Model predictive based UAV formation flight control : formulation, extension and experiment

Zhao, Weihua

2012

Zhao, W. (2012). Model predictive based UAV formation flight control : formulation, extension and experiment. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50733>

<https://doi.org/10.32657/10356/50733>

Model Predictive Based UAV Formation Flight Control: Formulation, Extension and Experiment

ZHAO Weihua#

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University

In partial fulfillment of the requirement for the degree of

Doctor of Philosophy

2012

DEDICATION

To

My supervisor

And

My parents

ACKNOWLEDGEMENTS

I am sincerely grateful to all the individuals who have contributed to the successful completion of my graduate studies and this research thesis.

First and foremost, I wish to express my deepest appreciation and gratitude to my supervisor, Professor Go Tiau Hiong, for his excellent guidance, constructive suggestions, exact insight and profound knowledge in supervising my research project. Without his encouragement and enlightenment that carried me on through difficulties, I could not come this far so smoothly for my graduate study.

I would thank Mr. Seet Thian Beng, Mr. Ng Tian Seng and Mr. Ting Li Yong for the valuable technical assistance. I would also be grateful to Mr. Chiew Soon Hooi, Mr. Wang Jian, Mr. Huang Mingxing, Mr. Guo Jingjun and Mr. Wan Chao for their help whenever needed.

Last but not least, I would like to express my infinite gratitude to my parents. Their unconditional love and support enabled me not to be lost in ambiguous and uneasy circumstances. All successful results in my research are from their supports and understanding.

Abstract

This thesis considers the problem of unmanned aerial vehicle (UAV) formation flight control. This problem has received significant attention recently in the control, robotics and UAV community due to its numerous potential applications ranging from simple to complex tasks. The focus of this thesis is on the development of formation flight control in hierarchical fashion based on model predictive control (MPC) approach. This approach is interesting in its own right and in this thesis three novel variants of MPC formation flight control approaches are developed. Each variant is developed to achieve improvement in terms of performance, robustness and/or safety.

The first variant utilizes multiplexed MPC to convert the computationally-heavy centralized formation flight control problem into computationally-feasible decentralized ones while addressing the robustness issue. Under this method, the whole centralized formation flight problem is divided into decentralized subsystems. One major advantage of this method is that the closed-loop stability of the whole formation flight system is always guaranteed even if a different updating sequence is used, which makes the scheme flexible and able to exploit the capability of each UAV fully. The obstacle avoidance scheme in MPC formation flight control is extended from the literature. By combining the spatial horizon and the time horizon, the small pop-up obstacles avoidance is transformed into additional convex position constraint in the MPC online optimization.

Considering the complexity of the UAV dynamics and in order to realize the formation flight, the second formation flight control variant proposes the combination of MPC and robust feedback linearization. A specific type of UAV, i.e. the quadcopter is considered in the formation flight. In order to achieve real-time MPC optimization, two modifications, i.e. the control input hold and variable prediction horizon, are made. Formation flight experiments are set up in the Vicon environment and the flight results demonstrate the effectiveness of the proposed formation flight architecture.

The last formation flight control variant proposes the combination of MPC with adaptive neural networks. In order to make the formation flight safer during extreme maneuvers, the reachability algorithm based on the safe maneuver envelope is proposed in the MPC framework. The “Pseudo-Control Hedging” technique is also implemented in the MPC formation flight planner resulting in the tight coupling between the attitude adaptive control and the MPC planner. Formation flight simulations are then conducted based on detail-modeled quadcopter in Simulink/Matlab.

In this work, the formation flight algorithms are validated throughout a series of simulations and for the cases where hardware realization is feasible, experiments using quadcopters as the test-bed vehicles. Technical challenges as well as recommendations for future work are highlighted.

Table of Contents

ACKNOWLEDGEMENTS	I
Abstract	II
Table of Contents	IV
List of Figures	VIII
List of Tables	X
Nomenclature	XI
Chapter 1 Introduction	1
1.1 Background & Motivation	1
1.2 Classes of Cooperative Controllers	3
1.3 Formation Flight Scenario & Research Objectives	8
1.4 Contribution of This Research	8
1.5 Organization of the Thesis	9
Chapter 2 Literature Review	11
2.1 Formation Flight Control Schemes	11
2.2 Formation Flight Controllers	12
2.2.1 Linear State Feedback Flight Controller	13
2.2.2 Nonlinear Formation Flight Controller	15
2.2.3 Model Predictive Control (MPC) for Formation Flight Control	16
2.3 Chapter Summary	18
Chapter 3 Model Predictive Control Approach for Formation Flight	20

3.1 Introduction	20
3.2 Model Predictive Control Development for Formation Flight Control.....	21
3.2.1 Motivation.....	21
3.2.2 Formation Kinematics	21
3.2.3 MPC Formation Flight Formulation	24
3.2.4 Formation Pair Stability.....	29
3.2.5 Inclusion of the Constraints in the MPC Formulation	31
3.3 Chapter Summary	34
Chapter 4 Robust Decentralized Formation Flight Control using Multiplexed	
MPC.....	35
4.1 Introduction	35
4.2 Formation Flight Control Strategy Overview.....	38
4.3 UAV Subsystem Dynamics and Overall Formation Flight System under	
MMPC	40
4.3.1 Dynamics of UAV Subsystem	40
4.3.2 Formation Flight System under MMPC	43
4.4 Robust Decentralized Formation Flight	45
4.4.1 Stability of the Formation Flight Under MMPC.....	50
4.5 Collision Avoidance Integration	52
4.5.1 Collision Avoidance of Large Apriorily-known Obstacles	53
4.5.2 Collision avoidance of small or non-apriorily known obstacles	55
4.6 Simulation Results.....	56
4.7 Concluding Remarks	64
Chapter 5 Specific Application to Quadcopters MPC Formation Flight.....	66

5.1 Introduction.....	67
5.2 Quadcopter dynamics.....	70
5.3 Two-Layer Formation Flight Framework	73
5.4 Top-Layer MPC-Based Formation Flight Control.....	74
5.4.1 MPC Formation Flight Problem Statement	76
5.4.2 Robust feasibility of the formation flight	79
5.4.3 Formation Stability and Finite Time Formation Reachability	81
5.5 Bottom-Layer Controller Design Based On Robust Feedback Linearization .	84
5.5.1 Robust Feedback Linearization Control of the Rotational Motion.....	85
5.5.2 Feedback Linearization Control on the Translational Motion	87
5.6 Stability Statement of the Two-Layer Formation Flight Control Framework .	88
5.7 Formation Flight Test.....	91
5.7.1 Experimental flight-test facility.....	92
5.7.2 Formation Flight Control Computational Setup	95
5.7.3 Experimental Results.....	95
5.8 Concluding Remarks.....	102

Chapter 6 Combination of Adaptive Neural Networks and Reachability

Algorithm in MPC Formation Flight Framework.....	104
6.1 Introduction.....	105
6.2 Input Saturation and Adaptive Controllers	106
6.3 Attitude Loop Controller Description	108
6.3.1 Dynamic Inversion Technique	108
6.3.2 The Pseudo-Control Signal.....	108
6.3.3 The NN Adaptive Element.....	109

6.4 Formation Flight Loop Description: Combining Reachability Analysis and PCH in MPC	112
6.4.1 Introduction to the Reachability Idea in Flight Adaptive Control.....	112
6.4.2 Maneuver Envelope and Reachability	113
6.4.3 Reachable Sets	116
6.4.4 Safe Maneuver Envelope Through Reachable Set Analysis and Computation in MPC Framework	118
6.4.5 Reachability and PCH Algorithm Implementation in the MPC Formation Flight Loop.....	119
6.5 Simulation Results.....	123
6.5.1 MPC Formation Flight Loop Setup	123
6.5.2 NN Adaptive Based Attitude Control Loop Setup	124
6.5.3 Simulation Result.....	125
6.6 Discussion and Concluding Remarks	129
Chapter 7 Conclusions and Recommendations	130
7.1 Conclusion	130
7.2 Recommended Further Research	131
APPENDIX A	134
APPENDIX B	135
Reference.....	137

List of Figures

Figure 1.1 Cooperative control metrics.....	5
Figure 2.1 Proud's formation flight control system (FFCS) Block Diagram.....	15
Figure 3.2 Formation flight control diagram and signals.....	29
Figure 3.3 Separations between Leader and W1	32
Figure 3.4 Velocity control input.....	33
Figure 3.5 Heading control input	33
Figure 4.1 The overall Formation Flight Control System under MMPC	38
Figure 4.2 Decentralized UAV formation flight control	40
Figure 4.3 Agent in the center of the cluster exchange information with most 6 neighbors	47
Figure 4.4 UAVs penetrating the small size obstacle	53
Figure 4.5 Collision avoidance with a small obstacle	55
Figure 4. 6 Collision Avoidance Flow chart.....	56
Figure 4.7 Triangle Obstacle Collision Avoidance.....	59
Figure 4.8 Formation Split and Obstacle avoidance	60
Figure 4.9 Decentralized Formation Flight with robustness and without robustness ...	61
Figure 4.10 Small-obstacle collision free formation flight	62
Figure 4.11 Multiplexed decentralized vs nominal decentralized scheme.....	63
Figure 4.12 3-D Formation Flight	63
Figure 5.1 Configuration of the Quadcopter	70
Figure 5.2 Two-layer formation flight framework	73
Figure 5.3 The time setting example for the control input holding MPC	74
Figure 5.4 Quadcopter during autonomous flight experiment in the flight-test facility of the UAV lab.....	91
Figure 5.5 Overview of the software architecture for the autonomous formation flight experiment in the flight-test facility of the UAV lab.....	94

Figure 5.6 3D formation flight trajectories	96
Figure 5.7 XY plane view of the formation flight trajectory	96
Figure 5.8 X positions of formation flight system	97
Figure 5.9 Y positions of formation flight system	98
Figure 5.10 Near 10 minutes formation flight	99
Figure 6.1 Formation flight: MPC combining NN adaptive control	106
Figure 6.2 Single Hidden Layer (SHL) Neural Network (NN)	110
Figure 6.3 Backwards and forwards reachable set.....	116
Figure 6.4 The safe envelope for a known safe set is defined by the intersection of the forwards and backwards reachable sets	117
Figure 6.5 Reachable sets for double integrator	119
Figure 6.6 online states constraints calculation based on reachable sets	120
Figure 6.7 online states constraints calculation based on reachable sets	124
Figure 6.8 XYZ position history	127
Figure 6.9 Euler angle history	127
Figure 6.10 Thrust and moments history	128
Figure 6.11 Pitch channel NN weights history	128

List of Tables

Table 1.1 Formation Flight Control Schemes	7
Table 1.2 Formation Flight Control Methods.....	7
Table 5.1: Key parameters of the formation flight system used for the formation flight experiments.....	92
Table 5. 2 Quadcopter Specific Parameters.....	93
Table 5.3: Computational load and the prediction horizon H with respect to different MPC input holding length N	101
Table 5.4: Comparison between the conventional MPC and the proposed MPC	102
Table 6.1 NN parameters.....	124

Nomenclature

$\mathbf{V}_L^B$	Lead's inertial velocity in the rotating reference frame;
$\mathbf{V}_{WL}^B$	Lead's velocity with respect to the wing, expressed in the rotating reference frame;
$\boldsymbol{\omega}_W^B$	Angular velocity of wing in the rotating reference frame;
$\mathbf{R}_{WL}^B$	Position of lead with respect to wing in the rotating reference frame;
$\mathbf{V}_W^B$	Wing's velocity in the rotating reference frame
τ_ψ	Time constant of the yaw autopilot
τ_v	Time constant of the speed autopilot
I	<i>Identity matrix</i>
$\mathbf{J}$	Cost function
$\mathbf{f}(\cdot)$	System dynamics
$\mathbf{h}(\cdot)$	Output function
T	Thrust
$[X \ Y \ Z]$	Position
$[\phi \ \theta \ \psi]$	Euler angles
$[p \ q \ r]$	Angular velocities
H	Prediction horizon
$\mathcal{Q}_f$	Target set
T_d	Model discretization sampling time
T_{MPC}	MPC updating interval
$\boldsymbol{\tau}_\eta$	Quadcopter control torque
$\mathbf{v}_\eta$	Pseudo-control signal
v_{ad}	Output of neural network

v_h	Hedge signal
$\sigma(.)$	Neural sigmoid activation function
Γ_v, Γ_w	Neural learning rates
λ_v, λ_w	Modification gain

Chapter 1 Introduction

1.1 Background & Motivation

Many scientists observed that birds fly frequently in “V” shape formation in nature, and they have proposed lots of reasons for the behavior. Formation flight of birds can simply be regarded as birds’ natural social behavior [1], or it can be driven by the need to avoid and defend against their predators [2]. The formation patterns exhibited in some bird formations become the motivations for the research of the aircraft flight formation.

Unmanned Aerial Vehicle (UAVs) will be even more demanded in the near future to autonomously perform complex, cooperative tasks such as intelligence surveillance, reconnaissance, and cooperative wide area search/destroy. As such, some advanced control algorithms for making decisions that meet team goals and mission objectives are needed for these UAVs. Cooperative control of UAVs is a complicated problem due to the task coupling, high degree of uncertainties in both dynamics and environment, limited information, etc.

The formation flight control problem has been extensively discussed in recent years with numerous applications on aircraft and spacecraft systems. The main motivation for UAV formation stems from the fact that the group performance should exceed the sum of the performance of the individual UAVs. Such cooperation is only possible when the UAVs have a high level of autonomy. Part of the growing interest

in UAVs stems from the increasing feasibility of achieving aircraft to aircraft communication, navigation, and collision avoidance due to advances in small-packaged and powerful processors, cameras, and wireless networking [3]. The cooperative UAV control problem is of further interest due to the challenges of distributed control over wireless networks, the highly non-linear flight dynamics, and the difficulty of real-time computer vision and depth perception.

Motivated by the formation of birds stated above, close formation flight of aircraft can be of benefit in many situations. The cooperative flight of a group of small, inexpensive UAVs (unmanned aerial vehicles) is of great usefulness in military and civilian applications, including mapping, patrolling, and search and rescue operations. These tasks may be repetitive or dangerous for humans, making them ideal for autonomous vehicles. In these kinds of applications, the group of UAVs becomes a mobile sensor network, and must have certain capabilities including aircraft to aircraft communication, navigation, and inter-agents collision avoidance. Similar to the bird formation, by flying in a tight formation, aircraft can achieve aerodynamic efficiency much greater than when flying alone. Such an effect can be of interest to commercial aviation to reduce fuel costs on long flights to common destinations. In addition to manned applications, formation flight can be applied for unmanned aerial vehicles (UAVs). For example, in order to operate on solar power alone and remain aloft overnight [4], these vehicles must gain enough energy during daylight for operation over 24 hours, as the energy for overnight operation must be stored during the daytime. Thus efficiency is a paramount concern for these UAVs. By increasing the aspect ratio of an aircraft's wings to decrease the induced drag, one

can improve the aerodynamics efficiency. However, the constraints of an extremely lightweight design imply that a vehicle with a very high aspect ratio would be quite fragile. An alternative way is to design several smaller aircrafts, each with aspect ratio a , and fly them in a formation. The effect would be that of a single craft with aspect ratio $n*a$ (n is the number of aircraft) in the idealized case of wing-tip to wing-tip flight.

In addition to gaining some aerodynamic advantages, formation flight has other benefits. It can also be useful for reconnaissance [5-6], atmospheric study, and as relay stations for telecommunication equipment [7]. A formation of high altitude aircraft would also be an ideal platform for wide aperture sensors [8]. If an aircraft in the formation experiences a failure, it can be recalled for repairs. The formation can just continue to fly without the faulty aircraft until a replacement aircraft can join the formation [3].

1.2 Classes of Cooperative Controllers

Although there are many ways to classify the cooperative controllers, a useful set of metrics is organization structure, information structure, and performance level. A particular cooperative controller for a specific problem can focus on one of these metrics although it will contain all the elements.

Figure 1.1 shows the three different measure metrics i.e. performance, information and organization, for cooperative controllers.

Centralized controller

Centralized means that all the state information from the UAVs is sent to a

centralized UAV/ground control station (GCS) which computes the individual control action for each UAV in the formation group. No independent action is taken by the UAVs. Centralized control can render optimal control action since the complex constraints and couplings between UAVs can be properly accounted for and not assumed away. Algorithms used for centralized control include dynamic programming [9], large linear programming [10] and nonlinear programming [11]. However, this kind of organization can lack significant robustness, and it is computationally complex which would impose high costs in computation and communication. Furthermore, it does not scale well due to the curse of dimensionality. Thus, a centralized controller would have to be tailored to each formation, i.e. if the number of wingmen in a formation were to change, the whole system controller would have to be redesigned. The traditional centralized controller is sitting in the upper-right back section of the cube in Figure 1.1.

Decentralized controller

In contrast to centralized controller, decentralized controller [12-14] needs little if any global information and it is implemented in every UAV. In general, decentralized controller's performance is low due to the absence of communication and little global coordination information. A weak disturbance on a UAV near the front of the formation can result in large oscillations on a few UAVs behind, so that preventing the formation from enjoying much of the potential gain in efficiency. Some techniques used are biological analogies/biomimetics [15], potential fields [16] and information consistency scheme [17-19], which is an asynchronous voting scheme with consistency guarantees. The lower-left front section of the cube in

Figure 1.1 presents the classical decentralized controller.

Hierarchical Controller

The upper-right section in Figure 1.1 employs decomposition, usually hierarchical. Now, only subsets of the vehicles send a subset of the local vehicle state to a team member. The team members have limited global information. While the vehicles have more autonomy and therefore robustness, it may be difficult to decompose the problem if there is appreciable task coupling. In general, this approach is more computable and scalable but is not as optimal as the centralized solution.

Predictive decentralized controller

Combining the predictive controller with decentralized organization can lead to

Figure 1.1 Cooperative control metrics

good performance as the centralized controller. Normally, in such situation, the information exchange is local, i.e. within the neighboring UAVs. Often times the individual UAVs are autonomous and capable of independent action but they share a

common objective. Since the optimization/control algorithm is predictive, this kind of controller depends very much on the UAV models and state estimation algorithm. The upper-left front section of Figure 1.1 is represents the predictive decentralized controller.

In real application, there is often no clear boundary between centralized and decentralized control organization. Generally, real-life formation flight control is affected by task coupling, dynamics/environmental uncertainty, imperfect communication and so on. The interaction of these factors renders the formation flight control a complex problem. Currently there is no universal theory of formation flight system that takes into account all these dimensions.

In formation control field, there are mainly three formation schemes:

1. Leader-follower: one agent is designated as the leader and the remaining agents follow the leader's motion offset by a relative position
2. Behavior-based: this control scheme places different weightings on certain behaviors for each agent which leads to the emergence of the group dynamics
3. Virtual structure: the formation is treated conceptually as a virtual structure with place-holders that represent the desired position for each agent.

Table 1.1 shows the advantages and disadvantages of each control scheme and Table 1.2 shows the various formation control methods used in the previous work.

Table 1.1 Formation Flight Control Schemes

	Leader-Follower	Behavioral approach	Virtual structure
Strength	Intuitive and easy to implement	It is natural to derive control strategies when agents have multiple competing objectives	Easy to prescribe a coordinated behavior for the group
Weakness	Leader is a single point of failure	Stability cannot generally be guaranteed	High communication capabilities are needed

Table 1.2 Formation Flight Control Methods

	Linear Mixer & PI Controller	Feedback Adaptive Controller	Optimal Controller
Strength	Intuitive and Easy to synthesize	Inclusion of the vortex effects	Handle the constraints
Weakness	Can not handle the constraints directly		May fall into infeasibility

From Table 1.1 and Table 1.2, the main issues in formation flight control are stability, complexity and implementation feasibility under constraints. These issues will be addressed in this thesis. The leader-follower scheme is adopted in this thesis

since this approach is easy to implement and easy to prescribe a coordinated behavior of the formation.

1.3 Formation Flight Scenario & Research Objectives

The formation flight scenario considered in this thesis is the case where only the leader aircraft knows the formation maneuver commands. In other words, the followers try to catch up with the leader in the presence of unknown leader commands and disturbances in order to keep the specified formation.

The eventual goal of this research is to come up with a stable, robust, novel and easy to implement control scheme/method which can be applied on general aircraft formation flight (including manned/unmanned aircraft, fixed wing/rotor wing aircraft). The main objective of this research is to develop formation flight control system based on model predictive control (MPC) to enable collision free tight formation flight. During the tight formation flight, all the agents should be able to keep the specified formation in the presence of disturbances and uncertainty while avoiding collision with each other and with the obstacles.

1.4 Contribution of This Research

The major contributions achieved in this research work are as follows:

- 1) First utilizes multiplexed MPC (MMPC) to convert the computationally-heavy centralized formation flight control problem into computationally-feasible decentralized ones while addressing the robustness issue. Under this method, the whole centralized formation flight problem is divided into decentralized subsystems. One major advantage of this method is that the

closed-loop stability of the whole formation flight system is always guaranteed even if a different updating sequence is used, which makes the scheme flexible and able to exploit the capability of each UAV fully.

- 2) The obstacle avoidance scheme in MPC formation flight control is extended from the literature. By combining the spatial horizon and the time horizon, the small pop-up obstacles avoidance is transformed into additional convex position constraint in the MPC online optimization.
- 3) Introduces the use of reachability analysis to augment the MPC formation flight framework. This augmentation, combined with adaptive neural networks and PCH algorithm, improves the safety of the formation flight especially during some extreme flight maneuvers or conditions, as validated through simulations.

1.5 Organization of the Thesis

The Ph.D. dissertation is organized as follows.

In Chapter 2, after giving the history of aircraft formation flight, some formation flight controllers and schemes are reviewed.

Chapter 3 details the MPC framework for UAV formation flight control problem. The formation kinematics is derived, formation pair stability is provided and formation flight simulations for centralized/decentralized control scheme are conducted under the MPC framework.

In Chapter 4, a new framework for UAV formation flight and coordination is introduced. After the whole formation flight problem is divided into decentralized

subsystems, the decentralized robust MPC formulation for each UAV subsystem with bounded input disturbances and model uncertainties is presented. The extension on obstacle avoidance scheme is also addressed in this chapter

Chapter 5 presents a practical formation flight control approach for quadcopter which combines the MPC and robust feedback linearization technique. After overview the formation flight framework, two modifications, i.e. the control input hold and variable prediction horizon are made on the MPC formation flight planner to allow real-time optimization. The formation flight tests are setup and conducted to verify the formation flight method.

Chapter 6 proposes a novel reachability algorithm in the MPC framework and combine it with adaptive an NN element which is suitable for extreme formation maneuver. The reachability algorithm is based on the safe operating envelope concept. The Pseudo Control Hedge technique is also implemented into the MPC formation controller to prevent the NN adaptive element to adapt the input saturation.

Finally, conclusions and recommendations are presented in Chapter 7.

Chapter 2 Literature Review

In this chapter, the works that have been done in the formation flight control field are reviewed. Although most of the literatures reviewed in this chapter are concerned with the fixed-wing aircraft formation flight, the control schemes and the formation flight control methods used can be applied to the rotary wing craft in some cases.

2.1 Formation Flight Control Schemes

In the field of coordination of multiple agents, there are roughly three control schemes reported in the literature, namely leader-following, behavioral, and virtual structures (or virtual leader). This section highlights the advantages and disadvantages of each flight formation control scheme.

The first one is the leader-follower scheme, where one agent is designated as the leader and the remaining agents follow the leader's motion offset by a relative position [18, 20-22]. In this scheme, the commands can be assumed to be sent to the leader only. The advantage of a leader-follower scheme is that the formation flight group maneuvers can be specified in terms of the leader's motion. It has a disadvantage in that the leader's motion is independent of the wingmen, which means if the following agents are unable to maintain a small tracking error, the leader does not "wait" and then the formation can be broken.

The second common approach to formation keeping is the behavior-based

scheme [16, 23]. This control scheme places different weightings on certain behaviors for each agent which leads to the emergence of the group dynamics. The advantage of the behavior-based schemes is that the group dynamics implicitly contain formation feedback by coupling the weightings of the actions. However, it comes with the main disadvantage in that it is difficult to prescribe formation maneuvers while guaranteeing its stability due to the emergence of group dynamics that are difficult to model.

The third approach is the virtual structure scheme [24-25]. In this scheme, the formation is treated conceptually as a virtual structure with place-holders that represent the desired position for each agent. The place-holders trace out trajectories for each agent to track as the formation, which is represented as a virtual rigid body, evolves in time. The advantage of this control scheme is that it is fairly easy to prescribe a coordinated behavior for the group. The disadvantage is that it requires the formation to act as a virtual structure, which limits the potential applications of this approach.

