An Integrated Fault-tolerant Model Predictive Control Framework for UAV Systems

by

Binyan Xu
B.Eng., Nanjing University of Aeronautics and Astronautics, 2016
M.A.Sc., Nanjing University of Aeronautics and Astronautics, 2019

A Dissertation Submitted in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

An Integrated Fault-tolerant Model Predictive Control Framework
for UAV Systems

by

Binyan Xu
B.Eng., Nanjing University of Aeronautics and Astronautics, 2016
M.A.Sc., Nanjing University of Aeronautics and Astronautics, 2019

Supervisory Committee

Dr. Yang Shi, Co-supervisor
Department of Mechanical Engineering

Dr. Afzal Suleman, Co-supervisor
Department of Mechanical Engineering

Dr. Pan Agathoklis, Outside Member
Department of Electrical and Computer Engineering

ABSTRACT

The application of unmanned aerial vehicles (UAVs) has considerably expanded over the past few decades, driven by their flexibility, efficiency, cost-effectiveness, and distinct advantages in executing tasks within dangerous and inaccessible environments. As the demand for UAVs grows, so does the expectation for their autonomy and reliability. Therefore, there is a need to enhance the efficiency and safety of UAV control systems.

This dissertation proposes the development of innovative control strategies applicable to both individual and multi-agent systems, aiming to effectively address control challenges in UAV applications, such as complex dynamics, inherent constraints, unexpected faults, and resource limitations. To achieve this objective, a unified framework to effectively integrate model predictive control (MPC) with fault-tolerant control (FTC) is proposed, with the primary focus on identifying and addressing theoretical and practical challenges associated with this integration.

The dissertation starts by providing a comprehensive introduction and systematic literature review, highlighting unresolved issues and gaps in fault-tolerant model predictive control (FTMPC). Essential mathematical preliminaries, including models and necessary theorems, are also discussed.

Next, a novel adaptive fault-tolerant MPC method for fault-tolerant tracking control of constrained nonlinear systems is presented. This design integrates an adaptive fault estimator into the Lyapunov-based MPC framework, thereby ensuring closed-loop control performance and system stability in the presence of actuator faults with reduced computational complexity.

The FTMPC framework is further extended by applying it to the trajectory tracking control problem of UAVs with input constraints and actuator faults. To tackle the unique UAV control challenges, it presents the design and stability analysis of a dual-loop, dual-rate hierarchical UAV control system. By implementing MPC only to the outer-loop at a slower sampling rate, it significantly reduces the computational demands of solving the MPC problem while maintaining the rapid response capabilities of the inner loop. Furthermore, the dual-sampling-rate issue is rigorously evaluated in the closed-loop analysis using singular perturbation theory, providing important guidelines for selecting control parameters based on the sampling frequency.

Furthermore, the fault-tolerant formation control problem of a multi-UAV system interconnected through a directed communication graph is investigated. With the de-

veloped adaptive distributed Lyapunov-based MPC method, the formation tracking control objective is achieved with partially known leader information and unexpected actuator faults. This design also significantly reduces communication and computational burdens by requiring only a single round of calculation and communication per control update.

Finally, unknown communication faults between agents in a nonlinear multi-agent system are addressed, instead of only considering the actuator faults that only affect individual local agents. To this end, a novel adaptive distributed observer-based DMPC method is developed, enhancing the resilience of distributed formation tracking in the presence of communication faults. This strategy is able to simplify the complexity of local MPC design by decomposing the original formation tracking control problem into several fully localized tracking control problems.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

As I bring the journey of my Ph.D. to its culmination, I am filled with deep gratitude to all those who have illuminated my path with their guidance, support, and inspiration.

First and foremost, I would like to express my sincerest thanks to my supervisors, Drs. Yang Shi and Afzal Suleman, whose expertise and insightful guidance have been the guiding lights throughout my journey in the vast ocean of research. My adventure began four years ago when they recognized my potential and welcomed me into their excellent teams. Throughout these years, they have not only been remarkable mentors but also great supporters who encouraged me to pursue my research with rigor and integrity. Beyond their academic guidance, their genuine concern for my personal development and their unwavering belief in my capabilities have continuously motivated me, pushing me to extend my limits and strive for higher goals. I am profoundly thankful for having the opportunity to learn from such distinguished and professional scholars.

I also owe a great deal of thanks to the members of the Applied Control and Information Processing Laboratory (ACIPL) and the Centre for Aerospace Research (CfAR) at the University of Victoria. I am particularly grateful to Drs. Jicheng Chen, Changxin Liu, Songling Zhuang, Kunwu Zhang, Qian Zhang, Henglai Wei, and Tianyu Tan, senior members of ACIPL. Their invaluable suggestions and expert insights have been fundamental in advancing my research endeavors. My gratitude also extends to my peers at ACIPL, Xinxin Shang, Tianxiang Lu, and Yue Song, who have been not only collaborators but also great friends. Their kindness and passion have not only enriched my daily study experience but have also created a supportive and inspiring learning environment. Additionally, special thanks go to Stephen Warwick, Sean Bazzocchi, Grant Howard, and Nic Richardson from CfAR, who provided essential technical support and introduced me to the exhilarating world of aerospace research and industry. Their expertise offered me a glimpse into the practical challenges and breakthroughs in real-world aerospace technology. Working alongside such talented individuals has greatly enriched my experience and influenced my career trajectory.

My heartfelt thanks also extend to my family and friends for their indispensable support. To my parents, thank you for your unconditional love and support. To my boyfriend, Yufan, thank you for your love, patience, understanding, and endless

encouragement. To my roommates, thank you for transforming our shared spaces into a home away from home. I am deeply grateful to all of you for always standing by me.

Finally, I would like to extend my appreciation for every opportunity that has crossed my path during this Ph.D. journey. Each experience, whether challenging or rewarding, shaped me into who I am today. I will forever cherish the eternal sunshine and the shimmering waves of beautiful Victoria, carrying these precious memories with me into future challenges and opportunities.

Acronyms

UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-off and Landing
MPC	Model Predictive Control
FTC	Fault-tolerant Control
FTMPC	Fault-tolerant Model Predictive Control
DMPC	Distributed Model Predictive Control
MAS	Multi-agent System
MIMO	Multi-input Multi-output
SISO	Single-input Single-output
FDI	Fault Detection and Isolation
FIFO	First-in First-out
MHE	Moving Horizon Estimation

Chapter 1

Introduction

This chapter sets the stage for the dissertation by introducing the research background and reviewing the methodologies relevant to the study. It also outlines the motivations driving the research. Finally, the organization of the subsequent chapters and their respective contributions are detailed.

1.1 Research Background and Significance

We begin by providing a general introduction of unmanned aerial vehicles (UAVs), highlighting the prevailing control challenges that this dissertation aims to explore. Following this, we delineate our research objectives and then introduce our proposed solution, fault-tolerant model predictive control.

1.1.1 Overview of UAVs

Unmanned aerial vehicles (UAVs) are aircraft that operate without a human pilot on board. They are either controlled autonomously by onboard computers or remotely manipulated by human operators. Based on wing types, UAVs can be classified into three categories: rotary wing UAVs, fixed wing UAVs, and hybrid UAVs, as shown in Figure 1.1. Rotary wing UAVs are known for their vertical take-off and landing (VTOL) capabilities and their ability to hover, allowing them to perform tasks in diverse environments, both indoors and outdoors. Fixed-wing UAVs, on the other hand, excel in long-endurance flights over extensive areas due to their aerodynamic efficiency. Hybrid UAVs combine the attributes of both rotary and fixed-wing designs, offering enhanced operational versatility. Owing to their flexibility, efficiency,

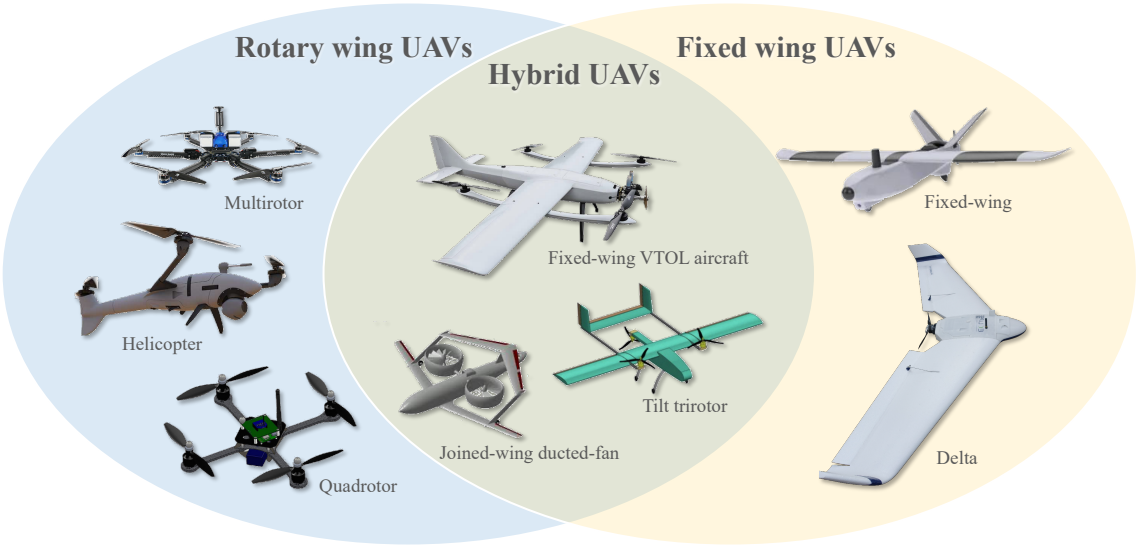

Figure 1.1: UAV classification based on wing types [1]

and cost-effectiveness, UAVs have emerged as valuable tools for tasks that may be monotonous, challenging, or hazardous for humans [1].

The initial development of UAVs was driven by military needs. As relevant technologies have improved and become commercialized, the application of UAVs has expanded significantly, now encompassing both military and civilian fields [2]. In the early years of their use, UAVs were primarily deployed and operated on an individual basis. During the last decade, the importance of interactions between UAVs has been recognized. This recognition, together with rapid advancements in communication networks and coordination algorithms, has facilitated the development of networked unmanned multi-vehicle systems [3]. The emergence of multi-UAV systems leads to improved efficiency and enhanced redundancy, thereby spurring a wider spectrum of UAV-assisted applications [4]. Figure 1.2 illustrates some of the scenarios in our modern life where UAV systems shine, including in smart cities and intelligent vehicles, smart agriculture, forest fire detection, disaster relief, and remote area search and rescue.

The development of sophisticated control systems is crucial for enhancing the efficiency and safety of UAVs. As demand for UAVs grows, so do the standards for their maneuverability and reliability. Such increasing requirements for UAV autonomy pose additional challenges and difficulties in UAV control, outlined as follows:

[1] UAV pictures included in this figure are from the Internet.

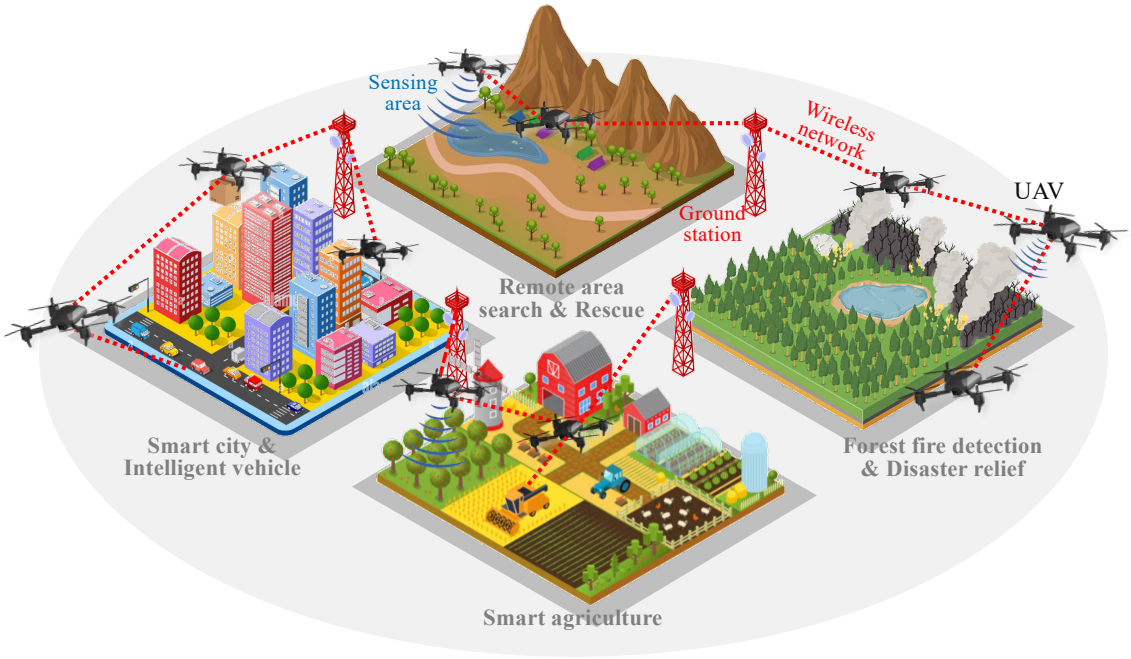

Figure 1.2: Application scenarios of UAV systems[2]

- **Complex dynamics model:** Establishing precise dynamic models for UAVs that remain effective across a wide range of operating conditions is challenging. A common feature shared by various aerial vehicles, including UAVs, is their multiple-input and multiple-output (MIMO), under-actuated, highly nonlinear, and strongly coupled dynamics. This dynamic complexity significantly increases the difficulty of developing UAV control systems. Specifically, traditional control techniques, designed for single-input and single-output (SISO), fully actuated, and linear systems, cannot not be directly applied. Instead, the employment of more sophisticated control strategies and structures is essential.

- **Input constraints:** UAV actuators, such as copters and aero-surfaces, have inherent physical limitations, which impose strict input constraints on UAV control systems. Such input constraints need to be taken into consideration during control development. If not properly addressed, they may lead to degraded or even unstable closed-loop performance.

[2]Vector images used in this figure are downloaded from Vecteezy (https://www.vecteezy.com/free-vector).

- **Inevitable faults and malfunctions:** Given the fact that UAVs incorporate a range of subsystems, sensors, actuators, and components that are susceptible to failures, the occurrence of faults is inevitable despite technological advancements. In addition, UAVs operate in complex environments where unforeseeable conditions and events can occur [5]. However, the control performance and closed-loop stability assurances that are valid under fault-free circumstances may not be sustained in the event of faults.

- **Limited onboard resources:** All UAV manipulation operations are managed by the onboard flight microcontroller or microcomputer, with minimal intervention from the ground station. Given the constraints of onboard devices, the available computation and communication resources are limited. Therefore, for single UAV control that demands a swift response, the computational complexity of the control solution must be minimized to ensure a sufficiently fast control update frequency. For multi-UAV control, it is equally crucial to mitigate the frequency and volume of communications between network controllers to conserve communication bandwidth and alleviate network load.

- **Sampled-data control nature:** UAVs inherently operate with continuous-time dynamics, such as flying through the air under continuous aerodynamic forces. However, due to the digital nature of onboard sensors and microprocessors, control commands are computed and executed at discrete intervals [6]. This disagreement between the discrete control and the continuous system evolution can lead to uncontrolled discretization errors. If not properly managed, these errors may result in performance degradation or even instability [7].

1.1.2 Research Scope and Focus

This dissertation is dedicated to the development of innovative control strategies that effectively address the identified control challenges within UAV systems, applicable to both individual UAVs and multi-UAV systems. The research objectives are as follows:

- To achieve optimized control performance of MIMO nonlinear models that capture the complex dynamics of UAVs, ensuring simultaneously real-time responsiveness and efficient resource utilization.

- To generate control actions that inherently consider input constraints, thereby enhancing the reliability of UAV operations within their operational limitations.
- To implement resilient control strategies that prevent faults from significantly jeopardizing the safety of the system, maintaining control performance not only under normal operating conditions but also in the presence of faults in system components or communication networks.
- To obtain satisfactory control performance while reducing computational complexity and communication burden, thereby accommodating the limited processing capabilities of UAV systems.
- To utilize specific mathematical tools to predict and estimate discretization errors, thereby mitigating the effects of sampled-data implementation in control systems.

To achieve the outlined research objectives, it is natural for us to consider the integration of model predictive control (MPC) and fault-tolerant control (FTC). MPC is an advanced control theory for the online computation of optimal control. Its optimization nature not only allows it to handle constraints systematically but also makes it particularly suitable for managing multi-variable nonlinear systems. FTC, on the other hand, focuses on preserving system performance and stability as close as possible to the desired level in the presence of faults.

In this dissertation, we aim to establish a unified framework that effectively integrates MPC with FTC. This integration is intended to enhance the resilience and effectiveness of UAV control systems, ensuring their reliability, particularly in scenarios involving unexpected faults. The primary focus of this dissertation is to identify and address the key challenges and obstacles associated with this integration. Additionally, practical issues related to implementing such integration in UAV systems are explored.

1.2 Literature Review

This chapter presents a literature review of relevant methodologies. It begins with a tutorial review in Section 1.2.1, which aims to establish a foundational understanding of MPC principles and introduce several commonly utilized MPC formulations. Then, Section 1.2.2 examines various studies that have implemented MPC in UAV

systems. Section 1.2.3 explores the existing results on distributed MPC that facilitate coordinated control among multiple agents or systems. Finally, Section 1.2.4 offers an extensive literature review on fault-tolerant model predictive control (FTMPC), highlighting the unresolved issues and gaps in this area.

1.2.1 Model Predictive Control

This section starts with a general overview of MPC, introducing its fundamental concepts and principles. It then progresses to introduce the standard formulation of MPC, followed by a detailed exploration of Lyapunov-based MPC—another commonly used stabilizing MPC formulation.

A. Overview of MPC

The idea of MPC initially emerged from industrial applications as a method for the online computation of optimal control. This approach was deemed necessary because deriving offline analytical solutions is impractical for most realistic cases, except for the simplest linear and unconstrained scenarios [8]. The formal theoretical development of MPC began in the 1980s, marked by the publication of the first work on generalized predictive control [9]. Since then, the field of MPC has experienced remarkable developments, both from theoretical and practical perspectives. Today, MPC is a prevalent control approach characterized by well-established theoretical foundations and a proven capability for high-performance control of complex systems. It has been successfully applied in a wide range of application areas, including industrial processes, power electronics, robotics, autonomous vehicles, and more [10–12]. This broad applicability underlines its versatility and effectiveness in addressing the demands of modern control.

MPC distinguishes itself from other control methods in its approach to determining control actions. At each sampling instant, MPC solves an online finite horizon optimal control problem using the current state of the plant as the initial state. This contrasts with traditional methods that typically establish an analytical feedback control policy offline for all possible states [13, 14]. In MPC, the solution to the open-loop optimization problem yields an optimal control sequence for a finite number of future control inputs, but only the first input in the sequence is implemented. This process is then repeated at the next sampling instant, using the newly available state measurement. The basic working philosophy of MPC, as described, is illustrated in

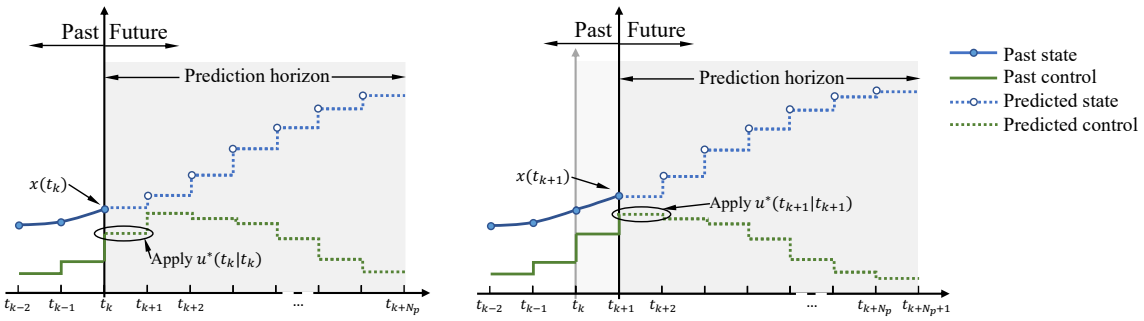

Figure 1.3: Basic working philosophy of MPC

Figure 1.3. Essentially, the three main components that constitute a typical MPC design include:

- **A plant model for prediction**: This model predicts the future evolution of the system based on the current or estimated state and a hypothetical future input trajectory. As a model-based control method, the efficacy of MPC heavily depends on the accuracy of this model. A precise model enables more effective control actions, underscoring the importance of having an adequately accurate prediction model that captures the main characteristics of the physical plant.

- **A finite-horizon optimization problem**: At each sampling instant, MPC solves an online finite-horizon optimization problem to determine the control actions. The optimization problem involves minimizing a performance cost defined over a finite future horizon, with the control inputs over this horizon being the free variables for optimization. The prediction model, along with input and state constraints and other relevant constraints, are seamlessly integrated into this constrained optimization problem. Solving this problem yields a trajectory of optimal future control inputs over the prediction horizon, guiding the system's actions toward the desired performance while satisfying all specified constraints.

- **A receding horizon scheme**: MPC operates in a receding horizon manner, meaning that only the first control input from the optimized control sequence is actually applied to the plant. After this implementation, the process is repeated at the next sampling instant, using the latest available system state information. This introduces a feedback loop into the MPC strategy, allowing it to adapt to

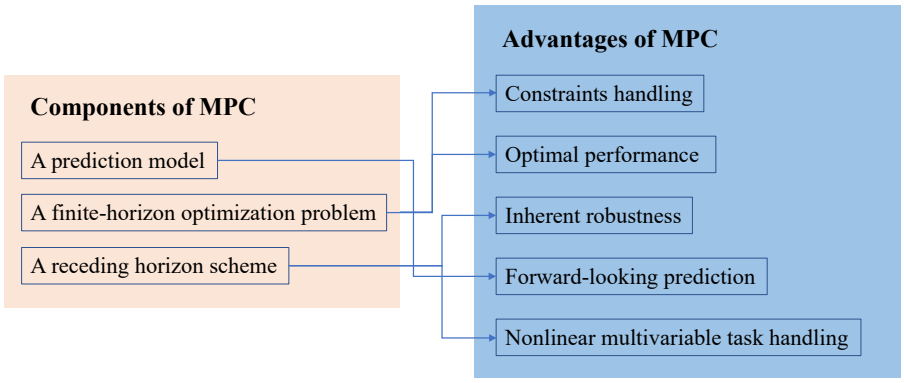

Figure 1.4: Main components and advantages of MPC

changes and disturbances by continually updating the initial conditions with real-time data.

It is illustrated in Figure 1.4 that these unique principles of MPC enable the following distinctive advantages of MPC:

- **Constraints handling:** One of the most distinctive characteristics of MPC is its capability of handling both the system state and control input constraints in a systematic and flexible way. These restrictions can be seamlessly integrated into the MPC framework by being included in the optimization problem.

- **Optimized control performance:** MPC is fundamentally centered around optimization, naturally resulting in optimal performance to some extent. Unlike traditional optimal control approaches that typically operate over an infinite horizon, MPC solves optimization control problems with a finite horizon. This allows MPC to maintain the benefits of optimal control strategies while significantly reducing computational complexity, particularly for those systems with nonlinear dynamics and constraints.

- **Inherent robustness:** The receding horizon implementation of MPC integrates a feedback mechanism. Such feedback allows MPC to adjust dynamically to new data at each sampling instant, significantly enhancing its robustness against uncertainties.

- **Forward-looking prediction:** In addition to determining the control action for the current instant, MPC also generates a forward-looking prediction based

on the current situation. The obtained predictions provide a valuable approximation of the future evolution of the plant system, enabling more informed and effective control decisions.

- **Nonlinear multi-variable task handling:** Typically, designing an analytical control law for nonlinear multi-variable systems requires that the dynamics model possesses a specific structure, such as the input affine form. However, MPC does not require this stringent structural condition due to its unique implementation scheme. This flexibility allows MPC to handle complex, multi-variable nonlinear systems without the need for assumptions about the system's structure.

B. Standard Formulation of MPC

A standard formation of an open-loop MPC optimization control problem at the time instant t_k is

$$\min_{u^p(t|t_k)} \int_{t_k}^{t_{k+N_p}} L(x^p(t|t_k), u^p(t|t_k))\mathrm{d}t + V_f(x^p(t_{k+N_p}|t_k)) \tag{1.1a}$$

subject to

$$\dot{x}^p(t|t_k) = f(x^p(t|t_k), u^p(t|t_k)) \tag{1.1b}$$
$$x^p(t_k|t_k) = x(t_k) \tag{1.1c}$$
$$u^p(t|t_k) \in \Omega_u \tag{1.1d}$$
$$x^p(t_{k+N_p}|t_k) \in \Omega_f \tag{1.1e}$$

where $N_p \in \mathbb{R}^+$ is the prediction horizon; $t \in [t_k, t_k + N_p]$; Ω_u represents the input constraint; Ω_f is the terminal set. To distinguish the internal variables of MPC for prediction from the actual system evolution, we denote them by a superscript p.

