Bergerman, M. "A Fault Tolerance Framework for Cooperative Robotic Manipulators" *Control Engineering Practice,* 2007, 15(5), 615-625.

[157] Trodden, P., & Richards, A. "Robust Distributed Model Predictive Control Using Tubes", In *the American Control Conference*, Minneapolis, USA, June 14-16 2006, 2034-2039.

[158] Weitz, L. A., & Hurtado, J. E., "Investigating Time-Delay Effects for Multivehicle Formation Control", *AIAA Journal of Aerospace Computing, Information, and Communication,* 2008, 5(9), 321-336.

[159] Zhang, Y. M., & Jiang, J., "Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems", *Annual Reviews in Control*, 2008, 32(2), 229-252.

[160] Azimi, A. Gordon, B. W, Rabbath, C. A., "Dynamic Scheduling of Receding Horizon Controllers with Application to Multiple Unmanned Hovercraft Systems", In *Proceedings of the American Control Conference,* New York, USA, July 2007, Page(s):3324 – 3329.

[161] H. Izadi, B. W. Gordon, C. A. Rabbath, "A Variable Communication Approach for Decentralized Receding Horizon Control of Multi-Vehicle Systems", In *Proceedings of the American Control Conference,* New York City, USA, July 11-13, 2007.

[162] T. Keviczky, B. Vanek, F. Borrelli, and G. J. Balas, "Hybrid decentralized receding horizon control of vehicle formations", *Proceedings of the American Control Conference*, Minneapolis, Minnesota, USA, June 14-16, 2006, pp. 3358-3363.

[163] H. Fukushima, K. Kon and F. Matsuno, "Distributed Model Predictive Control for Multi-Vehicle Formation with Collision Avoidance Constraints", In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, December 12-15, 2005.

[164] Kuwata, Y.; Richards, A.G.; Schouwenaars, T. & How, J.P., "Distributed Robust Receding Horizon Control for Multi-vehicle Guidance", *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 4, July 2007, Pages 627--641

[165] F. Borrelli, T. Keviczky and G. J. Balas, "Collision-Free UAV Formation Flight Using Decentralized Optimization and Invariant Sets", In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 14-17, 2004.

[166] F. Borrelli, T. Keviczky, K. Fregene, and G. J. Balas, "Decentralized Receding Horizon Control of Cooperative Vehicle Formations", In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, Dec. 12-15, 2005.

[167] Kirk D. E., "*Optimal Control Theory: An Introduction*", Dover Publications, April 2004.

[168] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos, "Effects of limited bandwidth communications channels on the control of multiple robots", In *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, HI, USA, Oct. 2001.

[169] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of Studies in Applied Mathematics Society for Industrial and Applied Mathematics (SIAM), 1994.

[170] J. Krozel, and M. Peters, "A decentralized control strategy for distributed air/ground traffic separation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, Aug. 14-17, 2000.

[171] J. Krozel, and M. Peters, "Decentralized control techniques for distributed air/ground traffic separation" NASA Report: TR 99RTO36-03, NASA Ames Research Center, Moffett Field, CA 94035.

[172] L. Pallottino, V.G. Scordio, E. Frazzoli, and A. Bicchi, "Decentralized cooperative policy for conflict resolution in multi-vehicle systems", *IEEE Trans. on Robotics*, 23(6):1170-1183, Dec. 2007.

[173] L.D. Baskar, B. De Schutter, and H. Hellendoorn, "Decentralized traffic control and management with intelligent vehicles", the 9th Trail Congress, Rotterdam, The Netherlands, Nov. 2006.

[174] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies" *Proceedings of the IEEE*, Vol. 91, NO. 12, Dec. 2003.

[175] L. Pallottino, V. Giovanni Scordio, and A. Bicchi, "Decentralized cooperative conflict resolution among multiple autonomous mobile agents", *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 14-17, 2004, pp 4758-4763.

[176] Y. Yoon, H. Kim, J. Shin, T. Choe, and Y. Park, "Communication in distributed model predictive collision avoidance", *1st International Conf. on Robot Communication and Coordination*, Athens, Greece, 2007.

[177] A. Richards, and J. How, "Decentralized model predictive control of cooperating UAVs", *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 14-17, 2004.

[178] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations", *IEEE Transactions on Control Systems Technology* vol. 16, No.1, Jan. 2008.

[179] D. Limon, T. Alamo, F. Salas, E.F. Camacho, "Input to state stability of min–max MPC controllers for nonlinear systems with bounded uncertainties", *Automatica* 42 (2006), 797 – 803.

[180] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in hard real time environment", *Journal of the Association for Computing Machinery*, vol. 20, No.1, 1973, pp. 40-61.