After comparing these three control schemes, the leader-follower scheme is chosen as the baseline of the investigation here, since this method is easy to prescribe to the formation maneuvers, and can guarantee the formation stability with the application of a proper formation controller. This control scheme also needs less communication channels than that of the virtual structure method.

2.2 Formation Flight Controllers

In this section the formation pair (one leader and one wingman aircraft)

controllers are reviewed. For formation flight control problem, normally the formation control scheme (centralized and decentralized) and the formation pair controller design is decoupled and treated separately.

2.2.1 Linear State Feedback Flight Controller

For the formation flight control problem, the controller type used in many references can be categorized into the linear proportional–integral–derivative (PID) controller [26-28]. In [26], a formation flight controller (i.e., a formation-hold autopilot for the wingman) is designed such that the formation's geometry is maintained in the face of the leader maneuver. The additional aerodynamic coupling effects introduced by tight formation flight are modeled. The most significant aerodynamic coupling effect introduced by tight formation entails the coupling of the lateral/directional channel into the altitude-hold autopilot channel.

In [27], the aircraft dynamics are modeled as first-order dynamic systems. It is assumed that the position of the leader (L) relative to Wingman (W) is available to W , and a formation flight control system is synthesized for the automatic maintenance of the formation. In that paper, W is able to maintain its position relative to L (station keeping), in the face of maneuvers by L . Additionally, L can “command” a change $(\Delta x_r, \Delta y_r)$ in the formation's parameters, i.e., a change in the L - W lateral and longitudinal separation (x_r, y_r) .

Design and flight testing of formation control laws using a set of YF-22 research UAVs have been addressed in [28]. The research UAV YF-22 is designed, built, and instrumented at West Virginia University, US and the model of the UAV

was identified with a batch least squares (BLS) technique. The formation controller was designed to have an inner-loop and outer-loop structure, where the outer-loop guidance control laws minimized the forward, lateral, and vertical distance error by controlling the engine propulsion and generating the desired pitch and roll angles to be tracked by the inner-loop controller. The horizontal components of the outer-loop controller were designed using a nonlinear dynamic inversion approach, while the vertical channel of the outer-loop controller and the inner-loop control laws were instead linear.

In these references [26-28], the basic idea of the formation controller design is output feedback just like Proud's [26] mixer PID control. The formation flight control strategy implemented was a two tiered structure. The upper tier is designed as guidance for the flight formation. This consists of velocity, heading, and altitude controls for the formation. The lower tier is specified to control the actual formation geometry. This consists of x , y and z separation distances between the leader and wingmen in the formation. The structure of this mixed PID formation flight controller is illustrated in Figure 2.1. The main weaknesses of this approach are the lack of capability to include the input/states constraints in the design and its relatively low range of flight condition validity.

Figure 2.1 Proud's formation flight control system (FFCS) Block Diagram

2.2.2 Nonlinear Formation Flight Controller

Many researchers used gain scheduling technique and adaptive control methods to deal with the model /aerodynamics uncertainties and disturbances during formation flight, and thus the formation flight controller becomes nonlinear. Several nonlinear formation flight control methodologies [29-32] are reviewed below. Compared with the linear formation flight controller, the nonlinear ones are able to compensate the model uncertainties and be operated in a wider operating range. However, most of the nonlinear formation flight controllers [29-32] still cannot directly deal with input/states constraints. Also the obstacle avoidance is often not considered in the framework.

In [29], the vortex forces produced by the leader aircraft are treated as unknown functions. An adaptive control law is derived for the position control of the wingman aircraft based on a back-stepping design technique. The control of multiple UAVs flying in a close-coupled formation for the purpose of drag reduction is studied in [32]. In [31], a decentralized nonlinear robust control system for formation flight of multiple UAVs was addressed. The forces produced by any aircraft are treated as

unknown functions, which cannot be parameterized. A nonlinear robust control system for the separation trajectory control of the wingman aircraft in the simplified wing coordinate system is derived. The uncertain functions and unmeasured variables are estimated using a high-gain observer for the synthesis of the control system. Since each wingman UAV synthesized its formation flight control law using its own state variables and the relative position of the preceding UAV with respect to the wingman UAV, the control scheme is decentralized in this case.

2.2.3 Model Predictive Control (MPC) for Formation Flight Control

In MPC or receding horizon control (RHC), the feedback control inputs are obtained by repeatedly solving an on-line open-loop optimal control problem. The problem is posed at each sampled time and by updating the initial condition with the current state, it is solved over a finite time horizon longer than the sampling time. This control method clearly falls into a different category than the conventional control where control laws are pre-computed. MPC is thus suited for cases where off-line computation of a control input is difficult or impossible. Furthermore, it is one of the few methods in control that can incorporate constraints on the output and inputs, as the constraints are directly invoked in the on-line optimization problem.

The description of MPC here follows [33] and [34]. Reference [33] gives a very comprehensive basis of MPC with constraints. Reference [34] also concerns with constrained MPC, but adequately discusses the nonlinear unconstrained case as well. Applications of MPC to fast dynamical control problems have begun to emerge over recent years [35]. Neural Network based MPC is applied to F-16 aircraft models in [36], robots in [37], and UAVs in [38-40].

Comparing to other control techniques, MPC offers a number of unique advantages for the autonomous formation flight control in terms of the capability to:

- Deal with the constrained multiple-input-multiple-output dynamics of the UAV system by directly using its mathematical model in the control loop design
- Consider the formation flight kinematics and dynamics of the quadcopter as an entire system, which results in an integrated formation flight framework;
- Give a local path planning function by combining future reference and the environment information such as obstacles

In recent years, there are some research works done on multi-vehicle control that uses MPC [41-42]. In [41], the author designs a distributed control law attenuating external disturbances coming into a formation, so that each vehicle can safely maintain sufficient clearance with all other vehicles. While conventional methods are limited to homogeneous formations, their decentralized model predictive control approach allows for heterogeneity in a formation. The scenario in [41] is different from ours in the way that they propose to use MPC to help to diminish battlefield stress, since the potential for accidents is increased by requirements to fly in close formation and under harsh conditions including poor weather, extremely low altitudes, low visibility, extreme temperatures, noises, vibration, blasts, flashed, radiation and battlefield air pollution. So in [41], each vehicle knows the formation trajectories and the MPC controllers are implemented to resist the effects of the environment, such as crosswind. Ref. [42] considers the control of interacting subsystems whose dynamics and constraints are uncoupled, but whose state vectors

are coupled non-separably in a single centralized cost function of a finite horizon optimal control problem. The method used in this paper is virtual structure approach, while the leader-follower method is applied in our research.

There are still some issues need further investigation in multiple-UAV cooperative control:

1. The update sequence of the UAV: most works either update all the UAVs' control inputs simultaneously [39, 43-44] or update each UAV's inputs as soon as its computation is finished [41, 45-47]. The question is: how should we update the inputs for each UAV and keep the whole system stable, this issue is addressed by introducing Multiplexed MPC framework in Chapter 4.
2. In the existing formation flight literature, to our best knowledge, the safety flight maneuver envelope is not guaranteed; in Chapter 6 we propose a novel reachability algorithm for the wingman UAV, this algorithm will prevent the loss of control during extreme flight maneuvers.

2.3 Chapter Summary

In this chapter, firstly, the three formation control schemes are compared and the leader-following scheme is selected as the baseline in this research. Secondly, in the formation control methods review, the linear, nonlinear feedback control methods and MPC based formation flight controller are covered. The main limitation of the traditional linear or nonlinear feedback control methods covered above is the lack of

capability to handle constraints. This, coupled with the advantage of having a predictive feature, leads to the selection of MPC as the focus framework of investigation here. Some issues in the existing MPC based formation flight control are pointed out which forms the main research parts in this thesis. The next Chapters will detail the control approaches developed based on MPC.

Chapter 3 Model Predictive Control Approach for Formation Flight

3.1 Introduction

In model predictive control (MPC), the current control action is calculated by solving on-line, at each sampling time, a finite horizon optimal control problem. Each optimization calculates an optimal control sequence and the first control action in the sequence is applied to the plant until the next sampling time. MPC has traditionally been applied to control plants whose dynamics are slow enough to permit a sampling rate amenable to optimal input computations between samples, e.g. chemical industry process plants [48]. These systems are usually also governed by strict constraints on states, inputs and/or combinations of both. With the advance of faster powerful computers, it has become possible to extend the MPC method to systems governed by faster dynamics, for example the UAV control systems [14, 38, 43, 45, 49].

In multi-vehicle problem, a general objective is to achieve/maintain a formation and perform joint maneuvers in the formation. Generally, the vehicles are individually governed by nonlinear and constrained dynamics. The objective of this chapter is to introduce a primary theoretical framework for the application of MPC to the multi-vehicle coordinated control problem.

3.2 Model Predictive Control Development for Formation Flight Control

3.2.1 Motivation

A generalized multi-vehicle formation flight control problem is considered in this section. Individual UAV dynamics are decoupled from one another but they become coupled by virtue of common objectives that involve the desired formations. This is still an open and challenging problem, particularly when accounting for issues such as various communication topologies between vehicles, environmental uncertainties (crosswind effects), nonlinear UAV dynamics, obstacles and limitations on the UAV dynamics. Pre-computation of control laws to achieve such objectives, possibly in the presence of input/output constraints, is not obvious and often not possible. This motivates the novel extension of MPC to the multi-vehicle formation flight control problem, where constraints can be directly included in an optimal control formulation.

Control development of a multi-vehicle experimental test-bed developed at NTU is also a key motivation for this research. The individual vehicle dynamics and inputs are subject to constraints and the objectives include real-time formation maneuvers while avoiding collision with obstacles. The MPC framework is a natural choice to meet these objectives.

3.2.2 Formation Kinematics

In order to derive the kinematics equations between the leader and the wingman aircraft, a spatial coordinate and a rotating coordinate are attached to earth and the wingman respectively. For the rotating coordinate attached to the wingman, we use the stability axis system, i.e. the x^B axis is chosen to coincide with the

velocity vector of the wingman, V_w , the Y^B axis is aligned with the starboard direction of the wingman and Z^B axis completes the right-handed Cartesian coordinate system. For the definition of the spatial coordinate, the X^S axis is chosen to point North, the Y^S axis points East and the Z^S axis completes the right-handed Cartesian coordinate system. These two coordinate systems are illustrated in Figure 3.1

Figure 3.1 formation flight coordinates

The leader-wingman separation, i.e. the kinematics equations derived here are referred to the rotating coordinate attaching to the wing aircraft. Using the Velocity Transformation Law, we can get the equations below:

$$\mathbf{V}_L^B = \mathbf{V}_{wL}^B + \mathbf{V}_w^B + \boldsymbol{\omega}_w^B \times \mathbf{R}_{wL}^B \quad (3.1)$$

where

$\mathbf{V}_L^B$ = lead's inertial velocity in the rotating reference frame;

$\mathbf{V}_{WL}^B$ = lead's velocity with respect to the wing, expressed in the rotating reference frame;

$\boldsymbol{\omega}_W^B$ = angular velocity of wing in the rotating reference frame;

$\mathbf{R}_{WL}^B$ = position of lead with respect to wing in the rotating reference frame;

$\mathbf{V}_W^B$ = wing's velocity in the rotating reference frame;

Note that:

$$\mathbf{V}_L^B = C_L^B \mathbf{V}_L = \begin{pmatrix} \cos \psi_e & -\sin \psi_e & 0 \\ \sin \psi_e & \cos \psi_e & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} V_L \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} V_L \cos \psi_e \\ V_L \sin \psi_e \\ 0 \end{pmatrix} \quad (3.2)$$

$$\mathbf{V}_W^B = \begin{pmatrix} V_W \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{\omega}_W^B = \begin{pmatrix} 0 \\ 0 \\ \dot{\psi}_W \end{pmatrix}, \mathbf{R}_{WL}^B = \begin{pmatrix} x^B \\ y^B \\ z^B \end{pmatrix} \quad (3.3)$$

where $\psi_e = \psi_L - \psi_W$.

So, from Eqs. (3.2) and (3.3), we get:

$$\mathbf{V}_{WL}^B = \begin{pmatrix} V_L \cos \psi_e \\ V_L \sin \psi_e \\ 0 \end{pmatrix} - \begin{pmatrix} -\dot{\psi}_W y^B \\ \dot{\psi}_W x^B \\ 0 \end{pmatrix} - \begin{pmatrix} V_W \\ 0 \\ 0 \end{pmatrix} \quad (3.4)$$

The components of $\mathbf{V}_{WL}^B$ can be expressed as:

$$\mathbf{V}_{WL}^B = i \cdot \dot{x}^B + j \cdot \dot{y}^B + k \cdot \dot{z}^B \quad (3.5)$$

So:

$$\dot{x}^B = V_L \cos \psi_e + \dot{\psi}_W y^B - V_W \quad (3.6)$$

$$\dot{y}^B = V_L \sin \psi_e - \dot{\psi}_W x^B \quad (3.7)$$

$$\dot{z}^B = 0 \quad (3.8)$$

Eqs (3.6)-(3.8) describe the kinematics of the formation, which will be used to develop our formation flight control law.

3.2.3 MPC Formation Flight Formulation

In this and the next Chapter, each UAV is assumed to be equipped with a flight control system (autopilot). Thus, in general, the controlled UAV system will be of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \bar{\mathbf{f}}(\mathbf{x}, \mathbf{u}), \mathbf{x}(t_0) = \mathbf{x}_0, t \in [t_0, \infty), t_0 \in \mathbb{R}^+ \\ \bar{\mathbf{f}} : \mathcal{X} \subset \mathbb{R}^n \times \mathcal{U} \subset \mathbb{R}^m &\rightarrow \mathbb{R}^n\end{aligned}\quad (3.9)$$

Let us consider k vehicles within the system as follows

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i), \mathbf{x}_i(t_0) = \mathbf{x}_{i0}, t \in [t_0, \infty), i = 1, \dots, k, \mathbf{x}_i \in \mathcal{X} \quad (3.10)$$

where f_i is a second order vector field for all $i = 1, \dots, k$. For convenience, we introduce the vector notation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \mathbf{x} \in \mathcal{X}^k, \mathbf{u} \in \mathcal{U}^k \quad (3.11)$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_k \end{bmatrix}, \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_k \end{bmatrix}, \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_k \end{bmatrix}, \mathcal{X}^k = \mathcal{X} \times \dots \times \mathcal{X}, \mathcal{U}^k = \mathcal{U} \times \dots \times \mathcal{U}$$

The state vector includes position, velocity, and acceleration.

The control objective is to steer the set of states to the equilibrium (specified formation shape). Although modeled separately, the UAV (closed-loop) dynamics become coupled by virtue of the formation requirement. Suppose that the formation is specified by the integrated cost in the MPC problem posed for each UAV; each cost would then incorporate some or all of the states and inputs of the other UAVs. For any such UAV, we refer to the other UAVs that are referenced in its formation

specification as neighbors. In this sense the UAVs are cooperative, i.e. the closed-loop dynamics of every UAV is directly affected by its neighbors in the formation.

The definition of a formation of UAVs can be given as:

Given a $\mathcal{U}^k$, controlled positively invariant set $\mathbb{M} \subset \mathbb{X}^k$ of Eq. (3.11) and a formation reference $\mathcal{X}_r(t) \in \mathbb{X}^k, \forall t \geq 0$, a k -UAV formation associated with Eq. (3.11) is denoted as $\mathcal{F}(\mathbb{M}, k, \mathcal{X}_r(t))$ and defined as

$$\mathcal{F}(\mathbb{M}, k, \mathcal{X}_r(t)) = \left\{ \mathbf{x} \in \mathbb{X}^k \mid (\mathbf{x}(t) - \mathcal{X}_r(t)) \in \mathbb{M}, \forall t \geq t_0 \right\}, \quad (3.12)$$

Combine the formation kinematics, i.e. Eq. (3.6)-(3.8) and the autopilots, i.e.

$$\dot{V} = -\frac{1}{\tau_v} (V - V_C) \quad (3.13)$$

$$\dot{\psi} = -\frac{1}{\tau_\psi} (\psi - \psi_C) \quad (3.14)$$

where τ_ψ and τ_v are the time constants of the corresponding autopilot.

Then linearize them around the nominal formation flight point $([\bar{x}, \bar{V}_w, \bar{V}_L, \bar{y}, \bar{\psi}_w, \bar{\psi}_L]^T)$,

the plant of the leader-follower pair is

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}_c \mathbf{x} + \mathbf{B}_c \mathbf{u} \\ \mathbf{y} &= \mathbf{C}_c \mathbf{x} \end{aligned} \quad (3.15)$$

where

$$\mathbf{A}_c = \begin{pmatrix} 0 & -1 & 1 & 0 & -\frac{\bar{y}}{\tau_\psi} & 0 \\ 0 & \frac{-1}{\tau_v} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{\tau_v} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{\bar{x}}{\tau_\psi} - \bar{V} \right) & \bar{V} \\ 0 & 0 & 0 & 0 & \frac{-1}{\tau_\psi} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_\psi} \end{pmatrix}$$

$$\mathbf{B}_c = \begin{pmatrix} 0 & \frac{\bar{y}}{\tau_\psi} \\ \frac{1}{\tau_v} & 0 \\ 0 & 0 \\ 0 & \frac{\bar{x}}{\tau_\psi} \\ 0 & \frac{1}{\tau_\psi} \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{C}_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

where $\bar{y}$, $\bar{x}$ and $\bar{v}$ denote the nominal value of the formation flight. The state vector is: $\mathbf{x} = [x, V_w, V_L, y, \psi_w, \psi_L]^T$ and the input vector is: $\mathbf{u} = [V_{wc}, \psi_{wc}]^T$, where the first two are manipulated variables and last two are the measured disturbances. The output vector is: $\mathbf{y} = [x, V_L - V_w, y, \psi_L - \psi_w]^T$

By using the “c2d” function in MATLAB, we can get the continuous model

(3.15) in a discrete form below:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}_1 \mathbf{x}(k) + \mathbf{B}_1 \mathbf{u}_1(k) \\ \mathbf{y}(k) &= \mathbf{C}_1 \mathbf{x}(k)\end{aligned}\quad (3.16)$$

Since the predictive control algorithm will in fact produce the changes $\Delta \mathbf{u}$ rather than $\mathbf{u}$, it is therefore convenient for many purposes to regard the ‘controller’ as producing the signal $\Delta \mathbf{u}$, and the ‘model’ as having this signal as its input. One way to change the input to $\Delta \mathbf{u}$ is to define the state vector [34]:

$$\boldsymbol{\xi}(k) = \begin{bmatrix} \Delta \mathbf{x}(k) \\ \mathbf{y}(k-1) \end{bmatrix}$$

Then the augmented nominal model can be written as:

$$\begin{aligned}\boldsymbol{\xi}(k+1) &= \mathbf{A} \boldsymbol{\xi}(k) + \mathbf{B} \Delta \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C} \boldsymbol{\xi}(k)\end{aligned}\quad (3.17)$$

where:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{C}_1 & \mathbf{I} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{C} = (\mathbf{C}_1 \quad \mathbf{I}) \quad (3.18)$$

It can be shown that the predictions based on the augmented state space model (3.17) is

$$\begin{pmatrix} \hat{\mathbf{y}}(k+1) \\ \hat{\mathbf{y}}(k+2) \\ \vdots \\ \hat{\mathbf{y}}(k+N) \end{pmatrix} = \begin{pmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{pmatrix} \boldsymbol{\xi}(k) + \begin{pmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{0} & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{CA}^{N-1} \mathbf{B} & \dots & \dots & \mathbf{CB} \end{pmatrix} \begin{pmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+N-1|k) \end{pmatrix} \quad (3.19)$$

or, in vector-matrix notation

$$\hat{\mathbf{Y}} = \Phi \xi(\mathbf{k}) + \mathbf{G} \hat{\mathbf{U}} \quad (3.20)$$

The cost function for the leader-wingman formation control pair can be defined as follows:

$$J = \sum_{i=N_1}^{N_2} [\hat{\mathbf{y}}(k+i|k) - \mathbf{w}(k+i|k)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta \hat{\mathbf{u}}(k+i-1|k)]^2 \quad (3.21)$$

where $\mathbf{w}(k+i|k)$ is the set point, N_2 is the prediction horizon, N_u is the control horizon and λ is the control weighting variable, which expresses the relative importance between the tracking errors ($\hat{\mathbf{y}} - \mathbf{w}$) and control effort ($\Delta \hat{\mathbf{u}}(k)$).

In our application, minimizing the cost function means the future outputs $\hat{\mathbf{y}}$: $[x, V_L - V_w, y, \psi_L - \psi_w]^T$ within the considered horizons should follow a pre-determined reference signal $\mathbf{w} : [0, 0, 0, 0]^T$ and, at the same time, the control effort $\Delta \hat{\mathbf{u}} : [V_{wc}, \psi_{wc}]^T$ necessary for doing so should be penalized. The parameters N_l and N_2 are the minimum and maximum prediction horizons.

This cost function can be written in vector form as

$$\mathbf{J} = (\hat{\mathbf{Y}} - \hat{\mathbf{W}})^T (\hat{\mathbf{Y}} - \hat{\mathbf{W}}) + \lambda \hat{\mathbf{U}}^T \hat{\mathbf{U}} \quad (3.22)$$

$\hat{\mathbf{U}}$ can be solved online as a function of the current states

$$\min_{\mathbf{U}} \mathbf{J}(\mathbf{k}, \hat{\mathbf{Y}}(\mathbf{k}), \hat{\mathbf{U}}) \quad (3.23)$$

subject to:

$$\begin{aligned} \hat{\mathbf{Y}} &= \Phi \xi(\mathbf{k}) + \mathbf{G} \hat{\mathbf{U}} \\ \Delta \mathbf{u}_{min} &\leq \Delta \hat{\mathbf{u}}(k+i|k) \leq \Delta \mathbf{u}_{max} \\ \mathbf{y}_{min} &\leq \hat{\mathbf{y}}(k+i|k) \leq \mathbf{y}_{max} \end{aligned} \quad (3.24)$$

The MPC formation flight control algorithm can be described as follows:

- 1) Obtain the current model output $\mathbf{y}(k)$;
- 2) Compute the required model input $\Delta \mathbf{u}(k)$ according to the cost function (3.22) and respect to the input, output constraints (3.24) as well;
- 3) Apply $\Delta \mathbf{u}(k)$ to the model.

The whole system is illustrated in Figure 3.2. In the formation flight, each wingman UAV is equipped with one MPC controller in order to keep the specified distances from the leader and the formation leader is commanded by the GCS.

Figure 3.2 Formation flight control diagram and signals

3.2.4 Formation Pair Stability

The stability of MPC system is well defined for various constraints and cost conditions, for linear and nonlinear systems. In this section the terminal constraints are applied to ensure the stability of the formation flight system.

Since the cost function, i.e. Eq. (3.22) is a function of the state variables $\mathbf{x}$ and

input variable $\mathbf{u}$, it can be expressed as follows:

$$\mathbf{J} = (\mathbf{Y} - \mathbf{W})^T (\mathbf{Y} - \mathbf{W}) + \lambda \mathbf{U}^T \mathbf{U} = \sum_{i=1}^{H_p} q(\mathbf{x}(k+i), \mathbf{u}(k+i-1))$$

where H_p is the prediction horizon, $q(\mathbf{x}, \mathbf{u}) \geq 0$, and $q(\mathbf{x}, \mathbf{u}) = 0$ only if $\mathbf{x} = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$. q is a decreasing function, subject to the terminal constraint $\mathbf{x}(k + H_p) = \mathbf{0}$. Let $\mathbf{J}^0(t)$ be the optimal value of $\mathbf{J}$ which corresponds to the optimal input variable $\mathbf{u}^0$ as evaluated at time t . For stability analysis, we assume that for each step, the optimal solution can be found.

Obviously, $\mathbf{J}^0(t) \geq 0$ and $\mathbf{J}^0(t) = 0$ only if $\mathbf{x}(t) = \mathbf{0}$.

$$\begin{aligned} \mathbf{J}^0(t+1) &= \min_{\mathbf{u}} \sum_{i=1}^{H_p} q(\mathbf{x}(t+1+i), \mathbf{u}(t+i)) \\ &= \min_{\mathbf{u}} \left\{ \sum_{i=1}^{H_p} q(\mathbf{x}(t+i), \mathbf{u}(t+i-1)) - q(\mathbf{x}(t+1), \mathbf{u}(t)) + q(\mathbf{x}(t+1+H_p), \mathbf{u}(t+H_p)) \right\} \\ &\leq -q(\mathbf{x}(t+1), \mathbf{u}^0(t)) + \mathbf{J}^0(t) + \min_{\mathbf{u}} \{q(\mathbf{x}(t+1+H_p), \mathbf{u}(t+H_p))\} \end{aligned}$$

Since we have used the terminal constraint $\mathbf{x}(k + H_p) = \mathbf{0}$, we can make $\mathbf{u}(t + H_p) = \mathbf{0}$ and stay at $\mathbf{x} = \mathbf{0}$, which gives $\min_{\mathbf{u}} \{q(\mathbf{x}(t+1+H_p), \mathbf{u}(t+H_p))\} = 0$.