In the MPC formulation (1.1), $L(x^p(t|t_k), u^p(t|t_k))$ defines a performance index to be minimized and $V_f(x^p(t_{k+N_p}|t_k))$ defines the terminal penalty cost. The constraint of (1.1b) is the prediction model, which is used for the MPC to predict the future evolution of the system, and the initial state of it is the current state measurement as defined in (1.1c). The constraint of (1.1d) takes into account the constraints on the control inputs. Constraint (1.1e) is the terminal constraint. The optimal solution of the MPC problem can be defined as $u^*(t|t_k)$ for $t \in [t_k, t_{k+N_p})$, which is applied to the

system for $t \in [t_k, t_{k+1})$, and at the next sampling instant t_{k+1}, when the new state measurement $x(t_{k+1})$ is available, the same control evaluation and implementation process is repeated.

The inclusion of the terminal penalty cost $V_f(x^p(t_{k+N_p}|t_k))$ and the terminal set Ω_f in the MPC formulation plays a critical role in ensuring the closed-loop stability and feasibility of the optimization problem [15]. The terminal penalty transforms the inherently finite horizon of MPC into an effective infinite horizon. With this manner, the infinite horizon can be divided into two intervals: one refers to the finite predicted time steps that are under the control of the predicted control inputs, while the subsequent infinite intervals are considered as controlled by a stabilizing feedback law, which is often taken to be the optimal control law in the absences of constraints [15, 16]. The terminal constraint set Ω_f is imposed to ensure that the predicted state at the end of the prediction horizon reaches an invariant set that, once entered, the state will never leave. Hence, the future feasibility of the receding horizon strategy can be ensured. The development of feedback control laws, along with the definition of corresponding terminal costs and constraints, has been a significant focus of MPC theoretical research. This MPC formulation has been extensively studied for both linear and nonlinear cases, with notable references including [16–25].

C. Lyapunov-based MPC

In addition to MPC that incorporates terminal ingredients, Lyapunov-based MPC offers an alternative stabilizing MPC formulation. The main insight of Lyapunov-based MPC is to guarantee stability by incorporating a contractive stability constraint that enforces the decay rate of the Lyapunov function. The typical formulation of a Lyapunov-based MPC problem is expressed as follows:

$$\min_{u^p(t|t_k)} \int_{t_k}^{t_{k+N_p}} L(x^p(t|t_k), u^p(t|t_k))\mathrm{d}t \tag{1.2a}$$

subject to

$$\dot{x}^p(t|t_k) = f(x^p(t|t_k), u^p(t|t_k)) \tag{1.2b}$$

$$x^p(t_k|t_k) = x(t_k) \tag{1.2c}$$

$$u^p(t|t_k) \in \Omega_u \tag{1.2d}$$

$$\frac{\partial V(x^p(t_k|t_k))}{\partial x^p} f(x^p(t_k|t_k), u^p(t_k|t_k)) \leqslant \frac{\partial V(x^p(t_k|t_k))}{\partial x^p} f(x^p(t_k|t_k), u^a(x^p(t_k|t_k))) \tag{1.2e}$$

where $t \in [t_k, t_k + N_p]$; $V(\cdot)$ is a candidate Lyapunov function, and $u^a(\cdot)$ is an auxiliary control law. The constraint of (1.2e) guarantees that the time derivative of the Lyapunov function at t_k is less than or equal to the derivative obtained under the auxiliary control policy u^a. This constraint allows the Lyapunov-based MPC to inherit the stability properties of the auxiliary control system. Furthermore, the feasibility of the Lyapunov-based MPC is also ensured, as the auxiliary control law provides a feasible solution for the optimization problem, referencing established results in [26, 27].

The fundamental distinction between Lyapunov-based MPC and traditional MPC methodologies lies in its reliance on an auxiliary controller to ensure the closed-loop stability and feasibility of the MPC problem. Unlike other approaches, Lyapunov-based MPC does not require terminal ingredients to guarantee stability. This characteristic is especially beneficial in the control of nonlinear systems, where selecting appropriate terminal costs and sets is challenging. First of all, Lyapunov-based MPC does not require the assumption that the system's linearization around the equilibrium point is stabilizable—a condition that is not always met [13, 15]. Furthermore, terminal regions approximated for nonlinear systems are often small, requiring longer prediction horizons to secure a sufficiently large region of attraction. However, longer prediction horizons can lead to increased computational loads, potentially restricting the application of traditional MPC in scenarios requiring rapid response, such as UAV control. Lyapunov-based MPC, instead, can stabilize the closed-loop system without terminal constraints, allowing for the freedom of choosing a shorter prediction horizon and thereby reducing the computational workload for control calculation.

In recent years, substantial progress has been made in investigating the theory and application of Lyapunov-based MPC for addressing a range of nonlinear control problems. The initial integration of Lyapunov-based control with MPC into a unified framework was proposed in [28]. Following this, a similar formulation known as contractive MPC was proposed in [29]. In both of the two early results, a Lyapunov function decay constraint is employed as the terminal constraint to guarantee the closed-loop stability. Lyapunov-based MPC is then formally proposed in [26], which requires the value of the Lyapunov function to decrease at the first time step rather than the last step of the prediction horizon. This approach has been applied to specialized applications such as trajectory tracking for underwater vehicles in [30]. To address plant-model mismatches, the Lyapunov-based MPC framework has been combined with robust and adaptive control techniques, enhancing its practical robustness against uncertainties and disturbances. By applying robust auxil-

iary controller design, Lyapunov-based MPC inherits the robustness to uncertainties and disturbances [31]. By integrating moving horizon estimation (MHE) with the Lyapunov-based MPC framework, an offset-free adaptive MPC for nonlinear systems subject to uncertainties is developed [32–34]. Additionally, the framework has been adapted for stochastic uncertainties through the development of stochastic Lyapunov-based feedback control in [35]. For systems with multiple candidate configurations, Lyapunov-based MPC can characterize the stability region of each configuration and thus guide the system state move in a pre-designed fashion. With this idea, Lyapunov-based MPC is successfully applied to switched hybrid systems [26] and fault-tolerant reconfiguration systems [31, 36, 37]. Furthermore, the approach has been applied to networked nonlinear control systems, addressing challenges like data losses and measurement delays [38,39], and extended to distributed settings for large-scale networked systems [40, 41, 41–43] and multi-agent systems [44–46].

1.2.2 Application of Model Predictive Control to UAVs

The design of high-performance control systems for UAVs is inherently challenging due to the under-actuated, strongly coupled, and highly nonlinear nature of their dynamics models. Significant research has been conducted to apply model predictive control to enhance UAV control performance. A common strategy in these studies is the adoption of a dual-loop hierarchical control architecture, which effectively addresses the under-actuated characteristics of UAV dynamics. Specifically, the outer loop stabilizes translational variables, including the position and the linear velocity, and generates a reference signal for the inner rotation control loop. Many studies have concentrated solely on the outer-loop control using MPC to optimize performance within certain constraints [47–51]. Others have extended MPC applications to both loops. For instance, [52] introduces a switched MPC scheme for controlling rotational dynamics. This result is further extended in [53], which innovatively omits the reliance on position measurements. Additionally, [54] designs two linear MPC methods tailored for both translational and rotational dynamics of a quadrotor UAV.

Research efforts also focus on reducing the computational demands of MPC. [55] develops an efficient nonlinear MPC-based controller for UAV trajectory tracking, utilizing a modified continuation/generalized minimum residual (C/GMRES) numerical algorithm and a Lyapunov-based MPC framework to achieve effective control with lower computational complexity. Similarly, [56] introduces a novel lateral guidance

law using a C/GMRES-based nonlinear MPC for UAV path following, applicable to any arbitrarily described differentiable function in the horizontal plane. These advancements highlight the ongoing efforts to refine MPC applications in UAV control, balancing performance optimization with computational efficiency.

1.2.3 Distributed Model Predictive Control

Distributed model predictive control (DMPC) provides a perfect solution for optimal synchronization problems of multi-agent systems with constraints [57]. It is a receding horizon optimal control strategy implemented in a distributed architecture, in which each agent is assigned its own finite-horizon optimization problem based on the current states of itself and of its neighbors, optimizes for its own control at each update instant, and exchanges information within its neighborhood. DMPC inherits the advantages of centralized model predictive control, including systematic handling of hard constraints, optimized control performance with lower computation complexity, inherent robustness against uncertainties, and the ability to cope with nonlinear multi-variable systems. On the other hand, the distributed architecture of DMPC makes full use of the independently actuated nature of multi-UAV systems, distributes the computational load, and reduces the communication requirement.

Numerous DMPC approaches have been proposed, and comprehensive summaries of these approaches can be found in review papers such as [58–60]. A key issue that arises with distributed architecture is that it is challenging to establish global stability and local feasibility with distributed design based on locally available information only. For the convenience of theoretical completeness and analytical convenience, many earlier investigations on DMPC specifically focus on systems with unconstrained and/or linearized dynamics [61–65]. To guarantee the feasibility and stability of a multi-agent control system with nonlinear dynamics and coupling constraints, one solution is to "robustify" the coupling constraints in a way that the constraint at each node is formulated to be satisfied for all possible behaviors of its neighbors, as proposed in [66] and [67]. By employing this strategy, couplings between subsystems can be treated as bounded disturbances, allowing for the application of various robust MPC methods. Such a robustification strategy, however, introduces conservatism. One way to mitigate such conservatism is to enable information exchange between optimizers, aiming to achieve consistency between the actual behavior of an agent and the predictions made by its neighboring agents. This concept has inspired the devel-

opment of iterative DMPC methods [68–70], in which all agents iteratively negotiate their local optimal decisions for each control update. Given the satisfaction of certain criteria, these iterations converge to a solution for the global optimization problem. The practical drawback of iterative DMPC is apparent in real-world applications, as it often results in slow update rates due to the high computational and communication requirements associated with multiple iterations. Another DMPC design with information exchange during the optimization process is sequential DMPC [71, 72], in which all subproblems are solved only once per step following a sequential order. While the sequential solution process reduces the computation and communication burden to a certain extent, it still incurs a delay of several steps before an agent can update its control action since it must wait for the solutions from all its preceding neighbors. The delay, whose magnitude depends on the size and complexity of the multi-agent system, can affect system performance, especially in scenarios that require fast responses or tight coordination among agents. Parallel DMPC, where all agents compute, execute, and exchange their control solutions in parallel, is more suitable for flight control scenarios demanding a fast control updating rate. As proposed in [73–75], parallel MPC methods typically involve the construction of assumed trajectories based on previously calculated optimal control inputs and the incorporation of consistency constraints to bound the mismatch between the actual trajectory and the assumed trajectory. It is also important to note that most of the currently existing results can only address the cooperative regulation problem that drives all agents toward a prior-known set point.

1.2.4 Fault-tolerant Model Predictive Control

This subsection provides a literature review of fault-tolerant model predictive control, starting with a general introduction to fault-tolerant control.

A. Overview of FTC

Control systems are vulnerable to various faults. Typical types of faults include actuator and sensor faults. Additionally, multi-agent control systems, which consist of multiple interacting agents, are susceptible to communication faults introduced by the communication network. Network issues like changes in topology and bandwidth limitations can disrupt the connectivity among agents, thereby posing significant risks to the safety of the systems.

FTC is a subfield of control engineering focused on ensuring that a system, typically in an engineering or technological context, continues to operate safely and effectively even in the presence of faults or failures within some of its components. FTC has received increasing attention over the last few decades due to the growing demand for reliability and safety in control systems [76]. Many new FTC methodologies and algorithms have been developed with successful applications in the context of industrial and chemical process control and flight control. All the existing FTC approaches can be divided into two main categories:

- **Passive FTC:** The development of passive FTC is fundamentally based on the robust control theory, which accommodates the worst-case scenarios involving unknown faults. Specifically, passive FTC involves the design of a controller that maintains stability and control performance robustness against a predefined spectrum of faults. Discussions on passive FTC can be referred to [77–83] and the references therein for recent development.

- **Active FTC:** Different from passive FTC, active FTC reacts to the occurrence of faults and adjusts the control actions actively. It relies on the fault detection and isolation (FDI) block to provide real-time information about the fault, such as its location, type, and size. With the fault information, active FTC approaches actively recover the system by fault accommodation or control reconfiguration. Fault accommodation indicates that the control actions are changed online by updating certain relevant parameters or modifying the control law to compensate for the effects caused by faults. Control reconfiguration activates the backup control configuration to switch off the faulty components in such a way as to maintain stability and recover the control performance. Please see [84] for a literature survey on active FTC.

Passive and active FTC strategies each have distinct advantages and limitations. Passive FTC is relatively straightforward, enabling easy implementation and rapid response. However, it is designed to anticipate specific failure modes, making it conservative and limiting its fault-tolerant capabilities. On the other hand, active FTC can manage a broader range of faults that passive systems cannot, though its effectiveness heavily depends on the accuracy of fault identification.

B. Classifications of FTMPC

Model predictive control is potentially a promising tool for achieving fault tolerance in the presence of unexpected faults. First, MPC exhibits inherent robustness against uncertainties due to its capability to recalculate control actions at each sampling interval. Any changes in plant dynamics directly in the ongoing computations. In addition, MPC has some degrees of self-reconfiguration capability, a unique feature named the "daisy-chaining" property in [85]. Methods that make use of the inherent properties of MPC fall into the category of passive FTC.

Passive FTMPC methods, while effective up to a point, have their limitations and may not fully mitigate the effects of severe faults. In such cases, active fault-tolerant approaches become necessary. Given that MPC is a model-based control method, its effectiveness heavily depends on the accuracy of the system model. When faults occur, they introduce unexpected changes to this model, necessitating the integration of fault detection or approximation techniques within the MPC framework. These techniques are crucial for maintaining high accuracy and performance of the control system despite the presence of faults.

As discussed above, FTMPC can also be categorized into passive and active designs. Passive FTMPC approaches utilize the inherent robustness of MPC to shield the system against a predefined set of faults. These methods rely on the ability of MPC to adjust control actions dynamically in response to deviations without specific modifications for fault handling. In contrast, active FTMPC approaches proactively address faults as they occur, aiming to restore system functionality through either accommodation or reconfiguration. The strategies employed in active FTMPC can be further differentiated based on how they utilize fault information to modify the MPC framework. These strategies include:

- **Online accommodation:** This approach involves adjusting the MPC model parameters and constraints in real-time based on fault information detected or estimated. The primary focus here is on updating the model to reflect the fault's impact accurately, compensating for disturbances without altering the control structure itself. This method relies heavily on the accuracy and timeliness of parameter estimation, ensuring that control actions remain effective despite the presence of faults.

- **Control reconfiguration:** Unlike online accommodation, control reconfiguration involves significant changes to the control loop itself. This strategy may include activating a backup control system, disconnecting faulty components,

Table 1.1: Classifications of FTMPC according to FTC mechanisms and system types

Fault tolerance mechanisms \ Type of systems dealt with		Linear systems	Nonlinear systems	Network systems
Passive FTC		[86] [87] [88] [89]	[90] [31] [91]	
Active FTC	Online accommodation	[86] [92] [93] [94] [95] [103] [104] [105] [106] [107] [111] [112]	[96] [97] [98] [99] [100] [108] [109] [110]	[101] [102]
	Control reconfiguration	[113] [114] [115] [116] [117] [122] [123] [124] [86]	[118] [119]	[120] [121]
	Accommodation + reconfiguration	[125] [126]		

Table 1.2: Classifications of FTMPC according to application fields

Field of applications	References
Autonomous aerial vehicles	[86]
Water transport and sewer networks	[86] [125] [115]
Power & energy systems	[113] [122]
Motor driving	[123] [124]
Wind turbines	[109]
Steer-by-wire systems	[105]

or adjusting control parameters such as reference signals and performance costs. The reconfiguration is especially useful when structural changes occur within the system or when the obtained fault information is imprecise or incomplete. It ensures that the control system remains functional and meets performance criteria under altered operational conditions.

- **Accommodation + reconfiguration:** Some studies explore integrating both accommodation and reconfiguration strategies. By combining real-time model updates with strategic control modifications, this approach offers a robust solution for maintaining control performance in the face of both predictable and unpredictable changes.

This classification of FTMPC results, as outlined in Table 1.1, effectively summarizes existing papers according to the fault tolerance mechanisms they employed and the types of systems they target. It is evident from this table that the majority of studies focus on linear systems, which are generally easier to address compared to nonlinear systems. Additionally, the table highlights a notable gap in research concerning FTMPC for networked or multi-agent systems, especially in the presence of faults in communication links.

Another classification provided in Table 1.2 summarizes the developed FTMPC

Table 1.3: Classifications of active FTMPC according to applied fault detection approaches

Applied fault detection approaches	References
Gaussian processes	[96] [100]
Moving horizon estimation	[97] [112]
Parameter estimator	[101]
Observer	[95] [103] [114]
External FDI module	[113] [94] [106] [107] [115] [116] [126] [123] [86]
Delta operator	[105]
Fault identification combined with MPC	[104]
Neural networks & modeling	[99] [108]
Fuzzy model	[110]

methods across a diverse range of engineering applications. As can be seen, most applications of FTMPC have so far been on aerospace, industrial, and mechanical systems. In particular, significant research efforts have been directed toward flight control and industrial process control, which are fields with perpendicularly high safety requirements.

The development of active FTMPC schemes significantly relies on the effectiveness of fault detection strategies, as detailed in the classification presented in Table 1.3. This classification distinguishes between approaches based on how they implement fault detection. One segment of research focuses on online data-driven methodologies, utilizing techniques like Gaussian processes, moving horizon estimation, parameter estimators, and state observers to extract critical post-failure information from system data. These methods enable real-time, adaptive responses to faults. Another substantial portion of research relies on external FDI modules that provide real-time failure information. This approach allows researchers to concentrate on enhancing the MPC's control responses, assuming reliable fault detection is handled by these external systems.

The existing research on FTMPC is also organized according to the specific MPC methods used, as shown in Table 1.4. Most studies employ the basic MPC approach, which, despite its widespread use, does not inherently guarantee closed-loop stability. Alternatively, some studies employ MPC formulations that guarantee convergence but only under nominal conditions —assuming an ideal scenario where faults are either immediately isolated or accurately estimated. This assumption can be overly optimistic, as it overlooks the practical challenges associated with real-time fault detection and the dynamic nature of system responses to faults. Significantly, the lit-

Table 1.4: Classifications of FTMPC according to applied MPC approaches

Applied MPC approaches	References
Basic MPC with no guarantee of stability	[127] [90] [87] [31] [89] [100] [96] [120] [101] [92] [98] [125] [113] [93] [94] [119] [105] [114] [107] [100] [108] [109] [126] [122] [121] [123] [110] [102] [124] [86]
MPC with terminal ingredients	[111] [115] [99] [106] [104] [103] [95] [97]
Lyapunov-based MPC	[31] [112]
Tube-based MPC	[116]
Min-max MPC	[91] [118]
Explicit MPC	[88]
Multiple MPC	[117]

erature lacks comprehensive evaluations of overall system performance that explicitly consider the potential estimation errors and the transient dynamics introduced by fault detection and estimation. The effectiveness of fault-tolerant MPC thus heavily depends on the speed and accuracy of the FDI schemes employed. This gap in the research underscores a critical area for future investigation, indicating the need for more studies that address these real-world complexities and enhance the reliability of MPC in fault-prone environments.

1.3 Motivations

Despite notable advances in MPC and FTMPC, several pertinent issues remain inadequately addressed, particularly when applied to UAV systems. There is a clear need for deeper investigation into the application and optimization of FTMPC to enhance the reliability and efficiency of UAV control. The following gaps in research have significantly motivated the studies presented in this dissertation:

- Although significant advancements have been made in FTMPC, most approaches still lack rigorous theoretical guarantees for closed-loop performance and stability, particularly when addressing nonlinear systems. Some can only assure stability under ideal conditions—when accurate fault information is obtained through additional fault estimation or detection strategies. This gap in theoretical rigor arises because accounting explicitly for the errors and transient dynamics in fault estimation and detection could complicate the real-time updating of terminal constraints and costs. Such complexity could also lead to a substantial increase in computational burden, which is generally undesirable in flight control systems. Therefore, developing a novel FTMPC strategy that

ensures closed-loop performance, exhibits less conservatism, and reduces computational complexity holds both theoretical significance and practical value for reliable UAV control. This need motivates the research presented in Chapter 3.

- When it comes to applying MPC to UAV control, the most prominent obstacle is still the contradiction between the high computational demands of solving the MPC optimization problem and the limited computational resources available on the onboard microcontrollers of UAVs. For MPC strategies that incorporate terminal ingredients to ensure system stability, a long prediction horizon is necessary to secure a sufficiently large region of attraction. However, this extended prediction horizon increases the computational time required to solve the MPC optimization problem. While such delays are manageable in process control scenarios characterized by slower dynamics, they pose significant challenges for flight control, since aerial vehicles are fast-evolving systems that require a relatively high control update frequency. Therefore, it is crucial to develop a unified framework for applying MPC to UAV control that not only achieves optimal control performance but also mitigates the computational complexity. This is the motivation behind the research in Chapter 4.

- Although numerous DMPC methods have been developed, their application to multi-UAV formation control encounters significant challenges due to inherent limitations. Firstly, vehicle-embedded hardware has limited computational resources, yet DMPC methods with terminal constraints need a long prediction horizon, demanding significant computation to ensure feasibility. Secondly, most DMPC approaches are designed for cooperative regulation tasks that aim to drive all agents toward a pre-set target, assuming full connectivity to the leader, which is unrealistic in formation control where only some followers may access the leader's information. Moreover, current DMPC algorithms often require extensive information sharing, either iteratively or sequentially, to attain the most up-to-date optimization results from neighbors for global stability. Such intensive communication requirements conflict with the limited bandwidth of wireless networks in multi-UAV systems. These research gaps highlight the necessity for developing a novel DMPC framework specifically tailored for multi-UAV formation tracking, which is the research focus of Chapter 5.

- Communication networks in MASs are critical for agent interactions and cooper-

ative behaviors, yet they also introduce significant vulnerabilities to the control system due to their susceptibility to unexpected events such as cyber-attacks and channel fading. If such communication faults are not effectively managed, they can severely impact the control system's performance and stability. Despite these challenges, the development of DMPC for multi-agent formation tracking control in the presence of communication link faults remains an underexplored area of research. This gap in the literature motivates the investigation in Chapter 6.

1.4 Organization and Contributions

The remainder of this dissertation includes one chapter on preliminary knowledge, four chapters presenting our research outcomes, and one concluding chapter. To elaborate, the content and contributions of each chapter are:

- **Chapter 2:** To ensure this dissertation is self-contained, we begin by proving essential mathematical preliminaries. This includes relevant mathematical models and theorems necessary to support the research findings that follow.

- **Chapter 3:** A novel adaptive FTMPC method is proposed in this chapter for nonlinear systems experiencing unexpected actuator faults and input constraints. Utilizing a Lyapunov-based MPC framework that incorporates an adaptive fault parameter estimator, the approach improves on existing adaptive MPC methods by simplifying the formulation. Notably, it eliminates the need for complex terminal ingredients, thereby reducing the computational burden. In addition, the proposed design uniquely addresses both multiplicative and additive uncertainties caused by actuator faults. Furthermore, stability of the closed-loop system and feasibility of the optimization process are rigorously proven. The derived stability conditions explicitly characterize the relationship between closed-loop performance, control parameter values, and the sampling period, offering valuable guidelines for parameter selection.

- **Chapter 4:** This chapter extends FTMPC method developed in Chapter 3, applying it to the trajectory tracking control problem of UAVs with input constraints and actuator faults. It presents the design and stability analysis of a dual-loop hierarchical controller. The hierarchical controller, developed based

on a dual-time-scale model decomposition, consists of two control loops: the outer-loop translation control and the inner-loop rotation control. For translation control, the fault-tolerant Lyapunov-based MPC is applied. For rotation control, feedback linearization is employed, and an adaptive fault-tolerant control analytical law is implemented. The hierarchical control system employs dual-rate sampling, with a slower rate for the computationally intensive outer-loop that applies MPC and a faster rate for the inner-loop, successfully reducing the control system's computational burden while maintaining its fast response. With the inter-sample behavior and the interconnection between the translational and rotational dynamics taken into account, the closed-loop stability of the dual-loop and dual-rate control system is rigorously proven using singular perturbation theory.

- **Chapter 5:** This chapter tackles the formation tracking control problem for multiple UAVs interconnected via a directed communication graph. We propose a novel DMPC framework, which equips each UAV with an individual controller featuring a hierarchical architecture composed of three sequentially connected control layers. This DMPC design enables all vehicles to achieve a predetermined geometric configuration and simultaneously track a dynamic virtual leader, who is only partially connected to the network, even in the presence of unexpected actuator faults. The design significantly enhances communication and computational efficiency by requiring only a single round of calculation and communication along the directed communication graph per control update. Furthermore, within the control framework, adaptive estimators are employed to estimate and compensate for unknown fault parameters, as well as prediction mismatches arising from non-real-time communication among optimizers.

- **Chapter 6:** In order to address the nonlinear multi-agent formation tracking control problem with input constraints and unknown communication faults, a novel adaptive distributed observer-based DMPC method is developed in this chapter. This design employs adaptive distributed observers in local control systems to estimate the leader's state, dynamics, and relative positioning with respect to the leader. Utilizing the estimated data as local references, the original formation tracking control problem can be decomposed into several fully localized tracking control problems, which can be efficiently solved by the local MPC controller. Through the incorporation of adaptive distributed

observers, this proposed design not only enhances the resilience of distributed formation tracking against communication faults but also simplifies the DMPC formulation.