Because $q(\mathbf{x}(t+1), \mathbf{u}^0(t)) \geq 0$, so $\mathbf{J}^0(t+1) \leq \mathbf{J}^0(t)$ is obtained.

$\mathbf{J}^0(t)$ can be considered as a Lyapunov function for the system, and hence by Lyapunov's stability theory, the equilibrium point $\mathbf{x} = \mathbf{0}$, $\mathbf{u} = \mathbf{0}$ is stable which means the separations between a pair of wingman and leader will converge to nominal values, i.e. the wingman achieve the nominal separation with the leader provided

each optimization is feasible.

3.2.5 Inclusion of the Constraints in the MPC Formulation

As stated above, MPC method can deal with constraints directly. In this section, a quick 60 degree heading change simulation will show the MPC constraints-dealing ability in our formation flight control case.

A generic fixed-wing UAV model is used in the simulation and the autopilot time-constants UAV are: $\tau_v = 5$ s and $\tau_\psi = 0.75$ s.

In the MPC controller, the predictive control horizon N_u and the predictive horizon N_2 are set as $N_u = 3$ and $N_2 = 20$. The sample time is 0.2 s. The control signal constraints and the output constraints are set as:

$$\begin{aligned}\Delta V_{\min} &\leq \Delta V_c \leq \Delta V_{\max} \\ \Delta \psi_{\min} &\leq \Delta \psi_c \leq \Delta \psi_{\max} \\ x_{\min} &\leq x \\ y_{\min} &\leq y\end{aligned}$$

The limits are selected as:

$$\begin{aligned}\Delta V_{\min} &= -0.2(m/s \text{ per step}) \\ \Delta V_{\max} &= 0.2(m/s \text{ per step}) \\ \Delta \psi_{\min} &= -0.07(rad / step) \\ \Delta \psi_{\max} &= 0.07(rad / step) \\ x_{\min} &= 1m \\ y_{\min} &= 1m,\end{aligned}$$

The nominal desired separation: $\bar{x} = 2m, \bar{y} = 2m$, and the formation pair is flying with speed of 2m/s, the leader is turning 60 degree immediately. In this simulation, in order to verify that the MPC can cope with constraints, that the leader does not have the same constraints is assumed. The response of the wingman is

shown in Figure 3.3, and the control inputs, i.e. the command velocity input and the command heading angle input to the wingman are shown in Figure 3.4 and Figure 3.5. In Figure 3.4, we notice that at about 0.2 second and 7 second the change of the velocity control input is big since the wingman is trying to maintain the formation, but due to the input constraint, the change of the velocity never bigger than 0.2 m/s per step.

Figure 3.3 Separations between Leader and W1

Figure 3.4 Velocity control input

Figure 3.5 Heading control input

From this simulation, we can say that the MPC approach can work well in our case, the input and output constraints can be easily taken care of.

3.3 Chapter Summary

In this chapter, the theories behind the MPC are reviewed. Then a generalized constrained MPC formulation is discussed for the UAV formation flight. The results here in general guarantee only nominal stability, as no uncertainty or disturbances are treated as part of the problem. Simulation results have been presented.

Chapter 4 Robust Decentralized Formation Flight Control using Multiplexed MPC

In the last Chapter, the formation pair MPC framework is introduced. In order to guarantee the whole formation flight is stable and break the whole centralized formation problem into small decentralized problems, this chapter introduces a new framework for unmanned aerial vehicles (UAVs) formation flight and coordination. Formulated using MMPC approach, the whole formation flight system can be considered as a linear periodic system with control inputs of each UAV subsystem as its periodic inputs. Divided into decentralized subsystems, the whole formation flight system is guaranteed stable if proper terminal cost and terminal constraints are added to each decentralized MPC formulation of the UAV subsystem. The decentralized robust MPC formulation for each UAV subsystem with bounded input disturbances and model uncertainties is also presented. Furthermore, an obstacle avoidance control scheme for any shape and size of obstacles, including the non-apriorily known ones, is integrated under the unified MPC framework. The results from simulations demonstrate that the proposed framework can successfully achieve robust collision-free formation flights.

4.1 Introduction

Many works have addressed the “decentralized MPC” in UAV cooperative control [39, 41, 45, 47, 50]. Although the control optimization for the UAVs are all

performed sequentially, these published works can be sorted in two groups based on their input update scheme:

Scheme 1: Update all the UAVs' control inputs simultaneously. For example, in the schemes proposed in [39, 43-44], the MPC solution is solved iteratively for each vehicle input, but it is assumed that all the UAVs' inputs are updated simultaneously when the iterations have been completed;

Scheme 2: Similar to scheme 1 except that it updates each UAV's input as soon as its computation is finished [41, 45-47].

In this chapter, motivated by the MMPC proposed in [51], the centralized formation flight problem is put into a periodic form in which only one UAV's inputs are updated at each time step. This substantially reduces the computation time and load. Furthermore, by neglecting the dynamical couplings between the individual vehicles, the centralized periodic formation flight system can be treated as a decentralized system. Suppose that a given UAVs formation flight control problem can be solved in not less than T seconds by applying the scheme 1. In the MMPC formation flight control scheme, we propose to update only one UAV's inputs at interval smaller than T . If the dynamics and optimization of each UAV is the same, then typically the update interval is T/m , where m is the number of UAVs in the formation group. The main motivation for this formation flight scheme is the belief that updating only one UAV's input at a faster time step will be beneficial for rapid response to commands from Ground Control Station (GCS) and to disturbances as compared to scheme 1. Compared to scheme 2, since we know the exact update

timing of each UAV in the current proposed scheme, it is easier for the human operator to schedule the formation tasks or to trouble-shoot when errors exist in the formation flight group. This scheme functions like an ‘update manager’ system which systematically handle and schedule the update of each UAV in the formation system. Additionally, this proposed scheme is useful in the heterogeneous formation flight scenario, where the leader UAV or the more capable UAV (equipped with more sensors) can be updated much more often than the other UAVs without affecting the stability of the overall formation flight system (since the formation stability does not depend on the update sequence).

Another contribution of this chapter is the extension of the obstacle avoidance algorithm in the formation flight MPC framework. Most of the existing collision avoidance scheme under the MPC formation flight treats the shape of the obstacles to be rectangle and does not take into account the small or non-apriorily known obstacles. The framework proposed will try to eliminate these limitations.

This chapter is organized as follows. Section 4.2 presents the overview of the proposed formation flight control strategy. The controlled UAV dynamics, the multiplexed formation flight system and stability requirements for each UAV subsystem are given in Section 4.3. The stable robust decentralized formation flight control is detailed in Section 4.4, including its problem formulation, the MPC controller design and stability analysis. In section 4.5, the obstacle avoidance control scheme is addressed. Some simulation results are presented in section 4.6. Finally, discussions and conclusions are given in section 4.7.

4.2 Formation Flight Control Strategy Overview

Assumption 1

Each UAV is equipped with autopilot system capable of waypoint tracking.

With this assumption, the dynamics of each UAV subsystem in the formation flight group can be represented by their controlled dynamics (with autopilot), which can be modeled as linear system or piecewise linear system.

Assumption 2

The couplings between the UAVs, if any, come only from the cost function and constraint formulations in the MPC optimization problem. The dynamical couplings, like induced vortices, are neglected.

With this assumption, the whole formation flight system can be formulated as a linear periodic system under the MMPC as will be detailed in Section 4.3.

Figure 4.1 The overall Formation Flight Control System under MMPC

The overall formation flight control scheme is represented in Figure 4.1. Each UAV in the formation is considered as a subsystem in the framework of the whole MMPC formation flight system. In the following section, we show that under the

MMPC framework, the whole formation flight system is stable if proper terminal cost and stabilizing feedback gain are added to the optimization problem formulation of each UAV subsystem.

The key idea of the proposed framework is to break the large MPC formation flight problem into decentralized problems of smaller size while the stability of the whole formation flight system is guaranteed. In order to achieve this, the framework utilizes two steps: first, to apply the MMPC update scheme by formulating the whole formation flight as a linear periodic system; and second, to break the centralized periodic system into decentralized systems consisting of individual UAV, which is possible under Assumption 2 above.

The use of robust decentralized model predictive control (RDMPC) scheme to update only one UAV subsystem at a time is proposed. Assume that there are m UAVs in the formation, so that at each time step T/m , only one UAV's control input set is updated. After m updates a fresh cycle of updates begins, so that each whole cycle of updates repeats with a cycle time T . Figure 4.1 shows the scenario for $m=3$.

Figure 4.2 Decentralized UAV formation flight control

Each RDMPC controller is associated with a different vehicle and computes the local control inputs based only on its states and that of its neighbors (shown in Figure 4.2). On each UAV, the current state and the model of its neighbors are used to predict their possible trajectories and move accordingly. The information exchange topology and inter-vehicle constraints are described by a graph structure in the problem formulation and depends on the task descriptions. The cost function depends on the formation's mission and includes terms that minimize relative distances and/or velocities between neighboring vehicles. The coupling constraints arise from inter-agent collision avoidance and communication range. The interaction graph is based on a "closest spatial neighbors" model.

4.3 UAV Subsystem Dynamics and Overall Formation Flight System under MMPC

4.3.1 Dynamics of UAV Subsystem

In this chapter, the inputs, outputs, and states constraints are all regarded as polytopes. Some pertinent definitions are given in the Appendix A. For more details,

readers may refer to [52].

Under Assumption 1, consider a set of m linear decoupled UAV dynamical systems, where the i -th vehicle dynamics is described by the discrete-time time-invariant state equation

$$\begin{aligned}\mathbf{x}_{k+1}^i &= \mathbf{f}^i(\mathbf{x}_k^i, \mathbf{u}_k^i) = (A^i + \hat{A}^i)\mathbf{x}_k^i + B^i\mathbf{u}_k^i + D\mathbf{w}_k^i \\ \mathbf{y}_k^i &= \mathbf{h}^i(\mathbf{x}_k^i, \mathbf{u}_k^i) = C^i\mathbf{x}_k^i\end{aligned}\quad (4.1)$$

where the state update function $\mathbf{f}^i: \mathbb{R}^6 \times \mathbb{R}^3 \rightarrow \mathbb{R}^6$ and output map $\mathbf{h}^i: \mathbb{R}^6 \rightarrow \mathbb{R}^6$ are linear or piecewise linear functions of its inputs. The states and inputs of the vehicle at time k are denoted by $\mathbf{x}_k^i \in \mathbb{R}^6$ and $\mathbf{u}_k^i \in \mathbb{R}^3$, respectively. In particular

$$\mathbf{u}_k^i = \begin{bmatrix} N_{cmd} \\ E_{cmd} \\ h_{cmd} \end{bmatrix} \quad \mathbf{y}_k^i = \begin{bmatrix} N \\ E \\ h \\ \dot{N} \\ \dot{E} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k,pos}^i \\ \mathbf{y}_{k,vel}^i \end{bmatrix} \quad \mathbf{x}_k^i = \begin{bmatrix} \mathbf{x}_{k,pos}^i \\ \mathbf{x}_{k,vel}^i \end{bmatrix} \quad (4.2)$$

and $\mathbf{x}_{k,pos}^i \in \mathbb{R}^3$ is the vector of N , E , h (north-east-down) coordinates and $\mathbf{x}_{k,vel}^i \in \mathbb{R}^3$ denotes a vector of states corresponding to N -, E - and h -axis velocity at time k . The subscript *cmd* means the current commanded position

In general, Eq. (4.1) subjects to the disturbance input

$$\mathbf{w}_k^i \in \mathcal{W} \subset \mathbb{R}^q \quad (4.3)$$

Additionally, as indicated in Eq. (4.1), there may also be model uncertainty in the system, indicated by the uncertainty set

$$\mathcal{A}^i = \left\{ \hat{A}^i \mid \frac{\|\hat{A}^i\|}{\|A^i\|} = \varepsilon \right\} \quad (4.4)$$

where the constant matrix A^i gives the average value of the state matrix, and $\hat{A}^i$ describes its statistical variation. For example, if we assume that the uncertainty in the state matrix is 10% of its nominal value, then $\varepsilon = \frac{\|\hat{A}^i\|}{\|A^i\|} = 0.1$.

Let $\mathcal{U}_k^i \in \mathbb{R}^{m^i}$, $\mathcal{X}^i \in \mathbb{R}^{n^i}$ and $\mathcal{Y}_k^i \in \mathbb{R}^{p^i}$ denote the set of feasible inputs, states and outputs of the i -th vehicle

$$\mathbf{u}_k^i \in \mathcal{U}^i, \mathbf{x}^i \in \mathcal{X}^i, \mathbf{y}_k^i \in \mathcal{Y}^i \quad k \geq 0 \quad (4.5)$$

where $\mathcal{Y}_k^i$ and $\mathcal{U}_k^i$ are given polytopes. In the case of UAV, Eq. (4.5) often indicates the constraints on the speed and acceleration of the vehicle as follows:

$$\begin{aligned} \mathbf{y}_{vel}^i \in \mathcal{Y}_v^i &= \left\{ \mathbf{z} \in \mathbb{R}^3 \mid \underline{\mathbf{y}}_{vel}^i \leq \mathbf{z}_i \leq \bar{\mathbf{y}}_{vel}^i \right\} \\ \dot{\mathbf{y}}_{vel}^i \in \mathcal{Y}_a^i &= \left\{ \mathbf{z} \in \mathbb{R}^3 \mid \underline{\dot{\mathbf{y}}}_{vel}^i \leq \mathbf{z}_i \leq \bar{\dot{\mathbf{y}}}_{vel}^i \right\} \end{aligned}$$

The set of m constrained systems are referred to as the multi-vehicle system. Let $\mathbf{x}_k \in \mathbb{R}^{\tilde{n}}$ with $\tilde{n} = \sum_i n^i$ and $\mathbf{u}_k \in \mathbb{R}^{\tilde{m}}$ with $\tilde{m} = \sum_i m^i$ be the vectors which collect the states and inputs of the multi-vehicle system at time k , i.e., $\mathbf{x}_k = [\mathbf{x}_k^1; \dots; \mathbf{x}_k^m]^T$, $\mathbf{u}_k = [\mathbf{u}_k^1; \dots; \mathbf{u}_k^m]^T$, with $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$. Given the particular command (target, formation shape etc), the equilibrium pair of the i -th vehicle is denoted by $(\mathbf{x}_e^i, \mathbf{u}_e^i)$, i.e. the i -th vehicle is in the specified formation position with respect to the neighboring vehicle, while $(\mathbf{x}_e, \mathbf{u}_e)$ denotes the corresponding

equilibrium for the multi-vehicle system, which means the required formation states.

4.3.2 Formation Flight System under MMPC

Under Assumption 2, the whole formation flight system can be expressed as

$$\begin{aligned}
\mathbf{x}_{k+1} &= \begin{bmatrix} \mathbf{x}_{k+1}^1 \\ \vdots \\ \mathbf{x}_{k+1}^m \end{bmatrix} = \begin{pmatrix} A^1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A^m \end{pmatrix} \begin{bmatrix} \mathbf{x}_k^1 \\ \vdots \\ \mathbf{x}_k^m \end{bmatrix} + \begin{pmatrix} B^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B^m \end{pmatrix} \begin{bmatrix} \mathbf{u}_k^1 \\ \vdots \\ \mathbf{u}_k^m \end{bmatrix} \\
&= A\mathbf{x}_k + B\mathbf{u}_k \\
&= A\mathbf{x}_k + \sum_{j=1}^m B_j \mathbf{u}_k^j
\end{aligned} \tag{4.6}$$

where B_j is the j -th column of B .

As we will be referring to the expression $(k \bmod m) + 1$ often in this section, it is convenient to introduce the indexing function

$$\sigma(k) = (k \bmod m) + 1$$

According to our UAV input update scheme, at each fixed time step only one UAV's input set, $\mathbf{u}_k^{\sigma(k)}$, will be updated. Under such scheme, the whole formation system can be represented as a periodic linear system with one UAV's input:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B_{\sigma(k)} \mathbf{u}_k^{\sigma(k)} \tag{4.7}$$

According to [51], in order to make the whole linear periodic system (4.7) closed-loop stable by using the MMPC scheme, the following two additional terms need to be added to the optimization problem formulation:

- Terminal cost term:

$$F(\mathbf{x}_{k+N}) = \min_{\mathbf{u}_k^{\sigma(k)}} \left\{ \sum_{i=N}^{\infty} \left(\|\mathbf{x}_{k+i+1} - \mathbf{x}_{k+i+1}^d\|_Q^2 + \|\mathbf{u}_{k+i}^{\sigma(k)}\|_R^2 \right) \mid \mathbf{x}_{k+1} = A\mathbf{x}_k + B_{\sigma(k)}\mathbf{u}_k^{\sigma(k)} \right\} \quad (4.8)$$

$$= \mathbf{x}_{k+N}^T \bar{P}_{\sigma(k+N)} \mathbf{x}_{k+N} + 2\bar{q}_{\sigma(k+N)}^T \mathbf{x}_{k+N} + \bar{r}_{\sigma(k+N)}$$

where $\bar{P}_{\sigma(k+N)}$ can be calculated according to Eq. (B-2) in Appendix B which presents some known results on unconstrained infinite-time linear quadratic (LQ) control of periodic system.

- Terminal states constraints:

$$\mathbf{x}_{k+N+1} \in \mathcal{X}_I(K_{\sigma(k)}) \quad (4.9)$$

where $\mathcal{X}_I(K_{\sigma(k)})$ denotes the set in which none of the constraints is active . This is the maximum positive invariant set for the linear periodic system (4.7), when a linear periodic feedback controller (B-3) is applied. $K_{\sigma(k)}$ can be calculated according to Eq.(B-4) in Appendix B.

Remark 1

The terminal cost and the linear terminal stabilization controller can be derived from the LQ control theory of periodic system (See Appendix B).

Remark 2

The stability requirements for the MMPC formation flight framework from Eqs. (4.8) and (4.9) actually need the state information of the multi-vehicle system, i.e. $\mathbf{x}_k$, to be included in the terminal cost calculation, but according to Assumption 2, since the dynamical effects between UAV systems are ignored, the terminal cost and terminal stabilization controller of each vehicle subsystem can be calculated

independently, i.e. using only the i -th vehicle and its neighbors' states information.

4.4 Robust Decentralized Formation Flight

A graph topology [53] is used in this chapter to present the connections among the UAVs in the formation group. A directed graph $\mathcal{G}$ consists of a set of vertices, denoted by $\mathcal{V}$, and a set of edges $\mathcal{A} \subset \mathcal{V}^2$, where $a = (\alpha, \beta) \in \mathcal{A}$ and $\alpha, \beta \in \mathcal{V}$. The first element of a is denoted $tail(a)$, and the second is denoted $head(a)$. We assume that $tail(a) \neq head(a)$ for all a , meaning that the graph has no self-loops. We also assume that each element of $\mathcal{A}$ is unique. A graph with the property that for any $(\alpha, \beta) \in \mathcal{A}$, the arc $(\beta, \alpha) \in \mathcal{A}$ is said to be undirected. If every possible arc exists, the graph is said to be complete.

In this chapter, a graph topology is used to represent the couplings among the UAVs and it is assumed that for every wingman UAV, there is at least one path connecting it with the leader UAV. This assumption guarantees the controllability of the formation flight group [54]. The i -th vehicle system is associated with the i -th node of the graph. If an edge $(i, j) \in \mathcal{A}$ connecting the i -th and j -th node is present, then the cost and the constraints of the optimal control problem will have a component which is a function of both $\mathbf{x}^i$ and $\mathbf{x}^j$. Before defining the optimal control problem, we define a graph (which can be time-varying) $\mathcal{G}(t) = (\mathcal{V}, \mathcal{A}(t))$ associated with the vehicle formation group.

Using the graph structure defined previously, the optimization problem can be formulated by letting $q^{i,j}(\mathbf{x}^i, \mathbf{x}^j)$ be the interconnection function between the i -th

and j -th UAV for $(i, j) \in \mathcal{A}$ and

$$q^{i,j}(\mathbf{x}^i, \mathbf{x}^j) \leq 0 \quad (4.10)$$

define the interconnection constraints between the i -th and the j -th vehicle, with

$$q^{i,j} : \mathbb{R}^{n^i} \times \mathbb{R}^{n^j} \rightarrow \mathbb{R}.$$

Remark 3

Depending on the group tasks, the interconnection function $q^{i,j}(\mathbf{x}^i, \mathbf{x}^j)$ can have several variant forms:

- For inter-agent collision avoidance, this constraints define non-convex requirements in the following way:

$$q^{i,j}(\mathbf{x}_k^i, \mathbf{u}_k^i, \mathbf{x}_k^j, \mathbf{u}_k^j) = d_{safe} - \left\| \mathbf{y}_{k,pos}^i - \mathbf{y}_{k,pos}^j \right\|_p \leq 0 \quad (4.11)$$

where the parameter d_{safe} in Eq. (4.11) represents a lower bound on the norm of relative position between neighboring vehicles. The subscript p denotes the type of norm which will be used in the optimization problem. $p = 2$ leads to a circular protection zones around vehicle and the optimization problem becomes quadratic. If $p = 1$ or ∞ , then the protection zones around vehicle are square.

- Since the group of vehicles is assumed to be equipped with a communication network, and that the connectivity of the network depends upon the relative distance between the neighboring vehicles, maintaining network connectivity constrains the maximum allowable distance between vehicles. These constraints define convex requirements in the following way:

$$q^{i,j}(\mathbf{x}_k^i, \mathbf{u}_k^i, \mathbf{x}_k^j, \mathbf{u}_k^j) = \left\| \mathbf{y}_{k,pos}^i - \mathbf{y}_{k,pos}^j \right\|_p - r_{com} \leq 0 \quad (4.12)$$

where r_{com} represents the radius of the detection cylinder, i.e., the maximum allowable distance between vehicles.

In general, guaranteed collision avoidance and network connectivity between any vehicle pair in a formation flight problem would necessitate the use of a complete graph for describing inter-vehicle constraints. However, for practical decentralization purposes, it is usually sufficient for each vehicle to consider only a neighboring subset of all vehicles to accomplish the formation flight. Furthermore, the allowed number of vehicles in these neighboring subsets might be limited. For example, the maximum number of similar agents whose detection cylinder is in contact with the detection cylinder of the computing agent is 6 (see Figure 4.3).

Figure 4.3 Agent in the center of the cluster exchange information with most 6 neighbors

Consider the following cost function for the i -th UAV subsystem at time step k :

$$J_k^i = \begin{cases} \sum_{t=0}^{N-1} \left(\left\| \mathcal{Q}_f \left(\mathbf{y}_{k+t|k}^i - \mathbf{y}_f^i \right) \right\| + \left\| R \left(\mathbf{u}_{k+t|k}^i - \mathbf{u}_f^i \right) \right\| + F \left(\mathbf{x}_{k+N} \right) \right), i=1 \\ \sum_{t=0}^{N-1} \left(\left\| \mathcal{Q}_l \left(\mathbf{y}_{k+t|k}^i - \mathbf{y}_{k+t|k}^j - \mathbf{L} \right) \right\| + \left\| R \left(\mathbf{u}_{k+t|k}^i - \mathbf{u}_f^i \right) \right\| + F \left(\mathbf{x}_{k+N} \right) \right), i \neq 1, (i, j) \in \mathcal{A}(k) \end{cases} \quad (4.13)$$

The leader in the formation fight group is denoted as the first UAV, i.e. $i=1$.

$\left\| \mathcal{Q}_f \left(\mathbf{y}_{k+t|k}^i - \mathbf{y}_f^i \right) \right\|$ represents the cost term driving the leader to the target position, the terminal cost term $F(\mathbf{x}_{k+N})$ comes from section II, which makes the decentralized MPC controller similar to the infinite horizon formulation. For the wingmen's cost function, i.e., $i > 1$, the formation shape is maintained by the cost term $\left\| \mathcal{Q}_l \left(\mathbf{y}_{k+t|k}^i - \mathbf{y}_{k+t|k}^j - \mathbf{L} \right) \right\|$ and $\mathbf{L}$ is the separation vector between neighboring UAVs. This cost term can also be used in formation reconfiguration purposes.

Given a certain graph interconnection structure $\mathcal{G}(t)$, for each UAV, the associated robust decentralized MPC controller (RDMPC) can be presented as:

$$\mathbf{u}_k^i = \arg \min J_k^i \quad (4.14)$$

subject to several constraints below:

1. System dynamics (without disturbance and model uncertainty)

$$\begin{aligned} \mathbf{x}_{k+t+1|k}^i &= A^i \mathbf{x}_{k+t|k}^i + B^i \mathbf{u}_{k+t|k}^i, t \geq 0 \\ \mathbf{y}_{k+t|k}^i &= C^i \mathbf{x}_{k+t|k}^i \\ \mathbf{x}_{k|k}^i &= \mathbf{x}_k^i \end{aligned} \quad (4.15)$$

2. State and control constraints

$$\mathbf{x}_{k+t|k}^i \in \mathcal{X}_t^i, \hat{\mathbf{u}}_{k+t|k}^i \in \mathcal{U}_t^i, t=0, 1, \dots, N \quad (4.16)$$

3. Stabilizing controller after the prediction horizon

$$\mathbf{u}_{k+t|k}^i = \bar{K}_i \mathbf{x}_{k+t|k}^i + \mathbf{g}_i, t \geq N \quad (4.17)$$

4. Interconnection constraints for i -th UAV and its neighboring UAV, i.e.,

$$(i, j) \in \mathcal{A}(k)$$

$$q^{i,j}(\mathbf{x}_{k+t+1|k}^i, \mathbf{u}_{k+t+1|k}^i, \mathbf{x}_{k+t+1|k}^j, \mathbf{u}_{k+t+1|k}^j) \leq 0, t \geq 0, (i, j) \in \mathcal{A}(k) \quad (4.18)$$

where the control sequence is given by $\mathbf{U}_k^i = [\mathbf{u}_{k|k}^i; \mathbf{u}_{k+1|k}^i; \dots; \mathbf{u}_{k+N-1|k}^i]^T$, and the double indices $(k + j | k)$ denote the prediction at time k of a value at time $k + j$.

Remark 4

- For the stabilizing controller around the equilibrium point, the closed-loop system equation can be expressed as:

$$\begin{aligned} \mathbf{x}_{k+1}^i &= (A^i + \hat{A}^i) \mathbf{x}_k^i + B(\bar{K}_i \mathbf{x}_k^i + \mathbf{g}_i) + D\mathbf{w}_k^i \\ &= (A^i + \hat{A}^i + B\bar{K}_i) \mathbf{x}_k^i + D\mathbf{w}_k^i + B\mathbf{g}_i \end{aligned}$$

The set of periodic stabilizing control gains $(\bar{K}_i, i = 1, \dots, m)$ for the whole formation flight system should not only satisfy the stability requirement of the linear periodic system, i.e., all the eigenvalues of Ψ_1 in Eq. (B-5) must lie inside unit circle, but also for each stabilizing gain $\bar{K}_i$, the eigenvalues of matrix $(A^i + \hat{A}^i + B\bar{K}_i)$ should lie inside the unit circle. This stringent requirement is because of the model uncertainty $\hat{A}^i$.

- Constraint 2 comes from the constraint tightening technique [55] to provide the margin for robustness

$$\mathcal{X}_t^i = \mathcal{X}^i \sim \mathcal{R}_t^i, \mathcal{U}_t^i = \mathcal{U}^i \sim (\bar{K}_i x_{k+l|k}^i + g_i) \mathcal{R}_t^i$$

where $\mathcal{R}_t^i$ denotes the set of states of the i -th UAV subsystem under the stabilizing control (B-3), which is reachable in t steps from the equilibrium point. Symbol $\sim$ denotes the sets subtraction operation.

- It can be shown [56] that $\mathcal{R}_\infty^i$ is the smallest invariant set and if the initial optimization problem is feasible, then the feasibility here will also be guaranteed.

Combining this scheme with the stability results in Section 3, the whole formation flight system is guaranteed to be stable.

4.4.1 Stability of the Formation Flight Under MMPC

Proposition: The multi-vehicle system with the Multiplexed Robust Decentralized MPC scheme is closed-loop stable.

Proof:

According to [33], if the following four conditions hold:

- (1) State constraints satisfy terminal constraint set
- (2) Control constraints satisfy terminal constraint set
- (3) The terminal constraint set is positively invariant under a local controller
- (4) The terminal cost is a local Lyapunov function

Then the closed-loop stability is obtained.

In our set-up, conditions (1)-(3) are satisfied by problem formulation. So we only have to check that condition (4) is satisfied, namely that $F(\mathbf{x}_{k+N+1}) - F(\mathbf{x}_{k+N}) \leq 0$.

According to [57], $F(\mathbf{x}_{k+N})$ as defined in (4.8) is the value function of the unconstrained infinite horizon periodic optimal control problem. As a consequence

$$F(\mathbf{x}_{k+N}) = F(\mathbf{x}_{k+N+1}) + \|\mathbf{x}_{k+N} - \mathbf{x}_{k+N}^d\|_Q^2 + \|\mathbf{u}_{k+N}^{\sigma(k)}\|_R^2$$

Hence

$$F(\mathbf{x}_{k+N+1}) - F(\mathbf{x}_{k+N}) = -\|\mathbf{x}_{k+N} - \mathbf{x}_{k+N}^d\|_Q^2 - \|\mathbf{u}_{k+N}^{\sigma(k)}\|_R^2 \leq 0$$

with equality satisfied if and only if $\mathbf{x}_{k+N} - \mathbf{x}_{k+N}^d = \mathbf{u}_{k+N}^{\sigma(k)} = 0$. This shows that $F(\mathbf{x}_{k+N})$ is a Lyapunov function in some neighborhood of the desired formation shape $\mathbf{x}^d$. It can be seen that this result, and its proof, depends only on using an appropriate terminal cost $F(\mathbf{x}_{k+N})$, and not at all on the details of the constrained optimization over the horizon of length N . There exist a number of techniques to calculate a suitable Lyapunov function to be used as a terminal cost function, for example the linear matrix inequality (LMI)-based method. Consequently the result and its proof hold for any other multiplexed scheme which involves constrained optimization over a finite horizon, providing that the terminal cost is properly defined. For example, for a 3-UAV fleet formation flight system, it may be beneficial to update the UAV subsystem in the sequence (1, 2, 1, 3, 1, 2, 1, 3...), thus updating UAV1, i.e. the leader twice as often as the other two UAVs. This update scheme will be very useful

when the leader UAV has more capabilities (equipped with more sensors) than the others and the leader can response faster to the GCS.

The proof above depends on the feasibility of the constrained optimization at each step, and the feasibility is guaranteed by the constraints tightening(4.16), so if the initial states are feasible, then the following decentralized MPC for each UAV will be feasible and the whole formation group is thus stable.

4.5 Collision Avoidance Integration

Mixed Integer Linear Programming (MILP) combined with MPC [43, 45-46, 50] has been applied in the collision avoidance of multiple agents' coordination. But there are some drawbacks in this collision avoidance method. Firstly, the obstacle's shape is normally modeled as rectangular in the problem formulation, while in the real-life environment, the shape of the obstacles can be of any form and formulating it as a box may waste the space that maybe useful for the vehicle to perform the necessary maneuver. Secondly, since the collision avoidance constraint is formulated based on the predicted future positions, in the situation where the obstacle's width is smaller than the vehicle's one step predictive travel distance (this can happen if the time step is long or the vehicle velocity is high), such algorithm may fail to prevent the collisions. Figure 4.4 illustrates such scenario, where the obstacle's (red) width is the same as the vehicle's longest travel distance in one predictive step, and since the MPC does not "see" the obstacle in its predicted positions, it leads the vehicles to the trajectories colliding the obstacle. To deal with this problem and the variety of the obstacle shapes, as well as to handle the non-apriorily known obstacles, the following

scheme is proposed.

Figure 4.4 UAVs penetrating the small size obstacle

Receding horizons can be divided into two families, i.e., temporal and spatial ones. The temporal horizons manage the time frames while the spatial horizons manage the physical areas. A temporal horizon, as called in most of the MPC literatures [58], is defined as the prediction horizon and unique for each vehicle ($h_i, i = 1, 2, \dots, n$). It sets the temporal windows in the future over which the optimizer of the cost functions and the collision avoidance constraints is applied, starting from the present time. The temporal horizon can be effectively used to avoid the obstacles whose size is larger than one-step predictive distance.

A number of spatial horizons can also be defined. One example can be the sensor detection horizon [59] that allows for detection of near-by small-size or pop-up obstacles.

4.5.1 Collision Avoidance of Large Apriorily-known Obstacles

A polytope is used to present various shapes of the obstacles. In order to ensure that every trajectory point of a UAV does not lie inside the boundaries of an obstacle,

binary variables are introduced in the constraint formulation. For the hexagon shape obstacle for example, each edge can be regarded as a hyperplane which divides the fly zone into two half-spaces, one is safe while the other is not. In the collision avoidance constraint formulation, the predicted future point on the trajectory of each UAV is attached with an array of binary variables. In the case of a hexagon-shaped obstacle, an array of 6 binary variables is attached to each predicted point. The mathematic formulation is

$$b_i = 1 \text{ iff } x \in \{x \mid a_i^T x < b_i\}$$

$$\sum_{i=1}^6 b_i \geq 1$$

where b is the binary variable array attached to the predicted position x . For each edge, a hyperplane $\{x \mid a_i^T x = b_i\}$ can be defined, and $\{x \mid a_i^T x < b_i\}$ denotes the safe half-space for the UAV. The formulation above means that the predicted position point should lie in the safe half-space.

The collision avoidance formulation here is an extension of that in [43], which takes care the rectangular obstacles. Note that more edges correspond to more binary variables in the formulation, and hence more time will be needed in the optimization computation. If this is a concern, a reduced shape-representation can be used. For example, a hexagon can be used to represent a round obstacle.

The whole formation flight control optimization problem becomes non-convex after adding the collision avoidance constraints. In order to solve the non-convex optimization problem more efficiently, bounds on the variables are added explicitly

$$\underline{x} \leq x \leq \bar{x} \quad (4.19)$$

where $\bar{x}$ and $\underline{x}$ denote the upper and lower bounds for the aircraft predicted position.

Figure 4.5 Collision avoidance with a small obstacle

4.5.2 Collision avoidance of small or non-apriorily known obstacles

In this part, the small obstacle's shape is specified as a circle or cylinder (note that if the shape of the obstacle is not a circle or cylinder, the collision avoidance algorithm can still be applied after some modifications, which will not be detailed in this section).

In the proposed approach, position constraints are computed using the intersection points between the vehicle spatial horizon H_i^* and the obstacle. Suppose the obstacles detection sensor's range H_i is larger than the distance that a vehicle can travel in one predictive step, so any unknown obstacle appearing in range H_i will be successfully detected as shown in Figure 4.5. A temporary and smaller vehicle spatial horizon H_i^* can be set, based on the distance between the vehicle, the obstacle center and the intersection points with the obstacle. A position constraint can be added into

the MPC formulation to deal with such scenario:

$$\text{norm}(\hat{x}_{k+1} - x_k) \leq H_i^* \quad (4.20)$$

where $\hat{x}_{k+1}$ and x_k denote the next predicted position and current position respectively.

Figure 4. 6 Collision Avoidance Flow chart

Note that this position constraint actually prevents the next vehicle predictive position to be located outside of H_i^* , which ensure that the UAV will not penetrate the small obstacle, and this constraint adds no difficulty to the optimization problem since it is a convex inequality constraints on the predicted states. Illustration of the whole collision avoidance control scheme is given in Figure 4. 6.

4.6 Simulation Results

This section presents some simulation results of the multiplexed collision-free decentralized control scheme. The vehicle model used in the simulations is a small-size quadcopter with autopilot installed. Most of the simulations are done in a 2D space, however it can be extended to a 3D space in a straightforward manner. The 2D representation is selected since it facilitates the visualization of the vehicle trajectories in a clearer manner. Throughout this section, the interconnection is

described using a directed graph, which is determined by the two closest visible neighbors to each vehicle.

It is important to note that the method proposed in this chapter can easily accommodate any other particular UAV dynamics described by higher fidelity, heterogeneous, higher order, more complex linear or piecewise-linear models. Within the multiplexed robust decentralized MPC framework, each vehicle is modeled as a low-order linear system that represents the controlled dynamics of the vehicle. The details of the simulation setup are elaborated below.

1. The predictive horizon N_u as $N_u = 3$. Assume there is no delay in the simulations and the sample time is set as $0.3s$. In the multiplexed robust decentralized scheme, the update interval between UAV subsystems is $0.1s$.
2. The following simulations are performed using multiplexed robust decentralized scheme, and the update cycle is sequential, i.e., (1, 2, 3, 1, 2, 3...) except that in Case 5, the update cycle is (1, 2, 1, 3, 1, 2, 1, 3...).
3. Terminal cost for Eq. (4.8) can be determined by associated LQ problem
4. Given the system dynamics and a “two-closest-neighbors” interconnection policy, each vehicle solves the decentralized optimization problem with the cost function defined in (4.13)
5. The vehicles in the formation have identical dynamics
6. Linear constraints on single vehicle’ velocity is:

$$|v| \leq [v_{x\max} \ v_{y\max} \ v_{z\max}]^T = [5 \ 5 \ 5]^T m$$

7. The non-convex interconnection constraints (collision avoidance) are represented by:

$$q^{i,j}(\mathbf{x}_k^i, \mathbf{u}_k^i, \mathbf{x}_k^j, \mathbf{u}_k^j) = d_{safe} - \left\| \mathbf{y}_{k,pos}^i - \mathbf{y}_{k,pos}^j \right\|_{\inf} \leq 0, \quad d_{safe} = 0.5m$$

For pop-up constraint, the position constraint $norm\left(\begin{bmatrix} \hat{x}_{i,k+1} \\ \hat{y}_{i,k+1} \end{bmatrix} - \begin{bmatrix} x_{i,k} \\ y_{i,k} \end{bmatrix}\right) \leq H_{i^*}$ is added.

8. Network connectivity constraints :

$$q^{i,j}(\mathbf{x}_k^i, \mathbf{u}_k^i, \mathbf{x}_k^j, \mathbf{u}_k^j) = \left\| \mathbf{y}_{k,pos}^i - \mathbf{y}_{k,pos}^j \right\|_{\inf} - d_{com} \leq 0, \quad d_{com} = 10 \text{ m},$$

9. The weights in the cost function are chosen to be $Q_f = I$ for all vehicles, where

I denotes the identity matrix. Other weights are chosen as $Q_l = I$ and $R = 0.1I$

Case I: Formation flight with triangular-shape obstacle

In Figure 4.7, the leader (black), wingman1 (blue), and wingman2 (green) are initially aligned along the y- axis when the leader is commanded to fly to the target position (25,0) m with the two wingman maintaining the formation, i.e. the commanded separation between each other is 1 m in the y-direction. Figure 4.7 shows that the formation is maintained and the fleet successfully avoids the triangular obstacle shown in red.

Figure 4.7 Triangle Obstacle Collision Avoidance

Case 2: Formation splitting

The scenario in Figure 4.8 is more complicated than the previous case. Initially, the leader (black), wingman1 (blue), wingman2 (green), wingman3 (cyan), wingman4 (magenta) are aligned on the y axis at $x=0$. Firstly the leader is commanded to fly to the target position at (15, 0) m, and the formation is commanded to maintain the V shape as indicated by the red dotted line in the figure. Secondly, the 5-vehicle formation is commanded split into two formation groups, i.e., group 1 (leader, wingman1 and wingman2) and group 2 (wingman3 and wingman4). The leader and wingman 3 are commanded to fly to the targets (50, 0) m and (50,-15) m respectively. Figure 4.8 indicates that the vehicles can execute the commands while successfully maintaining safety. Notice that there is overshoot on the trajectories; this is mainly due to the short predictive horizons. If the computational power is enough, better performance can be attained by setting the predictive horizon longer.

Figure 4.8 Formation Split and Obstacle avoidance

Case 3: Comparison of robust decentralized approach (with constraint tightening) with decentralized approach without robustness guarantee

In this simulation, the leader (blue) is commanded to visit two target points (green circles), and the wingman is trying to maintain the formation (3 m from the leader in y direction). Both UAVs are subjected to disturbances $\|\mathbf{w}_k^i\|_\infty < 1\text{ m}$ which are randomly injected to the x and y position. The simulation results for formation flight with and without robustness considerations are shown in Figure 4.9 a) and b) respectively. As expected, without robustness consideration, the decentralized flight controller loses its feasibility, so it stops when it hits the first obstacle. On the contrary, by using the constraint tightening and stabilizing feedback controller, the robust decentralized formation flight controller can successfully fulfill the mission.

Figure 4.9 Decentralized Formation Flight with robustness and without robustness

Case 4: Small-obstacle avoidance

In this simulation, we show the benefit of the small-obstacle avoidance algorithm. In Figure 4.10 a), the predicted future point in one step lies 1.5 m ahead, which is larger than the diagonal length of the obstacle (1.13 m), so without imposing the position constraint presented in Section 4.5, the Leader “ignores “ the obstacle and heading to the target position (green circle). While in Figure 4.10 b), during the simulation, the position constraint has been added to the optimization, so as expected, the formation can successfully avoid the small obstacle and reach the target position with no collision.

Figure 4.10 Small-obstacle collision free formation flight

Case 5: Multiplexed vs. nominal decentralized formation flight control

In this simulation, the performances of the multiplexed decentralized formation flight control and the nominal decentralized formation flight control are compared and presented in Figure 4.11. Initially the Leader is at position (0, 0), the Wingman1 and Wingman2 are at (0,-3) m and (0, -6) m respectively. At first, the Leader is commanded to visit target1 (the green circle) at (12, 0) m, but at time 1.3 s, the Leader is commanded to fly to the target2 (the black circle). In the multiplexed decentralized scheme, the UAV subsystems are updated in the sequence (1, 2, 1, 3, 1, 2, 1, 3...) which means that the Leader will response faster than the nominal decentralized scheme that updates the UAV subsystem when the whole input updates are available. The total simulation time in both cases are 3 seconds, as expected, the formation response in Figure 4.11 b) are faster than that in Figure 4.11 a). This multiplexed decentralized scheme will be very beneficial when the agents in the formation are heterogeneous. Using a specified updating sequence, the human

operator or the GCS can fully making use of the advantages of each UAV subsystem.

Figure 4.11 Multiplexed decentralized vs nominal decentralized scheme

Case 6: Formation flight in 3D

This simulation is use to demonstrate that the scheme can be easily extended to 3D case as can be seen from Figure 4.12. In the 3-D collision avoidance constraint formulation, if the UAV has hovering ability, several options can be chosen, for example, the vehicle can avoid the obstacles either in the X-Y, X-Z or Y-Z plane depending on the current situation and the group mission requirements.

Figure 4.12 3-D Formation Flight

4.7 Concluding Remarks

In this chapter, the MMPC approach is employed to systematically handle the multiple UAV formation flight control. In its implementation, careful consideration should be given to the followings:

- If the number of UAV in the formation is large, updating one UAV's inputs at each time step will be not acceptable since the update interval will be too long. In this case, update can be applied to a small group of UAVs at each time step to reduce the total update interval.
- Since in the MMPC scheme, updating time of each UAV subsystem is fixed, an additional synchronization clock is required in the formation system, which may add complexities to the formation flight system.

The 3-D Multiplexed decentralized vehicle formation flight control problem has been formulated using decomposition in a hierarchical fashion. Under the multiplexed decentralized scheme, the leader UAV can response faster to the GCS compare to the nominal decentralized scheme, i.e. when the whole group update their inputs simultaneously. Moreover, the closed-loop stability of the whole formation flight system is always guaranteed even if a different updating sequence is used which makes the scheme flexible and the capability of each UAV can be fully exploited. The stable collision-free control and the robustness have been proposed under the unified MPC framework. By integrating the spacial horizon, the collision avoidance of the small pop-up obstacles is transformed into a convex position constraint in the MPC problem formulation. Simulation results demonstrate that the proposed framework is able to maintain the desired formation geometry within the

constraints posted and guarantees collision-free flight.

Chapter 5 Specific Application to Quadcopters MPC Formation Flight

In the last chapter, each UAV in the formation is assumed to have an autopilot system installed and a general piecewise linear controlled-UAV model is used in the decentralized formation flight control. In order to realize real time formation flight, the MPC formation flight controller must also consider the UAV dynamics and needs to be modified to achieve real-time computation. In this chapter, the formation flight control is considered together with the UAV stabilization and quadcopter is used as our formation flight test-bed.

The proposed control scheme is a hierarchical method consisting of a top-layer model predictive controller (MPC) with a bottom-layer robust feedback linearization controller. The MPC controller generate the optimized collision-free state reference trajectory which satisfies all kind of constraints and robust to the input disturbances (as explained in Chapter 4), while the robust feedback linearization controller tracks the optimal state reference and suppress any tracking errors during the MPC update interval. In the top-layer MPC, two modifications, i.e. the control input hold and variable prediction horizon are made and combined to allow the real-time formation flight implementation. The whole system is proved to be stable, computationally feasible and able to reach the desired formation position in finite time. Formation flight experiments are set up in the Vicon motion-capture environment and the flight

results demonstrate the effectiveness of the proposed formation flight architecture.

5.1 Introduction

In this chapter, we try to develop formation flight control scheme with the following properties: guaranteed stability, intuitive, and easy to implement under constraints.

The main challenges of the formation flight control include:

- The ability to explicitly account for the UAV dynamics to guarantee a feasible trajectory;
- Tight integration between path planning and control for each UAV in the formation to ensure safety, maximize performance and flexibility;
- Satisfaction of the explicit performance objectives.

In the existing literature, most of the work done in formation flight control only consider the kinematics (path planning), for example the formation consensus problem[18, 60]. Thus far, little effort has been focused on developing a formation flight framework that combines the planning and position/Euler angle control for the wingman UAV, and most of the existing work only provided simulation results. Many works have addressed the MPC in UAV cooperative control [39, 41, 43-47, 50]. Although the MPC framework have been proposed in [47, 50], the further reduction of computational burden in control algorithms can always benefit the formation flight application. The extra computation power saved can be put on extending the prediction horizon, utilizing a more detailed UAV model, etc.

This chapter concentrates on the quadcopter formation flight scenario, i.e. leader and follower quadcopters. To be specific, we are trying to design a formation flight control framework for the follower quadcopter which should keep the specified formation shape with respect to its immediate leader quadcopter, while at the same time reject the disturbances and try to keep collision-free flight.

Since there are still difficulties in directly applying MPC into plants with very fast dynamics like the quadcopters, this chapter proposes a formation flight framework for the follower quadcopter which explores the advantages of MPC while being able to control the fast quadcopter dynamics. Instead of attempting to implement a single MPC as the formation flight control system, the proposed framework employs a two-layer control structure where the top-layer MPC generates the optimal states trajectory by exploiting the quadcopter model and environment information, and the bottom-layer robust feedback linearization controller is designed based on exact dynamics inversion of the quadcopter to track the optimal trajectory provided by the top-layer MPC controller in the presence of disturbances and uncertainties. These two layer controllers are both designed in a robust manner and running parallel but at different time scales. The top-layer MPC controller which is implemented using the open source algorithm[61-62] runs at a low sampling rate allowing enough time to perform real-time optimization, while the bottom-layer controller performs at a much higher sampling rate to respond the fast dynamics and external disturbances.

The piecewise constant control scheme (input hold) and the variable prediction horizon length are adopted and combined in the top-layer MPC. The piecewise

constant control allows the real-time optimization occurring at scattered sampling time without losing the prediction accuracy. Moreover, it reduces the number of control variables to be optimized which helps to ease the workload of the real-time formation flight optimization. The variable prediction horizon length [63] is very suitable for the formation flight control problem which can be regarded as a transient control problem with a predetermined target set (the specified formation position). Compared to the fixed prediction horizon length, the variable prediction horizon version further saves the computational energy, for example when the follower quadcopter is already near the formation position, the prediction horizon length needed will be much shorter. The feasibility, stability and finite time reachability of this formation flight framework is investigated particularly focusing on the modified MPC.

Another contribution in this chapter is to provide an experimental solution to realize the proposed formation flight control algorithm using quadcopters. Formation flight tests on the quadcopter are carried out by utilizing the indoor test facility where the quadcopter can perform maneuvers under the supervision of the ground control station (GCS). The GCS is constructed to observe the quadcopter states and integrate the open source optimization solvers [62] with the real-time controller under the Matlab/Simulink environment.

The rest part of this chapter is organized as follows: Section 5.2 presents the quadcopter dynamical model; in Section 5.3, the two-layer formation flight control architecture is introduced; Section 5.4 details the top-layer modified MPC formation flight controller, including the input hold and variable prediction length. The robust

feasibility, stability and finite-time formation position reachability are proved; the bottom-layer robust feedback linearization controller is formulated in Section 5.5, followed by the stability statement for the whole two-layer flight control structure in Section 5.6; Section 5.7 devotes to the introduction of the flight test settings and formation flight test results; lastly this paper concludes in Section 5.8.