- **Chapter 7:** This chapter provides concluding remarks and discusses several remaining issues that suggest directions for future research.

All the numerical simulations are performed in MATLAB, utilizing YALMIP [128], a toolbox for modeling and optimization. The *optimizer* function in YALMIP is used to construct and solve the MPC optimization problem efficiently, where the solver IPOPT [129] is employed for nonlinear programming.

Chapter 2

Mathematical Preliminaries

Mathematics plays a crucial role in control design and analysis, offering tools for modeling the behavior of dynamic systems, designing controllers to achieve desired performance, and evaluating closed-loop control performance. Therefore, in this chapter, we provide mathematical preliminaries for this dissertation. In Section 2.1, mathematical models of the systems and control problems considered in this dissertation are given. In Section 2.2, some mathematical concepts and theorems are provided, as they are useful tools for control development and analysis in the subsequent chapters.

2.1 Mathematical Modeling

Mathematical modeling plays a crucial role in the development of control strategies, facilitating the understanding of system behavior. In this dissertation, we investigate both general nonlinear systems and the specific system of UAVs. We explore configurations involving a single agent as well as multiple agents working together. Additionally, various fault scenarios, including actuator malfunctions and communication disruptions, are considered. Therefore, below, we present mathematical models describing system dynamics, interactions among agents in multi-agent configurations, and the considered faults.

2.1.1 Modeling of Systems

This dissertation investigates the control problems of two distinct yet connected types of systems. Chapter 3 examines a broader category of general nonlinear systems, while Chapter 6 focuses on its multi-agent counterparts. Chapters 4 and 5 tackles the

specific control challenges posed by UAVs, exploring both single-UAV and multi-UAV scenarios, respectively. Hence, we now provide the mathematical modelings of these two types of systems.

A. General MIMO Nonlinear Systems

The first type of system is a class of control-affine MIMO nonlinear systems whose dynamics can be mathematically described by the following set of differential equations:

$$\dot{\xi} = a(\xi) + B(\xi)u \tag{2.1a}$$

$$y = h(x) \tag{2.1b}$$

where $\xi = \begin{bmatrix}\xi_1 & \xi_2 & \cdots & \xi_{n_\xi}\end{bmatrix}^\top \in \mathbb{R}^{n_\xi}$, $u = [u_1 \ u_2 \ \cdots \ u_n]^\top \in \mathbb{R}^n$ and $y = [y_1 \ y_2 \ \cdots \ y_n]^\top \in \mathbb{R}^n$ are the system state, the control input and the output, respectively; $a(\xi) = [a_1(\xi) \ a_2(\xi) \cdots \ a_{n_\xi}(\xi)\Big]^\top$; $B(\xi) = [b_1(\xi) \ b_2(\xi) \cdots \ b_n(\xi)]$ in which $b_i(\xi) = [b_{i1}(\xi) \ b_{i2}(\xi) \ \cdots \ b_{in}(\xi)]^\top$ for $i = 1, 2, \cdots, n_\xi$; $h(\xi) = [h_1(\xi) \ h_2(\xi) \ \cdots \ h_n(\xi)]^\top$. All entries of the vector functions $a(x)$ and $h(x)$ and the $n_\xi \times n$ matrix function $B(x)$ are sufficiently smooth.

Finding a direct solution to the control problem for nonlinear systems like (2.1) is often challenging. One commonly adopted method involves employing a change of coordinates followed by the design of a feedback control law to transform the nonlinear system into either a fully or partially linear system. We now provide the process of coordinate transformation and the resulting equivalent canonical system. The subsequent chapters, Chapters 3 and 6, will proceed with control design based on the canonical system directly.

To obtain the canonical form, we must understand how the input u enters the system and affects the output y. In this regard, we introduce the following concept of relative degree:

Definition 2.1. *(**System relative degree** [130]) The ith output of the system (2.1), y_i, is said to have relative degree r_i, if*

$$\mathfrak{L}_B h_i(\xi) = \mathfrak{L}_B \mathfrak{L}_a h_i(\xi) = \mathfrak{L}_B \mathfrak{L}_a^2 h_i(\xi) = \cdots = \mathfrak{L}_B \mathfrak{L}_a^{r_i-2} h_i(\xi) = 0 \tag{2.2a}$$

$$\mathfrak{L}_B \mathfrak{L}_a^{r_i-1} h_i(\xi) \neq 0 \tag{2.2b}$$

where

$$\mathfrak{L}_a h_i(\xi) = \frac{\partial h_i(\xi)}{\partial \xi} a(\xi) = \frac{\partial h_i(\xi)}{\partial \xi_1} a_1(\xi) + \frac{\partial h_i(\xi)}{\partial \xi_2} a_2(\xi) + \cdots + \frac{\partial h_i(\xi)}{\partial \xi_m} a_m(\xi)$$

is the standard Lie derivative of a scalar function $h_i(\xi)$ with respect to the vector function $a(\xi)$. The Lie derivative notations can be extended using the following chain laws to represent repeated derivatives with respect to the same vector field or with respect to matrix functions:

$$\begin{aligned}
\mathfrak{L}_a^2 h_i(\xi) &= \mathfrak{L}_a \mathfrak{L}_a h_i(\xi) = \frac{\partial \left(\mathfrak{L}_a h_i(\xi)\right)}{\partial \xi} a(\xi) \\
\mathfrak{L}_a^k h_i(\xi) &= \mathfrak{L}_a \mathfrak{L}_a^{k-1} h_i(\xi) = \frac{\partial \left(\mathfrak{L}_a^{k-1} h_i(\xi)\right)}{\partial \xi} a(\xi) \\
\mathfrak{L}_B h_i(\xi) &= [\mathfrak{L}_{b_1} h_i(\xi) \ \mathfrak{L}_{b_2} h_i(\xi) \ \cdots \ \mathfrak{L}_{b_n} h_i(\xi)] \\
\mathfrak{L}_B \mathfrak{L}_a h_i(\xi) &= [\mathfrak{L}_{b_1} \mathfrak{L}_a h_i(\xi) \ \mathfrak{L}_{b_2} \mathfrak{L}_a h_i(\xi) \ \cdots \ \mathfrak{L}_{b_n} \mathfrak{L}_a h_i(\xi)] \\
\mathfrak{L}_B \mathfrak{L}_a^k h_i(\xi) &= \left[\mathfrak{L}_{b_1} \mathfrak{L}_a^k h_i(\xi) \ \mathfrak{L}_{b_2} \mathfrak{L}_a^k h_i(\xi) \ \cdots \ \mathfrak{L}_{b_n} \mathfrak{L}_a^k h_i(\xi)\right]
\end{aligned}$$

This definition of relative degree is in light of taking time derivatives of the output. For the system (2.1), its total relative degree is the total number of times we have to differentiate each output y_i, $i = 1, 2, \cdots, n$ before the input u appears explicitly, which is $\sum_{i=1}^{n} r_i$. The necessary and sufficient condition for the existence of a transformation capable of completely linearizing (2.1) from the input/output point of view is that its relative degree equals the order of the system. To ensure the input/output linearizability, we made the following assumption

Assumption 2.1. *For the convenience of theoretical analysis, we assume that the relative degree for each output y_i, $i = 1, 2 \cdots, n$ in (2.1) is identical and denoted as r, such that $r_1 = r_2 = \cdots = r_n = r$. To ensure input/output complete linearization, we further assume that the total relative degree of the system equals its order, that is, $\sum_{i=1}^{n} r_i = nr = n_\xi$.*

Lemma 2.1. *Under Assumption 2.1, there exist a global diffeomorphism*

$$x_1 = y = h(\xi) \tag{2.3a}$$
$$x_2 = \dot{y} = \mathfrak{L}_a h(\xi) = [\mathfrak{L}_a h_1(\xi)\ \mathfrak{L}_a h_2(\xi)\ \cdots\ \mathfrak{L}_a h_n(\xi)]^\top \tag{2.3b}$$
$$\vdots$$
$$x_r = y^{(r-1)} = \mathfrak{L}_a^{r-1} h(\xi) = \left[\mathfrak{L}_a^{r-1} h_1(\xi)\ \mathfrak{L}_a^{r-1} h_2(\xi)\ \cdots\ \mathfrak{L}_a^{r-1} h_n(\xi)\right]^\top \tag{2.3c}$$

that transforms the original plant (2.1) into an equivalent strict-feedback system in lower triangular form with no zero dynamics as follows

$$\begin{cases} \dot{x}_1 = x_2 \\ \qquad\vdots \\ \dot{x}_{r-1} = x_r \\ \dot{x}_r = f(x) + G(x)u \\ y = x_1 \end{cases} \tag{2.4}$$

where $x = \left[x_1^\top\ x_2^\top\ \cdots\ x_r^\top\right]^\top \in \mathbb{R}^{rn}$ *is the transformed state vector with each component* $x_i \in \mathbb{R}^n$, $i = 1, 2, \cdots, r$; $f(x) = \mathfrak{L}_a^r h(\xi) = [\mathfrak{L}_a^r h_1(\xi)\ \mathfrak{L}_a^r h_2(\xi)\ \cdots\ \mathfrak{L}_a^r h_n(\xi)]^\top : \mathbb{R}^{rn} \to \mathbb{R}^n$ *and* $G(x) = \mathfrak{L}_B \mathfrak{L}_a^{r-1} h(\xi) = \left[\mathfrak{L}_B \mathfrak{L}_a^{r-1} h_1(\xi)^\top\ \mathfrak{L}_B \mathfrak{L}_a^{r-1} h_2(\xi)^\top \cdots \mathscr{L}_B \mathfrak{L}_a^{r-1} h_n(\xi)^\top\right]^\top : \mathbb{R}^{rn} \to \mathbb{R}^{n\times n}$, $i, j = 1, 2, \cdots, n$.

B. UAV Dynamics

The UAVs studied in this dissertation are of the rotary-wing aircraft type. A typical rotary-wing UAV is equipped with four rotors symmetrically distributed at an equal distance from the center, as illustrated in Figure 2.1. These four rotors, configured to spin in opposite directions in pairs, generate upward-lifting forces denoted as F_1, F_2, F_3, and F_4. By adjusting the spinning speed of each rotor, the necessary force and torque for maneuvering the vehicle to execute specific translation and rotation behaviors can be generated.

In order to model the UAV dynamics, including both translational and rotational motions, two reference frames are introduced: an earth-fixed inertial frame with origin 0^e and axes x^e, y^e, z^e, and a body-fixed frame with 0^b and x^b, y^b, z^b. The position of the quadrotor is represented by the coordinate values of the center of mass in the earth-fixed frame, denoted as $\zeta = [x\ y\ z]^\top$. The linear velocity of the vehicle with

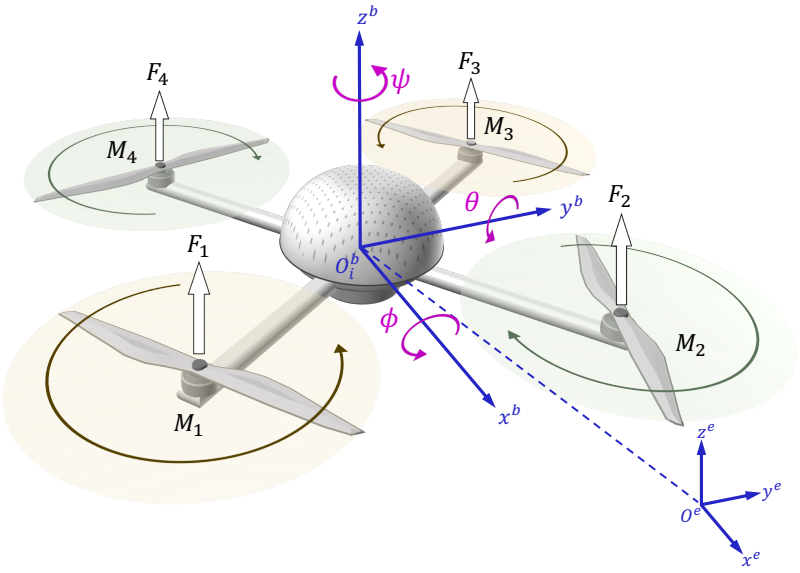

Figure 2.1: Configuration of a typical rotary-wing UAV

respect to the body-fixed frame is denoted as $v = [v_x \ v_y \ v_z]^\top$. The attitude of the quadrotor, describing the rotation of the body-fixed frame with respect to the earth-fixed frame, is denoted by Euler angles as $\eta = [\phi \ \theta \ \psi]^\top$, in which ϕ, θ, ψ are the roll, pitch, and yaw angles. $\omega = [\omega_p \ \omega_q \ \omega_r]^\top$ is a vector of angular velocities with respect to the body-fixed frame. By using the Lagrangian method, the UAV dynamic model can be established, as represented by the following differential equation sets [131]:

$$\dot{\zeta} = v \tag{2.5a}$$

$$m\dot{v} = -mg + r(\eta)F \tag{2.5b}$$

$$\dot{\zeta} = W(\eta)\omega \tag{2.5c}$$

$$J\dot{\omega} = -\omega \times J\omega + \tau \tag{2.5d}$$

with

$$r(\eta) = \begin{bmatrix} \sin\theta\cos\psi\cos\phi + \sin\psi\sin\phi \\ \sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}$$

$$W(\eta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\psi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}$$

where m is the vehicle mass; $g = [0\ 0\ g^z]^\top$ with g^z being the gravitational acceleration; $J = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$ is a diagonal matrix of moments of inertia; F is the total thrust force applied along z^b; $\tau = [\tau_\phi\ \tau_\theta\ \tau_\psi]^\top$ is a vector of rotation torques.

Both the lift force F_i and the rotation torque τ_i arise from the rotors equipped on the vehicle, serving as the system's actuators. For a typical quadrotor UAV, as illustrated in Fig. 2.1, the total lift force and the rotational torques are derived from the sum and the differences of the forces and moments produced by the four rotors:

$$\begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -l & l & l & -l \\ -l & -l & l & l \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{2.6a}$$

$$\tau_\psi = -M_1 + M_2 - M_3 + M_4 \tag{2.6b}$$

where l is the distance from each motor to the center of mass; F_j and M_j for $j = 1, 2, 3, 4$ denote the upward forces and the rotational moments of the jth rotor, which are proportional to the rotor's rotation rate as

$$F_j = c_l w_j \tag{2.7a}$$

$$M_j = c_r w_j,\ \ j = 1, 2, 3, 4 \tag{2.7b}$$

in which, w_j is the pulse-width modulation (PWM) input to the DC motor of the jth rotor; c_l and c_r are the lift and resistance coefficients.

Remark 2.1. *The established model (2.5) reveals that the UAV dynamics is a second-order 3-dimensional nonlinear system, falling into the category of nonlinear systems described in (2.4).*

Remark 2.2. *A set of solutions to (2.5), denoted as $(\zeta^*, \upsilon^*, \eta^*, \omega^*, F^*, \tau^*)$, establishes the equilibrium point of UAV dynamics when $\phi^* = \theta^* = 0$, $\upsilon^* = \omega^* = \tau^* = 0$ and $F^* = mg$. This equilibrium point corresponds to the UAV hovering conditions, where the overall lift force exactly counteracts the gravitational force, rotational torques are absent, and both roll and pitch angles are zero. Under these conditions, the UAV maintains a stable hover at position ζ^* and remains a constant yaw angle ψ^*.*

2.1.2 Modeling of Communication

We also explore MAS control and multi-UAV cooperation in Chapters 5 and 6. Therefore, in addition to dynamics modeling, it is also necessary to find a structured representation of the interactions among agents [132]. Typically, this is accomplished through directed or undirected graphs.

A. Graph Representation

For example, given a MAS with M agents, a directed graph, defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ can be employed for communication modeling. In this representation,

- **Node:** Agents are often represented as nodes in a graph. $\mathcal{V} = \{1, 2, \cdots, M\}$ is the set of nodes with each node representing an agent.
- **Edges:** Graph edges represent communication channels or links between agents. These edges denote the ability of agents to exchange information, messages, or signals with each other. $\mathcal{E} = \{(j, i) | i, j \in \mathcal{V}, i \neq j\}$ is the set of edges, indicating communication links between agents. For an undirected graph, if $(i, j) \in \mathcal{E}$, then so is (j, i), indicating that the communication link between agent i and agent j is bidirectional. In contrast, for a directed graph, the edge (i, j) in $\mathcal{E}$ means that agent j can obtain information from agent i, but not necessarily vice verse.

B. Neighborhood Representation

In this dissertation, we only deal with MASs connected via a directed communication graph. To further describe the unidirectional inter-agent communication, we introduce notions of "in-neighbors" and "out-neighbors":

- **In-neighbors:** The in-neighbor set of agent i is defined as $\mathcal{N}_i^- = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$, containing the neighboring nodes that have an edge pointing toward i for sending information to i.
- **Out-neighbors:** Similarly, the set of out-neighbors is defined as $\mathcal{N}_i^+ = \{l \in \mathcal{V} | (i, j) \in \mathcal{E}\}$, containing all the nodes that receive information from i.

C. Virtual Leader Representation

In order to facilitate efficient coordination and collaboration among multiple agents or nodes within a network, a virtual leader is often introduced in many graph representations. The representation is

- **Leader node:** The leader can be labeled as node 0 in the graph. The extended set of nodes is defined as $\mathcal{E}^0 = \mathcal{E} \cup \{0\}$.
- **Pinning edge:** The connectivity between the leader node 0 to the other M follower nodes can be represented as pinning edges. The set of pinning edges is defined as $\mathcal{E}^0 = \{(0, i) | i \in \mathcal{V}\}$. Here, $(0, i) \in \mathcal{E}^0$ means that agent i has direct access to the leader 0.

D. Associated Matrices

In graph theory, matrices play a significant role in representing and analyzing graphs. The most commonly used matrices are the adjacency matrix, the Laplacian matrix, and the pinning matrix, defined as follows:

- **Adjacency matrix:** The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathcal{R}^{M \times M}$ of a directed graph $\mathcal{G}$ is defined such that a_{ij} is positive if $(j, i) \in \mathcal{E}$, and $a_{ij} = 0$ if $(j, i) \notin \mathcal{E}$. In general, $a_{ii} = 0$ without further indication, as self-edges are typically not allowed in graph representation. Note that a_{ij} denotes the weight for edge (j, i). If it is an unweighted graph, then a_{ij} is set equal to 1 or 0.
- **Laplacian matrix:** The Laplacian matrix of $\mathcal{G}$ is denoted as $\mathcal{L} = [l_{ij}] \in \mathcal{R}^{M \times M}$, where $l_{ii} = \sum_{j=1}^{M} a_{ij}$ and $l_{ij} = -a_{ij}$ for $i \neq j$.
- **Pinning matrix:** The pinning matrix is employed when there is a virtual leader 0 in addition to the m follower nodes. It is denoted as $\mathcal{B} = \text{diag}(b_1, b_2, \cdots, b_M)$, where $b_i > 0$ if $(0, i) \in \mathcal{E}^0$ and $b_i = 0$ otherwise. b_i denotes the weight for the pinning edge. If the weight is not relevant, then b_i is either 1 or 0.

These matrices offer a convenient way of storing graph data and performing computations on graphs using standard matrix algorithms. They facilitate the efficient representation of various graph properties, allowing for the application of graph theory concepts to multi-agent control. Later in Section 2.2, some important properties of graph theory will be introduced.

2.1.3 Modeling of Faults

One of the primary research objectives of this dissertation is to ensure system performance even in the presence of faults or malfunctions. Fault modeling is an essential aspect of developing fault tolerance strategies. In this dissertation, we consider two common types of faults: Actuator faults on a specific component and communication faults that happen between different components or subsystems. Here, modeling of these two types of faults is provided.

A. Actuator Faults

Actuators, including motors, valves, pistons, and other mechanical or electromechanical devices, are responsible for translating control signals into physical motions or actions to achieve desired system performance. Actuator faults refer to any unexpected deviations or failures in the behavior or performance of these devices, which can lead to undesirable effects on the controlled system. Actuator faults can be modeled as multiplicative and additive uncertainties in the control channel, as follows:

$$u(t) = \Theta_f(t)u_c(t) + \vartheta_{f^0}(t) \tag{2.8}$$

where $u \in \mathbb{R}^n$ is the actual control actions applied to the plant system, while $u_c \in \mathbb{R}^n$ is the control signals, serving as the input of the actuator model; $\Theta_f = \text{diag}\{\vartheta_{f_1}, \vartheta_{f_2}, \cdots, \vartheta_{f_n}\} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with each element indicating the corresponding actuator efficiency; $\vartheta_{f^0} = \begin{bmatrix}\vartheta_{f_1^0} & \vartheta_{f_2^0} & \cdots & \vartheta_{f_n^0}\end{bmatrix}^\top$ representing the time-varying uncontrolled portion of the control input that is produced by actuators.

This actuator fault modeling is common in FTC studies [133–135]. With this model, multiple common working patterns of actuators can be represented:

- **Healthy operation with no faults:** When the ith actuator is working normally without any faults, $\vartheta_{f_i} = 1$ and $\vartheta_{f_i^0} = 0$.
- **Partial loss of effectiveness:** In the case of partial loss of effectiveness in the ith actuator, the multiplicative parameter $\vartheta_{f_i} \in (0, 1]$ represents the actual actuator efficiency.
- **Additive bias fault:** Additive faults could happen on the faulty actuator, leading to a continuous bias $\vartheta_{f_i^0}$ in its output.

The actuator fault modeling proposed in (2.8) is also effective in encapsulating

common actuator faults in quadrotor UAV systems. Such faults may reduce the efficiency of lift and resistance coefficients or cause uncontrolled deviations in PWM inputs, which can be mathematically represented as follows:

$$c_l^f = \vartheta_{c_l} c_l \tag{2.9a}$$

$$c_r^f = \vartheta_{c_r} c_r \tag{2.9b}$$

$$w_j^f = w_j + \vartheta_{w_j^0},\ j = 1, 2, 3, 4 \tag{2.9c}$$

Here, ϑ_{c_l} and ϑ_{c_r} denote scaling factors that quantify the reduced efficiency of lift and rotational functions under fault conditions, generally ranging from 0 to 1. $\vartheta_{w_l^0}$ the unintentional components added to the PWM signals, potentially caused by electronic malfunctions, signal interference, or other issues impacting actuator performance. In the presence of these faults, the relationship between actual forces and torques and their corresponding control commands can be similarly expressed as:

$$\begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \vartheta_{c_l} & 0 & 0 & 0 \\ 0 & \vartheta_{c_l} & 0 & 0 \\ 0 & 0 & \vartheta_{c_l} & 0 \\ 0 & 0 & 0 & \vartheta_{c_r} \end{bmatrix} \begin{bmatrix} u_F \\ u_{\tau_\phi} \\ u_{\tau_\theta} \\ u_{\tau_\psi} \end{bmatrix} + \begin{bmatrix} \vartheta_{F^0} \\ \vartheta_{\tau_\phi^0} \\ \vartheta_{\tau_\theta^0} \\ \vartheta_{\tau_\psi^0} \end{bmatrix} \tag{2.10}$$

where

$$\begin{bmatrix} \vartheta_{F^0} \\ \vartheta_{\tau_\phi^0} \\ \vartheta_{\tau_\theta^0} \\ \vartheta_{\tau_\psi^0} \end{bmatrix} = \begin{bmatrix} c_l^f & c_l^f & c_l^f & c_l^f \\ -lc_l^f & lc_l^f & lc_l^f & -lc_l^f \\ -lc_l^f & -lc_l^f & lc_l^f & lc_l^f \\ -c_r^f & c_r^f & -c_r^f & c_r^f \end{bmatrix} \begin{bmatrix} \vartheta_{w_1^0} \\ \vartheta_{w_2^0} \\ \vartheta_{w_3^0} \\ \vartheta_{w_4^0} \end{bmatrix}$$

where u_F and $u_\tau = [u_{\tau_\phi}\, u_{\tau_\theta}\, u_{\tau_\psi}]^\top$ are the expected force and torque commands determined by the local controllers of UAV i.

B. Communication Link Faults

Apart from faults that happen on specific components or subsystems, we also take account of faults in the communication network. They can be modeled as uncertain-

ties on edge weights as [136]

$$a_{ij}^f(t) = a_{ij} + \vartheta_{ij}^a(t) \tag{2.11a}$$

$$b_i^f(t) = b_i + \vartheta_i^b(t) \tag{2.11b}$$

where a_{ij} and b_i are the ideal weights of general and pinning edges, and ϑ_{ij}^a and ϑ_i^b denote corrupted weights caused by communication faults. It can be seen from the modeling that communication faults cause the edge weights to be time-varying and even unknown.

This fault model covers the following types of communication faults:

- **Channel manipulation attack:** The unknown corrupted weights ϑ_{ij}^a and ϑ_i^b are capable of modeling a range of cyber attacks. The unknown corrupted weights ϑ_{ij}^a and ϑ_i^b can simulate various cyber attacks. These attacks involve infiltrating communication channels and manipulating the shared information between vehicles.

- **Fading channel:** The corrupted communication weights can also represent the effect of a fading channel, resulting in a decrease in the values of the communication weights.

2.2 Mathematical Concepts and Theorems

In this section, some fundamental concepts and theorems concerning signals and systems are covered. Firstly, we introduce concepts of norms and several properties for matrices, followed by the definition of function continuity. Then, we review the concepts of stability and Lyapunov's stability theorem. Furthermore, we present the definition and properties of stable sliding surfaces. Finally, we review some important results in graph theory.