5.2 Quadcopter dynamics

In the quadcopter configuration (Figure 5.1), the throttle input is the sum of the thrusts of each motor. Pitch movement is controlled by increasing (reducing) T_1 while reducing (increasing) T_3 . The roll movement is controlled by increasing (reducing) T_4 while reducing (increasing) T_2 . The yaw is controlled by increasing (decreasing) the speed of the T_2 and T_4 while decreasing (increasing) T_1 and T_3 . To describe the quadcopter dynamics two coordinate systems are required,

Figure 5.1 Configuration of the Quadcopter

i.e., the earth fixed frame and the mobile body frame whose dynamic behavior can be described relative to the earth fixed frame: in which the Newton's law is valid. The earth fixed frame is denoted by O_{NED} (North-East-Down) The mobile body frame is denoted by O_{xyz} , and has its origin coincident with the quadcopter's center of gravity. These two frames are illustrated in Figure 5.1.

The translational dynamics[64] is given as follows:

$$\begin{aligned} m\ddot{X} &= (c\psi s\theta c\phi + s\psi s\phi)\mathbf{T} \\ m\ddot{Y} &= (s\psi s\theta c\phi - c\psi s\phi)\mathbf{T} \\ m\ddot{Z} &= mg + (c\theta c\phi)\mathbf{T} \end{aligned} \quad (5.1)$$

where $[X \ Y \ Z]$ is the position of the center of gravity in the frame O_{NED} , $[\phi \ \theta \ \psi]$ are the Euler angles, $c \cdot = \cos(\cdot)$, $s \cdot = \sin(\cdot)$ and $\mathbf{T} = -(T_1 + T_2 + T_3 + T_4)$.

and rotational dynamics:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{M_x}{I_{11}} \\ \frac{M_y}{I_{22}} \\ \frac{M_z}{I_{33}} \end{pmatrix} - \begin{pmatrix} \frac{(I_{33} - I_{22})qr}{I_{11}} \\ \frac{(I_{11} - I_{33})pr}{I_{22}} \\ \frac{(I_{22} - I_{11})pq}{I_{33}} \end{pmatrix} \quad (5.2)$$

where I_{11} , I_{22} , and I_{33} are the moment of inertial of the xyz axes in the body frame O_{xyz} , $[p \ q \ r]$ are the angular velocities and $[M_x \ M_y \ M_z]$ are the moments around each axis of the O_{xyz} frame.

Using the strap-down equation [65]

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan \theta s \phi & \tan \theta c \phi \\ 0 & c \phi & -s \phi \\ 0 & s \phi / c \theta & c \phi / c \theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (5.3)$$

The rotational Eq. (5.2) in terms of Euler angles $\boldsymbol{\eta} = [\phi \ \theta \ \psi]^T$ can be written in general form, as follows [66]:

$$\mathbf{M}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} = \boldsymbol{\tau}_{\boldsymbol{\eta}} \quad (5.4)$$

where $\mathbf{M}(\boldsymbol{\eta}) = \begin{pmatrix} I_{11} & 0 & -I_{11}s\theta \\ 0 & I_{22}c^2\phi + I_{33}s^2\phi & (I_{22} - I_{33})c\phi s\phi c\theta \\ -I_{11}s\theta & (I_{22} - I_{33})c\phi s\phi c\theta & I_{11}s^2\theta + I_{22}s^2\phi c^2\theta + I_{33}c^2\phi c^2\theta \end{pmatrix}$, and

$$\mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

where

$$\begin{aligned} c_{11} &= 0 \\ c_{12} &= (I_{22} - I_{33})(\dot{\theta}c\phi s\phi + \dot{\psi}s^2\phi c\theta) + (I_{33} - I_{22})\dot{\psi}c^2\phi c\theta - I_{11}\dot{\psi}c\theta \\ c_{13} &= (I_{33} - I_{22})\dot{\psi}c\phi s\phi c^2\theta \\ c_{21} &= (I_{33} - I_{22})(\dot{\theta}c\phi s\phi + \dot{\psi}s^2\phi c\theta) + (I_{22} - I_{33})\dot{\psi}c^2\phi c\theta + I_{11}\dot{\psi}c\theta \\ c_{22} &= (I_{33} - I_{22})\dot{\phi}c\phi s\phi \\ c_{23} &= -I_{11}\dot{\psi}c\phi s\phi + I_{22}\dot{\psi}s^2\phi c\theta s\theta + I_{33}\dot{\psi}c^2\phi c\theta s\theta \\ c_{31} &= (I_{22} - I_{33})\dot{\psi}c^2\theta s\phi c\phi - I_{11}\dot{\theta}c\theta \\ c_{32} &= (I_{33} - I_{22})(\dot{\theta}s\phi c\phi s\theta + \dot{\phi}s^2\phi c\theta) + (I_{22} - I_{33})\dot{\phi}c^2\phi c\theta + I_{11}\dot{\psi}c\theta s\theta - I_{22}\dot{\psi}s^2\phi c\theta s\theta - I_{33}\dot{\psi}c^2\phi c\theta s\theta \\ c_{33} &= (I_{22} - I_{33})\dot{\phi}c\phi s\phi c^2\theta - I_{22}\dot{\theta}s^2\phi c\theta s\theta - I_{33}\dot{\theta}c^2\phi c\theta s\theta + I_{11}\dot{\theta}c\theta s\theta \end{aligned}$$

By combining Eq. (5.1) and Eq. (5.4), we can represent the quadcopter model into a state-space compact form:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (5.5)$$

5.3 Two-Layer Formation Flight Framework

The proposed MPC scheme eases the computational burden by (i) increasing the computational interval to give more time for the real-time optimization (ii) reducing the number of decision variables to be optimized and (iii) utilizing the variable prediction horizon length. However, the MPC strategy becomes an open-loop optimal control within the MPC update interval T_{MPC} , and thus it is understandable that during this interval, the control would not perform optimally as designed. Within the interval T_{MPC} , the MPC cannot suppress any tracking error. Experiments have shown that the bandwidth of the MPC may not be adequate for stabilizing and controlling the quadcopter which has fast dynamics.

Figure 5.2 Two-layer formation flight framework

In order to overcome these difficulties in the implementation of the real-time MPC, a two-layer framework is adopted in the formation flight control. The top-layer controller is the MPC, which provides optimized state reference $\mathbf{x}_m$, whereas the low-level controller is a robust feedback linearization controller that can provide stability around the optimized state reference $\mathbf{x}_m$ in the presence of disturbances and uncertainties. The top-layer MPC controller runs at a lower sampling rate T_{MPC} due to

the computation time caused by solving the mixed integer linear program. In contrast, the bottom-layer controller works at a much higher sampling rate to reject disturbances and increase the control bandwidth of the whole formation flight system. The control structure is shown in Figure 5.2.

In the implementation, the bottom-layer controller measures real quadcopter states $\mathbf{x}$ and compares them with the state reference $\mathbf{x}_m$ from the top-layer MPC. The error signals are used to generate the motor control signals.

5.4 Top-Layer MPC-Based Formation Flight Control

Figure 5.3 The time setting example for the control input holding MPC

The traditional MPC is either derived based on a continuous system model or a discrete counterpart. Although a continuous-time model is much more natural and accurate in terms of describing the behavior of a dynamical system, the corresponding MPC algorithm involves continuously solving optimization problem (OP) which is difficult to implement for real-time optimization. In this research, the discrete MPC is used and further more two modifications are made and combined together to implement the real-time MPC formation flight control.

First modification: in order to reduce the computational power for the real-time MPC, a modified MPC strategy in which the control inputs are kept constant, i.e. zero-order holding, for several discretization intervals is introduced. In the traditional MPC setting, the MPC updating interval T_{MPC} is equal to the model discretization sampling time, T_d . With this method, the error between the discrete model and continuous model increases monotonically with T_d . However, the small T_d will increase the MPC computational burden, as there are more variables to be decided with respect to the same prediction length. So in order to reduce the computational burden, $T_{MPC} = NT_d$ is used in the real-time formation flight MPC setup, which implies that the values of the control inputs remain constant over N integration steps T_d . As illustrated in Figure 5.3, the control holding horizon is set to 3 steps which is the same as the prediction horizon H . Within the period of T_{MPC} , the control inputs are set to be constant, while the integration of the system equations follows the fast discrete sampling time T_d . This setting maintains the accuracy of the prediction but significantly reduces the number of the variables in the MPC optimization. For instance, when $N = 3$ and $H = 3$, in the conventional MPC there are $N \times H = 9$ variables in the optimization, while in Figure 5.3, only 3 variables need to be optimized.

Second modification: the formation flight MPC controller combines the constraint tightening technique which guarantees the robustness with a variable horizon length [63]. MPC with variable prediction horizon length is very suitable for

transient control problems like formation flight control, i.e. the follower quadcopter's target set $\mathcal{Q}_f$ is around the specified formation flight position. The significance of this is that the target set need not be invariant, and the variable horizon length MPC controller guarantees robust feasibility and finite-time entry of the target set, i.e. formation flight position. The other advantage is that by setting the predication horizon H as an optimization variable, it can also save computational efforts. For example, when the follower quadcopter is near the specified formation position, the prediction horizon H may be short as 1 or 2, while for the fixed H (commonly $H = 5$ or above), the MPC will always compute the same length trajectory which is a waste of the computational energy.

5.4.1 MPC Formation Flight Problem Statement

For the follower quadcopter, the aim is to maintain a specified distance from the leader quadcopter and at the same time try to avoid the obstacles during formation.

The discrete dynamics of the quadcopter during hover can be written as:

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\end{aligned}\tag{5.6}$$

where $\mathbf{x}(k) = (\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ X \ \dot{X} \ Y \ \dot{Y} \ Z \ \dot{Z})^T$, $\mathbf{y}(k) = (X \ \dot{X} \ Y \ \dot{Y} \ Z \ \dot{Z})^T$, $\mathbf{u}(k) = (\mathbf{T}, \boldsymbol{\tau}_\eta)^T$, and $\mathbf{w}(k) \in \mathcal{W}$ is the disturbance vector. State matrix $\mathbf{A}$ and input matrix $\mathbf{B}$ are obtained by linearizing Eq.(5.5) around the hover fly condition.

Under the piecewise constant control input and variable horizon setting, a

discrete MPC is employed for the formation flight control. By defining the leader quadcopter trajectory as $\mathbf{x}_l$ and the distance between the leader and follower as $\mathbf{D}$, then the formation error is defined as $\mathbf{x}_e = \mathbf{x}_l - \mathbf{D} - \mathbf{x}$. The performance index to be minimized can be formulated as:

$$J(k) = \sum_{i=0}^{H(k)-1} \left[\sum_{j=0}^N l(\mathbf{x}(k+iN+j|k), \mathbf{u}(k+iN+j|k)) + 1 \right] \quad (5.7)$$

where $H(k)$ is the predicted horizon to entry of $\mathcal{Q}_f$ at step k , N is the control input holding horizon, $l(\mathbf{x}(k), \mathbf{u}(k))$ is the stage cost function in the following form,

$$l(\mathbf{x}(k), \mathbf{u}(k)) = \|\mathbf{x}_e\|_Q + \|\mathbf{u}\|_R \quad (5.8)$$

where $\|\cdot\|_Q$ and $\|\cdot\|_R$ denote weighted norms. Typical choices for the norms would be quadratic forms $\mathbf{x}_e^T Q \mathbf{x}_e$ or weighted one-norms $|Q \mathbf{x}_e|$.

The formation flight problem $\mathcal{P}_{\text{formation}}(\mathbf{x}(k), \mathcal{Y}, \mathcal{Q}_f, \mathcal{W})$ for the follower quadcopter that minimizes the performance index (5.7) and subjects to system dynamics, input disturbances and various constraints can be stated as

$$\mathbf{x}_m, \mathbf{u}_m = \arg \min J(k) \quad (5.9)$$

subject to $\forall j \in \{0, 1, \dots, NH\}$:

$$\begin{aligned}
\mathbf{x}(k+j+1|k) &= \mathbf{A}\mathbf{x}(k+j|k) + \mathbf{B}\mathbf{u}(k+j|k) \\
\mathbf{y}(k+j|k) &= \mathbf{C}\mathbf{x}(k+j|k) \\
\mathbf{x}(k|k) &= \mathbf{x}(k) \\
\mathbf{y}(k+j|k) &\in \mathcal{Y}(j) \\
\mathbf{x}(k+HN(k)+1|k) &\in \mathcal{Q}_f(HN(k)+1)
\end{aligned} \tag{5.10}$$

where the double indices $(k+j|k)$ denote the prediction at time k of a value at time $k+j$. The constraint sets $\mathcal{Y}(j)$ are tightened by the following recursion [55]:

$$\begin{aligned}
\mathcal{Y}(0) &= \mathcal{Y} \\
\mathcal{Y}(j+1) &= \mathcal{Y}(j) \sim \mathbf{CL}(j)\mathcal{W}
\end{aligned} \tag{5.11}$$

where the operator ‘ $\sim$ ’ denotes the Pontryagin difference [67]. $\mathbf{L}(j)$ is defined as the state transition matrix for the closed-loop system under a candidate control law

$$\mathbf{u}(j) = \mathbf{K}\mathbf{x}(j)$$

In this case,

$$\begin{aligned}
\mathbf{L}(0) &= \mathbf{I} \\
\mathbf{L}(j+1) &= (\mathbf{A} + \mathbf{BK})\mathbf{L}(j)
\end{aligned} \tag{5.12}$$

Remark 5

The control input holding horizon N plays an important role in the modified MPC formulation. If $N=1$, the modified MPC reverts to the conventional discrete MPC. For the given prediction length, the increase of N can reduce the number of variables to be optimized, hence significantly reduce the computational burden.

Definition 1: Robust Feasibility

Assume that the formation flight MPC optimization problem can be solved from the initial state of the system. Then the system is robust-feasible, if for all future disturbances $\mathbf{w}(k) \in \mathcal{W} \forall k \geq 0$, the formation flight optimization problem can be solved at every subsequent step.

Algorithm 1 MPC formation flight

1. If $\mathbf{x}(k) \in Q_f$, i.e. the follower quadcopter reached the specified formation position, stop the MPC
2. Solve optimization $\mathcal{P}_{formation}(\mathbf{x}(k), \mathcal{Y}, Q_f, \mathcal{W})$
3. Pass the optimized $\mathbf{x}_m$ as reference to the bottom-layer controller
4. Increment the time step and repeat the process

5.4.2 Robust feasibility of the formation flight

Theorem 1: *Robust feasibility of the formation flight MPC control for the follower quadcopter*

Given an initial feasible solution to the $\mathcal{P}_{formation}(\mathbf{x}(k), \mathcal{Y}, Q_f, \mathcal{W})$, by applying the *Algorithm 1*, then the formation flight problem for follower quadcopter is robustly feasible.

Proof:

In order to proof this theorem, similar to that of [55], we need to show that feasibility of $\mathcal{P}_{formation}(\mathbf{x}(k_0), \mathcal{Y}, Q_f, \mathcal{W})$ at some time k_0 will ensure feasibility of the subsequent formation problem $\mathcal{P}_{formation}(\mathbf{x}(k_0 + N), \mathcal{Y}, Q_f, \mathcal{W})$ for all

disturbances $\mathbf{w}(k_0) \in \mathcal{W}$.

Assume $\mathcal{P}_{\text{formation}}(\mathbf{x}(k_0), \mathcal{Y}, \mathcal{Q}_f, \mathcal{W})$ is feasible with a horizon of $H(k_0)$. Then it has a feasible (not necessarily optimal) solution, denoted by $*$, with states $\mathbf{x}^*(k_0+j|k_0)$, $j \in \{0, \dots, H(k_0)(N+1)\}$, inputs $\mathbf{u}^*(k_0+j|k_0)$, $j \in \{0, \dots, NH(k_0)\}$ and output $\mathbf{y}^*(k_0+j|k_0)$, $j \in \{0, \dots, NH(k_0)\}$ satisfying all of the constraints (5.10). Now, to prove feasibility of the subsequent optimization, a candidate solution is constructed and then shown to satisfy the constraints accordingly. Consider the following candidate solution, denoted by $\hat{\cdot}$, for $\mathcal{P}_{\text{formation}}(\mathbf{x}(k_0+N), \mathcal{Y}, \mathcal{Q}_f, \mathcal{W})$

$$\hat{H}(k_0+N) = H(k_0) - 1 \quad (5.13)$$

$$\begin{aligned} \hat{\mathbf{u}}(k_0+j+N|k_0+N) &= \mathbf{u}^*(k_0+j+N|k_0) + \mathbf{KL}(j+N-1)\mathbf{w}(k_0), \\ &\forall j \in \{0, \dots, N(H(k_0)-1)\} \end{aligned} \quad (5.14)$$

$$\begin{aligned} \hat{\mathbf{x}}(k_0+j+N|k_0+N) &= \mathbf{x}^*(k_0+j+N|k_0) + \mathbf{L}(j+N-1)\mathbf{w}(k_0), \\ &\forall j \in \{0, \dots, NH(k_0)\} \end{aligned} \quad (5.15)$$

$$\begin{aligned} \hat{\mathbf{y}}(k_0+j+N|k_0+N) &= \mathbf{C}\hat{\mathbf{x}}(k_0+j+N|k_0+N), \\ &\forall j \in \{0, \dots, N(H(k_0)-1)\} \end{aligned} \quad (5.16)$$

where $\mathbf{K}$ denotes a stabilizing feedback law and the corresponding state transition matrices $\mathbf{L}(j)$ are defined in (5.12). Note that the candidate solution here is one step shorter than the previous solution. To prove that this candidate solution (5.13)-(5.16) is feasible, it is necessary to show that it satisfies all the constraints (5.10) at time step

$k = k_0 + N$ for any disturbance $\mathbf{w}(k_0) \in \mathcal{W}'$.

Satisfaction of dynamics constraints (the first Eq. of (5.10)):

Since the previous plan satisfied the dynamics equation, we know

$$\begin{aligned} \mathbf{x}^*(k_0 + j + N + 1 | k_0) &= \mathbf{A}\mathbf{x}^*(k_0 + j + N | k_0) + \mathbf{B}\mathbf{u}^*(k_0 + j + N | k_0) \\ &\quad \forall j \in \{0, \dots, N(H(k_0) - 1)\} \end{aligned} \quad (5.17)$$

Substituting on both sides from the definition of the candidate solution Eq. (5.14) and Eq. (5.15), we get,

$$\begin{aligned} &\hat{\mathbf{x}}(k_0 + j + N + 1 | k_0 + N) - \mathbf{L}(j + N)\mathbf{w}(k_0) \\ &= \mathbf{A}[\hat{\mathbf{x}}(k_0 + j + N | k_0 + N) - \mathbf{L}(j + N - 1)\mathbf{w}(k_0)] + \\ &\quad \mathbf{B}[\hat{\mathbf{u}}(k_0 + j + N | k_0 + N) - \mathbf{K}\mathbf{L}(j + N - 1)\mathbf{w}(k_0)] \\ &= \mathbf{A}\hat{\mathbf{x}}(k_0 + j + N | k_0 + N) + \mathbf{B}\hat{\mathbf{u}}(k_0 + j + N | k_0 + N) \\ &\quad - (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{L}(j + N - 1)\mathbf{w}(k_0) \end{aligned} \quad (5.18)$$

Using the definition of the state transition matrices $\mathbf{L}$ in Eq. (5.12), Eq. (5.18) is reduced to:

$$\hat{\mathbf{x}}(k_0 + j + N + 1 | k_0 + N) = \mathbf{A}\hat{\mathbf{x}}(k_0 + j + N | k_0 + N) + \mathbf{B}\hat{\mathbf{u}}(k_0 + j + N | k_0 + N) \quad (5.19)$$

which satisfies the dynamics constraint. Satisfaction of the other constraints in Eq.

(5.10) can be proved in the same manner which is omitted here.

5.4.3 Formation Stability and Finite Time Formation Reachability

Theorem 2: formation stability and finite time formation reachability

If the formation flight problem $\mathcal{P}_{\text{formation}}(\mathbf{x}(k_0), \mathcal{Y}, \mathcal{Q}_f, \mathcal{W})$ has a feasible solution

with cost $J^*(\mathbf{x}(k_0))$ and

$$\begin{aligned} \gamma = 1 + \sum_{j=0}^{N-1} \left[\left\| \mathbf{x}_e^*(k_0 + j | k_0) \right\|_Q + \left\| \mathbf{u}^*(k_0 + j | k_0) \right\|_R \right] - \\ \sum_{i=0}^{H(k_0)-2} \sum_{j=0}^N \left(\left\| \mathbf{L}(j+N-1) \mathbf{w}(k_0) \right\|_Q + \left\| \mathbf{K} \mathbf{L}(j+N-1) \mathbf{w}(k_0) \right\|_R \right) \geq 0 \end{aligned} \quad (5.20)$$

then the formation flight control for the follower quadcopter is stable and furthermore the follower quadcopter will enter the specified formation shape set Q_f in $N_{\max}$ steps

or fewer, where $N_{\max}$ is the integer part of $\frac{J^*(\mathbf{x}(k_0))}{\gamma}$.

Proof:

Theorem 1 showed that if a solution $\mathcal{P}_{\text{formation}}(\mathbf{x}(k_0), \mathcal{Y}, Q_f, \mathcal{W})$ exists for the formation flight problem then a particular candidate solution Eq. (5.13)-(5.16) $\mathcal{P}_{\text{formation}}(\mathbf{x}(k_0 + N), \mathcal{Y}, Q_f, \mathcal{W})$ is feasible for the subsequent problem. The cost of this candidate solution $\hat{J}(\mathbf{x}(k_0 + N))$ can be used to bound the optimal cost at a particular time step $J^*(\mathbf{x}(k_0 + N))$ relative to the optimal cost at the preceding step $J^*(\mathbf{x}(k_0))$. A Lyapunov-like argument based on this function will then be used to prove entry to the formation flight target set Q_f . $J^*(\mathbf{x}(k_0))$ can be written as

$$\begin{aligned} J^*(\mathbf{x}(k_0)) &= \sum_{i=0}^{H(k_0)-1} \left[\sum_{j=0}^N l(\mathbf{x}^*(k_0 + iN + j | k_0), \mathbf{u}^*(k_0 + iN + j | k_0)) + 1 \right] \\ &= \sum_{i=0}^{H(k_0)-1} \sum_{j=0}^N \left(\left\| \mathbf{x}_e^*(k_0 + iN + j | k_0) \right\|_Q + \left\| \mathbf{u}^*(k_0 + iN + j | k_0) \right\|_R \right) + H(k_0) \end{aligned} \quad (5.21)$$

The cost of the candidate solution in Eqs. (5.13)-(5.16) is,

$$\begin{aligned}
\hat{J}(\mathbf{x}(k_0+N)) &= \sum_{i=0}^{\hat{H}(k_0+N)-1} \left[\sum_{j=0}^N l(\hat{\mathbf{x}}(k_0+j+N | k_0+N), \hat{\mathbf{u}}(k_0+j+N | k_0+N)) + 1 \right] \\
&= \sum_{i=0}^{\hat{H}(k_0+N)-1} \sum_{j=0}^N \left(\|\hat{\mathbf{x}}_e(k_0+iN+j+N | k_0+N)\|_Q + \|\hat{\mathbf{u}}(k_0+iN+j+N | k_0+N)\|_R \right) + \hat{H}(k_0+N) \\
&= \sum_{i=0}^{H(k_0)-2} \sum_{j=0}^N \left(\|\mathbf{x}_e^*(k_0+iN+j+N | k_0) + \mathbf{L}(j+N-1)\mathbf{w}(k_0)\|_Q + \right. \\
&\quad \left. \|\mathbf{u}^*(k_0+iN+j+N | k_0) + \mathbf{KL}(j+N-1)\mathbf{w}(k_0)\|_R \right) + H(k_0) - 1
\end{aligned} \tag{5.22}$$

The cost of the candidate solution can be bounded from above using the triangle inequality, applied to each term in the summation:

$$\begin{aligned}
\hat{J}(\mathbf{x}(k_0+N)) &\leq \sum_{i=0}^{H(k_0)-2} \sum_{j=0}^N \left(\|\mathbf{x}_e^*(k_0+iN+j+N | k_0)\|_Q + \|\mathbf{L}(j+N-1)\mathbf{w}(k_0)\|_Q + \right. \\
&\quad \left. \|\mathbf{u}^*(k_0+iN+j+N | k_0)\|_R + \|\mathbf{KL}(j+N-1)\mathbf{w}(k_0)\|_R \right) + H(k_0) - 1
\end{aligned} \tag{5.23}$$

This bound can now be expressed in terms of the previous optimal cost in (5.21),

$$\begin{aligned}
\hat{J}(\mathbf{x}(k_0+N)) &\leq J^*(\mathbf{x}(k_0)) - \sum_{j=0}^{N-1} \left(\|\mathbf{x}_e^*(k_0+j | k_0)\|_Q + \|\mathbf{u}^*(k_0+j | k_0)\|_R \right) + \\
&\quad \sum_{i=0}^{H(k_0)-2} \sum_{j=0}^N \left(\|\mathbf{L}(j+N-1)\mathbf{w}(k_0)\|_Q + \|\mathbf{KL}(j+N-1)\mathbf{w}(k_0)\|_R \right) - 1 \\
&= J^*(\mathbf{x}(k_0)) - \gamma
\end{aligned} \tag{5.24}$$

Note that at time instant k_0+N , the real-time optimization will be performed to minimize the cost function. Thereby, the cost of the candidate solution forms an upper bound on the optimal cost,

$$J^*(\mathbf{x}(k_0+N)) \leq J^*(\mathbf{x}(k_0)) \tag{5.25}$$

Next, $J^*(\mathbf{x})$ is adopted as the Lyapunov function for the proposed MPC formation flight. The stability condition based on Lyapunov theory requires:

$$J^*(\mathbf{x}(k_0+N)) \leq J^*(\mathbf{x}(k_0)) \quad (5.26)$$

To achieve (5.26), we can impose a condition on (5.24), i.e. $\gamma \geq 0$ set such that :

$$J^*(\mathbf{x}(k_0+N)) \leq \hat{J}(\mathbf{x}(k_0+N)) \leq J^*(\mathbf{x}(k_0)) - \gamma \quad (5.27)$$

So the stability condition (5.20) is derived.

If the follower quadcopter remains outside the formation flight target set for Z steps, the cost is:

$$J^*(\mathbf{x}(k_0+ZN)) \leq J^*(\mathbf{x}(k_0)) - Z\gamma \quad (5.28)$$

And the left-hand side would be negative if $Z > N_{\max}$. This is a contradiction, as the cost must be positive, by construction. Hence the follower quadcopter must enter the formation flight in $N_{\max}$ steps or fewer.

Remark 6

From Theorem 2, the robustness is actually depend on the disturbances, so based on the measured or estimated disturbance, the stability can be verified at beginning of each formation flight.

5.5 Bottom-Layer Controller Design Based On Robust Feedback Linearization

While the MPC controller could in principles be used to close the formation

loop around the quadcopter dynamics, it is often more practical to close the formation flight loop around the augmented dynamics. This ensures that the dynamics response of the quadcopter is more consistent, i.e. reduce the effects of modeling uncertainty and disturbance. At the same time, the control augmentation reduces the effective order of the vehicle dynamics, allowing more simple dynamic models to be used for the real-time MPC optimization. It also lowers the requirement on the bandwidth of the formation flight control loop.

The bottom-layer controller consists of two cascaded loops, namely the position loop with a relative degree of two, and an attitude control loop which also with a relative degree of two. Since the quadcopter is a highly nonlinear, multi-variable and strongly coupled system, a nonlinear controller based on robust feedback linearization has been developed.

Feedback linearization as a baseline control strategy can exploit the physical capability of the quadcopter, and therefore accomplish complex, highly curved three-dimensional trajectories. With adequate knowledge of the plant dynamics, the control approach can transform the nonlinear system into an equivalent linear system without any simplification, through exact state transformation and suitable control inputs. [68].

5.5.1 Robust Feedback Linearization Control of the Rotational Motion

The inputs of the attitude loop are the desired Euler angles, and output signals are the commanded torques on the quadcopter. The rotational movements dynamic model (5.4) is used to develop the robust feedback linearization based attitude controller. τ_n joints the control torques and external disturbances, and is redefined as:

$$\boldsymbol{\tau}_\eta = \boldsymbol{\tau}_{\eta a} + \boldsymbol{\tau}_{\eta b} \quad (5.29)$$

where $\boldsymbol{\tau}_{\eta a}$ is the applied torques vector and $\boldsymbol{\tau}_{\eta b}$ represents the total effect of system modeling errors and external disturbances.

The following control law is proposed for the rotational dynamics subsystem:

$$\boldsymbol{\tau}_{\eta a} = \mathbf{M}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) + (\mathbf{K}_D\dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_P\tilde{\boldsymbol{\eta}}) \quad (5.30)$$

where $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_r - \boldsymbol{\eta}$, $\boldsymbol{\eta}_r$ is the desired Euler angles.

This control law can be split up into two different parts: the first one consists of the first two term of Eq. (5.30) which are designed in order to compensate the system dynamics (5.4). The second part consists of two terms including the error vector $\tilde{\boldsymbol{\eta}}$ and its derivative, $\dot{\tilde{\boldsymbol{\eta}}}$. These two terms of the control law enable robust tracking in the presence of the input disturbance $\boldsymbol{\tau}_{\eta b}$.

By substituting the expression of the control law from Eq. (5.30) into rotational dynamics Eq.(5.4), the following dynamic equation of the attitude error is obtained:

$$\mathbf{K}_D\dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_P\tilde{\boldsymbol{\eta}} = -\boldsymbol{\tau}_{\eta b} \quad (5.31)$$

To verify the gain matrices $\mathbf{K}_D$ and $\mathbf{K}_P$ are robust to the input disturbance $\boldsymbol{\tau}_{\eta b}$, the linear matrix inequality (LMI) technique can be utilized. Eq. (5.31) can be rewritten in the following form:

$$\dot{\tilde{\boldsymbol{\eta}}} = \mathbf{A}_{\tilde{\boldsymbol{\eta}}}\tilde{\boldsymbol{\eta}} + \mathbf{g}(\tilde{\boldsymbol{\eta}}) \quad (5.32)$$

where, $A_{\tilde{\eta}} = -\mathbf{K}_D^{-1}\mathbf{K}_P$ and $g(\tilde{\eta}) = -\mathbf{K}_D^{-1}\boldsymbol{\tau}_{\eta b}$.

With the chosen $\mathbf{K}_D$ and $\mathbf{K}_P$, the state matrix $A_{\tilde{\eta}}$ can be calculated and the disturbance term $g(\tilde{\eta})$ can be bounded by

$$\|g(\tilde{\eta})\| \leq r \|\tilde{\eta}\| \quad (5.33)$$

with known r . In order to ensure the error dynamics (5.32) is exponentially stable, we need to find a quadratic Lyapunov function $V(\tilde{\eta}) = \tilde{\eta}^T \mathbf{P} \tilde{\eta}$ with $\mathbf{P} > 0$ and

$$\dot{V}(\tilde{\eta}) \leq -\alpha V(\tilde{\eta}) \quad (5.34)$$

for all $\tilde{\eta}$ ($\alpha > 0$ is given).

By using S-procedure [69], combining Eqs. (5.33) and (5.34), the necessary and sufficient conditions for the existence of quadratic Lyapunov function can be expressed as linear matrix inequality (LMI),

$$\mathbf{P} > 0, \quad \begin{pmatrix} A_{\tilde{\eta}}^T \mathbf{P} + \mathbf{P} A_{\tilde{\eta}} + \alpha \mathbf{P} + \tau r^2 I & \mathbf{P} \\ \mathbf{P} & -\tau I \end{pmatrix} \leq 0 \quad (5.35)$$

in variables $\mathbf{P}$, τ . Note that solving this LMI to find $\mathbf{P}$ is a better and systematical method than for example solving the Lyapunov equation $A_{\tilde{\eta}}^T \mathbf{P} + \mathbf{P} A_{\tilde{\eta}} + I = 0$ and hoping the resulting $\mathbf{P}$ works.

5.5.2 Feedback Linearization Control on the Translational Motion

In the position loop, the input command is the desired position in the world coordinate, while the control signals are given as input for the attitude loop.

Since the yaw control can be decoupled from the position control, the heading angle ψ can be taken as a known parameter. From the translational dynamics (5.1), solving for the pitch, roll and thrust leads to the inverse equations:

$$\begin{aligned} \mathbf{T} &= -m\sqrt{\ddot{X}^2 + \ddot{Y}^2 + (\ddot{Z} - g)^2} \\ \phi &= \sin^{-1}\left(\frac{m\ddot{X}}{\mathbf{T}}\sin(\psi) - \frac{m\ddot{Y}}{\mathbf{T}}\cos(\psi)\right) \\ \theta &= \cos^{-1}\left(\frac{m\ddot{Z} - mg}{\mathbf{T}\cos(\phi)}\right) \end{aligned} \quad (5.36)$$

and the following robust PD controller can be applied:

$$\begin{aligned} \ddot{X} &= \ddot{X}_r + k_{xd}(\dot{X}_r - \dot{X}) + k_{xp}(X_r - X) \\ \ddot{Y} &= \ddot{Y}_r + k_{yd}(\dot{Y}_r - \dot{Y}) + k_{yp}(Y_r - Y) \\ \ddot{Z} &= \ddot{Z}_r + k_{zd}(\dot{Z}_r - \dot{Z}) + k_{zp}(Z_r - Z) \end{aligned} \quad (5.37)$$

The stability of the translational dynamics subsystem can be guaranteed by using the Lyapunov function technique[68]. Note that in both cascade controllers design process, there are no simplifications in the dynamics nor small angle approximation is used.

5.6 Stability Statement of the Two-Layer Formation Flight Control Framework

The flight formation control framework for the follower quadcopter consists of two layers: top-layer MPC for the desired trajectory planning and bottom-layer robust feedback linearization controller for trajectory stabilization and tracking. Previously the stability of the MPC and the robust feedback linearization controller with bounded disturbances has been discussed separately. Then the stability of the whole

system needs to be investigated and justified.

In the absence of model uncertainty and disturbances, the follower quadcopter states $\mathbf{x}(t)$ should always maintain the formation flight preterm. In reality, within an MPC sampling interval $t \in [t_k, t_{k+1})$, the actual state of the quadcopter will diverge from predicted states $\mathbf{x}_m(t)$ due to unmodelled dynamics and/or disturbances and furthermore the leader quadcopter may also change its trajectory within this interval. However, within a surrounding aerea of $\mathbf{x}_m(t)$, the error dynamics can be described by a linearised system, where the surrounding is defined as $\Omega(\mathbf{x}_m) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_m\| \leq \bar{e}\}$, where $\bar{e}$ is a small positive scalar.

Theorem 3. *The two-layer control ensures that for any perturbed state $\mathbf{x}(t) \in \Omega(\mathbf{x}_m)$. $t \in (t_k, t_{k+1})$, caused by a bounded constant disturbance $\mathbf{d}$, its divergence $\mathbf{x}_\Delta(t)$ is bounded for all $t > 0$.*

Proof. The behavior of the divergence $\mathbf{x}_\Delta(t)$ can be described by the following equation,

$$\dot{\mathbf{x}}_\Delta = A_{cl}\mathbf{x}_\Delta + E\mathbf{d} \quad (5.38)$$

where A_{cl} is the lineaized state matrix of Eq.(5.5) with the feedback linearization controller applied. E is a disturbance input matrix. As at the time $t = t_k$, the top-layer MPC reset the error state to zero, i.e. $\mathbf{x}_\Delta = 0$, the state response in the period $t \in (t_k, t_{k+1})$ can be written as:

$$\mathbf{x}_{\Delta}(t) = A_{cl}^{-1} \left(e^{A_{cl}(t-t_k)} - I \right) E \mathbf{d} \quad (5.39)$$

Therefore, there exists a bound on $\mathbf{x}_{\Delta}$, which may depend on the magnitude of the disturbance. Furthermore, the top-layer MPC resets the error state to zero at $t = t_{k+1}$, i.e. $\mathbf{x}_{\Delta}(t_{k+1}^+) = 0$. After MPC synchronises the real quadcopter states $\mathbf{x}$ and MPC state $\mathbf{x}_m$, the overall stability is guaranteed as MPC runs in an closed-loop fashion.

Furthermore, because of the robust design of the bottom layer controller, the disturbance happened during the MPC sampling interval can be rejected at a high running frequency. The whole formation flight system is verified to be stable under many flight tests which are presented in the following flight test section.

5.7 Formation Flight Test

Figure 5.4 Quadcopter during autonomous flight experiment in the flight-test facility of the UAV lab

The real-time formation flight system was implemented and evaluated in the UAV lab of Nanyang Technological University of Singapore. The lab was setup to enable evaluation of trajectory planning and control algorithms within controlled conditions. The UAV lab setup allows the verification and testing of the combination of the UAV hardware and software components, which lead to a good understanding and comparisons of different components contribute to the overall flight behavior and performance. Figure 5.4 shows a picture of the facility with the quadcopter during autonomous flying experiments, and Figure 5.5 provides a block diagram view of the system used for the experimental validation for the formation flight. Table 5.1 gives the control settings that were used for the formation flight demonstration.

Table 5.1: Key parameters of the formation flight system used for the formation flight experiments

Control input holding horizon N	5
Discretization time T_d	0.04
MPC sampling time T_s	0.2
Weighting matrix Q	$\text{diag}(0.1 \cdot \text{ones}(6,1) \ 1 \cdot \text{ones}(6,1))$
Weighting matrix R	$\text{diag}(0.02 \cdot \text{ones}(4,1))$
Model used in MPC	Linearization around hovering the point
Stabilization controller of MPC	LQR
Maximum horizontal velocity	2m/s
Maximum vertical velocity	1.2m/s
Follower quadcopter onboard bottom-layer controller update rate	Euler angle loop: 1000 Hz Position loop: 50 Hz
Vicon data fetching	50 Hz
Xbee telemetry	About 50 Hz
Obstacle size	100X100 mm square

5.7.1 Experimental flight-test facility

The Hummingbird [70] quadcopter (the specific parameters are listed in Table 5.

2) was used as a testbed to verify the proposed formation flight framework. The flight

test facility is build around the Vicon motion tracking system [71] which provides the position measurement with an update rate of 100 Hz and accuracy level of 1 mm. The test quadcopter's position data are provided in real time by Vicon-MX motion tracking cameras. This setup eliminates the need for on-board navigation sensors, like GPS for outdoor navigation.

Table 5. 2 Quadcopter Specific Parameters

Mass (kg)	0.68	$I_{11}(= I_{22})$	0.007	Propeller	8X4.5
Motor arm length (m)	0.17	I_{33}	0.012	Flight time	20 minutes

This test platform adopts a number of pieces of commercial-off-the-shelf equipment and combines them effectively into the Matlab/Simulink environment, which provides a seamless way from control analysis and design to numerical simulation and experimental verifications.

5.7.2 Formation Flight Control Computational Setup

The attitude, position control loops and sensor data fusion algorithm are implemented onboard on the ARM 7 processor in the quadcopter. The onboard attitude and position control are running at 1 kHz and 100 Hz, respectively.

The ground control computer, HP 4411s (has Intel Core Duo CPU, 3GB memory), is responsible for fetching the position data from the Vicon system, computing the MPC formation flight control commands, sending the reference states (desired position, velocity) via the Xbee telemetry system to the follower quadcopter and providing basic user interface functionalities such as switching between different tasks and operational modes. The flight data is recorded in Simulink using a customized S-function.

5.7.3 Experimental Results

During the flight test, since there is no air conditioner in the UAV hangar, the fans on the wall cause additional disturbance to the quadcopter. The maximum of this disturbance is about 1 m/s.

To exercise the formation flight system and demonstrate the capabilities of the proposed control architecture, a wand is used as the virtual leader and the follower quadcopter is commanded to maintain $L=[0, 0.7, -0.9]$ m relative to the virtual leader in the global frame, and at the same time try to avoid the obstacle in the flight space.

Figure 5.6 3D formation flight trajectories

Figure 5.7 XY plane view of the formation flight trajectory

Figure 5.6 shows the 3D trajectories of the formation flight with obstacle avoidance. During the experiment, the virtual leader (the wand hold by the researcher) was purposely “flying” around the obstacle to test the robustness and consistence of the proposed formation flight control algorithm. The follower quadcopter can successfully maintain the flight formation with respect to the virtual leader while keeping itself safe from the obstacle. The formation trajectories is also plotted in X-Y plane (see Figure 5.7) in order to clearly show the obstacle collision-free flight of the follower quadcopter.

Figure 5.8 X positions of formation flight system

Figure 5.9 Y positions of formation flight system

Figure 5.8 and Figure 5.9 show the X and Y positions respectively. The solid blue line is the wand trajectory and the red dot line is the commanded trajectory generated by the MPC formation flight top-layer controller. The experiment demonstrated that by using the proposed MPC controller, smooth optimal trajectory can be generated real-time and the formation shape is guaranteed. As can be seen in both X and Y position, the actual trajectory of the follower quadcopter (the dash black line) can successfully tracking the optimal trajectory from MPC. So by combining the MPC top-layer control with bottom-layer robust feedback linearization controller, tight formation flight with obstacle avoidance in the flight field can be achieved.

Variability in the performance is inevitable because of the modeling uncertainties and disturbances, such as the rotor wake. Even with these effects, the

performance of the proposed formation flight control is relatively consistent. A significant number of formation flight tests have been conducted (a formation flight demo video is attached) and Figure 5.10 shows the trajectories of the formation flight for about 10 minutes which verifies the stability, consistency and robustness of the proposed method.

Figure 5.10 Near 10 minutes formation flight

The computational load of the proposed formation flight scheme was also evaluated. The CPU time for each MPC update was used as the measurement for the formation flight computational performance. During all the flight testing, the overall computing time was small enough to allow the real-time implementation.

To give a general idea of the computational load of the proposed formation flight scheme, the computational time for the MPC at each optimization step and their corresponding prediction horizon H are evaluated in simulation of 50 different initial

points. To illustrate the effects of the MPC input hold N , different values of N ($N = \{5, 6, 7\}$) are simulated. The simulations were performed on the same laptop HP 4411s (has Intel Core Duo CPU, 3GB memory) which is used for the formation flight test. As illustrated in Table 5.3, the computational time is acceptable for the real-time formation flight implementation and the prediction horizon length H is also finite during the simulations and all the formation flight tests. This confirms Theorem 2 which guarantees that the follower quadcopter enters the formation position within the finite time steps. In addition, in order to compare the proposed modified MPC formation flight control scheme with the original one, i.e. without input holding and with fixed prediction length, another 50 simulations are carried out. The setting for the proposed MPC is $T_d = 0.04s$, $N = 5$ and $T_s = 0.2$. From Table 5.3, the mean prediction length is 10, so the average prediction time is $T_s \times 10 = 2s$. In order to give the same prediction time, the original MPC has to use the prediction length of 50 steps. The computational time and formation error for both MPC are listed in Table 5.4, although the formation error is larger when the proposed MPC is applied, for the formation flight scenario, the computational load is reduced significantly by combining the input hold and variable prediction length. For the conventional MPC, the computational time is not satisfactory and may not satisfy on-line implementation since the maximum time (2.95s) is even larger than the prediction time (2s).

Table 5.3: Computational load and the prediction horizon H with respect to different MPC input holding length N

MPC input holding length (N)	Max computation time (s)	Min computation time (s)	Mean computation time (s)	Std. computation time (s)
5	0.17	0.03	0.13	0.021
6	0.18	0.04	0.14	0.022
7	0.18	0.04	0.14	0.025
MPC input holding length (N)	Max prediction length H	Min prediction length H	Mean prediction length H	Std. prediction length H
5	18	1	10	3
6	17	1	11	2
7	17	1	13	3

Table 5.4: Comparison between the conventional MPC and the proposed MPC

	Max computation time (s)	Min computation time (s)	Mean computation time (s)	Std. computation time (s)	Formation error (m)
Proposed MPC	0.17	0.03	0.13	0.02	0.13
Conventional MPC	2.95	0.32	1.75	0.13	0.06

Finally, there is room to improve the formation flight performance. First the The MPC optimization code used in the experiments is modified from the open source [61] and [62]. This code can be further optimized for the formation flight, for example the warm-start method can be used and in the optimization the total number of Newton steps can be limited to further reduce the computational burden. Second, a single board computer like the BeagleBoard [72] or the Overo Fire board[73] can be equipped to the onboard quadcopter such that the MPC top-layer controller can be run onboard to reduce the Xbee transmission latency. Moreover when the computation is sufficiently efficient, the Vicon update rate can be increased to 200HZ, which will cause the overall system bandwidth to increase further.

5.8 Concluding Remarks

This chapter addressed and demonstrated a novel formation flight architecture for the quadcopters based on top-layer MPC and bottom-layer robust feedback linearization approach. In the top-layer MPC, the input holding and variable

prediction horizon are combined to reduce the computational burden and the formation flight stability, optimization feasibility and finite time formation reachability are proved. The formation flight experiments were performed in the UAV lab and the test results verified the effectiveness of the proposed formation flight architecture.

Chapter 6 Combination of Adaptive Neural Networks and Reachability Algorithm in MPC Formation Flight Framework

Based on the formation flight approach presented in the last chapter, this chapter presents a novel formation flight control method which augments the feedback linearization with the neural networks (NN) adaptive control and adds reachability analysis within the MPC. This approach aims to ensure safety for extreme formation flight maneuver by giving the follower UAV safer control commands. The advantages of this method are: 1) the combination of MPC and reachability analysis makes the formation flight safer; 2) the parametric uncertainty and unmodelled dynamics of the UAV system can be dealt with the NN adaptive control. However, the disadvantage is the computation is relatively heavy.

In order to prevent the follower quadcopter from loss of control during extreme formation maneuvers, a novel reachability algorithm is proposed in the MPC formation flight controller which constraints the states more during extreme flight maneuvers. The reachability algorithm calculated the safe maneuver envelope at each sampling time. In order to prevent the adaptive NN elements from adapting the inputs saturations, the “Pseudo-Control Hedging” technique is implemented in the MPC formation flight planner, which resulting the tight coupling between the attitude adaptive control and the MPC planner.

6.1 Introduction

The flight control architecture presented in this chapter consists of two layers: the bottom layer for attitude tracking based on adaptive NN augmentation of the feedback linearization (proposed in Chapter 5), and the upper layer for optimal formation trajectory generation. Based on the desired formation position and the states of the UAV, the MPC controller calculates the safe operating envelop which determines the instantaneous flight state constraints for the real-time optimization, then MPC controller generates the desired optimal attitude angles of the UAV body axis which are the inputs for the bottom layer adaptive NN controller. The flight control structure is shown in Figure 6.1.

The connection between the upper layer MPC flight control and the bottom layer attitude control is the “Pseudo-Control Hedging (PCH)” technique from [74] and the real-time state constraints adjustment based on the reachability analysis/calculation. Based on the idea of PCH, the method proposed in this chapter not only prevents the adaptive element of an adaptive control system from trying to adapt to the input characteristics (the motor characteristics like saturation), but also forms a safe maneuver envelope determination through reachability analysis which makes the formation flight safer.

A plain-language conceptual description of this method is: by estimating the amount the plant did not move due to some system (input) characteristics, the upper layer MPC formation flight controller will reduce this amount in the optimization process. At the same time, by determining the safe operating envelope through the reachability analysis, the on-line optimization also adjusts the states’ constraints on-

line which makes the MPC upper layer adaptive too.

6.2 Input Saturation and Adaptive Controllers

When designing adaptive flight control system, normally the controller designer only considers full authority controllers, and there is very less considerations related to input dynamics, saturation, and other system input characteristics. In this

Figure 6.1 Formation flight: MPC combining NN adaptive control
section, some improvements in adaptive control theory are reviewed.

Input saturation present a significant problem for adaptive control, and sometimes such saturation will lead to even worse control results than using a non-adaptive controller. Saturation violates any affinity in control assumptions, which is common in the control literature. It also violates the assumption that the sign of the effect of the control is known/non-zero, since the effect of additional control input action is zero when saturation is encountered. Input saturation and rate saturation can cause an adaptation law to be “tricked” by unexpected effects. This kind of behavior is analogous to integrator windup in a PID controller. Like integrator windup, these

effects can dramatically reduce the domain of attraction of a commanded equilibrium, and in the worst case, this behavior can preclude any command or disturbance that would involve an input saturation, even for just a brief period.

Adaptive controllers that consider input saturation are limited in the literature. The methods reported include: command alteration to prevent or avoid saturation, slowing/halting the effective adaptation rate as saturation is entered, reference model tracking error modification and Pseudo-Control Hedging (PCH) technique.

Command alteration: this has been demonstrated in an adaptive control setting [75]. This method is able to determine exactly how much the command signal needs to be modified to prevent a specific adaptive controller from exceeding saturation limits [76].