This review is not exhaustive but aims to provide essential mathematical tools for the design and analysis in the subsequent chapters. Since most of the results given in this chapter are standard in the nonlinear systems and control literature, they are presented without proof. For those interested in detailed proofs, classic books such as [130, 132, 137–139] can be referred to.

2.2.1 Norms for Vectors and Matrices

The magnitude of signals provides essential information about the behavior and characteristics of the system under study. With signal magnitude, control engineers can understand how the system responds to different inputs and disturbances, tune the controllers, and assess the stability of the system and the effectiveness of the control strategies. Therefore, appropriate measurement and interpretation of signal magnitudes are critical for the design, analysis, and optimization of control systems.

Norms, which map vector or matrix spaces to the non-negative real numbers, are effective mathematical tools to quantify signal magnitudes. This section defines the norms used in this dissertation for quantifying the size and magnitudes of vectors and matrices, followed by introducing the associated properties that will be employed in the subsequent chapters.

A. Norms for Vectors

In the vector space, the class of p-norms is defined by

$$\|x\|_p = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}, & \text{for } 1 \leqslant p < \infty \\ \max_{1 \leqslant i \leqslant n} |x_i|, & \text{for } p = \infty \end{cases} \tag{2.12}$$

where $x = [x_1 \; x_2 \; \cdots \; x_n]^\top$ in a vector on the n-dimensional Euclidean space $\mathbb{R}^n$. All these norms are equivalent in the sense that they all define the same topology, but the most commonly used one among them is the 2-norm $\|x\|_2$, also known as the Euclidean norm. The Euclidean norm gives the ordinary distance from x to the origin. Hence, this norm can be denoted as $\|x\|$ for simplicity and written as the square root of the matrix product of a vector with its transpose:

$$\|x\| = \sqrt{x^\top x} \tag{2.13}$$

Another useful notation is $\|x\|_P^2$, which presents the weighted square norm of x, defined as:

$$\|x\|_P^2 = x^\top P x \tag{2.14}$$

where P is an $n \times n$ matrix.

A. Norms for Matrices

Norms for matrices are typically extensions of vector norms. For an $m \times n$ matrix A, the induced p-norms of A can be expressed as

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \tag{2.15}$$

When $p = 2$, the induced matrix norm is also called the spectral norm, which is the square root of the largest eigenvalue of $A^\top A$

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)} \tag{2.16}$$

The Frobenius norm is an extension of the vector $2-$norm and is often easier to compute than induced norms. This norm can be defined by

$$\|A\|_F = \sqrt{\mathrm{tr}(A^\top A)} \tag{2.17}$$

where the $\mathrm{tr}(\cdot)$ is the sum of diagonal entries.

A. Important Properties of Norms

In the control design and analysis, certain properties regarding norms for vectors and matrices are useful. Below are some important properties that will be applied in the subsequent chapters:

- **Non-negativity:** The norm of any vector or matrix is always non-negative and equals zero only when the vector and matrix itself are zero. This ensures that the norm serves as an appropriate measure of signal magnitude.

- **Norm Equivalence:** All p-norms are equivalent in the sense that there exist positive constants c_1 and c_2 such that

  $$c_1\|x\|_{p_1} \leqslant \|x\|_{p_2} \leqslant c_2\|x\|_{p_1} \tag{2.18}$$

  for all $x \in \mathbb{R}^n$. In particular

  $$\|x\|_\infty \leqslant \|x\|_2 \leqslant \|x\|_1 \leqslant \sqrt{n}\|x\|_2 \leqslant n\|x\|_\infty \tag{2.19}$$

- **Subadditivity/Triangle inequality:** As one of the fundamental properties

of norms, the triangle inequality states that for any vectors x and y, the norm of the sum of these vectors is less than or equal to the sum of their individual norms. Mathematically, it can be expressed as

$$\|x + y\|_p \leqslant \|x\|_p + \|y\|_p \tag{2.20}$$

This property is useful in evaluating system stability and designing robust control strategies, as it can be used to scale the combined effect of multiple signals or disturbances.

- **Homogeneity:** This property signifies that scaling the input vector by a scalar factor also scales the norm of the vector by the same factor. For any vector x and a scalar α, homogeneity is

$$\|\alpha x\|_p = |\alpha| \|x\|_p \tag{2.21}$$

Homogeneity helps understand how the system responds to changes in input magnitudes and also helps tune control parameters accordingly.

- **Matrix 2-norm inequality:** It states that for any matrix A and any vector x, the 2-norm of the matrix-vector product Ax is bounded by the product of the Frobenius norm of A and the 2-norm of x:

$$\|Ax\|_2 \leqslant \|A\|_F \|x\|_2 \tag{2.22}$$

This inequality relates the vector 2-norm to the Frobenius matrix norm.

- **Submultiplicativity:** This property asserts that the norm of the product of two matrices is bounded by the product of their individual norms. For matrices A and B, their norms satisfies

$$\|AB\| \leqslant \|A\| \|B\| \tag{2.23}$$

It's worth noting that not all norms satisfy the submultiplicative property. However, many commonly used norms, such as the induced matrix norms and the Frobenius norm, do satisfy this property. This property of matrix norms is also useful in analyzing the effects of control on system performance.

2.2.2 Properties of Matrices

In the subsequent control design and analysis, some useful facts regarding real matrices are going to be used. They are now summarized below:

- **Positive (semi-)definiteness:** A square symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive (semi-)definite (denoted by $A \succ 0 (\succeq 0)$), if and only if all the eigenvalues of A are positive(nonnegative) . If A is a diagonal matrix and all the diagonal elements are positive(nonnegative), then A is positive (semi-)definite.

- **Principal minors criterion:** A square symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive definite, if and only if all its principal minors are strictly positive.

- **Quadratic form inequality:** For any vector $x \in \mathbb{R}^n$, the following inequality holds:

$$\lambda_{\min}(Q)\|x\|^2 \leqslant x^\top Q x \leqslant \lambda_{\max}(Q)\|x\|^2 \tag{2.24}$$

  where $Q \in \mathbb{R}^{n\times n}$ is a symmetric Quadratic positive semi-definite matrix, and $\lambda_{\min}(Q)$ and $\lambda_{\max}(Q)$ are the smallest and largest eigenvalues of Q.

2.2.3 Lipschitz Continuity

Functions are mathematical representations used to model relationships between signals, such as inputs and outputs. Lipschitz continuity is a strong form of function continuity. It is useful in characterizing the behaviors of nonlinear functions, as it can limit how fast the function can change. The definition of Lipschitz continuity is as follows

Definition 2.2. *(**Lipschitz continuity**) Let Ω_x be a subset of $\mathbb{R}^n$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be locally Lipschitz continuous if there exists a positive constant L such that the following inequality holds*

$$|f(x_a) - f(x_b)| \leqslant L\|x_a - x_b\| \tag{2.25}$$

for all $x_a, x_b \in \Omega_x$.

From the definition, it can be concluded that the function is Lipschitz continuous when the magnitude of the slope of the line segment connecting any two points $f(x_a)$

and $f(x_b)$ on the function is limited by a maximum value. This boundedness criterion ensures that the function does not exhibit excessively steep or uncontrolled variations between any pair of points. Furthermore, the Lipschitz constant, which quantifies the degree of this continuity, is defined as the maximum slope of the function in Ω_x.

2.2.4 Stability Theorem

Stability is the primary requisite for all control systems. Among various stability analysis approaches, Lyapunov stability is widely employed, and we utilize it extensively throughout this dissertation. Next, we are going to provide a summary of the stability concepts and theorems.

A. Definitions of Stability

We start by giving the definitions of Lyapunov stability and asymptotic stability. For a nonlinear system described by

$$\dot{x} = f(x, t) \tag{2.26}$$

where $x \in \mathbb{R}^n$, $f(\cdot) : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^n$ is locally Lipschitz continuous in x and piecewise continuous in t. When studying the system stability, we are actually examining the properties of its equilibrium point, defined as

Definition 2.3. *A state x^* is an equilibrium point of the system (2.26), if it satisfies*

$$f(x^*, t) = 0 \tag{2.27}$$

for all $t \geqslant t_0$. This ensures that once the system state reaches x^*, it will remain at x^* forever.

For convenience, we present all definitions and theorems for the case when the equilibrium point is the origin, that is, $x^* = 0$, without loss of generality.

Definition 2.4. *The equilibrium point of the system $x^* = 0$ is said to be*

- ***stable**, if for any given $R > 0$ there exists $r(R, t_0) > 0$ such that*

$$\|x(t_0)\| < r(R, t_0) \Rightarrow \|x(t)\| < R \tag{2.28}$$

  for all $t \geqslant t_0$.

- ***asymptotically stable***, *if it is stable at* t_0 *and there exists* $r(t_0) > 0$ *such that*

$$\|x(t_0)\| < r(t_0) \Rightarrow x(t) \to 0 \text{ as } t \to \infty \tag{2.29}$$

- ***globally asymptotically stable***, *if it is stable at* t_0 *and there exists* $r(t_0) > 0$ *such that*

$$\|x(t_0)\| < \delta(t_0) \Rightarrow x(t) \to 0 \text{ as } t \to \infty \tag{2.30}$$

  for all $\delta(t_0) > 0$.

- ***unstable***, *if it is not stable at* t_0.

When the constant $r(R, t_0)$ *in (2.28),* $r(t_0)$ *in (2.29) , or* $\delta(t_0)$ *in (2.30) is independent of* t_0, *the corresponding stability property can be regard to hold* ***uniformly***.

B. Lyapunov's Stability Theorem

Having defined the stability and asymptotic stability of equilibrium points, the next task is to find ways to determine stability. Lyapunov's stability theorem is one of the most effective methods, as it can determine system stability directly from the relationships between $f(x,t)$ and a positive definite function $V(x,t)$ without the need to solve complicated nonlinear systems analytically. Let us introduce the Lyapunov's direct theorem:

Theorem 2.1. *(**Lyapunov's direct theorem*** [139]*) Consider the system (2.26) with the equilibrium point being* $x^* = 0$. *Let* $V(x,t) : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^+$ *be a continuously differentiable, positive definite function. There exist two class* $\mathcal{K}$ *functions* $\underline{\alpha}$ *and* $\overline{\alpha}$ *such that* $V(x,t)$ *satisfies*

$$\underline{\alpha}(\|x\|) \leqslant V(x,t) \leqslant \overline{\alpha}(\|x\|) \tag{2.31}$$

If the derivative of $V(x,t)$ *along the solutions of the system (2.26) satisfies*

$$\dot{V}(x,t) = \frac{\partial V(x,t)}{\partial t} + \frac{\partial V(x,t)}{\partial x} f(x,t) \leqslant 0 \tag{2.32}$$

for all $t \geqslant t_0$ *and for all* $x \in \Omega_x \subset \mathbb{R}^n$, *then the system of (2.26) is stable. If (2.32)*

can be written as

$$\dot{V}(x,t) \leqslant -\alpha(\|x\|) \tag{2.33}$$

where α is a class $\mathcal{K}$ function, then the system (2.26) is asymptotically stable. Moreover, if Ω_x is expanded to the whole space $\mathbb{R}^n$, then the above conclusions hold globally. Furthermore, if $\underline{\alpha}$ and $\overline{\alpha}$ are class $\mathcal{K}_\infty$ functions, then the stability properties hold uniformly.

C. Barbalat's Lemma

In adaptive control problems, it is often the case that only the negative semi-definite condition like (2.32) can be satisfied. To obtain asymptotic stability, the following Barbalat's lemma is useful.

Lemma 2.2. *(**Barbalat's lemma** [130]) Let $\phi(t) : [t_0, \infty) \to \mathbb{R}$ be a uniformly continuous function of t. If $\lim_{t\to\infty} \int_{t_0}^{t} \phi(t)\mathrm{d}\tau$ exists and is finite, then $\phi(t) \to 0$ as $t \to 0$.*

2.2.5 Sliding Surface

For higher-order systems, the concept of sliding surfaces can be applied. For a tracking error $e \in \mathbb{R}^n$, a filtered tracking error s_e can be defined as

$$s_e = \lambda_0 e + \lambda_1 \dot{e} + \cdots + \lambda_{r-2} e^{(r-2)} + e^{(r-1)} = \sum_{i=0}^{r-2} \lambda_i e^{(i)} + e^{(r-1)} \tag{2.34}$$

where λ_0, λ_1, $\cdots$, λ_{r-2} are positive constants chosen such that the polynomial $s^{(r-1)} + \lambda_{r-2}s^{(r-2)} + \cdots + \lambda_1 s + \lambda_0$ is Hurwitz. With this definition, the tracking error e can be expressed as $e = H(s)s_e$ with $H(s) = 1/\left(s^{(r-1)} + \lambda_{r-2}s^{(r-2)} + \cdots + \lambda_1 s + \lambda_0\right)$ being a proper stable transfer function, which implies that $e(t) \to 0$ as $s_e \to 0$. Therefore, we have the following proposition:

Proposition 2.1. [138] *With the definition in (2.34), $s_e = 0$ defines a time-varying sliding surface in $\mathbb{R}^n$ on which the tracking error e asymptotically converges to zero.*

In (2.34), if λ_0, λ_1, $\cdots$, λ_{r-2} are chosen such that $\lambda_0 = \lambda^{(r-1)}$, $\lambda_1 = (r-1)\lambda^{(r-2)}$, $\cdots$, $\lambda_{r-2} = (r-1)\lambda$ with $\lambda > 0$, then the filtered tracking error s_e becomes

$$s_e = \left(\frac{\partial}{\partial t} + \lambda\right)^{r-1} e \tag{2.35}$$

and the following additional proposition holds

Proposition 2.2. [138] *With the definition in (2.35), we have the following properties*

(i) If s_e is bounded by $\|s_e\| \leqslant \rho_s$ with $\rho_s > 0$ and $e(0) \in \Omega_e$, then $e(t) \in \Omega_e$, in which

$$\Omega_e = \left\{ e(t) | \|e(t)\| \leqslant \frac{2}{\lambda^{(r-1)}} \rho_s \right\} \tag{2.36}$$

(ii) If $\|s_e\| \leqslant \rho_s$ and $e(0) \notin \Omega_e$, then $e(t)$ will converge to Ω_e with a time-constant $r - 1/\lambda$.

2.2.6 Graph Theory

Graph theory provides a rich mathematical framework essential for analyzing the structural and dynamic properties of multi-agent systems. In this dissertation, we particularly focus on the control problem of MAS over directed graphs. Therefore, in this subsection, some important algebraic graph theory and matrix properties associated with directed graphs are introduced.

Lemma 2.3. [140, 141] *Let $\mathcal{L}$ be the nonsymmetric Laplacian matrix associated with the directed graph $\mathcal{G}$ of order M. Then, $\mathcal{L}$ has at least one zero eigenvalue and all its nonzero eigenvalues have positive real parts. Furthermore, $\mathcal{L}$ has a simple zero eigenvalue and all other eigenvalues have positive real parts if and only if $\mathcal{G}$ has a directed spanning tree, respectively.*

Lemma 2.4. [141] *Suppose that $z = [z_1^\top \ z_2^\top \ \cdots \ z_M^\top]^\top$ with $z_i \in \mathcal{R}^n$. Let $\mathcal{L}$ be the nonsymmetric Laplacian matrix associated with the directed graph $\mathcal{G}$. Then the following five conditions are equivalent:*

(i) Node k is the only node that has directed paths to all other nodes in $\mathcal{G}$;

(ii) The directed graph $\mathcal{G}$ has a directed spanning tree and node k has no neighbor;

(iii) The directed graph $\mathcal{G}$ has a directed spanning tree and every entry of the kth row of $\mathcal{L}$ is zero;

(iv) Consensus is reached for the closed-loop system $\dot{z} = -(\mathcal{L} \otimes I_n)z$. In particular, for all $z_i(0)$ and all $i = 1, 2, \cdots, M$, $z_i(t) \to z_k(0)$ as $t \to \infty$.

Lemma 2.5. [132] *Let $\mathcal{G}$ be the directed graph for M followers, labeled as agents or followers 1 to M. Let $\mathcal{L}$ be the nonsymmetric Laplacian matrix associated with the directed graph $\mathcal{G}$. Suppose that in addition to the M followers, there exists a leader, labeled as agent 0, whose connection to the M followers can be described by a pinning matrix $\mathcal{B} = \text{diag}(b_1, b_2, \cdots, b_M)$, where $b_i > 0$ if the ith follower can receive information from the leader and $b_i = 0$ otherwise. Let $\mathcal{L}_\mathcal{B} = \mathcal{L} + \mathcal{B}$. Then, all eigenvalues of $\mathcal{L}_\mathcal{B}$ have positive real parts if and only if in the directed graph $\mathcal{G}$ the leader has directed paths to all followers.*

Chapter 3

Fault-tolerant Adaptive Lyapunov-based Model Predictive Control for Nonlinear Systems with Actuator Faults and Input Constraints

Fault-tolerant model predictive control (FTMPC) provides an optimal solution for addressing the control challenges posed by multi-variable nonlinear dynamics, particularly in the presence of input constraints and stringent safety requirements, such as in UAV flight control. This chapter proposes a novel adaptive FTMPC design that integrates adaptive parameter estimation with the Lyapunov-based MPC framework. The proposed adaptive method is less conservative with reduced computational complexity, aiming to ensure the closed-loop performance of nonlinear systems despite unexpected actuator faults and input constraints.

3.1 Introduction

As previously discussed in Section 1.2.4, FTMPC can be classified into two categories: passive FTMPC, which utilizes the inherent robustness of MPC [31, 87, 88] or the self-reconfiguration capability of MPC [85, 142], and active FTMPC, which adjusts control policies [98, 105, 108, 114] or modifies control configurations [117, 121, 124]

in response to real-time fault detection [117, 122] or online fault parameter estimation [98, 105, 108, 114, 121]. The limitation of passive strategies is that they cannot fully compensate for fault effects due to the inevitable discrepancies between the real dynamics and the predictive model. In contrast, active FTMPC proactively adjusts to faults, thereby enhancing system resilience and enabling it to handle a broader range of faults. However, most existing active FTMPC methods lack rigorous theoretical guarantees for closed-loop performance and stability. These methods often lack formal stability proofs, or they can only assure stability under ideal conditions—when accurate fault information is presumed to be available or can be fully estimated.

Therefore, developing novel active FTMPC strategies that ensure closed-loop performance, reduce conservatism, and mitigate computational complexity holds both theoretical significance and practical interest. Since common types of faults, such as actuator faults, can typically be modeled as uncertain parameters in control channels, adaptive estimation—an effective solution for addressing parametric uncertainties—can be integrated into the MPC framework. This introduction of adaptive strategies allows predictions to be processed with estimated fault parameters, thereby enabling the system's accommodation to unknown faults. Significant research contributions in adaptive MPC include the development of strategies for estimating constant parametric uncertainties [143], employing persistent excitation [144], and applying adaptive MPC to single-loop systems [145]. Additional research has also explored the use of comparison models [146], adaptive MPC in constrained discrete-time linear systems [147], and its extension to continuous nonlinear systems [148].

In this chapter, we propose a novel adaptive FTMPC for a class of nonlinear systems with input constraints and unexpected actuator faults. The Lyapunov-based MPC framework is employed with the incorporation of an adaptive fault parameter estimator. Both the feasibility of the MPC optimization problem and the stability of the closed-loop control system are rigorously evaluated. The main contributions of this chapter include:

- Compared to existing adaptive MPC algorithms [143–148], this chapter presents a simpler formulation of adaptive MPC. By employing a Lyapunov-based contractive stability constraint, our approach ensures closed-loop stability without the need for terminal constraints and terminal costs, which are challenging to design for nonlinear dynamics. This flexibility also permits the selection of shorter prediction horizons, thereby reducing computational complexity.

- Compared with the result in [149], which also explores adaptive Lyapunov-based MPC, our proposed design can handle both multiplicative and additive uncertainties caused by the actuator faults. Additionally, we consider hard input constraints and rigorously prove the feasibility of MPC's constrained optimization problem.

- The discrepancy between the discrete updates of control actions and the continuous evolution of plant dynamics can lead to unexpected degradation in closed-loop control performance. By utilizing Lyapunov stability theory and Lipschitz continuity, we quantify such discretization error during stability analysis. This provides sufficient stability conditions, offering guidelines for choosing appropriate control parameters based on the given sampling frequency.

The remainder of this chapter is organized as follows: Section 3.2 mathematically formulates the control problem and objectives; Section 3.3 details the FTMPC design; Section 3.4 provides a closed-loop stability analysis of the proposed control system; Section 3.5 offers two simulation examples to validate the proposed design; Finally, Section 3.6 presents the conclusions of this chapter.

3.2 Problem Formulation

Before presenting the main result of this chapter, we first formulate the control problem addressed. This includes providing mathematical descriptions of the system under study, the input constraint, and the actuator fault. Subsequently, the control objective of this study is formulated mathematically.

3.2.1 System Description

In this chapter, let us consider a typical high-order Multi-Input Multi-Output (MIMO) nonlinear system in strict-feedback canonical form as

$$\begin{cases} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{r-1} &= x_r \\ \dot{x}_r &= f(x) + G(x)u \\ y &= x_1 \end{cases} \tag{3.1}$$

where $x = \left[x_1^\top \; x_2^\top \; \cdots \; x_r^\top\right]^\top$ is the system state vector with $x_i = y^{(i-1)} \in \mathbb{R}^n$ for $i = 1, 2, \cdots, r$; $u = [u_1 \; u_2 \; \cdots \; u_n]^\top \in \mathbb{R}^n$ is the control input; $y = [y_1 \; y_2 \; \cdots \; y_n]^\top \in \mathbb{R}^n$ is the output; $f(x) = [f_1(x) \; f_2(x) \; \cdots \; f_n(x)]^\top : \mathbb{R}^{rn} \to \mathbb{R}^n$ is a vector function; $G(x) = [g_1(x) \; g_2(x) \; \cdots \; g_n(x)] : \mathbb{R}^{rn} \to \mathbb{R}^{n \times n}$ is a square matrix function with $g_i(x) : \mathbb{R}^{rn} \to \mathbb{R}^n$ for $i = 1, 2, \cdots, n$. The following assumption is necessary and common:

Assumption 3.1. *All entries of $f(x)$ and $G(x)$ are sufficiently smooth and locally Lipschitz functions of x satisfying $f(0) = 0$ and $G(0) \neq 0$.*

3.2.2 Input Constraint and Actuator Fault

Given the constraints on input, the control input of (3.1) is confined to a nonempty compact convex set, defined as:

$$u \in \Omega_u \triangleq \{u \in \mathbb{R}^n | \; -\underline{u} \leqslant u \leqslant \overline{u}\} \tag{3.2}$$

where $\underline{u} = [\underline{u}_1 \; \underline{u}_2 \; \cdots \; \underline{u}_n]^\top \in \mathbb{R}^{n+}$ and $\overline{u} = [\overline{u}_1 \; \overline{u}_2 \; \cdots \; \overline{u}_n]^\top \in \mathbb{R}^{n+}$.

Moreover, we consider the potential occurrence of unexpected actuator faults affecting one or more control channels of (3.1). By recalling the modeling of actuator fault (2.8) in Chapter 2, the original plant model (3.1) can be modified as

$$\begin{cases} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{r-1} &= x_r \\ \dot{x}_r &= f(x) + G(x)\Theta^{-1}u_c + G(x)\vartheta^0 \\ y &= x_1 \end{cases} \tag{3.3}$$

where $u_c = [u_{c_1} \; u_{c_2} \; \cdots \; u_{c_n}]^\top$ is the control command to be designed later; $\Theta = \text{diag}(\vartheta_1, \vartheta_2, \cdots, \vartheta_n) = \Theta_f^{-1}$ is a diagonal matrix of containing the multiplicative uncertainties and $\vartheta^0 = [\vartheta_1^0 \; \vartheta_2^0 \; \cdots \; \vartheta_n^0]^\top = \vartheta_f^0$ is a vector of the additive uncertainties.

By this means of actuator fault modeling, both incipient and abrupt faults can be described. Considering there is no additional fault detection mechanism in the control system, the real-time values of these fault parameters remain unknown. In this work, just like many works in literature, only bounded and slow time-varying changes in actuator signals are considered, ensuring that actuators will not suffer a complete failure in the event of a fault. As such, the following assumption holds:

Assumption 3.2. *It is assumed that the uncertain parameters ϑ_i, $i = 1, \cdots, n$ are bounded within the interval $\vartheta_i \in [\frac{1}{\overline{\vartheta}_i}, \overline{\vartheta}_i]$ with $\overline{\vartheta}_i > 1$. Furthermore, ϑ_i^0 is assumed to be bounded in its absolute value by $|\vartheta_i^0| \leqslant \overline{\vartheta}_i^0$ with $\overline{\vartheta}_i^0 \in \mathbb{R}^+$ being the maximum acceptable magnitude. Additionally, the changes of ϑ_i^0 and ϑ_i^0 are slowly time-varying such that their derivatives can be regarded as 0 in the closed-loop analysis without losing much of the analytical accuracy.*

To guarantee the satisfaction of the input constraint (3.2) in the presence of actuator faults, the control command u_c needs to be restricted as follows

$$u_c \in \Omega_{u_c} \triangleq \{u_c \in \mathbb{R}^n | -\underline{u}_c \leqslant u_c \leqslant \overline{u}_c, i = 1, 2, \cdots, n\} \tag{3.4}$$

where $\underline{u}_c = \overline{\Theta}^{-1}\left(\underline{u} - \overline{\vartheta}^0\right)$ and $\overline{u}_c = \overline{\Theta}^{-1}\left(\overline{u} - \overline{\vartheta}^0\right)$ with $\overline{\Theta} = \text{diag}(\overline{\vartheta}_1, \overline{\vartheta}_2, \cdots, \overline{\vartheta}_n)$ and $\overline{\vartheta}^0 = [\overline{\vartheta}_1^0 \ \overline{\vartheta}_2^0 \ \cdots \ \overline{\vartheta}_n^0]^\top \in \mathbb{R}^n$. Ω_{u_c} is a tightened nonempty convex set of Ω_u

3.2.3 Control Objective

The control objective of this work is to design a fault-tolerant optimal controller u_c for the nonlinear system (3.3), such that the output y tracks a desired trajectory denoted as $y^d \in \mathbb{R}^n$ and all the signals of the closed-loop system remain bounded. In addition, the tracking performance and the closed-loop stability are maintained in the presence of the input constraint (3.2) and the actuator fault modeled in (2.8).

Assumption 3.3. *The prespecified desired trajectory y^d exhibits slow time variation with respect to the plant system. It is piece-wise continuous and synchronized with the control update frequency. Additionally, It is assumed that y^d is at least r-times differentiable with respect to time, with its time derivatives denoted as $\dot{y}^d, \ddot{y}^d, \cdots, y^{d(r)}$.*

To quantify the trajectory tracking performance, we define the output tracking error as

$$e = y - y^d \tag{3.5}$$

Then, a filtered error in sliding surface form can be defined:

$$s = \lambda_0 e + \lambda_1 \dot{e} + \cdots + \lambda_{r-2} e^{(r-2)} + e^{(r-1)} = \sum_{i=0}^{r-2} \lambda_i e^{(i)} + e^{(r-1)} \tag{3.6}$$

where $\lambda_0, \lambda_1, \cdots, \lambda_{r-2} \in \mathbb{R}^+$ define the bandwidth of the error dynamics, whose values are appropriately chosen such that $s^{(n-1)} + \lambda_{r-2}s^{(n-2)} + \cdots + \lambda_1 s + \lambda_0$ is a Hurwitz polynomial.

Referring back to Proposition 2.1 given in Chapter 2, $s = 0$ serves as a sliding surface for output tracking. Therefore, the tracking control objective can be more precisely defined as regulation and maintenance of the filtered error s in close proximity to the origin, 0. Taking into account the restricted control efforts as well, the cost function for output trajectory tracking control can be formulated in a quadratic form:

$$L(s, u_c) = \|s\|_Q^2 + \|u_c\|_R^2 \tag{3.7}$$

where Q and R are $n \times n$ positive-definite and symmetric weighting matrices, whose magnitudes correspond to the penalization of the deviation of output from the desired trajectory and the actuator's energy consumption, respectively.

Finally, the control objective of this chapter can be formulated. It aims to determine an optimal output tracking control command u_c by minimizing the cost function (3.7) subject to the input constraint while ensuring the closed-loop stability with unexpected actuator faults.

3.3 Control Design

In this section, the core contribution of our work-an adaptive MPC framework designed for fault-tolerant tracking control-is presented. Initially, we derive an auxiliary controller tailored for the nominal scenario. Following this, utilizing the Lyapunov stability characteristic inherent in the auxiliary control scheme, we proceed to construct an adaptive fault-tolerant controller by combining adaptive parameter estimation with Lyapunov-based model predictive control. The developed control strategy aims to refine the performance metrics, as encapsulated in the cost function (3.7), while simultaneously guaranteeing the stability of the closed-loop control system under actuator faults.

Figure 3.1 illustrates the structure of the developed control system. The diagram also reveals that the model predictive controller updates the control commands discretely, implementing a sample-and-hold manner, whereas the plant system evolves in a continuous time frame. To elucidate the sampled-data characteristic of the

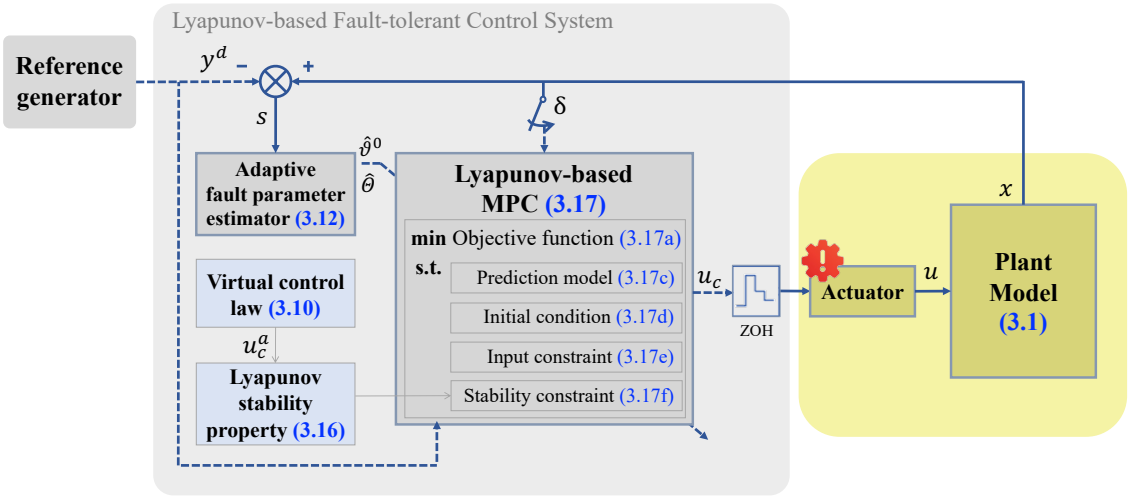

Figure 3.1: Block diagram of the Lyapunov-based fault-tolerant control system

MPC-based control system, we can denote the sequence of sampling instants as $\{t_k = \delta k | k \in \{0, 1, 2, \cdots\}, \delta \in \mathbb{R}^+\}$, where t_k denotes the specific instants in time at which the system's stats are sampled and the control decisions are updated, and δ represents the fixed sampling interval.

3.3.1 Auxiliary Control Law

In order to establish the stability constraint for the model predictive control problem, the first step is to find ways to define the stability criteria of the closed-loop control system. Here, we employ Lyapunov's direct method (Theorem 2.1) and focus on the nominal case, where the parameter uncertainties and the input constraint are disregarded. First of all, the following positive-definite function can be chosen as a Lyapunov function candidate:

$$V_s = \frac{1}{2} \|s\|^2 \tag{3.8}$$

Differentiating it and recalling the system model (3.3) yield

$$\begin{aligned}\dot{V}_s &= s^\top \dot{s} \\ &= s^\top \left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)} + e^{(r)} \right) \\ &= s^\top \left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)} + f(x) + G(x)\Theta^{-1} u_c + G(x)\vartheta^0 - y^{d(r)} \right)\end{aligned} \tag{3.9}$$

We need to find an auxiliary stabilizing control law that ensures the derivative of V_s satisfies the condition (2.33) as given in Theorem 2.1. This can be accomplished by designing an analytical control law using feedback linearization as follows:

$$u_c^a = \Theta G^{-1}(x) v_c^a \tag{3.10a}$$

$$v_c^a = -cs - \sum_{i=0}^{r-2} \lambda_i e^{(i+1)} - f(x) - G(x)\vartheta^0 + y^{d(r)} \tag{3.10b}$$

where the control gain $c \in \mathbb{R}^+$ is a strictly positive constant. By substituting this auxiliary controller u_c^a to (3.9), it has that

$$\dot{V}_s = -c\|s\|^2 \tag{3.11}$$

which aligns with the asymptotic stability condition delineated in (2.33), with the class $\mathcal{K}$ function spas $c|s|^2$. This analysis confirms that the auxiliary control law (3.10) delivers a stabilizing solution for the closed-loop tracking control system under nominal conditions.

3.3.2 Lyapunov-based Fault-tolerant MPC

After establishing the Lyapunov stability property through the derivation of an auxiliary controller, we now proceed to develop the Lyapunov-based fault-tolerant MPC framework. This framework includes an adaptive parameter estimator designed for fault accommodation and a Lyapunov-based MPC controller for online control optimization. Together, this integrated framework can determine optimized and stabilizing control actions in real-time, while simultaneously ensuring fault tolerance against the unknown actuator fault. The development details are presented below.

A. Adaptive Parameter Estimator

For handling the parameter uncertainty, a parameter estimator is required to integrate with the controller. The estimate of the multiplicative uncertain parameter Θ is denoted as $\hat{\Theta} = \text{diag}(\hat{\vartheta}_1, \hat{\vartheta}_2, \cdots, \hat{\vartheta}_n)$, and the estimate of ϑ^0 is denoted as $\hat{\vartheta}^0 = \left[\hat{\vartheta}_1^0, \hat{\vartheta}_2^0, \cdots, \hat{\vartheta}_n^0\right]^\top$. The adaptive laws for updating $\hat{\vartheta}_i, i = 1, 2, \cdots, n$ and $\hat{\vartheta}^0$ are designed as

$$\dot{\hat{\vartheta}}_i = -k_i \hat{\vartheta}_i^{-1} s^\top g_i(x) u_{c_i}, \ i = 1, 2, \cdots, n \tag{3.12a}$$

$$\dot{\hat{\vartheta}}^0 = k^0 G^\top(x) s \tag{3.12b}$$

where k_i, $i = 1, \cdots, n$, and k^0 are user-define positive constants.

This adaptive parameter estimator is also derived based on the Lyapunov stability theory. The Lyapunov function employed is:

$$V = V_s + V_\vartheta \tag{3.13}$$

where V_s has been given in (3.8) and

$$V_\vartheta = \frac{1}{2k^0} \left\| \tilde{\vartheta}^0 \right\|^2 + \sum_{i=1}^{n} \frac{1}{2k_i \vartheta_i} \tilde{\vartheta}_i^2 \tag{3.14}$$

Here, $\tilde{\vartheta}^0$ and $\tilde{\vartheta}_i$ are the estimation errors and defined as $\tilde{\vartheta}^0 = \left[\tilde{\vartheta}_1^0, \tilde{\vartheta}_2^0, \cdots, \tilde{\vartheta}_n^0\right]^\top = \left[\hat{\vartheta}_1^0 - \vartheta_1^0, \hat{\vartheta}_2^0 - \vartheta_2^0, \cdots, \hat{\vartheta}_n^0 - \vartheta_n^0\right]^\top$ and $\tilde{\Theta} = \text{diag}(\tilde{\vartheta}_1, \tilde{\vartheta}_2, \cdots, \tilde{\vartheta}_n)$ with $\tilde{\vartheta}_i = \hat{\vartheta}_i - \vartheta_i$, $i = 1, 2, \cdots, n$.

Differentiating V along the closed-loop dynamics and substituting the adaptive laws (3.12) yields

$$\begin{aligned}
\dot{V} =& s^\top \left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)} + f(x) + \sum_{i=1}^{n} \vartheta_i^{-1} g_i(x) u_{c_i} + G(x)\vartheta^0 - y^{d(r)} \right) + \frac{1}{k^0} \dot{\hat{\vartheta}}^{0\top} \tilde{\vartheta}^0 + \sum_{i=1}^{n} \frac{1}{k_i \vartheta_i} \tilde{\vartheta}_i \dot{\hat{\vartheta}}_i \\
=& s^\top \left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)} + f(x) + \sum_{i=1}^{n} \frac{\hat{\vartheta}_i}{\vartheta_i} \hat{\vartheta}_i^{-1} g_i(x) u_{c_i} + G(x)\vartheta^0 - y^{d(r)} \right) + s^\top G(x) \tilde{\vartheta}^0 \\
& - \sum_{i=1}^{n} \frac{\tilde{\vartheta}_i}{\vartheta_i} \hat{\vartheta}_i^\top s^\top g_i(x) u_{c_i} \\
=& s^\top \left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)} + f(x) + G(x)\hat{\Theta}^{-1} u_c + G(x)\hat{\vartheta}^0 - y^{d(r)} \right)
\end{aligned} \tag{3.15}$$

By incorporating the auxiliary control law (3.10) with Θ and ϑ^0 replaced by their estimates $\hat{\Theta}$ and $\hat{\vartheta}^0$ into the right hand side of (3.15), we obtain the following equation

$$s^\top \left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)} + f(x) + G(x)\hat{\Theta}^{-1} u_c^a + G(x)\hat{\vartheta}^0 - y^{d(r)} \right) = -c\|s\|^2 \tag{3.16}$$

This equation reveals that the auxiliary control law, in conjunction with the adaptive estimator, ensures robust stability and convergence of the sliding mode error s despite fault-induced uncertainties. The term $-c\|s\|^2$ explicitly formulates the desired decay rate for an appropriate chosen Lyapunov function V, which is crucial for guaranteeing the attenuation of errors. This formulation forms a critical basis for the development of the upcoming Lyapunov-based MPC design.

B. MPC Problem Formulation

We then proceed to the MPC problem formulation, which constitutes a finite-horizon constrained optimization problem with a forward-looking prediction horizon $T > \delta$. At the time instant t_k, the predictive controller receives the current measurements of $x(t_k)$, the desired output trajectory $y^d(t)$ and its derivatives for $t \in [t_k, t_k+T]$, and the current estimated parameters $\hat{\Theta}(t_k)$ and $\hat{\vartheta}^0(t_k)$. Using this information as the initial conditions, we can formulate a finite-horizon optimal control problem as follows

$$\min_{u_c^p(t|t_k)} \int_{t_k}^{t_k+T} \left(\|s^p(t|t_k)\|_Q^2 + \|u_c^p(t|t_k)\|_R^2 \right) \mathrm{d}t \tag{3.17a}$$

with

$$s^p(t|t_k) = \sum_{i=0}^{r-2} \lambda_i \left(x_{i+1}^p(t|t_k) - y^{d(i)}(t) \right) + \left(x_r^p(t|t_k) - y^{d(r-1)}(t) \right) \tag{3.17b}$$

subject to

$$\begin{cases} \dot{x}_1^p(t|t_k) = x_2^p(t|t_k) \\ \qquad\qquad \vdots \\ \dot{x}_{r-1}^p(t|t_k) = x_r^p(t|t_k) \\ \dot{x}_r^p(t|t_k) = f(x^p(t|t_k)) + G(x^p(t|t_k))\left(\hat{\Theta}^{-1}(t_k)u^p(t|t_k) + \hat{\vartheta}^0(t_k)\right) \end{cases} \tag{3.17c}$$

$$x^p(t_k|t_k) = x(t_k) \tag{3.17d}$$

$$u_c^p(t|t_k) \in \Omega_{u^c} \tag{3.17e}$$

$$s^{p\top}(t_k|t_k)\Bigg(\sum_{i=0}^{r-2}\lambda_i\Big(x_{i+2}^p(t_k|t_k) - y^{d(i+1)}(t_k)\Big) + f(x^p(t_k|t_k)) + G(x^p(t_k|t_k))\Big(\hat{\Theta}^{-1}(t_k)u_c^p(t_k|t_k) + \hat{\vartheta}^0(t_k)\Big) - y^{d(r)}(t_k)\Bigg) \leqslant -c\,\|s^p(t_k|t_k)\|^2 \tag{3.17f}$$

where the internal variables are denoted by a superscript p to distinguish them from the actual system signals. In the optimization problem (3.17), constraint (3.17c) serves as the prediction model to predict the system's future evolution with $\hat{\Theta}$ and $\hat{\vartheta}^0$ estimated by (3.12). Constraint (3.17d) specifies the initial conditions of the prediction model. Compliance with the input constraint is ensured by (3.17e). The Lyapunov-based stability constraint (3.17f), whose construction is inspired by (3.11), is designed to enforce the decay of the Lyapunov function at a rate that is no slower than the decay achieved by the auxiliary controller u_c^a.

Before moving forward, it is necessary to demonstrate the feasibility of the optimization problem (3.17). Feasibility in this context implies that there exists at least one feasible solution such that the input constraint (3.17e) and the stability constraint (3.17f) are both satisfied. In the following lemma, we present a feasible solution:

Lemma 3.1. *There always exists a feasible solution to the optimization problem (3.17), constructed as*

$$u_c^0(t|t_k) = \text{sat}\left(v_c(t|t_k), \underline{u}_c, \overline{u}_c\right) \tag{3.18a}$$

$$v_c(t|t_k) = \hat{\Theta}(t_k)G^{-1}(x^p(t|t_k))\left(v_c^a(t|t_k) + v_c^d(t|t_k)\right) \tag{3.18b}$$

$$v_c^a(t|t_k) = -cs^p(t|t_k) - \sum_{i=0}^{r-2}\lambda_i\Big(x_{(i+2)}^p(t_k|t_k) - y^{d(i+1)}(t_k)\Big) - f(x^p(t|t_k)) - G(x^p(t|t_k))\hat{\vartheta}^0(t_k) + y^{d(r)}(t) \tag{3.18c}$$

$$v_c^d(t|t_k) = -k_s\underline{\chi}^{-1}\text{sgn}\left(s^p(t|t_k)\right) \tag{3.18d}$$

for $t \in [t_k, t_k + T]$. *In (3.18a),* $\text{sat}\left(v_c(t), \underline{u}_c, \overline{u}_c\right)$ *is a saturation function written as*

$$\text{sat}\left(v_c(t), \underline{u}_c, \overline{u}_c\right) = \chi\left(v_c(t), \underline{u}_c, \overline{u}_c\right) v_c(t) \tag{3.19}$$

with

$$\chi\left(v_c(t), \underline{u}_c, \overline{u}_c\right) = \begin{bmatrix} \chi_1\left(v_{c_1}(t), \underline{u}_{c_1}, \overline{u}_{c_1}\right) & 0 & \cdots & 0 \\ 0 & \chi_2\left(v_{c_2}(t), \underline{u}_{c_2}, \overline{u}_{c_2}\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \chi_n\left(v_{c_n}(t), \underline{u}_{c_n}, \overline{u}_{c_n}\right) \end{bmatrix}$$

$$\chi_i\left(v_{c_i}(t), \underline{u}_{c_i}, \overline{u}_{c_i}\right) = \begin{cases} \frac{\overline{u}_{c_i}}{v_{c_i}(t)}, & \text{if } v_{c_i}(t) \geqslant \overline{u}_{c_i} \\ 1, & \text{if } -\underline{u}_{c_i} < v_{c_i}(t) < \overline{u}_{c_i} \\ -\frac{\underline{u}_{c_i}}{v_{c_i}(t)}, & \text{if } v_{c_i}(t) \leqslant -\underline{u}_{c_i} \end{cases}, \quad i = 1, 2 \cdots, n$$

and v_{c_i}, $\underline{u}_{c_i}$ *and* $\overline{u}_{c_i}$ *being the* i*th element of* v_c, $\underline{u}_c$ *and* $\overline{u}_c$. $\chi\left(v_c, \underline{u}_c, \overline{u}_c\right)$ *is the control input saturation degree indicator, and it is clear that all of its diagonal elements vary between* $(0, 1]$. v_c^a *developed in (3.18c) resembles the expression of the auxiliary nominal controller (3.10) with the unknown parameters replaced by their estimates.* v_c^d *in (3.18d) is the discontinuous portion of the feasible input, where* $\underline{\chi} \in \mathbb{R}^{n \times n}$ *is the lower bound of* $\chi\left(v_c, \underline{u}_c, \overline{u}_c\right)$ *satisfying*

$$O_n < \underline{\chi} \leqslant \chi\left(v_c, \underline{u}_c, \overline{u}_c\right) \leqslant I_n \tag{3.20}$$

The user-defined positive parameter k_s *is chosen appropriately such that*

$$\left\| \left(\chi\left(v_c, \underline{u}_c, \overline{u}_c\right) - I_n\right) v_c^a \right\|_1 \leqslant k_s \tag{3.21}$$

Proof. Due to the inclusion of the saturation function, the control profile constructed in (3.18) naturally satisfies the input constraint (3.17e). Substituting $u_c^0(t_k|t_k)$ into

the right half of the inequality in (3.17f) gives

$$
\begin{aligned}
&{s^p}^\top(t_k|t_k)\Bigg(\sum_{i=0}^{r-2}\lambda_i\Big(x^p_{(i+2)}(t_k|t_k)-y^{d(i+1)}(t_k)\Big)+f(x^p(t_k|t_k))+G(x^p(t_k|t_k))\hat{\Theta}^{-1}(t_k)u_c^0(t_k|t_k)\\
&+G(x^p(t_k|t_k))\hat{\vartheta}^0(t_k)-y^{d(r)}(t_k)\Bigg)\\
&={s^p}^\top(t_k|t_k)\Bigg(\sum_{i=0}^{r-2}\lambda_i\Big(x^p_{(i+2)}(t_k|t_k)-y^{d(i+1)}(t_k)\Big)+f(x^p(t_k|t_k))+v_c^a(t_k)+G(x^p(t_k|t_k))\hat{\vartheta}^0(t_k)\\
&\quad-y^{d(r)}(t_k)-k_s\chi\left(v_c(t_k),\underline{u}_c,\overline{u}_c\right)\underline{\chi}^{-1}\mathrm{sgn}(s^p(t_k|t_k))+\left(\chi\left(v_c(t_k),\underline{u}_c,\overline{u}_c\right)-I_n\right)v_c^a(t_k)\Bigg)\\
&=-c\|s^p(t_k|t_k)\|-k_s{s^p}^\top(t_k|t_k)\left(\chi\left(v_c(t_k),\underline{u}_c,\overline{u}_c\right)\underline{\chi}^{-1}-I_n\right)\mathrm{sgn}(s^p(t_k|t_k))\\
&\quad-k_s{s^p}^\top(t_k|t_k)\mathrm{sgn}(s^p(t_k|t_k))+{s^p}^\top(t_k|t_k)\left(\chi\left(v_c(t_k),\underline{u}_c,\overline{u}_c\right)-I_n\right)v_c^a(t_k)
\end{aligned}
\tag{3.22}
$$

Recalling inequality (3.20), it can be obtained that $I_n \leqslant \chi\left(v_c(t_k),\underline{u}_c,\overline{u}_c\right)\underline{\chi}^{-1}$. Further with (3.21), we can have that

$$
\begin{aligned}
&{s^p}^\top(t_k|t_k)\Bigg(\sum_{i=0}^{r-2}\lambda_i\Big(x^p_{(i+2)}(t_k|t_k)-y^{d(i+1)}(t_k)\Big)+f(x^p(t_k|t_k))+G(x^p(t_k|t_k))\hat{\Theta}^{-1}(t_k)u_c^0(t_k|t_k)\\
&+G(x^p(t_k|t_k))\hat{\vartheta}^0(t_k)-y^{d(r)}(t_k)\Bigg)\\
&\leqslant -c\|s^p(t_k|t_k)\|-k_s\|s^p(t_k|t_k)\|_1+\|\left(\chi\left(v_c(t_k),\underline{u}_c,\overline{u}_c\right)-I_n\right)v_c^a(t_k)\|_1\|s^p(t_k|t_k)\|_1\\
&\leqslant -c\|s^p(t_k|t_k)\|
\end{aligned}
\tag{3.23}
$$

which satisfies the Lyapunov-based stability constraint (3.17f). Therefore, it can be concluded that u_c^0 is a feasible solution to the optimization problem (3.17). □

C. Receding Horizon Implementation

Given the proven feasibility of the optimization problem (3.17), an optimal control profile $u_c^*(t|t_k)$ for $t \in [t_k, t_k+T]$ can be found by solving it at t_k. The found optimal solution is then implemented in a receding horizon manner. In this regard, $u_c^*(t|t_k)$ is applied to the system only until the next measurement is available. The actually applied control command is defined as

$$
u_c(t) = u_c^*(t_k|t_k) \tag{3.24}
$$

for $t \in [t_k, t_{k+1})$ with $t_{k+1} = t_k + \delta$ representing the next sampling instant. When

the new measurement is updated at t_{k+1}, the optimization problem (3.17) will be solved again with t_k replaced by t_{k+1}, and a new optimal control profile $u_c^*(t|t_{k+1})$ for $t \in [t_{k+1}, t_{k+1}+T]$ will be found. The newly found optimal control profile, in turn, updates the applied control command u_c.

With the initial condition (3.17d) specified and the stability condition (3.17f) satisfied, the following inequality holds for $t \in [t_k, t_{k+1})$

$$\begin{aligned} & s^\top(t_k)\left(\sum_{i=0}^{r-2}\lambda_i e^{(r+1)}(t_k) + f(x(t_k)) + G(x(t_k))\left(\hat{\Theta}^{-1}(t_k)u_c(t) + \hat{\vartheta}^0(t_k)\right) - y^{d(r)}(t)\right) \\ & \leqslant -c\left\|s(t_k)\right\|^2 \end{aligned} \tag{3.25}$$

This inequality will be recalled later in the closed-loop stability analysis.

3.4 Closed-loop Stability Analysis

This section carries out the closed-loop stability analysis of the proposed control system. We begin with the selection of a Lyapunov function candidate, from which we can derive the upper bound of its time derivative along the closed-loop control system. Then, the main theorem for concluding the closed-loop stability of the tracking control system and characterizing the stability conditions is presented, accompanied by a comprehensive proof.