- Slowing/halting the adaptation when saturation is entered: A common ad-hoc approach for most adaptive control methods is to simply stop adaptation completely when any input saturates. In addition to the lack of stability results, this approach created the potential problem of not being able to “adapt out of” an undesired equilibrium condition in which an input is unnecessarily saturated. In the category of slowing adaptation, there are results that bound the feedback control by some form of a squashing function, such as [77], where adaptation is effectively being slowed because the absolute upper bound on plant input is approached only asymptotically.
- Error signal augmentation: this method is often applied to Model Reference Adaptive Control (MRAC). For this method, the effect of the saturation

nonlinearity on the reference model tracking error is removed by adding a signal derived from the actual plant input to the error signal [78-81].

- Pseudo-Control Hedging (PCH) technique: This method is proposed by Eric Johnson [74] for adaptive control using NNs (not limited to NNs). This method is most related to augmented error signal, in that it also relies on removing input characteristics from the error signal used for adaptation.

6.3 Attitude Loop Controller Description

6.3.1 Dynamic Inversion Technique

The quadcopter dynamics for the attitude loop control is the same as in Eq. (5.4) in chapter 5 which can be put into the following form,

$$\begin{aligned}\ddot{\boldsymbol{\eta}} &= \mathbf{M}(\boldsymbol{\eta})^{-1} (\boldsymbol{\tau}_{\eta} - \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})) \\ &= \mathbf{f}_{\eta}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\tau}_{\eta})\end{aligned}\tag{6.1}$$

An approximate dynamic inversion element is developed to determine actuator (torque) commands of the form,

$$\boldsymbol{\tau}_{\eta_{cmd}} = \hat{\mathbf{f}}_{\eta}^{-1}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \mathbf{v}_{\eta})\tag{6.2}$$

where $\mathbf{v}_{\eta}$ is the pseudo-control signal, and presents a desired angular acceleration that is to be achieved by $\boldsymbol{\tau}_{\eta_{cmd}}$.

$$\mathbf{v}_{\eta} = \hat{\mathbf{f}}_{\eta}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\tau}_{\eta_{cmd}})\tag{6.3}$$

6.3.2 The Pseudo-Control Signal

In this chapter, a Single Hidden Layer (SHL) NN will be trained online to approximate the model error function

$$\Delta(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\tau}_{\boldsymbol{\eta}_{cmd}}) = \mathbf{f}_{\boldsymbol{\eta}}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\tau}_{\boldsymbol{\eta}_{cmd}}) - \hat{\mathbf{f}}_{\boldsymbol{\eta}}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\tau}_{\boldsymbol{\eta}_{cmd}}) \quad (6.4)$$

The complete pseudo-control signal for the attitude control system in Figure 6.1 is,

$$\mathbf{v}_{\boldsymbol{\eta}} = \mathbf{v}_{\boldsymbol{\eta}_{crm}} + \mathbf{v}_{\boldsymbol{\eta}_{pd}} - \mathbf{v}_{\boldsymbol{\eta}_{ad}} \quad (6.5)$$

The second term of the right-hand-side ($\mathbf{v}_{\boldsymbol{\eta}_{pd}}$) is the output of proportional-derivative compensation acting on the reference model tracking error,

$$\mathbf{v}_{\boldsymbol{\eta}_{pd}} = \left(\mathbf{K}_{\boldsymbol{\eta}_p} \quad \mathbf{K}_{\boldsymbol{\eta}_d} \right) \mathbf{e}_{\boldsymbol{\eta}_{rm}} \quad (6.6)$$

where $\mathbf{K}_{\boldsymbol{\eta}_p} > 0, \in \mathcal{R}^{3 \times 3}$ and $\mathbf{K}_{\boldsymbol{\eta}_d} > 0, \in \mathcal{R}^{3 \times 3}$ are diagonal matrices containing desired second-order linear reference model error dynamics, chosen such that

$$A_{\boldsymbol{\eta}} = \begin{pmatrix} 0 & I \\ -\mathbf{K}_{\boldsymbol{\eta}_p} & -\mathbf{K}_{\boldsymbol{\eta}_d} \end{pmatrix} \quad (6.7)$$

is Hurwitz. The reference model tracking error is determined by

$$\mathbf{e}_{\boldsymbol{\eta}_{rm}} = \begin{pmatrix} \boldsymbol{\eta}_{rm} - \boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}}_{rm} - \dot{\boldsymbol{\eta}} \end{pmatrix} \quad (6.8)$$

6.3.3 The NN Adaptive Element

Single Hidden Layer (SHL) NNs are universal approximators in that they can approximate any smooth nonlinear function to within arbitrary accuracy, given a sufficient number of hidden layer neurons and input information. In this section, a SHL NN is trained online to cancel model error/disturbances with feedback, as described in [82-84]. Figure 6.2 shows the structure of a SHL NN.

Figure 6.2 Single Hidden Layer (SHL) Neural Network (NN)

Now the adaptive term are designed to cancel error term Δ in (6.4) as

$$v_{ad} = \hat{\mathbf{W}}^T \boldsymbol{\sigma}(\hat{\mathbf{V}}^T \boldsymbol{\gamma}) \quad (6.9)$$

where $\hat{\mathbf{W}}$, $\hat{\mathbf{V}}$ are input and output weights matrix estimates, and $\boldsymbol{\sigma}(\cdot)$ denotes the vector containing the hidden layer neurons. The input vector ($\boldsymbol{\gamma}$) to the NN is defined as:

$$\boldsymbol{\gamma}(t) = (1, y_c(t), y(t), y(t-d), \dots, y(t-(n_1-1)d), u(t), u(t-d), \dots, u(t-(n_1-r-1)d))^T \quad (6.10)$$

$d > 0$ is a positive time delay, $n_1 \geq n$.

Each hidden layer neuron $\sigma(\cdot)$ is a sigmoid activation function which

represents the characteristics of the neuron:

$$\sigma(x) = \frac{1}{1 + e^{-ax}} \quad (6.11)$$

The factor a is the activation potential and a random value between 0 and 1 is used in this research.

The update law for the input and output weight matrices, i.e., V and W is based on the following equation from [85],

$$\begin{aligned} \dot{\hat{\mathbf{V}}} &= -\Gamma_v \left[\gamma \left(z \hat{\mathbf{W}}^T \hat{\boldsymbol{\sigma}}' \right) + \lambda_v |z| \hat{\mathbf{V}} \right] \\ \dot{\hat{\mathbf{W}}} &= -\Gamma_w \left[\left(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}' \hat{\mathbf{V}} \gamma \right) z + \lambda_w |z| \hat{\mathbf{W}} \right] \end{aligned} \quad (6.12)$$

where Γ_v and Γ_w are the learning rates, λ_v and λ_w are the σ - modification gain, z is a linear combination of the tracking error e and the filtered state ζ . $\hat{\boldsymbol{\sigma}}'$ is a matrix containing derivatives of the sigmoid vector as,

$$\hat{\boldsymbol{\sigma}}' = \begin{pmatrix} 0 & \dots & 0 \\ \frac{\partial \sigma(z_1)}{z_1} & a_{22} & 0 \\ & \ddots & \\ 0 & & \frac{\partial \sigma(z_{n2})}{z_{n2}} \end{pmatrix}$$

In the quadcopter application of the NN, since there are noises and disturbances in the sensor information and errors in the data fusion algorithm, therefore, one should shut the adaptation mechanism off for small tracking errors. Specifically, we replace an adaptation law Eq. (6.12) by:

$$\begin{aligned}
\dot{\hat{\mathbf{V}}} &= \begin{cases} -\Gamma_v \left[\gamma \left(z \hat{\mathbf{W}}^T \hat{\boldsymbol{\sigma}}' \right) + \lambda_v |z| \hat{\mathbf{V}} \right] & |e| \geq \Delta \\ 0 & |e| < \Delta \end{cases} \\
\dot{\hat{\mathbf{W}}} &= \begin{cases} -\Gamma_w \left[\left(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}' \hat{\mathbf{V}} \gamma \right) z + \lambda_w |z| \hat{\mathbf{W}} \right] & |e| \geq \Delta \\ 0 & |e| < \Delta \end{cases}
\end{aligned} \tag{6.13}$$

where Δ is the size of the dead-zone.

6.4 Formation Flight Loop Description: Combining Reachability Analysis and PCH in MPC

6.4.1 Introduction to the Reachability Idea in Flight Adaptive Control

In the last section, adaptive control algorithm based on NN has been employed in the quadcopter attitude control system. This kind of adaptive method is able to deal with external disturbances, and dynamics changes in the system due to possible changes in flight conditions.

For UAV system, which parts of the state space are safe to operate during the flight are still a question needs to be addressed, even when the dynamics of the UAV are completely understood or assumed known. This question is of fundamental importance in the safety verification of control systems and system validation.

Statistics show that the majority of accidents in aviation nowadays are due to Loss-of-Control (LOC). LOC is equally a real problem for the UAV system since nowadays UAV plays an important role in both military and civilian life, and the sensors and equipments installed in UAV are becoming more and more expensive.

In this section, the safe maneuvering/operating envelope for quadcopter dynamics is addressed through reachable sets. The safe maneuvering set is defined as

the intersection between the forward and backwards reachable sets for a given set of a-priori known safe states. This method has been applied to solve fluid flow problems [86]. Furthermore, the PCH technique is also implemented in the formation flight MPC framework.

6.4.2 Maneuver Envelope and Reachability

The conventional definition of the flight envelope is “[the flight envelope] describes the area of altitude and airspeed where an airplane is constrained to operate.” [87]. Various limitations on the aircraft performance are combined to defined the boundaries of the flight envelope, for example engine power, structure considerations and acceleration/speed limits. A common way to present the flight envelope for the manned aircraft is the doghouse diagram which relates the altitude, velocity and possibly other variables at which the aircraft can safely fly.

Flight Envelope Protection

In order to prevent LOC-related accidents, envelope protection is one of the most promising techniques. This technique tries to prevent the aircraft from exceeding the safe maneuvering envelope. A prerequisite for flight-envelop protection systems in a manned aircraft is a fly-by-wire (FBW) system. In the FBW, the pilot’s input are sent to a computer which calculates the desired commands, i.e., there’s no feed-through link between the pilot and the control actions. Similarly for UAV system, FBW means that in the control system design, there should be a command reference system between the pilot and the control action. This is normally the case for UAV when model reference adaptive control (MRAC) is applied. Before applying the flight-envelope protection, one must get the accurate knowledge of the maneuvering

envelope, both to safeguard against excursions and LOC events, and not place too conservative constraints on the performance of the aircraft. In addition, when the flight envelope is known, the maneuvering space can be presented to the ground control station (GCS).

Safe Maneuvering/operating Envelope

The major problem with the conventional definition of flight envelope is that only constraints on quasi-stationary aircraft states are taken into account, for example during coordinated turns and cruise flight. Additionally, constraints posed on the aircraft states by environment are not part of the conventional definition. The UAV dynamical behavior especially for some agile/acrobatic UAVs like helicopter, quadcopter can pose additional constraints on the flight envelope, for example when a quadcopter flies fast forward, it cannot immediately fly backwards. Therefore, an extended definition of the flight envelope is required for UAV system which will be named Safe UAV Maneuver Envelope (SUME).

Definition 2 Safe UAV Maneuver Envelope (SUME)

The Safe UAV Maneuver Envelope is the part of the state space for which safe operation of the UAV can be guaranteed and external constraints will not be violated.

The Safe UAV Maneuver Envelope is defined by the intersection of four envelopes:

- Dynamic Envelope: Constraints posed on the envelope by the dynamic behavior of the UAV, due to its aerodynamics and kinematics.

- Formation Envelope: Constraints due to the inter-agents connections. When a UAV is an agent in a formation flight group, depending on its neighborhood UAVs' state and the formation topology, there will be additional constraints like inter-agents collision avoidance, formation keeping and connection maintaining.
- Structural Envelope: Constraints posed by the airframe material, structure and so on. These constraints are defined through maximum loads that the airframe can take.
- Environmental Envelope: Constraints due to the environmental in which the UAV operates, for example the wind conditions, constraints on terrain and no-fly zones.

These envelopes can be put into the same MPC formation flight framework in which the constraints will be time-varying during the online optimization process. We are mainly focus on the first two types, i.e., the dynamic envelope and formation envelope because these two envelopes are dominating the dynamics during extreme formation maneuver. Constraints posed on the UAV by dynamic flight and formation envelopes are for example a maximum bank angle when it flies forward in order to prevent the UAV from entering a potentially hazardous phenomenon. These kinds of constraints are not fixed, they are dependent on the UAV's flight states and the formation states. Thus, in the formation flight MPC formulation we need to derive how to measure these envelopes during flight, and calculate the constraints accordingly which result in a adaptive MPC formation flight scheme.

In the following section, reachable sets will be introduced and followed by the reachability formulations in MPC framework.

6.4.3 Reachable Sets

Reachable set analysis is an extremely useful tool in safety verification of systems. The reachable set describes the set that can be reached from a given initial set within a certain amount of time, or the set of states that can reach a given target set given a certain time. The dynamics of the system can be evolved backwards and

Figure 6.3 Backwards and forwards reachable set

forwards in time resulting in the backwards and forwards reachable sets respectively. The difference between these two sets is illustrated in Figure 6.3. For a forwards reachable set, the initial conditions are specified and the set of all states that can be reached along trajectories that start in the initial set are determined. For the backwards reachable sets, a set of target states is defined, and a set of states from which trajectories start that can reach that target set are determined.

Assume that the dynamics of the system are given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \quad (6.14)$$

where $\mathbf{x} \in \mathcal{R}^n$ is the state of the system, $\mathbf{u} \in \mathcal{R}^m$ is the control input, and $\mathbf{d} \in \mathcal{R}^q$ is

the disturbance input. Then the formal definitions of the backwards and forwards reachable sets are given by the following definitions.

Definition 3 (Backwards Reachable Set). The backwards reachable set $\mathcal{B}(\tau)$ at time τ ($0 \leq \tau \leq t_f$), of the system (6.14) starting from the target set $\mathcal{T}_0$, is the set of all states $\mathbf{x}(\tau)$, such that there exists a control input $\mathbf{u}(t)$ ($\tau \leq t \leq t_f$), for all disturbance inputs $\mathbf{d}(t)$ ($\tau \leq t \leq t_f$), for which some $\mathbf{x}(t_f) \in \mathcal{T}_0$ are reachable from $\mathbf{x}(\tau)$ along a trajectory satisfying (6.14).

Definition 4 (Forwards Reachable Set). The forwards reachable set $\mathcal{F}(\tau)$ at time τ ($0 \leq \tau \leq t_f$) of the system (6.14) starting from the initial set $\mathcal{T}_0$, is the set of all states $\mathbf{x}(\tau)$, such that there exists a control input $\mathbf{u}(t)$ ($\tau \leq t \leq t_f$), for all disturbance inputs

Figure 6.4 The safe envelope for a known safe set is defined by the intersection of the forwards and backwards reachable sets

$\mathbf{d}(t)$ ($\tau \leq t \leq t_f$), for which $\mathbf{x}(\tau)$ is reachable from some $\mathbf{x}(0) \in \mathcal{T}_0$ along a trajectory

satisfying (6.14).

6.4.4 Safe Maneuver Envelope Through Reachable Set Analysis and Computation in MPC Framework

Now, the safe envelope for a given system with a preliminary set of safe states can be found by the intersection of the forwards and backwards reachable set of this safe set. States that are part of both these sets can be reached from the safe set, and can reach the safe set within a certain time. Therefore, if the initial/target set is known to be safe, then all states that are part of both the forwards and backwards reachable sets can be considered safe as well. This is illustrated by Figure 6.4. For example, a UAV can enter a spin starting from a certain initial condition, and the spin trajectory would then be included in the forwards reachable set. If recovery to a safe flight condition from the spin is possible, the spin trajectory, or part of it, is also included in the backwards reachable set.

Reachability computation in MPC framework

Using Multi-Parametric Toolbox (MPT) [88], N -steps reachable sets can be computed for linear and hybrid systems assuming the system input either belongs to some bounded set of inputs, or when the input is driven by some given explicit control law.

Here we illustrate the reachability computation on the double integrator for example and later on we will implement this computation within the online formation flight computation for the follower quadcopter.

The states/outputs of the double integrator system are: position and speed, with the system inputs: acceleration, the input constraint $[-1,1]$, and the output constraint

$[-5,5]$. In this case, the system matrices become:

$$A_{double} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, B_{double} = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The MPC controller is used here to drive the states to the origin, the initial state set is

Figure 6.5 Reachable sets for double integrator

$\mathcal{T}_0 = \{x_1, x_2 \mid 2 \leq x_1 \leq 4, -1 \leq x_2 \leq 1\}$, the 5 step reachability set is calculated using the command “mpt_reachSets” in MPT toolbox. The reachable sets (green) as well as the set of initial conditions (red) are depicted on top of the controller regions (yellow) in Figure 6.5.

6.4.5 Reachability and PCH Algorithm Implementation in the MPC Formation Flight Loop

Following the reachability analysis in the last section, in this section a practical

algorithm is introduced to make use of the safe operating set based on which the time-varying states constraints will be calculated online in MPC. In addition, the PCH algorithm is also implemented in the MPC formation flight optimization framework.

States constraints online calculation based on safe operating set

The UAV system consists of many dynamic resources like, pitch speed, roll speed, forwards acceleration/speed, backwards acceleration/speed, etc. Normally extreme usage of one resource will limit the use of other resources, for example, when a quadcopter flies forwards in full speed (accompanied by large pitch angle), then it is very dangerous for it to perform large roll.

Figure 6.6 online states constraints calculation based on reachable sets

In order to prevent such LOC in the UAV system, we propose an algorithm to calculate the states constraints online based on the safe operating set. The algorithm flow (pseudo code) is shown in Figure 6.6. There are totally five steps in this algorithm, in which steps 1, 2 can be implemented with the help of MPT toolbox, so only steps 3 and 4 will be detailed.

Step 3 Calculate the minimum volume ellipsoid covering the safe operating set

Here we consider the problem of finding the minimum volume ellipsoid E_s that contains the safe operating set $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \mathcal{R}^n$. An ellipsoid covers $\mathcal{S}$ if and only if it covers its convex hull, so finding the minimum volume ellipsoid E_s that covers $\mathcal{S}$ is the same as finding the minimum volume ellipsoid containing the polyhedron.

To characterize E_s , it will be convenient to parameterize a general ellipsoid as

$$E = \{\mathbf{v} \mid \|A\mathbf{v} + \mathbf{b}\| \leq 1\}$$

i.e., the inverse image of the Euclidean unit ball under an affine mapping.

We can assume without loss of generality that A is positive definite, in which case the volume of E is proportional to $\det A^{-1}$ (determinant of A^{-1}). So the problem of computing the minimum volume ellipsoid E_s containing $\mathcal{S}$ can be expressed as

$$\begin{aligned} \min \quad & \log \det A^{-1} \\ \text{subject to} \quad & \|A\mathbf{x}_i + \mathbf{b}\|_2 \leq 1, \quad i = 1, \dots, m \end{aligned} \tag{6.15}$$

where the variables are $A \succ 0$, and $\mathbf{b} \in \mathcal{R}^n$. The objective and constraint functions are both convex in A and b , so the problem (6.15) is convex.

Step 4 Calculate the state constraints for the short axes of the minimum volume ellipsoid E_s

From step 4, the minimum volume ellipsoid E_s that contains the safe operating set can be calculated using convex optimization, so A and b can be obtained.

Using singular value decomposition (SVD) for matrix A , we can find the eigen axes and the associated eigenvalues of the minimum volume ellipsoid E_s . To calculate the states constraints to prevent LOC of UAV, the following formula is used

$$\begin{aligned}\bar{x}' &= \frac{\lambda}{\lambda_{\max}} \bar{x} \\ \underline{x}' &= \frac{\lambda}{\lambda_{\max}} \underline{x}\end{aligned}\tag{6.17}$$

where $\bar{x}$ and $\underline{x}$ are the nominal upper and lower constraints respectively, λ is the eigenvalues corresponds to which state direction we would like to modify the constraints, $\lambda_{\max}$ is the maximum eigenvalues of the ellipsoid E_s .

Combining PCH into MPC formation flight framework

To get the PCH signal (v_h), an estimated actuator position ($\hat{\tau}_\eta$) is determined based on a motor model/servo model or a measurement. In cases where the actuator position is measured, it can be regarded known ($\hat{\tau}_\eta = \tau_\eta$). This estimate is used to get the difference between commanded pseudo-control and the achieved pseudo-control

$$\hat{v}_h = \hat{\mathbf{f}}_{\eta}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\tau}_{\eta_{cmd}}) - \hat{\mathbf{f}}_{\eta}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \hat{\boldsymbol{\tau}}_{\eta}) \quad (6.17)$$

With the addition of PCH, in the formation flight MPC optimization process, the output of the MPC ($\boldsymbol{\eta}$) will be reduced by

$$\boldsymbol{\eta}_h = \hat{\mathbf{f}}_{\eta}^{-1}(\dot{\boldsymbol{\eta}}, \hat{\boldsymbol{\tau}}_{\eta}, \hat{v}_h) \quad (6.18)$$

6.5 Simulation Results

This section presents some simulation results of the formation flight control method proposed in this chapter. The vehicle model used in the simulations is a small-size quadcopter whose dynamics are governed by Eq. (5.1) and Eq. (5.2). The modeling of the quadcopter is done in the Matlab/Simulink environment.

6.5.1 MPC Formation Flight Loop Setup

The formation flight control layer is implemented as a “Level-2 M-file S-Function” block in Simulink (see in Figure 6.7), the dynamics equation used in the optimization is Eq. (5.1). At each update time instant, the MPC formation flight controller takes the position of the leader, and the states of the follower as inputs, and generates the optimized commanded Euler angles as the outputs. The safe maneuver envelope, the minimum volume ellipsoid covering the safe operating set are calculated inside of each MPC optimization process. The detailed implementation is via the free software CVX [61] and YALMIP [62] toolbox.

Figure 6.7 online states constraints calculation based on reachable sets

6.5.2 NN Adaptive Based Attitude Control Loop Setup

The attitude loop take the optimized commanded Euler angles from the MPC formation flight loop as inputs and generates the thrust and torques to stabilize the quadcopter. The PD controller parameter $\mathbf{K}_{\eta_p}$ and $\mathbf{K}_{\eta_d}$ are set to be

$$\mathbf{K}_{\eta_p} = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 196 \end{pmatrix}, \mathbf{K}_{\eta_d} = \begin{pmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 30 \end{pmatrix}$$

The parameters for the NN are set as below,

Table 6.1 NN parameters

No of neurons	Learning rate(Γ_v, Γ_w)	σ - modification gain	Adaptation dead-zone
---------------	--	---------------------------------	-------------------------

3	$\Gamma_V = 10, \Gamma_W = 1$	$\lambda_V = 0.2, \lambda_W = 0.2,$	$\Delta = 2^\circ$
---	-------------------------------	-------------------------------------	--------------------

6.5.3 Simulation Result

In order to show the effectiveness of the proposed MPC + NN adaptive formation flight method, one leader and one follower quadcopter formation flight simulation is conducted in Matlab/Simulink environment. The leader quadcopter is controlled by the joystick and the follower quadcopter is implemented with the MPC + NN adaptive formation flight controller trying to maintain the formation distance with the leader quadcopter.

The formation shape is $[3, 3, 0]$, i.e. the follower quadcopter should keep 3 meter-distance with the leader quadcopter in both X, Y direction and fly at the same height with the leader.

Initially, the leader quadcopter is at position $(3, 3, 0)$. During the first two seconds, the leader ascends to 1.4 meter height (there is no XY direction movement due to safety considerations). The leader flies in the X direction for 2s, and when it reached 3s it begins to fly in the Y direction. In less than 3 s, the leader quadcopter flies 10 m in the X direction and 3.5 m in the Y direction with the speed larger than 3m/s. This kind of maneuver can be considered as dangerous since it requires the quadcopter to pitch at a large angle and suddenly perform a large-angle roll. To demonstrate the effectiveness of the reachability algorithm augmentation in MPC, two simulations with the some leader trajectory are conducted, one is with the reachability algorithm augmentation, and the other is without the reachability

algorithm.

Figure 6.8 shows the 3D position history of the leader and follower quadcopter. As we can see with the reachability algorithm, the follower quadcopter can successfully track and maintain the formation shape. On the contrary, without the reachability algorithm in the MPC formation flight controller, the follower quadcopter crashes at about 4s which can be seen in Z position plot. Figure 6.9 shows the time history of Euler angles in both simulations. During the first 3 seconds, the Euler angles are the same from both simulations, especially the pitch angles (θ) are very close to each other, which tells us that when the quadcopter flies only in one direction, i.e. only pitch, the MPC reachability algorithm yields identical result as the standard MPC formation flight algorithm. But when the follower quadcopter tries to catch the leader also in Y direction (but requiring large roll motion), the MPC reachability algorithm gives us a safe choice as it does not allow the pitch and roll angles to become simultaneously large at the same time. Thrust and moments applied to the follower quadcopter can be seen in Figure 6.10. For the MPC with reachability algorithm, the thrust and the moment in y (M_y) axis are both saturated during the formation flight, i.e. the thrust is saturated between 5.1s and 5.7s, M_y is saturated from 5.2s to 5.4s, but due to the PCH implemented in the MPC, the NN adaptation still works well as can be seen in Figure 6.11. In Figure 6.11, the pitch channel NN weights are plotted, and it can be seen that all the weights are bounded.