3.4.1 Lyapunov Function

The Lyapunov function, as specified in (3.13), and its time derivative, incorporating adaptive estimating laws, are outlined in (3.15). RBy revisiting the inequality expressed in (3.25), adding and subtracting it to and from (3.15), we can reformulate $\dot{V}(t)$ for the interval $t \in [t_k, t_{k+1})$ as follows:

$$\begin{aligned} \dot{V}(t) \leqslant & -c\|s(t)\|^2 + c\Big(\|s(t)\|^2 - \|s(t_k)\|^2\Big) + s^\top(t)\left(\sum_{i=0}^{r-2}\lambda_i e^{(i+1)}(t) + f(x(t)) + G(x(t))\hat{\Theta}^{-1}u_c(t)\right. \\ & \left. + G(x(t))\hat{\vartheta}^0(t) - y^{d(r)}(t)\right) - s^\top(t_k)\left(\sum_{i=0}^{r-2}\lambda_i e^{(i+1)}(t_k) + f(x(t_k)) + G(x(t_k))\hat{\Theta}^{-1}(t_k)u_c(t)\right. \\ & \left. + G(x(t_k))\hat{\vartheta}^0(t_k) - y^{d(r)}(t)\right) \end{aligned} \tag{3.26}$$

In equation (3.26), we investigate the behavior of the Lyapunov function V over the time interval $[t_k, t_{k+1})$. This investigation aims to understand and predict the system's evolution in continuous time, despite the discrete nature of control updates. To manage the system's inter-sample behavior, it is essential to establish a bound on its progression. To achieve this, we invoke the concept of Lipschitz continuity (2.2), which provides a tool for bounding and quantifying the system's rate of change. Under Assumption 3.1, there must exist positive Lipschitz constants L_x, L_{ϑ_i} and L_{ϑ^0} such that

$$\left|\|s(t)\|^2 - \|s(t_k)\|^2\right| \leqslant L_{s^2} \|x(t) - x(t_k)\| \tag{3.27a}$$

$$\begin{aligned}
&\left| s^\top(t)\left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)}(t) + f(x(t)) + G(x(t))\hat{\Theta}^{-1}u_c(t) + G(x(t))\hat{\vartheta}^0(t) - y^{d(r)}(t)\right)\right. \\
&\left. - s^\top(t_k)\left(\sum_{i=0}^{r-2} \lambda_i e^{(i+1)}(t_k) + f(x(t_k)) + G(x(t_k))\hat{\Theta}^{-1}(t_k)u_c(t) + G(x(t_k))\hat{\vartheta}^0(t_k) - y^{d(r)}(t)\right)\right| \\
&\leqslant L_x \|x(t) - x(t_k)\| + \sum_i^n L_{\vartheta_i} \left\|\hat{\vartheta}_i(t) - \hat{\vartheta}_i(t_k)\right\| + L_{\vartheta^0} \left\|\hat{\vartheta}^0(t) - \hat{\vartheta}^0(t_k)\right\|
\end{aligned} \tag{3.27b}$$

We also have positive constants M_x, M_{ϑ_i} and M_{ϑ^0} satisfying

$$\|x(t) - x(t_k)\| \leqslant M_x \delta \tag{3.27c}$$

$$\left\|\hat{\vartheta}_i(t) - \hat{\vartheta}_i(t_k)\right\| \leqslant M_{\vartheta_i}\delta,\ i = 1, 2, \cdots, n \tag{3.27d}$$

$$\left\|\hat{\vartheta}^0(t) - \hat{\vartheta}^0(t_k)\right\| \leqslant M_{\vartheta^0}\delta \tag{3.27e}$$

By substituting (3.27) into (3.26), we have that

$$\dot{V}(t) \leqslant -c\|s(t)\|^2 + (\kappa_1 + c\kappa_2)\,\delta \tag{3.28}$$

where $\kappa_1 = L_x M_x + \sum_i^n L_{\vartheta_i} M_{\vartheta_i} + L_{\vartheta^0} M_{\vartheta^0}$, and $\kappa_2 = L_{s^2} M_x$.

Remark 3.1. *The term $(\kappa_1 + c\kappa_2)\,\delta$ in (3.28) constrains the system's evolution between two consecutive control updates, which is directly proportional to the sampling period and the changing rate of the system. This approach provides insights into how the choice of sampling period affects the stability and performance of the control system, thereby guiding the design of appropriate control parameters to maintain desired system behavior.*

3.4.2 Closed-loop Stability and Stability Conditions

Theorem 3.1 below summarizes the main result of the closed-loop analysis, concluding that the proposed control framework can achieve tracking control with certain sufficient stability conditions.

Theorem 3.1. *Consider the nonlinear system (3.3) in the closed loop under the developed fault-tolerant Lyapunov-based MPC framework with the adaptive parameter estimation law (3.12) and the MPC problem (3.17). The stability condition is established as follows:*

$$-2c(\rho_s - \mu) + (\kappa_1 + c\kappa_2)\,\delta \leqslant 0 \tag{3.29}$$

in which c is the positive user-determined control gain and the adaptive estimation gains; δ is the sampling period; ρ_s^0 and ρ_s are positive constants, defining level sets of the Lyapunov function, characterizing the initial region and the terminal region of s. If $s(t_0) \in \Omega_{\rho_s^0} \triangleq \{s|V \leqslant \rho_s^0\}$, $\rho_s \leqslant \rho_s^0$, and the stability condition (3.29) is satisfied by choosing an appropriate control gain, then, s of the closed-loop system is always bounded and ultimately converges to $\Omega_{\rho_s} \triangleq \{s|V \leqslant \rho_s\}$.

Proof. According to the boundedness of the fault parameters as well as the satisfaction of the control constraints, the chosen Lyapunov function V can be upper bounded by

$$V \leqslant \frac{1}{2}\|s\|^2 + \mu \tag{3.30}$$

where $\mu = \sum_{i=1}^{n} \frac{k_i}{2\underline{\vartheta}_i}\rho_{\vartheta_i} + \frac{k^0}{2}\rho_{\vartheta^0}$ with ρ_{ϑ_i} and ρ_{ϑ^0} being the level values of the stability regions for $\tilde{\vartheta}_i^2$ and $\|\tilde{\vartheta}^0\|^2$.

Combining (3.30) and the obtained time derivative of V in (3.28), we can derive that

$$\dot{V} \leqslant -2c\,(V - \mu) + (\kappa_1 + c\kappa_2)\,\delta \tag{3.31}$$

If the condition (3.29) is satisfied, then it can be derived that $\dot{V} < 0$ for all $s \in \{s|\rho_s^0 < V \leqslant \rho_s\}$ and $\dot{V} = 0$ for $V = \rho_s$. Therefore, it implies that s converges to Ω_{ρ_s} without leaving the stability region $\Omega_{\rho_s^0}$ as t approaches ∞ □

Following Theorem 3.1 and recalling Proposition 2.2, we can finally demonstrate the ultimate boundedness of the output tracking error e. Essentially, this means that

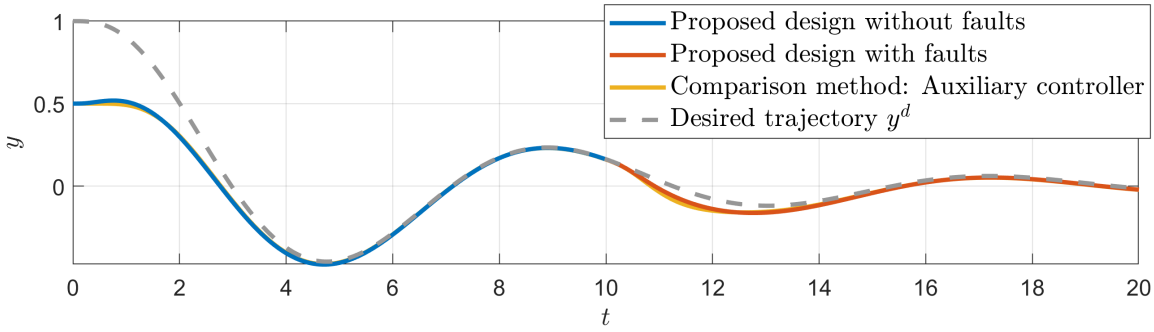

Figure 3.2: Output tracking performance in Example 1 with comparison method 1

the system's output follows the desired trajectory closely, with the error not exceeding a certain limit, hence fulfilling the control objective of output tracking control.

3.5 Simulation Study

We carry out a series of simulations in order to evaluate the performance of the proposed method in achieving robust tracking control and fault tolerance. Two examples are considered for the verification: a third-order nonlinear numerical system and a simplified kinematic model of UAVs.

In each case study, the performance of our proposed fault-tolerant Lyapunov-based MPC is compared with two alternative methods: the auxiliary analytical control law (comparison method 1) detailed in Section 3.3.1 and a non-adaptive MPC design (comparison method 2). This comparison intends to highlight the advantages of our developed control framework, particularly in terms of control optimization subject to input constraints and online accommodation against unexpected actuator faults.

3.5.1 Example 1: A Third-order Nonlinear System

Consider a third-order nonlinear system described by

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= x_1 x_2 + x_3 - x_1^3 + u \\ y &= x_1 \end{cases} \tag{3.32}$$

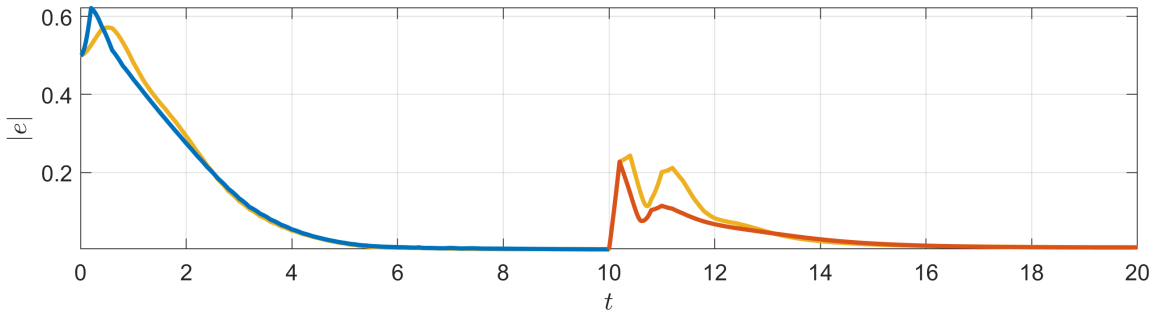

Figure 3.3: Tracking error in Example 1 with comparison method 1

where $x_1, x_2, x_3, u, y \in \mathbb{R}$ with initial conditions $y(0) = x_1(0) = 0.5$, $x_2(0) = 0$, and $x_3(0) = 0$. The original control input constraint for u is defined as

$$\Omega_u = \{u | -2.5 \leqslant u \leqslant 3.75\} \tag{3.33}$$

Taking the maximum actuator faults into consideration, the constraint for u_c must be revised to become more restrictive as

$$\Omega_{u_c} = \{u_c | -1 \leqslant u_c \leqslant 2\} \tag{3.34}$$

The objective is to manipulate the system output, y, to track a desired trajectory y^d. The desired trajectory y^d is governed by

$$\dddot{y}^d(t) = -y^d(t) - 1.16\dot{y}^d(t) - 2\ddot{y}^d(t) \tag{3.35}$$

with initial conditions set at$y^d(0) = 1$, $\ddot{y}^d(0) = 0$ and $\dot{y}^d = 0$. Also, in this case study, an actuator fault is introduced at $t = 0$ with $\vartheta = 0.9$ and $\vartheta^0 = -1$.

To this end, the proposed fault-tolerant Lyapunov-based MPC is applied. The involved control parameters are chosen as $\lambda_0 = 1$, $\lambda_1 = 2$, $c = 2$, $k = 5$, $\gamma = 0.01$, $k^0 = 0.5$, and $\gamma^0 = 0.01$. The MPC operates with a control update frequency of 5 Hz. The weighting factors in the cost function for optimization are chosen as $Q = 10$ and $R = 0.1$ to achieve balanced tracking control performance.

A. Comparison with Auxiliary Controller

To effectively highlight the capabilities of the newly developed Lyapunov-based MPC control strategy, we compared the simulation results from this strategy with those obtained using the auxiliary analytical control law. Figures 3.2-3.5 provide the

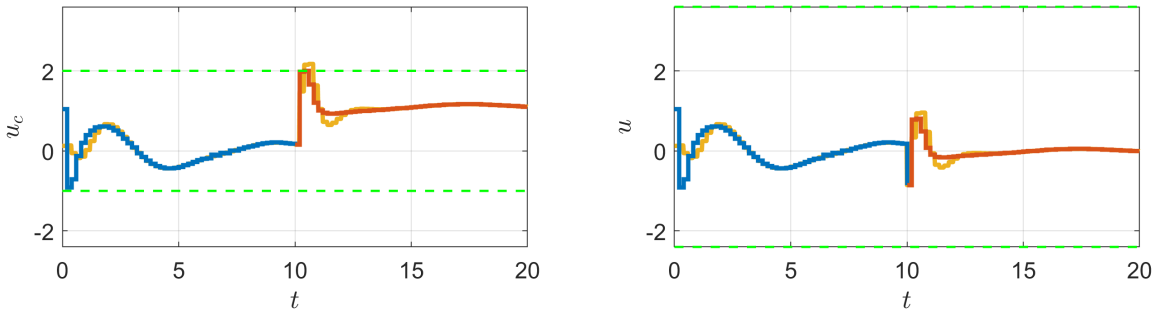

Figure 3.4: Control command (left) and actual input (right) in Example 1 with comparison method 1

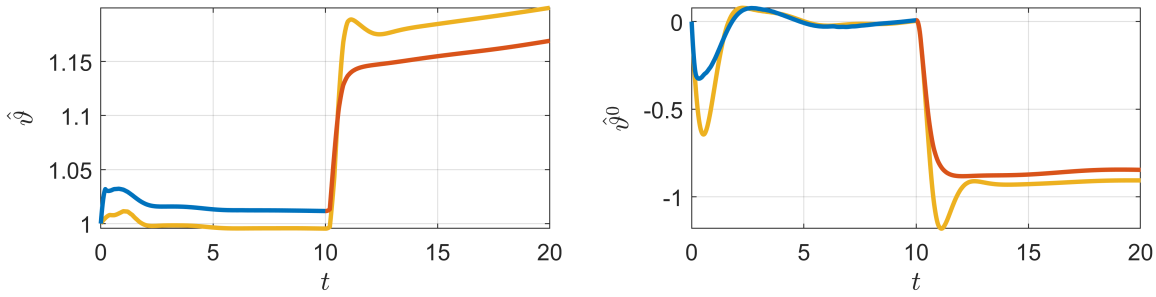

Figure 3.5: Fault parameter estimation in Example 1 with comparison method 1

results, where fault-free conditions are represented by blue lines, while red lines indicate faulty scenarios. Figures 3.2 and 3.3 illustrate, respectively, the output tracking performance and the convergence of the output tracking error e. These figures clearly indicate that the output tracks the desired trajectory with both high accuracy and rapid response. Notably, at the 10th seconds, although the tracking performance temporarily declines due to actuator faults, it promptly recovers, underscoring the fault tolerance capability of the control strategy.

The left half of Figure 3.4 shows the control command u_c, while its right half shows the actual input u, with their constraints illustrated by green dashed lines. It demonstrates that the developed control strategy actively responds to the actuator fault, effectively maintaining necessary control efforts for accurate output tracking while satisfying the constraints. Figure 3.5 gives the online estimation process of fault parameters for compensating for the fault's effect. The comparison between our method and the conventional analytical approach, indicated by yellow lines, reveals that both methods achieve comparable tracking performance. However, our MPC-based approach consistently maintains control input constraints, unlike the analytical solution, which tends to exceed the maximum allowable bounds.

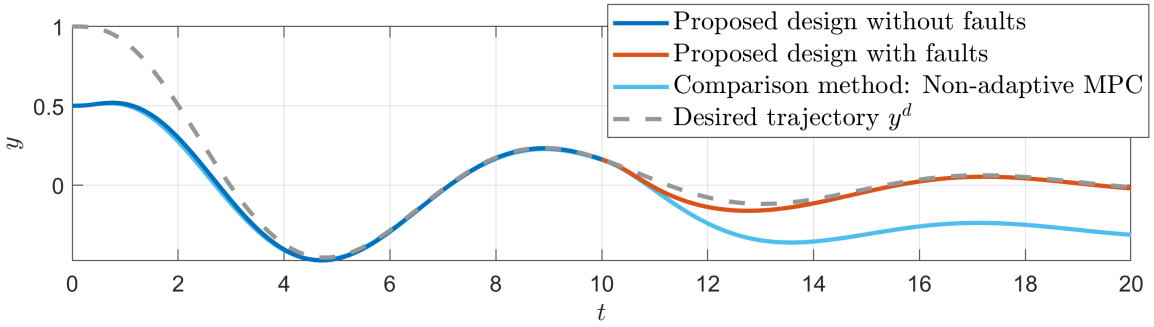

Figure 3.6: Output tracking performance in Example 1 with comparison method 2

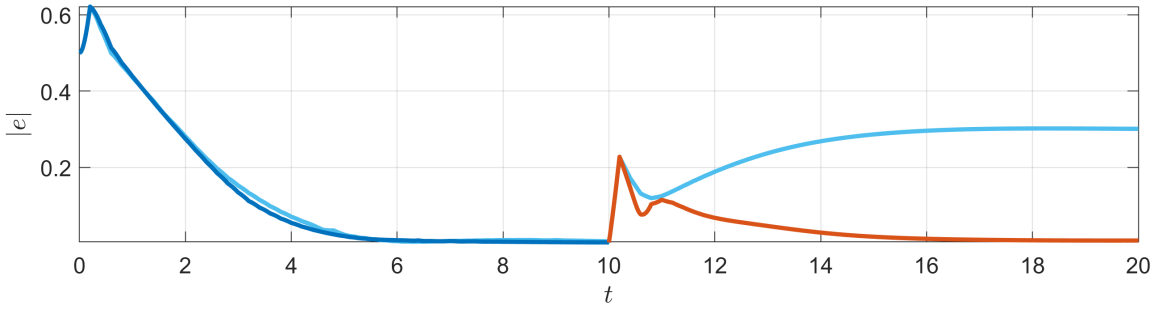

Figure 3.7: Tracking error in Example 1 with comparison method 2

B. Comparison with a Non-adaptive MPC

The simulation results are also compared with those obtained when the adaptive parameter estimator is removed. As shown in Figures 3.6-3.7, without the adaptive mechanism, the output begins to deviate from the desired trajectory after the actuator fault occurs. This demonstrates the critical role that the adaptive parameter estimator plays in maintaining the accuracy and stability of the system's output, particularly in the face of unexpected situations such as actuator faults.

3.5.2 Example 2: A UAV Kinematic Model

Apart from the numerical example, we also consider the following simplified kinematic model of UAVs for horizontal flight only:

$$\dot{x} = R(\psi)u \tag{3.36}$$

in which $x = [\zeta_x \ \zeta_y \ \psi]^\top$ is the state vector with ζ_x and ζ_y denoting the vehicle's position in the 2-dimensional horizontal plane while ψ denoting yaw angle, the direction the

UAV is facing; $u = [v_x \ v_y \ \omega_\psi]^\top$ is the input vector with v_x and v_y representing the component of linear velocity in the x and y directions, while ω_ψ representing the angular velocity around the yaw axis; $R(\psi)$ is the rotation matrix in yaw, defined as

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\Psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The initial condition of this model is set at $x(0) = [12 \ 0 \ 0]^\top$. The constraint for u is

$$\Omega_u = \left\{ u \,\middle|\, \begin{bmatrix} -6.5 & -4 & -4 \end{bmatrix}^\top \leqslant u \leqslant \begin{bmatrix} 6.5 & 4 & 4 \end{bmatrix}^\top \right\} \tag{3.37}$$

And the tightened constraint for u_c is

$$\Omega_{u_c} = \left\{ u_c \,\middle|\, \begin{bmatrix} -5 & -3 & -3 \end{bmatrix}^\top \leqslant u_c \leqslant \begin{bmatrix} 5 & 3 & 3 \end{bmatrix}^\top \right\} \tag{3.38}$$

The desired trajectory to be tracked is defined as

$$x^d = \begin{bmatrix} \zeta_x \\ \zeta_y \\ \psi \end{bmatrix} = \begin{bmatrix} 10\cos(\frac{\pi}{60}t)\cos(\frac{\pi}{15}t) \\ 10\cos(\frac{\pi}{60}t)\sin(\frac{\pi}{15}t) \\ -\frac{\pi}{2}\cos(\frac{\pi}{7}t) \end{bmatrix} \tag{3.39}$$

When $t = 15$, the system experiences an actuator fault, resulting in the actuator parameters to $\Theta = \text{diag}(0.9, 0.9, 0.9)$ and $\vartheta^0 = [-0.3 \ -0.3 \ -0.3]^\top$.

The control parameters for the system are set as follows: $c = 2$, $k = 5$, $\gamma = 0.01$, $k^0 = 0.5$, and $\gamma^0 = 0.01$. The MPC operates at a control update frequency of 5 Hz. In the optimization's cost function, the weighting matrices are defined as $Q = \text{diag}(5, 5, 1)$ and $R = \text{diag}(0.1, 0.1, 0.1)$.

A. Comparison with Auxiliary Controller

Figures 3.8-3.10 provide the obtained simulation results, with the auxiliary analytical controller's performance indicated by yellow lines. These comparisons reveal that, under the proposed FTMPC-based control strategy, the system achieves a faster convergence in tracking the desired trajectory. Furthermore, strict satisfaction of the input constraints is rigorously enforced. This simulation study underscores specifically the inherent advantage of MPC's optimization capabilities, which allow for op-

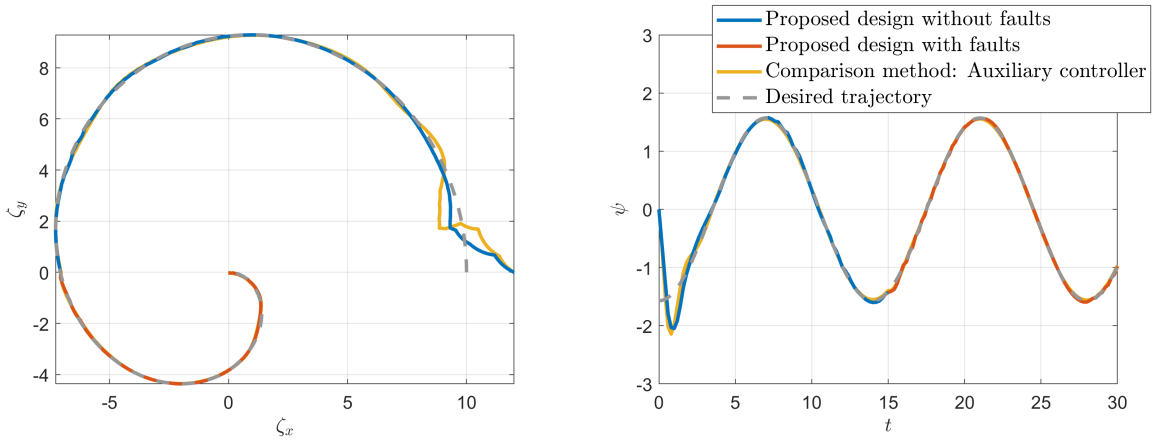

Figure 3.8: Position tracking performance in Example 2 with comparison method 1

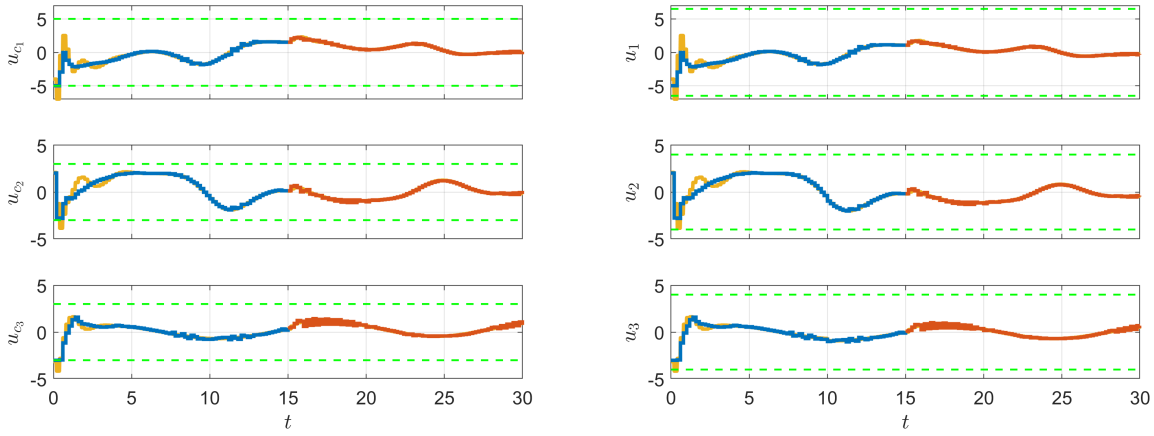

Figure 3.9: Control command (left) and actual input (right) in Example 2 with comparison method 1

timized performance through the careful design of the cost function and the explicit management of hard constraints on system signals.

B. Comparison with a Non-adaptive MPC

The simulation results are also evaluated against scenarios where the adaptive parameter estimator was omitted. Figure 3.11 illustrates that, in the absence of the adaptive mechanism, the system's output starts diverging from the desired trajectory under an actuator fault. This highlights the essential function of the adaptive parameter estimator in preserving both the precision and stability of the system's output in the presence of uncertainties such as unexpected faults.

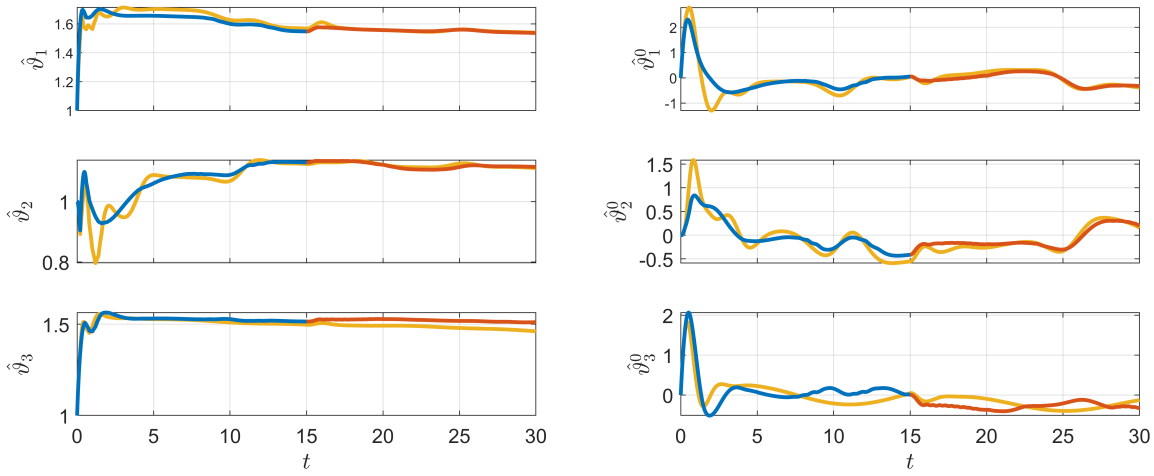

Figure 3.10: Fault parameter estimation in Example 2 with comparison method 1

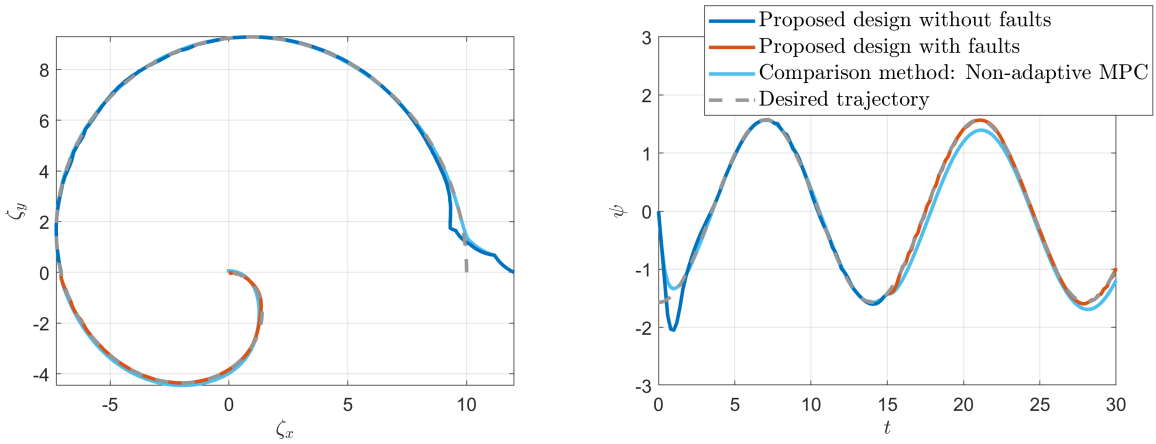

Figure 3.11: Position tracking performance in Example 2 with comparison method 2

3.6 Conclusions

In this chapter, a new adaptive fault-tolerant MPC is developed for a class of nonlinear systems with input constraints and unforeseen actuator faults. By integrating a Lyapunov-based MPC framework with an adaptive fault parameter estimator, the approach guarantees both the feasibility of the MPC optimization and the stability of the closed-loop control system. It proposes a simplified adaptive MPC formulation that eliminates the need for terminal constraints and costs, making it less computationally intensive and more straightforward for nonlinear dynamics. The method is also distinguished from other adaptive MPC methods by its ability to manage both multiplicative and additive uncertainties induced by unknown actuator faults. Furthermore, the approach incorporates Lyapunov stability theory and Lips-

chitz continuity to effectively address the impacts of discretization errors on control performance, providing sufficient stability conditions as guidance for tuning control parameters and sampling frequency. The effectiveness and superiority of the proposed adaptive FTMPC design are validated through two simulation studies.

This chapter focuses on the control problem for a broad class of general nonlinear systems. The subsequent chapters will explore the application of this FTMPC framework to trajectory tracking control of UAVs and formation tracking control of multi-UAV systems.

Chapter 4

Dual-rate Hierarchical Fault-tolerant Adaptive Model Predictive Control for UAVs with Actuator Faults

In this chapter, the fault-tolerant Lyapunov-based MPC framework developed in Chapter 3 is expanded to tackle the resilient tracking control problem of quadrotor UAVs. Given the unique control challenges and computation limitations of UAVs, this research applies a dual-loop hierarchical control architecture with dual sampling rates. This architecture is designed to manage the MPC's computational load while simultaneously maintaining the control performance.

4.1 Introduction

As previously introduced in Section 1.1.1, rotary-wing aerial vehicles are characterized by their under-actuated and highly nonlinear mechanical dynamics. The challenges of stabilizing or tracking the trajectory of such UAVs have been explored through various approaches in existing literature, ranging from classical linear control schemes [150, 151] to more sophisticated nonlinear control solutions [152–157]. Some results apply dynamic extension that integrates the motor's dynamics to address the under-actuated nature [152, 153]. However, this approach can increase the system's sensitivity to external disturbances and sensor noises, and it often requires

the measurement of the motor's states [2], which is not available in most cases. A better solution for addressing the under-actuated issue is to apply hierarchical control, where cascade controllers are developed successively to control the translational and rotational motions separately. With this control architecture, various nonlinear control techniques have been effectively applied to quadrotor UAVs, such as sliding mode control [155,156], backstepping control [157], and predictive control [54,55,158].

Hierarchical control is developed upon model decomposition by neglecting the interaction between different control loops. Design based on the decomposed model can potentially lead the system performance far from its desired outcomes. As for a dual-loop hierarchical UAV control system, achieving overall stability requires carefully chosen control gains for both loops, ensuring that the rotational dynamics converge more rapidly than the translational dynamics. However, tuning the control parameters is not an easy task. Singular perturbation theory [130, 159] provides an effective tool to quantify how "high" the inner-loop rotation control gain should be in view of stability. This theory employs time-scale separation to formalize the control design and the stability analysis in two steps: the first step is to design controllers based on the decomposed model; the second step is to analyze the closed-loop stability by examining the full dynamics with the disregarded interaction taken back into consideration. Despite the theoretical maturity of singular perturbation theory, there are only a few rigorous examinations of stability concerns in hierarchical UAV control designs [160–162]. Notably, these studies all focus on continuous-time control and do not take into account the sampled-data nature of flight control systems.

Further, another problem that arises with the hierarchical control design is that the simplified translational subsystem is non-affine in control. A common solution in literature is a two-step control design: the first step develops a newly defined virtual control vector including the actual control inputs; the second step calculates the actual inputs, which are the desired rotation angles and the required thrust force, from the control vector obtained in the first step. MPC can be a more appropriate alternative for translation control design. Different from other conventional control methods that design analytical feedback control policies offline, MPC determines the control action at each sampling instant by solving a finite horizon optimal control problem online [13,163]. Therefore, applying MPC to the translation control of UAVs not only brings a certain degree of optimal tracking performance but also determines the actual inputs in one shot.

This chapter presents the design and stability analysis of a dual-loop hierarchical

control system for UAVs subject to unexpected actuator faults. Translation control and rotation control are successively designed by considering a dual-time-scale decomposition of the translational and rotational dynamics. For the design of the translation controller, we propose a novel fault-tolerant Lyapunov-based model predictive control scheme based on the introduction of an adaptive parameter estimator. For rotation control, we integrate feedback linearization and adaptive parameter estimation. The developed translation control and rotational control are all implemented in a sample-and-hold fashion with different sampling rates. By using singular perturbation theory, the closed-loop stability of the dual-loop and dual-rate UAV control system is proven with the sufficient stability conditions provided. Results from numerical simulations are given to illustrate the trajectory tracking and fault-tolerance performance of the proposed design. The main contributions of this chapter are four-fold:

- This chapter presents a novel adaptive fault-tolerant MPC scheme by combining the Lyapunov-based MPC framework with adaptive estimation. Both multiplicative and additive actuator faults are addressed. The proposed control design can also be applied to a class of nonlinear systems with input constraints and unexpected actuator faults to achieve fault tolerance and optimal control performance.

- The proposed dual-loop and dual-rate control design provides a more practical way to apply the computation-intensive MPC to real-time flight control. With the dual-loop control architecture, MPC is applied to the outer-loop that only involves translation control so that the computation amount for solving the MPC problem is greatly reduced. Under the dual-rate sampling setup, the outer-loop is sampled with a slower sampling rate so that the burden of solving the MPC problem per iteration is mitigated. On the other hand, the inner-loop is sampled with a higher rate, limiting the uncontrolled inter-sampled behaviors and thereby ensuring the overall control performance.

- The proposed MPC-based scheme is able to determine the translation control actions in one step. Other existing solutions, such as the control strategies proposed in [161, 162, 164], require two steps — a virtual control vector is firstly designed, and actual control inputs are solved from that. Our approach avoids the singularity problem that might be caused by solving the nonlinear transcendental function in the second step.

- The closed-loop stability is evaluated rigorously. By taking account of the interconnection between subsystems and the inter-sample system evolution, we explicitly characterize the relationship between the closed-loop performance and the selection of control parameters and sampling periods. The obtained sufficient stability conditions specify the maximal singular perturbation parameter, thereby quantifying how "high" at least the control gain of the inner-loop should be for ensuring the overall stability. Following the stability conditions, we can also select the designed parameters together with the sampling periods so that practicable computation effort and guaranteed closed-loop performance can be attained simultaneously.

The rest of this chapter is organized as follows. In Section 4.2, we present the mathematical model of the UAV dynamics with constraints and actuator faults and then point out the control objectives. Section 4.3 decomposes the UAV dynamics in two time-scales by using singular perturbation theory and presents the hierarchical control design developed successively based on the decomposed model. In Section 4.4, the closed-loop stability of the full dynamics is rigorously analyzed with the inter-sample behaviors, and the interconnection between the translational and rotational dynamics is taken into consideration. In Section 4.5, simulation results are provided to demonstrate the effectiveness of the proposed control framework and algorithms. Concluding remarks are made in Section 4.6.

4.2 Problem Formulation

In this chapter, we formulate the tracking control problem for UAVs using mathematical descriptions.

4.2.1 UAV Dynamics

The UAV we consider is a quadrotor, six-degree-of-freedom rigid-body aerial vehicle with four rotary wings. Referring back to the UAV dynamics modeled in (2.5), and assuming a condition of steady hovering, the system addressed in this study can

be described as follows:

$$\begin{cases} \dot{\zeta} &= v \\ m\dot{v} &= -mg + r(\eta)F \end{cases} \tag{4.1a}$$

$$\begin{cases} \dot{\zeta} &= \omega \\ J\dot{\omega} &= -\omega \times J\omega + \tau \end{cases} \tag{4.1b}$$

with

$$r(\eta) = \begin{bmatrix} \sin\theta\cos\psi\cos\phi + \sin\psi\sin\phi \\ \sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}$$

where $\zeta = [x\ y\ z]^\top$ denotes the position of the vehicle; $\eta = [\phi\ \theta\ \psi]^\top$ describes the vehicle's rotation; $v = [v_x\ v_y\ v_z]^\top$ and $\omega = [\omega_p\ \omega_\eta\ \omega_r]^\top$ are the vector of linear and angular velocities; m is the vehicle mass; $g = [0\ 0\ g^*]^\top$ with g^* being the gravitational acceleration; $J = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$ is a diagonal matrix of moments of inertia; F and τ are the inputs of this system: F is the total thrust force and $\tau = [\tau_\phi\ \tau_\eta\ \tau_\psi]^\top$ is the vector of rotation torques.

4.2.2 Input and State Constraints

Given the physical limitation of the rotors, the inputs should comply with the following constraints:

$$F \in \left\{ F \in \mathbb{R} \mid |F - mg| \leqslant \overline{F} \right\} \tag{4.2a}$$

$$\tau \in \left\{ \tau \in \mathbb{R}^3 \mid \|\tau\| \leqslant \overline{\tau} \right\} \tag{4.2b}$$

where $\overline{F}, \overline{\tau}$ are positive constants specifying acceptable changing ranges for F and τ.

On the other hand, acrobatic maneuvers of the vehicle should be avoided for the sake of flight safety. This requires the rotation angles to evolve within the following set

$$\eta \in \Omega_\eta \triangleq \left\{ \eta \in \mathbb{R}^3 | \ \|\eta\| \leqslant \overline{\eta} \right\} \tag{4.3}$$

where $\overline{\eta}$ is the upper bound of the Euclidean norm of η.

4.2.3 Actuator Faults

As the actuators of the UAV control system, rotors have the possibility of malfunctions such as low voltage supply or blade defection. Similar to the modeling in (2.10), the impacts of common actuator failures can be modeled as either multiplicative or additive uncertainties in the output of the actuators as follows:

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \underbrace{\left[\begin{array}{c|ccc} \vartheta_{c_l} & 0 & 0 & 0 \\ \hline 0 & \vartheta_{c_l} & 0 & 0 \\ 0 & 0 & \vartheta_{c_l} & 0 \\ 0 & 0 & 0 & \vartheta_{c_r} \end{array}\right]}_{\Theta_\tau} \begin{bmatrix} u_F \\ u_\tau \end{bmatrix} + \underbrace{\begin{bmatrix} \vartheta_{F^0} \\ \hline \vartheta_{\tau_\phi^0} \\ \vartheta_{\tau_\theta^0} \\ \vartheta_{\tau_\psi^0} \end{bmatrix}}_{\vartheta_{\tau 0}} \tag{4.4}$$

where u_F and $u_\tau = [u_{\tau_\phi}\ u_{\tau_\theta}\ u_{\tau_\psi}]^\top$ are the control commands to be determined later in Section 4.3. In this work, no fault detection or identification device is equipped in the flight control system, so the real-time values of fault parameters are unknown.

Assumption 4.1. *The multiplicative parameters ϑ_{c_l} and ϑ_{c_r} are bounded by an upper bound of 1 and a lower bound denoted as $\underline{\vartheta} \in (0, 1)$. The unknown fault parameters ϑ_{F^0} and ϑ_{τ^0} are bounded above by $\overline{\vartheta}_{F^0} \in \mathbb{R}^+$ and $\overline{\vartheta}_{\tau^0} \in \mathbb{R}^+$ in their absolute value and Euclidean norm — that is $|\vartheta_{F^0}| \leqslant \overline{\vartheta}_{F^0}$ and $\|\vartheta_{\tau^0}\| \leqslant \overline{\vartheta}_{\tau^0}$. Also, these fault parameters are differentiable and slowly varying so their time derivatives can be regarded as 0 in the performance evaluation without losing much of the analytical accuracy.*

Remark 4.1. *Even though the exact fault information, like the time of occurrence or the magnitude, is hard to obtain without fault detection mechanisms, it is practical to estimate a rough impact range of the faults from historical data. Therefore, the above assumption that the unknown fault parameters have known bounds is reasonable.*

In order to guarantee the satisfaction of the input constraints (4.2), even in the most extreme cases of faults, the control commands u_F and u_τ should be further restricted to the following tightened sets

$$u_F \in \Omega_F \triangleq \{u_F \in \mathbb{R} \mid |u_F - mg| \leqslant \overline{u}_F\} \tag{4.5a}$$

$$u_\tau \in \Omega_\tau \triangleq \left\{u_\tau \in \mathbb{R}^3 \middle| \ \|u_\tau\| \leqslant \overline{u}_\tau\right\} \tag{4.5b}$$

where $\overline{u}_F = \overline{F} - \overline{\vartheta}_{F^0}$ and $\overline{u}_\tau = \overline{\tau} - \overline{\vartheta}_{\tau^0}$.

4.2.4 Tracking Control Objectives

This work aims to develop a control scheme for generating the control commands u_F and u_τ to

- Drive the UAV modeled by (4.1) to track a prespecified flight trajectory denoted as $\zeta^d = [x^d \ y^d \ z^d]^\top$;
- Stabilize the rotation of the UAV.

in the presence of the input constraints (4.5), the state constraint (4.3), and the actuator faults (4.4).

Assumption 4.2. *The specified desired trajectory ζ^d is slowly time-varying with respect to the UAV dynamics. ζ^d is piece-wise continuous with the outer-loop sampling period. Its first- and second-order derivatives exist, denoted as $\dot{\zeta}^d$ and $\ddot{\zeta}^d$.*

4.3 Hierarchical Control Design

In this section, we present the fault-tolerant tracking control design for UAVs. The discussion begins with the decomposition of the model, followed by an explanation of the hierarchical control architecture and the dual-rate control updating manner. Subsequent sections will detail the development of the outer translation control loop and the inner rotation control loop.

4.3.1 Model Decomposition

In the UAV model (4.1), (4.1a) represents the translational dynamics while (4.1b) represents the rotational dynamics. To enable the control design with hierarchical architecture, model decomposition of (4.1a) and (4.1b) is required, which is legitimized by assuming that (4.1b) converges much faster than (4.1a). The difference in convergence rate, from another perspective, can be regarded as a separation in time-scales. Therefore, we consider the problem in a dual-time-scale context below.

For the fast-varying rotational dynamics, a "stretched" time variable $\breve{t}$ is introduced by defining $\breve{t} = t/\varepsilon$ with t representing the standard time scale and $\varepsilon \in (0, 1]$. The time derivatives of a system state vector x with respect to $\breve{t}$ and t should meet

the following scaling relationship

$$\frac{\mathrm{d}x}{\mathrm{d}\breve{t}} = \varepsilon \frac{\mathrm{d}x}{\mathrm{d}t} \tag{4.6}$$

Then, we introduce new notations of η and ω in the fast time scale as $\breve{\eta} \triangleq \eta$ and $\breve{\omega} \triangleq \varepsilon\omega$. Thereupon, the UAV dynamics (4.1) with actuator faults (4.4) can be rewritten in the following dual-time-scale format

$$\begin{cases} \dot{\zeta} &= v \\ \dot{v} &= -g + \vartheta_{\zeta}^{-1} r(\eta) u_F + \vartheta_{\zeta^0} \end{cases} \tag{4.7a}$$

$$\begin{cases} \varepsilon\dot{\breve{\eta}} &= \breve{\omega} \\ \varepsilon\dot{\breve{\omega}} &= f_{\eta}(\omega) + \varepsilon^2 \Theta_{\eta}^{-1} u_{\tau} + \vartheta_{\eta^0} \end{cases} \tag{4.7b}$$

where $f_{\eta}(\omega) = -\breve{\omega} \times J\breve{\omega}$; $\vartheta_{\zeta}^{-1} = m\vartheta_{c_l}^{-1}$, $\vartheta_{\zeta^0} = m^{-1} r(\eta)\vartheta_{F^0}$, $\Theta_{\eta}^{-1} = J\Theta_{\tau}^{-1}$, and $\vartheta_{\eta^0} = J^{-1}\vartheta_{\tau^0}$. This is a standard singularly perturbed system with the perturbation parameter ε.

Model decomposition is produced by setting ε to 0. As $\varepsilon = 0$, transients of $\breve{\eta}$ and $\breve{\omega}$ diminish instantaneously. Assuming that the rotation controller is well designed to drive η to its reference signal, denoted as η^r, η in the slower-varying translational subsystem (4.7a) can be approximated by η^r. As a result, we obtain the following reduced-order translational dynamics

$$\begin{cases} \dot{\zeta} &= v \\ m\dot{v} &= -g + \vartheta_{\zeta}^{-1} r(\eta^r) u_F + \vartheta_{\zeta^0} \end{cases} \tag{4.8}$$

which is decoupled from the rotational dynamics (4.7b) evolving in the fast time-scale.

4.3.2 Dual-rate Hierarchical Architecture

Fig. 4.1 illustrates the block diagram of the developed hierarchical control system, in which an outer-loop translation control and an inner-loop rotation control are designed separately. The two control loops work in series: Firstly, the outer-loop translation control is developed to generate the required thrust force command u_T and q^r for steering the vehicle to perform certain specified translation behaviors; then, the inner-loop rotation control is developed to manipulate the actual attitude q to track q^r rapidly. The detailed design of the two control loops will be given later in

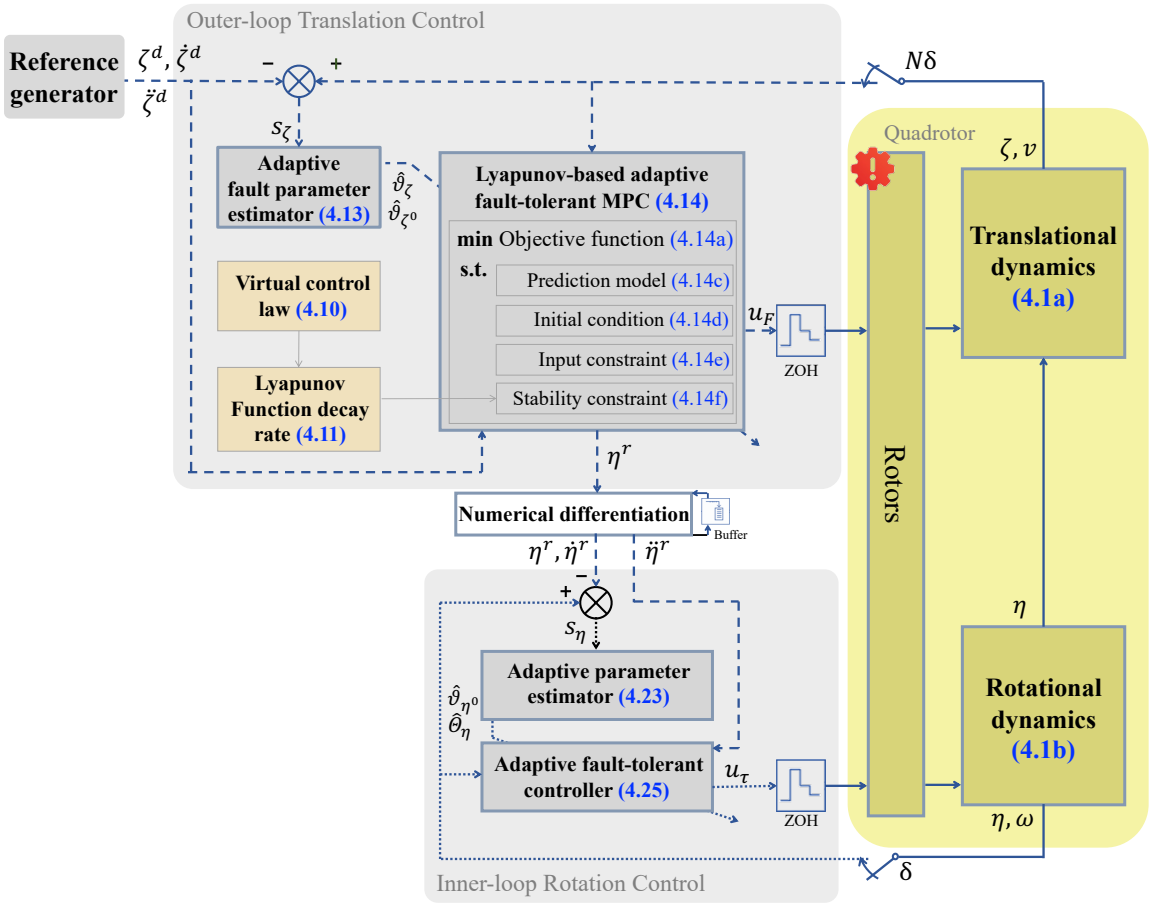

Figure 4.1: Block diagram of the hierarchical control system

Section 4.3.3 and 4.3.4, respectively.

The developed inner-loop and outer-loop controllers are all implemented in a sample-and-hold fashion, meaning that the system is updated periodically at sampling instants, and the computed control action is held until the next state measurement is available. Given the expected dual-time-scale behavior of the translation and rotation control loops, it is natural to set different sampling rates for them. Hence, the sampling period for the fast-varying rotation control inner-loop is defined as δ, and the outer-loop sampling period is $N\delta$, with N being a positive integer. Under this dual-sampling-rate setting, the translation control is updated once while the rotation control is updated N times. The sequence of the sampling instants of the inner-loop is denoted as $\{k_i\delta | k_i \in \{0, 1, 2, \cdots\}\}$. Correspondingly, the sampling instant sequence of the outer-loop is $\{k_o\delta | k_o \in \{0, N, 2N, \cdots\}\}$.