Figure 6.8 XYZ position history

Figure 6.9 Euler angle history

Figure 6.10 Thrust and moments history

Figure 6.11 Pitch channel NN weights history

6.6 Discussion and Concluding Remarks

As stated early in this chapter, the reachability algorithm is computationally heavy and the computational time in the simulation is about 3 times compared to the algorithm without reachability analysis. Currently due to the onboard computational limitations, we are not able to test it on the real quadcopter but this will be implemented in the near future.

In this chapter, a novel formation flight control approach, i.e. MPC combined with adaptive NN for UAV formation flight control is proposed. The reachability algorithm augmentation in the MPC formation planner makes the follower UAV safer in some extreme formation maneuver and the PCH algorithm is also implemented in the MPC framework to prevent NN adaptive element in the inner loop adapting to the inputs saturations. Simulation results demonstrate effectiveness of proposed formation flight control method.

Chapter 7 Conclusions and Recommendations

7.1 Conclusion

This thesis has presented novel contributions in the application of MPC framework to solve leader-follower UAV formation flight control problem.

Firstly, a generalized MPC formulation is constructed for the UAV formation pair flight. The ability to handle realistic constraints in the formulation is especially highlighted.

Secondly, the application of multiplexed MPC scheme for formation flight control is introduced. Under the multiplexed decentralized scheme, the leader UAV can respond faster to the GCS as compared to the nominal decentralized scheme, i.e. when the whole group update their inputs simultaneously. Moreover, the closed-loop stability of the whole formation flight system is always guaranteed even if a different updating sequence is used, which makes the scheme flexible and able to exploit the capability of each UAV. Integration of the obstacle-avoidance scheme into the MPC problem formulation has also been successfully demonstrated. By transformation into a convex position constraint in the MPC problem formulation, the obstacle-avoidance scheme has been extended to include the capability to avoid small pop-up obstacles.

Thirdly, a hierarchical control approach combining a top-layer model predictive controller (MPC) with a bottom-layer robust feedback linearization controller has been proposed for implementation to improve on the response speed and reduce

computational requirements. Real-time formation flight in Vicon motion-capture environment is further achieved by implementing two modifications on the MPC framework, i.e. the control input hold and variable prediction. These modifications are necessary in order to achieve the MPC formation flight in real time. Moreover, the computational power saved can be used for catering other higher level algorithms. The formation flight experiments demonstrate the robustness and effectiveness of the proposed control architecture.

Last but not least, this thesis introduces the use of reachability analysis to augment the MPC formation flight framework. This augmentation, combined with adaptive neural networks with PCH algorithm, improves the safety of the formation flight especially during some extreme flight maneuvers or conditions, as validated through simulations. Since the safe maneuver envelope needs to be computed during each MPC time step, real-time implementation of this approach needs much more computational power than what the current flight control hardware can provide. However, with the fast advancement in the computing hardware technology, the potential usefulness of the approach might be harnessed fully in the near future.

7.2 Recommended Further Research

Experimental verification of the theoretical results

The flight experiments presented in Chapter 5 validated the implementation of the MPC formation flight framework to some extent. However, in the future, the more advanced methods proposed in Chapter 4 and Chapter 6 should also be experimentally validated. Unfortunately, UAV testbed is generally expensive; and

furthermore there is no commercially available testbed for UAV formation flight. The MMPC formation flight method will require more than three heterogeneous UAVs for demonstration and the method proposed in Chapter 6 requires the augmentation of the UAV autopilot system with NN adaptive elements and a high level computer on a module for the reachability calculations. Currently, customized platform and hardware to address such need are being developed according to the roadmap shown in Figure 7.1.

Figure 7.1 Customized flight testbed

Extension of the adaptive formation flight control

The idea proposed in Chapter 6 can be further pursued. In Chapter 6, the reachability algorithm in MPC only uses some states information to calculate the safe maneuver set. The information on the environment/dynamics uncertainties are not included. For adaptive NN, the adaptive output v_{ad} is a measure of the disturbances and model uncertainty level, and thus this signal is an important indication on how well the follower quadcopter can maintain the formation shape. To include the adaptive output signal v_{ad} into the MPC reachability algorithm, several challenging steps need to be overcome. First, the adaptive output signal v_{ad} should be combined with other state information to calculate the safe maneuver envelope. Second, the stability of the whole system (MPC and adaptive NN) needs to be examined and addressed. Last, simulations and formation flight test should be conducted to validate the approach.

APPENDIX A

Definitions for Polyhedron and Polytope

Definition 5 (polyhedron): A convex set $\mathcal{Q} \in \mathbb{R}^n$ given as an intersection of a finite number of closed half-spaces

$$\mathcal{Q} = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{Q}^x \mathbf{x} \leq \mathbf{Q}^c \right\}, \mathbf{Q}^x \in \mathbb{R}^{n \times n}, \mathbf{Q}^c \in \mathbb{R}^n$$

is called polyhedron.

Definition 6 (polytope): A bounded polyhedron $\mathcal{P} \in \mathbb{R}^n$

$$\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{P}^x \mathbf{x} \leq \mathbf{P}^c \right\}, \mathbf{P}^x \in \mathbb{R}^{n \times n}, \mathbf{P}^c \in \mathbb{R}^n$$

is called polytope.

One of the fundamental properties of polytope is that it can also be described by its vertices

$$\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \sum_{i=1}^{vp} \alpha_i \mathbf{V}_p^{(i)}, 0 \leq \alpha_i \leq 1, \sum_{i=1}^{vp} \alpha_i = 1 \right\},$$

where $\mathbf{V}_p^{(i)}$ denotes the i -th vertex of $\mathcal{P}$, and vp is the total number of vertices of $\mathcal{P}$.

Therefore, in the constraints or obstacles representations, we can either use the half-space form or vertex form to denote the polytopes.

APPENDIX B

Linear Quadratic Control Theory of Periodic System

Consider the system (4.7) and the quadratic cost function

$$J_k = \sum_{i=0}^{\infty} \left(\left\| \mathbf{x}_{k+i+1} - \mathbf{x}_{k+i+1}^d \right\|_Q^2 + \left\| \mathbf{u}_{k+i}^{\sigma(k)} \right\|_R^2 \right) \quad (\text{B-1})$$

where $\mathbf{x}_{k+i+1}^d$ is the desired trajectory which can include information of formation shape or the target position for the leader UAV. $Q = Q^T \geq 0$, $R = R^T > 0$ are given state cost and input cost matrices.

Using dynamic programming technique [9], the cost function is minimized by the cost-to-go function $V_k(z)$ (which gives the minimum LQ cost-to-go starting from state z at time k):

$$V_k(z) = z^T \bar{P}_k z + 2\bar{q}_k^T z + \bar{r}_k \quad (\text{B-1})$$

where $\bar{P}_i$, $\bar{q}_i$, and $\bar{r}_i$ ($i=1, \dots, m$) is the solutions of the following discrete-time periodic Riccati-like equations:

$$\begin{aligned} P_k &= Q + A^T P_{k+1} A - A^T P_{k+1} B_{\sigma(k)} \left(R + B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} \right)^{-1} B_{\sigma(k)}^T P_{k+1} A \\ q_k &= \left(A + B_{\sigma(k)} K_{\sigma(k)} \right) q_{k+1} - Q \mathbf{x}_k^d \\ r_k &= r_{k+1} + \mathbf{x}_k^d Q \mathbf{x}_k^d + q_{k+1}^T B_{\sigma(k)} g_{\sigma(k)} \end{aligned} \quad (\text{B-2})$$

by setting:

$$\mathbf{u}_k^{\sigma(k)*} = K_{\sigma(k)} \mathbf{x}_k + g_{\sigma(k)} \quad (\text{B-3})$$

with

$$\begin{aligned}
K_{\sigma(k)} &= -\left(R + B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} B_{\sigma(k)}\right)^{-1} B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} A \\
g_{\sigma(k)} &= -\left(R + B_{\sigma(k)}^T \bar{P}_{\sigma(k+1)} B_{\sigma(k)}\right)^{-1} B_{\sigma(k)}^T \bar{q}_{k+1}
\end{aligned} \tag{B-4}$$

Remark

The set of periodic control gains $\{K_j : j=1, \dots, m\}$ is stabilizing, i.e., the monodromy matrix

$$\Psi_1 = \Phi_m \Phi_{m-1} \cdots \Phi_2 \Phi_1 \tag{B-5}$$

has all its eigenvalues inside the unit circle, where

$$\Phi_j = A - B_j K_j$$

It is a standard fact that this is a necessary and sufficient condition for closed-loop stability of a linear periodic system, and that the eigenvalues of the monodromy matrix are invariant under cyclic permutations of the Φ_j matrices.

Reference

1. Emlen, J.T., *Flocking behavior in birds*. The Auk, 1952. **69**(2): p. 160-170.
2. Vine, I., *Risk of visual detection and pursuit by a predator and the selective advantage of flocking behaviour*. Journal of Theoretical Biology, 1971. **30**(2): p. 405-422.
3. Zelinski, S., T.J. Koo, and S. Sastry. *Hybrid system design for formations of autonomous vehicles*. in *IEEE Conference on Decision and Control*. 2003. Field, CA, USA IEEE.
4. Chichka, D. and J.L. Speyer. *Solar-powered, formation-enhanced aerial vehicle systems for sustained endurance*. 1998: IEEE.
5. No, T.S., Y. Kim, M.J. Tahk, and G.E. Jeon, *Cascade-type guidance law design for multiple-UAV formation keeping*. Aerospace Science and Technology, 2010.
6. Ousingsawat, J. and M. Campbell. *Establishing trajectories for multi-vehicle reconnaissance*. 2004.
7. de Freitas, E.P., T. Heimfarth, A.M. Ferreira, C.E. Pereira, F.R. Wagner, and T. Larsson. *Decentralized Task Distribution among Cooperative UAVs in Surveillance Systems Applications*. in *2010 Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*, 3-5 Feb. 2010. 2010. Piscataway, NJ, USA: IEEE.
8. Jang, J.S. and C.J. Tomlin. *Design and Implementation of a Low Cost, Hierarchical and Modular Avionics Architecture for the DragonFly UAVs*. in *AIAA Guidance, Navigation, and Control Conference*. 2002. Monterey.
9. Bertsekas, D., *Dynamic programming and optimal control, vol. II*. 2007: Athena Scientific.
10. Betts, J.T., *Practical methods for optimal control using nonlinear programming*. 2001: Society for Industrial Mathematics.
11. Borrelli, F., D. Subramanian, A.U. Raghunathan, and L.T. Biegler. *MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles*. in *American Control Conference*. 2006. Minneapolis, USA.
12. Langbort, C. and R. Chandra, *A state space approach to control of interconnected systems*. 2003: Springer, IMA Book Series.
13. Ayanian, N. and V. Kumar, *Decentralized Feedback Controllers for Multiagent Teams in Environments With Obstacles*. Robotics, IEEE Transactions on, 2010. **26**(5): p. 878-887.
14. Izadi, H.A., B.W. Gordon, and Y. Zhang, *Decentralized Receding Horizon Control for Cooperative Multiple Vehicles Subject to Communication Delay*. Journal of Guidance, Control, and Dynamics, 2009. **32**(6).
15. Passino, K.M., *Biomimicry for optimization, control, and automation*. 2004: Springer Verlag.
16. Paul, T., T. Krogstad, and J. Gravdahl. *UAV formation flight using 3D potential field*. 2008.
17. Chen, G., F.L. Lewis, and L. Xie, *Finite-time distributed consensus via binary control protocols*. Automatica, 2011. **In Press, Corrected Proof**.
18. Guerrero, J.A., G. Romero, and R. Lozano. *Robust consensus tracking of leader-based multi-agent systems*. in *American Control Conference (ACC)*. 2010. Baltimore, MD

19. Jiang, F. and L. Wang, *Finite-time weighted average consensus with respect to a monotonic function and its application*. Systems & Control Letters, 2011. **In Press, Corrected Proof**.
20. Jie, M., R. Wei, and M. Guangfu, *Distributed Coordinated Tracking With a Dynamic Leader for Multiple Euler-Lagrange Systems*. Automatic Control, IEEE Transactions on, 2011. **56**(6): p. 1415-1421.
21. Chena, J., D. Sun, J. Yang, and H. Chen, *A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme*. The International Journal of Robotics Research, 2009.
22. Saffarian, M. and F. Fahimi. *A novel leader-follower framework for control of helicopter formation*. 2007.
23. Hengster-Movrić, K., S. Bogdan, and I. Draganjac, *Multi-Agent Formation Control Based on Bell-Shaped Potential Functions*. Journal of Intelligent and Robotic Systems, 2010. **58**(2): p. 165-189.
24. Cao, Y. and W. Ren, *Distributed Coordinated Tracking with Reduced Interaction via a Variable Structure Approach*. Automatic Control, IEEE Transactions on, 2011. **PP**(99): p. 1-1.
25. Shiyu, Z., Z. Rui, W. Chen, and D. Quanxin, *Design of Time-constrained Guidance Laws via Virtual Leader Approach*. Chinese Journal of Aeronautics, 2010. **23**(1): p. 103-108.
26. Pachter, M., J. D'Azzo, and A. Proud, *Tight formation flight control*. Journal of Guidance, Control, and Dynamics, 2001. **24**(2): p. 246-254.
27. Pachter, M., J. D'azzo, and J. Dargan, *Automatic formation flight control*. Journal of Guidance, Control, and Dynamics, 1994. **17**(6): p. 1380-1383.
28. Gu, Y., B. Seanor, G. Campa, M.R. Napolitano, L. Rowe, S. Gururajan, and S. Wan, *Design and flight testing evaluation of formation control laws*. Control Systems Technology, IEEE Transactions on, 2006. **14**(6): p. 1105-1112.
29. Singh, S., P. Chandler, C. Schumacher, S. Banda, and M. Pachter. *Adaptive feedback linearizing nonlinear close formation control of UAVs*. in *American Control Conference*. 2000. Chicago, USA.
30. Singh, S., P. Chandler, C. Schumacher, S. Banda, and M. Pachter, *Nonlinear adaptive close formation control of unmanned aerial vehicles*. Dynamics and Control, 2000. **10**(2): p. 179-194.
31. Singh, S.N., R. Zhang, P. Chandler, and S. Banda, *Decentralized nonlinear robust control of UAVs in close formation*. International Journal of Robust and Nonlinear Control, 2003. **13**(11): p. 1057-1078.
32. Schumacher, C. and R. Kumar. *Adaptive control of UAVs in close-coupled formation flight*. in *American Control Conference*. 2000. Chicago, USA.
33. Mayne, D., J. Rawlings, C. Rao, and P. Scokaert, *Constrained model predictive control: Stability and optimality*. Automatica, 2000. **36**(6): p. 789-814.
34. Maciejowski, J.M., *Predictive control: with constraints*. 2002: Pearson education.
35. Low, K.S. and H. Zhuang, *Robust model predictive control and observer for direct drive applications*. Power Electronics, IEEE Transactions on, 2000. **15**(6): p. 1018-1028.
36. Akpan, V.A. and G.D. Hassapis, *Nonlinear model identification and adaptive model predictive control using neural networks*. ISA Transactions. **In Press, Corrected Proof**.

37. Dold, J. and O. Stursberg. *Distributed predictive control of communicating and platooning vehicles*. in *2009 Joint 48th IEEE Conference on Decision and Control (CDC) and 28th Chinese Control Conference (CCC 2009)*, 15-18 Dec. 2009. 2009. Piscataway, NJ, USA: IEEE.
38. Eklund, J.M., J. Sprinkle, and S.S. Sastry, *Switched and Symmetric Pursuit/Evasion Games Using Online Model Predictive Control With Application to Autonomous Aircraft*. *Control Systems Technology*, IEEE Transactions on, 2011. **PP**(99): p. 1-17.
39. Kuwata, Y. and J.P. How, *Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP*. *IEEE Transactions on Control Systems Technology*, 2010. **19**(2): p. 423 -431.
40. Saffarian, M. and F. Fahimi, *A model predictive framework for autonomous 3D formation flight of helicopter groups*. *Control and Intelligent Systems*, 2009. **37**(Copyright 2010, The Institution of Engineering and Technology): p. 220-6.
41. Chung, H., *Autonomous formation flight of helicopters: Model predictive control approach*. 2006, University of California, Berkeley.
42. Dunbar, W.B. and R.M. Murray, *Distributed receding horizon control for multi-vehicle formation stabilization*. *Automatica*, 2006. **42**(4): p. 549-558.
43. Richards, A. and J. How. *Decentralized model predictive control of cooperating UAVs*. in *IEEE Conference on Decision and Control*. 2004. Atlantis, Paradise Island, Bahamas.
44. Venkat, A., J. Rawlings, and S. Wright, *Plant-wide optimal control with decentralized MPC*. *Dynamics and Control of Process Systems* 2004. 2004, Boston, USA. 589-594.
45. Keviczky, T., F. Borrelli, K. Fregene, D. Godbole, and G. Balas, *Decentralized receding horizon control and coordination of autonomous vehicle formations*. *IEEE Transactions on Control Systems Technology*, 2008. **16**(1): p. 19-33.
46. Kuwata, Y., J. How, and C. MIT. *Robust Cooperative Decentralized Trajectory Optimization using Receding Horizon MILP*. in *American Control Conference*. 2007. New York, NY
47. Richards, A. and J. How. *Implementation of robust decentralized model predictive control*. in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2005. San Francisco, USA.
48. Qin, S.J. and T.A. Badgwell, *A survey of industrial model predictive control technology*. *Control Engineering Practice*, 2003. **11**(7): p. 733-764.
49. Xu, B., D.J. Stilwell, and A.J. Kurdila. *A receding horizon controller for motion planning in the presence of moving obstacles*. in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. 2010.
50. Schouwenaars, T., J. How, and E. Feron. *Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees*. in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2004. Providence, USA.
51. Keck-Voon, L., J. MACIEJOWSKI, and W. Bing-Fang. *Multiplexed Model Predictive Control*. in *16th IFAC World Congress*. 2005. Prague.
52. Ziegler, G., *Lectures on polytopes*. 1995: Springer.
53. Diestel, R., *Graph theory, volume 173 of Graduate Texts in Mathematics*. Vol. 91. 2005: Springer, Heidelberg. 92.
54. Cai, N. and Y.S. Zhong, *Formation controllability of high-order linear time-invariant swarm systems*. *Control Theory & Applications, IET*, 2010. **4**(4): p. 646-654.

55. Chisci, L., J. Rossiter, and G. Zappa, *Systems with persistent disturbances: predictive control with restricted constraints*. Automatica, 2001. **37**(7): p. 1019-1028.
56. Weihua, Z. and T. Go. *3-D Formulation of Formation Flight Based on Model Predictive Control with Collision Avoidance Scheme* in *48th AIAA Aerospace Sciences Meeting*. 2010. Orlando, Florida: American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344, USA.
57. Bittanti, S., P. Colaneri, and G. De Nicolao, *The difference periodic Riccati equation for the periodic prediction problem*. IEEE Transactions on Automatic Control, 1988. **33**(8): p. 706-712.
58. Maciejowski, J., *Predictive control: with constraints*. 2002: Pearson education.
59. Fasano, G., D. Accardo, A. Moccia, C. Carbone, U. Ciniglio, F. Corrado, and S. Luongo, *Multi-Sensor-Based Fully Autonomous Non-Cooperative Collision Avoidance System for Unmanned Air Vehicles*. AIAA Journal of Aerospace Computing, Information, and Communication, 2008. **5**(10): p. 338.
60. Cao, Y. and W. Ren, *Multi-vehicle coordination for double-integrator dynamics under fixed undirected/directed interaction in a sampled-data setting*. International Journal of Robust and Nonlinear Control, 2010. **20**(9): p. 987-1000.
61. Grant, M., S. Boyd, and Y. Ye, *CVX: Matlab software for disciplined convex programming, version 1.21*. <http://cvxr.com/cvx>, April 2011.
62. Lofberg, J. *YALMIP: A toolbox for modeling and optimization in MATLAB*. in *IEEE International Symposium on Computer Aided Control Systems Design*. 2004. Taipei.
63. Scokaert, P. and D. Mayne, *Min-max feedback model predictive control for constrained linear systems*. Automatic Control, IEEE Transactions on, 1998. **43**(8): p. 1136-1142.
64. Wang, J., T. Bierling, L. Höcht, F. Holzapfel, S. Klose, and A. Knoll, *Novel Dynamic Inversion Architecture Design for Quadcopter Control*, in *Advances in Aerospace Guidance, Navigation and Control*, F. Holzapfel and S. Theil, Editors. 2011, Springer Berlin Heidelberg. p. 261-272.
65. Zipfel, P. and W. Schiehlen, *Modeling and Simulation of Aerospace Vehicle Dynamics*. Applied Mechanics Reviews. Vol. 54. 2001: AIAA. B101.
66. Castillo, P., R. Lozano, and A.E. Dzul, *Modelling and Control of Mini-flying Machines*. 2005, New York: Springer.
67. Kolmanovsky, I. and E. Gilbert. *Maximal output admissible sets for discrete-time systems with disturbance inputs*. in *American Control Conference*. 1995. Seattle, WA, USA
68. Slotine, J. and W. Li, *Applied nonlinear control*. 1991, New York: Prentice-Hall Englewood Cliffs, NJ.
69. Boyd, S. and L. Vandenberghe, *Convex optimization*. 2004: Cambridge Univ Press.
70. Ascending, T.G. 2010; Available from: <http://www.asctec.de/>.
71. Vicon Motion System. Available from: <http://www.vicon.com/>.
72. BeagleBoard. <http://beagleboard.org/>. Available from: <http://beagleboard.org/>.
73. Gumstix. Available from: http://www.gumstix.com/store/product_info.php?products_id=227.
74. Johnson, E.N., *Limited authority adaptive flight control*. 2000.

75. Wang, H. and J. Sun. *Modified model reference adaptive control with saturated inputs*. 1992: IEEE.
76. Leonessa, A., W.M. Haddad, T. Hayakawa, and Y. Morel, *Adaptive control for nonlinear uncertain systems with actuator amplitude and rate saturation constraints*. International Journal of Adaptive Control and Signal Processing, 2009. **23**(1): p. 73-96.
77. Rovithakis, G.A. *Nonlinear adaptive control in the presence of unmodelled dynamics using neural networks*. 1999: IEEE.
78. Monopoli, R.V. *Adaptive control for systems with hard saturation*. 1975: IEEE.
79. Ohkawa, F. and Y. Yonezawa, *Discrete model reference adaptive control system for a plant with input amplitude constraints*. INT. J. CONTR., 1982. **36**(5): p. 747-754.
80. Karason, S.P. and A.M. Annaswamy, *Adaptive control in the presence of input constraints*. Automatic Control, IEEE Transactions on, 1994. **39**(11): p. 2325-2330.
81. Annaswamy, A. and J.E. Wong, *Adaptive control in the presence of saturation nonlinearity*. International Journal of Adaptive Control and Signal Processing, 1997. **11**(1): p. 3-19.
82. Calise, A., B. Yang, and J. Craig, *Augmenting adaptive approach to control of flexible systems*. Journal of Guidance, Control, and Dynamics, 2004. **27**(3): p. 387-396.
83. Calise, A.J., N. Hovakimyan, and M. Idan, *Adaptive output feedback control of nonlinear systems using neural networks*. Automatica, 2001. **37**(8): p. 1201-1211.
84. Calise, A.J. and R.T. Rysdyk, *Nonlinear adaptive flight control using neural networks*. Control Systems Magazine, IEEE, 1998. **18**(6): p. 14-25.
85. Calise, A.J., Y. Bong-Jun, and J.I. Craig. *Augmentation of an existing linear controller with an adaptive element*. in *American Control Conference, 2002. Proceedings of the 2002*. 2002.
86. Losasso, F., F. Gibou, and R. Fedkiw. *Simulating water and smoke with an octree data structure*. 2004: ACM.
87. Eshelby, M., G. Ruijgork, M. Asselin, and J. Roskam, *Elements of Airplane Performance*. 1996.
88. Kvasnica, M., P. Grieder, M. Baotic, and M. Morari, *Multi-parametric toolbox (MPT)*. Lecture notes in computer science, 2004. **2993**: p. 448-462.