In the subsequent design and analysis, we use notations with the time variable t

in round brackets to represent signals evolving continuously and use those with the sampling numbers k_i and k_o in square brackets to represent the measurements at discrete time instants. For example, in the rotation control loop, $\eta[k_i]$ represents the measurement of the continuous state variable $\eta(t)$ at the sampling instant $t = k_i\delta$. In the translation control loop, $\zeta[k_o]$ is the measurement of $\zeta(t)$ at $t = k_o\delta$.

4.3.3 Outer-loop Translation Control

The translation control, forming the outer-loop of the hierarchical control system, aims to realize tracking of the desired trajectory and to enforce a certain level of optimal performance subject to constraints and unknown actuator faults. Motivated by the translation tracking objectives, a filtered tracking error is introduced

$$s_\zeta = \lambda_\zeta e_\zeta + e_v \tag{4.9}$$

where $e_\zeta = \zeta - \zeta^d$ and $e_v = v - \dot{\zeta}^d$ are the position and velocity tracking errors. λ_ζ is chosen appropriately such that the polynomial $s + \lambda_\zeta$ is Hurwitz. Referring back to Proposition 2.1 in Chapter 2, Then, $s_\zeta = 0$ defines the sliding surface for the translation tracking error e_ζ.

Therefore, the translation tracking control objective of the outer loop is equivalent to the regulation of s_ζ in close proximity to the origin. The detailed control design procedure is presented below.

A. Auxiliary Control Law

Before formulating the MPC problem, we need to first study the stability property achieved by an auxiliary control design that is developed to provide a solution to stabilize the translational dynamics (4.8) in the nominal case. This auxiliary control can be designed as

$$u_\zeta^a \triangleq r(\eta^r)u_F = \vartheta_\zeta \left(-c_\zeta s_\zeta - \lambda_\zeta e_v + g - \hat{\vartheta}_{\zeta^0} + \ddot{\zeta}^d\right) \tag{4.10}$$

where c_ζ is a strictly positive constant as the control gain of the virtual controller; ϑ_ζ and ϑ_{ζ^0} are provisionally assumed to be know.

To characterize the closed-loop stability under this auxiliary control design (4.10),

a Lyapunov function candidate is selected

$$V_{s_\zeta} = \frac{1}{2}\|s_\zeta\|^2 \tag{4.11}$$

Differentiating V_ζ along the reduced-order translation dynamics (4.8) and substituting (4.10) yields

$$\begin{aligned}\dot{V}_{s_\zeta}^{-} &= s_\zeta^\top \left(\lambda_\zeta e_v - g + \vartheta_\zeta^{-1} r(\eta^r) u_F + \vartheta_{\zeta^0} - \ddot{\zeta}^d\right) \\ &= -c_\zeta \|s_\zeta\|^2 \end{aligned} \tag{4.12}$$

where the superscript $^-$ implies that the derivative is obtained along the reduced model. This result explicitly characterizes a desired decay rate of V_{s_ζ} with the auxiliary controller u_ζ^a.

B. Lyapunov-based Fault-tolerant MPC

The developed Lyapunov-based fault-tolerant MPC framework consists of an adaptive online estimator and a Lyapunov-based MPC controller. The adaptive laws for estimating ϑ_ζ and ϑ_{ζ^0} within the interval $t \in [k_o\delta, (k_o+N)\delta)$ are designed as

$$\dot{\hat{\vartheta}}_\zeta(t) = -k_\zeta \hat{\vartheta}_\zeta^{-1}[k_o] u_F[k_o] r(\eta^r[k_o])^\top s_\zeta[k_o] \tag{4.13a}$$

$$\dot{\hat{\vartheta}}_{\zeta^0}(t) = k_{\zeta^0} s_\zeta[k_o] \tag{4.13b}$$

where k_ζ and k_{ζ^0} are positive constants.

We now present the MPC problem formulation. To distinguish the model used for prediction from the real system model, we mark the internal variables of MPC with a superscript of p. MPC determines η^r and u_F by solving a finite horizon optimization problem at each sampling instant. Taking the measured states $\zeta[k_o]$ and $v[k_o]$, the estimated parameter $\hat{\vartheta}_\zeta[k_o]$ and $\hat{\vartheta}_{\zeta^0}[k_o]$, ζ^d and its derivatives over the prediction horizon $[k_o\delta, (k_o+N_pN)\delta]$ as input elements, the MPC problem at $t = k_o\delta$ is constructed by

$$\min_{\eta^{r^p}, u_F^p} \int_{k_o\delta}^{(k_o+N_pN)\delta} \left(\left\|s_\zeta^p(t|k_o\delta)\right\|_Q^2 + \left\|\eta^{r^p}(t|k_o\delta)\right\|_{R_\eta}^2 + |u_F^p(t|k_o\delta) - mg|_{R_F}^2\right) \mathrm{d}t \tag{4.14a}$$

with

$$s_\zeta^p(t|k_o\delta) = \lambda_\zeta \left(\zeta^p(t|k_o\delta) - \zeta^d(t)\right) + \left(v^p(t|k_o\delta) - \dot{\zeta}^d(t)\right) \tag{4.14b}$$

subject to

$$\begin{cases} \dot{\zeta}^p(t|k_o\delta) = v^p(t|k_o\delta) \\ \dot{v}^p(t|k_o\delta) = -g + \hat{\vartheta}_\zeta^{-1}[k_o] r(\eta^{r^p}(t|k_o\delta)) u_F^p(t|k_o\delta) + \hat{\vartheta}_{\zeta^0}[k_o] - \ddot{\zeta}^d(t) \end{cases} \tag{4.14c}$$

$$\begin{cases} \zeta^p(k_o\delta|k_o\delta) = \zeta[k_o] \\ v^p(k_o\delta|k_o\delta) = v[k_o] \end{cases} \tag{4.14d}$$

$$\begin{cases} \eta^{r^p}(t|k_o\delta) \in \Omega_\eta \\ u_F^p(t|k_o\delta) \in \Omega_F \end{cases} \tag{4.14e}$$

$$\begin{aligned} & s_\zeta^{p^\top}(k_o\delta|k_o\delta) \left(\lambda_\zeta \left(v^p(k_o\delta|k_o\delta) - \dot{\zeta}(k_o\delta)\right) - g + \hat{\vartheta}_\zeta^{-1}[k_o] r(\eta^{r^p}(k_o\delta|k_o\delta)) u_F^p(k_o\delta) \right. \\ & \left. + \hat{\vartheta}_{\zeta^0}[k_o] - \ddot{p}^d(k_o\delta|k_o\delta)\right) \leqslant -c_\zeta \|s_\zeta^p(k_o\delta|k_o\delta)\|^2 \end{aligned} \tag{4.14f}$$

where N_p is the number of samplings contained within the prediction horizon; Q and R_η are positive-definite and symmetric weighting matrices; R_F is a positive weighting scalar.

In the optimization problem, (4.14c) is the prediction model to predict the future evolution of ζ^p and v^p with their initial values defined as (4.14d). In the prediction model, $\hat{\vartheta}_\zeta$ and $\hat{\vartheta}_{\zeta^0}$ are estimates of the unknown parameters updated by (4.13). (4.14e) ensures the input constraints. The construction of the stability contractive constraint (4.14f) is inspired by (4.12), enforcing a decay in the value of the Lyapunov function by at least the rate achieved by the auxiliary analytical controller u_F^a during the first sampling interval of the prediction horizon.

The solution of the optimization problem (4.14), denoted as η^{r^p*} and u_T^{p*}, is implemented in a receding horizon manner. That is, the calculated optimized solution is applied to the system until the next measurement is available. Hence, the translation control command of the outer loop is defined as

$$\begin{cases} \eta^r(t) = \eta^{r^p*}(k_o\delta|k_o\delta) \\ u_F(t) = u_F^{p*}(k_o\delta|k_o\delta) \end{cases} \tag{4.15}$$

for $t \in [k_o\delta, (k_o+N)\delta)$. This process is repeated, and the translation control command will be updated at the next outer-loop sampling instant at $t = (k_o + N)\delta$.

With the initial condition (4.14d) and the stability condition (4.14f) by the translation control, the following inequality holds for all $t \in [k_o\delta, (k_o+N)\delta)$

$$s_\zeta^\top[k_o]\left(\lambda_\zeta e_\zeta[k_o] - g + \hat{\vartheta}_\zeta^{-1} r(\eta^r[k_o])u_F(t) + \hat{\vartheta}_{\zeta^0}[k_o] - \ddot{\zeta}^d[k_o]\right) \leqslant -c_\zeta \|s_\zeta[k_o]\|^2 \quad (4.16)$$

which will be referred later in Section 4.4 for stability analysis.

Lemma 4.1. *There always exists a feasible solution for the optimization problem (4.14), which can be constructed as*

$$\eta^{r^0}(t|k_o\delta) = \begin{bmatrix}\phi^0(t|k_o\delta)\\ \theta^0(t|k_o\delta)\\ \psi^0(t)\end{bmatrix} = \begin{bmatrix}\arctan\left(\frac{u^0_{\zeta_1}\sin\psi^0 - u^0_{\zeta_2}\cos\psi^0}{\sqrt{u^{0^2}_{\zeta_1}+u^{0^2}_{\zeta_2}+u^{0^2}_{\zeta_3}}}\right)\\ \arctan\left(\frac{u^0_{\zeta_1}\cos\psi^0 + u^0_{\zeta_2}\sin\psi^0}{u^0_{\zeta_3}}\right)\\ \psi^0\end{bmatrix} \quad (4.17a)$$

$$\begin{aligned} u_{F^0}(t|k_o\delta) =& u^0_{\zeta_1}(\sin\theta^0\cos\psi^0\cos\phi^0 + \sin\psi^0\sin\phi^0) + u^0_{\zeta_2}(\sin\theta^0\sin\psi^0\cos\phi^0 - \cos\psi^0\sin\phi^0)\\ &+ u^0_{\zeta_3}\cos\theta^0\cos\phi^0 \end{aligned} \quad (4.17b)$$

where $t \in [k_0\delta, (k_0+N_pN)\delta)$; ϕ^0 could be any desired yaw angle satisfying (4.3); $u^0_{\zeta_1}$, $u^0_{\zeta_2}$, and $u^0_{\zeta_3}$ are elements of the virtual controller u^0_ζ, developed as

$$u^0_\zeta(t|k_o\delta) = \text{sat}\left(v_\zeta(t|k_o\delta), -\overline{u}_F + mg, \overline{u}_F + mg\right) \quad (4.18a)$$

$$v_\zeta(t|k_o\delta) = \hat{\vartheta}_\zeta[k_o]\left(v^a_\zeta(t|k_o\delta) + v^d_\zeta(t|k_o\delta)\right) \quad (4.18b)$$

$$v^a_\zeta(t|k_o\delta) = -c_\zeta s^p_\zeta(t|k_o\delta) - \lambda_\zeta\left(v^p(t|k_o\delta) - \dot{\zeta}(t)\right) + g - \hat{\vartheta}_{\zeta^0}[k_o] + \ddot{\zeta}^d(t) \quad (4.18c)$$

$$v^d_\zeta(t|k_o\delta) = -k_s\underline{\chi}^{-1}\text{sgn}\left(s^p_\zeta(t|k_o\delta)\right) \quad (4.18d)$$

where $\text{sat}\left(v_\zeta, -\overline{u}_F + mg, \overline{u}_F + mg\right)$ *is a standard saturation function, equaling to χv_ζ with the function χ defined the same as in (3.19); and $\underline{\chi}$ is the lower bound of the function χ; the gain parameter k_s is chosen such that $k_s \geqslant \left\|(\chi - I_n)\, v^a_\zeta\right\|_1$.*

Proof. The proof of Lemma 4.1 is similar to the proof of Lemma 3.1 and is therefore omitted here for brevity. □

4.3.4 Inner-loop Rotation Control

We now turn to the inner-loop control design for the rotational subsystem (4.7b) in the fast time-scale. With η^r determined by the outer-loop translation control, the

rotational control intends to drive η to track η^r.

Similarly, a sliding mode tracking error of the rotation motion is defined in the fast time-scale

$$\breve{s}_\eta = \lambda_\eta \breve{e}_\eta + \breve{e}_\omega \tag{4.19}$$

where $\breve{e}_\eta = \breve{\eta} - \eta^r$ and $\breve{e}_\omega = \breve{\omega} - \varepsilon \dot{\eta}^r$; λ_η is chosen to be strictly positive. Recalling Proposition 2.1, we can have $\breve{e}_\eta \to 0$ as $t \to 0$ when $\breve{s}_\eta = 0$. Therefore, our objective for rotation tracking control is to keep $\breve{s}_\eta$ converge to and stay near the origin.

To achieve the control objective, an adaptive fault-tolerant rotation control law can be developed as

$$u_\tau = \hat{\Theta}_\eta \left(\frac{1}{\varepsilon^2} \left(-c_\eta \breve{s}_\eta - \lambda_\eta \breve{e}_\omega - f_\eta(\breve{\omega}) \right) - \hat{\vartheta}_{\eta^0} + \ddot{\eta}^r \right) \tag{4.20}$$

where c_η is strictly positive; $\hat{\Theta}_\eta = \text{diag}\{\hat{\vartheta}_\phi, \hat{\vartheta}_\theta, \hat{\vartheta}_\psi\}$ and $\hat{\vartheta}_{\eta^0}$ are estimates of Θ_η and ϑ_{η^0}, updated by the following adaptive laws

$$\dot{\hat{\vartheta}}_\phi = -\varepsilon k_\eta \breve{s}_\eta^\top \text{diag}\{\hat{\vartheta}_\phi^{-1}, 0, 0\} u_\tau \tag{4.21a}$$

$$\dot{\hat{\vartheta}}_\theta = -\varepsilon k_\eta \breve{s}_\eta^\top \text{diag}\{\hat{\vartheta}_\theta^{-1}, 0, 0\} u_\tau \tag{4.21b}$$

$$\dot{\hat{\vartheta}}_\psi = -\varepsilon k_\eta \breve{s}_\eta^\top \text{diag}\{\hat{\vartheta}_\psi^{-1}, 0, 0\} u_\tau \tag{4.21c}$$

$$\dot{\hat{\vartheta}}_{\eta^0} = \varepsilon k_{\eta^0} \breve{s}_\eta \tag{4.21d}$$

where k_η and k_{η^0} are positive constants.

Under the sample-and-hold implementation setup, the rotation control law is updated discretely. The updated rotation control law for $t \in [k_i\delta, (k_i+1)\delta)$ is

$$u_\tau(t) = \hat{\Theta}_\eta[k_i] \left(\frac{1}{\varepsilon^2} \left(-c_\eta \breve{s}_\eta[k_i] - \lambda_\eta \breve{e}_\omega[k_i] - f_\eta(\breve{\omega}[k_i]) \right) - \hat{\vartheta}_{\tau_\eta}[k_i] + \ddot{\eta}^r[k_i] \right) \tag{4.22}$$

The sampled-data adaptive laws for $t \in [k_i\delta, (k_i+1)\delta)$are

$$\dot{\hat{\vartheta}}_\phi(t) = -\varepsilon k_\eta \breve{s}_\eta^\top[k_i] \text{diag}\{\hat{\vartheta}_\phi^{-1}[k_i], 0, 0\} u_\tau[k_i] \tag{4.23a}$$

$$\dot{\hat{\vartheta}}_\theta(t) = -\varepsilon k_\eta \breve{s}_\eta^\top[k_i] \text{diag}\{\hat{\vartheta}_\theta^{-1}[k_i], 0, 0\} u_\tau[k_i] \tag{4.23b}$$

$$\dot{\hat{\vartheta}}_\psi(t) = -\varepsilon k_\eta \breve{s}_\eta^\top[k_i] \text{diag}\{\hat{\vartheta}_\psi^{-1}[k_i], 0, 0\} u_\tau[k_i] \tag{4.23c}$$

$$\dot{\hat{\vartheta}}_{\eta^0}(t) = \varepsilon k_{\eta^0} \breve{s}_\eta[k_i] \tag{4.23d}$$

Remark 4.2. *Higher-order time derivatives of η^r are required for generating the rotation control signal. $\dot{\eta}^r$ and $\ddot{\eta}^r$ can be approximated by doing numerical differentiation with η^r and its historical data, as is done in* [162].

If presented under the standard time-scale of the outer-loop, the counterpart of $\breve{s}_\eta$ is

$$s_\eta = \frac{\breve{s}_\eta}{\varepsilon} = \frac{\lambda_\eta}{\varepsilon} e_\eta + e_\omega \tag{4.24}$$

The rotation control law (4.20) can be rewritten into the following expression in the outer-loop time-scale

$$u_\tau = \hat{\Theta}_\eta \left(-\frac{c_\eta}{\varepsilon} s_\eta - \frac{\lambda_\eta}{\varepsilon} e_\omega - f_\eta(\omega) - \hat{\vartheta}_{\eta^0} + \ddot{\eta}^r \right) \tag{4.25}$$

Remark 4.3. *Expression (4.25) explicitly indicates that the singular perturbation parameter ε formalizes the high-gain property of the rotational control law — the smaller ε is set to be, the higher the control gain is. Therefore, the maximal admissible value of ε, which will be given later in Section 4.4, can quantify exactly how "high" the rotation control gain should be in view of the overall stability.*

4.4 Closed-loop Stability Analysis

The outer-loop and inner-loop controllers are designed based on the decomposed model (4.8) and (4.7b) *without* considering the interaction of the translational and rotational dynamics. In this section, we evaluate the overall closed-loop stability of the full dynamics with the disregarded interaction taken back into consideration. Furthermore, the uncontrolled inter-sample behaviors of the sample-data system are considered.

4.4.1 Lyapunov Function

We take the following steps to analyze the stability. At first, Lyapunov functions for the translational dynamics and rotational dynamics are constructed respectively, and their derivatives with the developed translation and rotation control schemes are evaluated. After that, a composite Lyapunov function is formed by adding the two Lyapunov functions together.

A. Lyapunov Function for Outer-loop

A Lyapunov function candidate for the outer loop can be selected as

$$V_\zeta = V_{s_\zeta} + \frac{1}{2k_\zeta \vartheta_\zeta}\tilde{\vartheta}_\zeta^2 + \frac{1}{2k_{\zeta^0}}\|\tilde{\vartheta}_{\zeta^0}\|^2 \tag{4.26}$$

where V_{s_ζ} is the Lyapunov function for nominal case and has been defined in (4.11); $\tilde{\vartheta}_\zeta$ and $\tilde{\vartheta}_{\zeta^0}$ are adaptive estimation errors, defined as $\tilde{\vartheta}_\zeta = \hat{\vartheta}_\zeta - \vartheta_\zeta$ and $\tilde{\vartheta}_{\zeta^0} = \hat{\vartheta}_\zeta - \vartheta_{\zeta^0}$.

Differentiating V_ζ along the full-order translational dynamics (4.1a) gives

$$\dot{V}_\zeta = \dot{V}_\zeta^- + D \tag{4.27}$$

where

$$\begin{aligned}\dot{V}_\zeta^- &= \dot{V}_{s_\zeta}^- + \frac{1}{k_\zeta \vartheta_\zeta}\tilde{\vartheta}_\zeta \dot{\hat{\vartheta}}_\zeta + \frac{1}{k_{\zeta^0}}\tilde{\vartheta}_{\zeta^0}^\top \dot{\hat{\vartheta}}_{\zeta^0} \\ D &= s_\zeta^\top \vartheta_\zeta^{-1}\left(r(\eta) - r(\eta^r)\right)\left(u_F + \vartheta_{\zeta^0}\right)\end{aligned}$$

in which $\dot{V}_{s_\zeta}^-$ is the derivative of V_{s_ζ} along the reduced translational dynamics (4.8), whose expression has been given in (4.12); D describes the deviation of the actual derivative of V_{s_ζ} from $\dot{V}_{s_\zeta}^-$, resulting from the neglected transient dynamics of η towards η^r.

Since the control is implemented in a sample-and-hold setup, it is natural to look at the derivative of V_ζ between two adjacent sampling instants. For $t \in [k_i\delta, (k_i+1)\delta)$, $\dot{V}_\zeta$ can be further written as

$$\dot{V}_\zeta(t) = \dot{V}_\zeta^-[k_o] + D[k_o] + \left(\dot{V}_\zeta^-(t) - \dot{V}_\zeta^-[k_o]\right) + \left(D(t) - D[k_o]\right) \tag{4.28}$$

where

$$\begin{aligned}\dot{V}_\zeta^-[k_o] =& s_\zeta^\top[k_o]\left(\lambda_\zeta e_v[k_o] - g + \vartheta_\zeta^{-1}(t) r(\eta^r(t)))u_F(t) + \vartheta_{\zeta^0}(t) - \ddot{\zeta}^d(t)\right) + s_\zeta^\top[k_o]\tilde{\vartheta}_{\zeta^0}[k_o] \\ &- \frac{\tilde{\vartheta}_\zeta[k_o]}{\vartheta_\zeta(t)}\hat{\vartheta}_\zeta^{-1}[k_o] s_\zeta^\top[k_o] r(\eta^r(t)) u_F(t) \\ D[k_o] =& s_\zeta^\top[k_o]\vartheta_\zeta^{-1}[k_o]\left(r(\eta[k_o]) - r(\eta^r(t))\right)\left(u_F(t) + \vartheta_{\zeta^0}[k_o]\right)\end{aligned}$$

Substituting the adaptive control law(4.13) and (4.16) to the right-hand side of

$\dot{V}_\zeta^-[k_o]$ yields

$$\begin{aligned}\dot{V}_\zeta^-[k_o] =& s_\zeta^\top[k_o]\left(\lambda_\zeta e_v[k_o] - g + \hat{\vartheta}_\zeta^{-1}[k_o] r(\eta^r(t))) u_F(t) + \hat{\vartheta}_{\zeta^0}[k_o] - \ddot{\zeta}^d(t)\right) \\ \leqslant& - c_\zeta \|s_\zeta[k_o]\|^2 \end{aligned} \tag{4.29}$$

Then, we denote the stability regions for the sliding mode tracking errors s_ζ and $\breve{s}_\eta$ as $\Omega_{\rho_\zeta} = \{s_\zeta | \|s_\zeta\|^2 \leqslant \rho_\zeta\}$ and $\Omega_{\rho_\eta} = \{\breve{s}_\eta | \|\breve{s}_\eta\|^2 \leqslant \rho_\eta\}$. For all $s_\zeta \in \Omega_{\rho_\zeta}$, $\breve{s}_\eta \in \Omega_{\rho_\eta}$, $u_F \in \Omega_F$, using the Lipschitz continuity, it has that

$$\begin{aligned}\left|\dot{V}_\zeta^-(t) - \dot{V}_\zeta^-[k_o]\right| \leqslant& L_{s_\zeta}\|s_\zeta(t) - s_\zeta[k_o]\| + L_{\hat{\vartheta}_{\zeta^0}}\|\hat{\vartheta}_{\zeta^0}(t) - \hat{\vartheta}_{\zeta^0}[k_o]\| \\ &+ L_{\hat{\vartheta}_\zeta}|\hat{\vartheta}_\zeta(t) - \hat{\vartheta}_\zeta[k_o]| \end{aligned} \tag{4.30a}$$

$$|D(t) - D[k_o]| \leqslant L_{s_\zeta}^D\|s_\zeta(t) - s_\zeta[k_o]\| + L_{s_\eta}^D\|s_\eta(t) - s_\eta[k_o]\| \tag{4.30b}$$

and

$$\|s_\zeta(t) - s_\zeta[k_o]\| \leqslant M_{s_\zeta} N\delta \tag{4.30c}$$

$$\|s_\eta(t) - s_\eta[k_o]\| \leqslant M_{s_\eta} N\delta \tag{4.30d}$$

$$|\hat{\vartheta}_\zeta(t) - \hat{\vartheta}_\zeta[k_o]| \leqslant M_{\hat{\vartheta}_\zeta} N\delta \tag{4.30e}$$

$$\|\hat{\vartheta}_{\zeta^0}(t) - \hat{\vartheta}_{\zeta^0}[k_o]\| \leqslant M_{\hat{\vartheta}_{\zeta^0}} N\delta \tag{4.30f}$$

where L_{s_ζ}, $L_{\hat{\vartheta}_{\zeta^0}}$, $L_{\hat{\vartheta}_\zeta}$, $L_{s_\zeta}^D$, $L_{s_\eta}^D$, M_{s_ζ}, M_{s_η}, $M_{\hat{\vartheta}_\zeta}$ and $M_{\hat{\vartheta}_{\zeta^0}}$ are positive Lipschitz constants that can be obtained by evaluating the bounds of dynamics within the stability region and the input constraints.

Let $\alpha^* = \max_{t\geqslant 0} \|\partial r(\eta(t))/\partial \eta(t)\|$, we can obtain

$$\begin{aligned} D[k_o] &\leqslant \left(\frac{\overline{F}}{m} + g\right) \|s_\zeta[k_o]\| \, \|r(\eta[k_o]) - r(\eta^r[k_o])\| \\ &\leqslant \left(\frac{\overline{F}}{m} + g\right) \alpha^* \, \|s_\zeta[k_o]\| \, \|e_\eta[k_o]\| \\ &\leqslant \frac{2\alpha^*\left(\overline{F} + mg\right)}{m\lambda_\eta} \|s_\zeta[k_o]\| \, \|\breve{s}_\eta[k_o]\| \end{aligned} \tag{4.30g}$$

By substituting (4.29), 4.30a) together into (4.28), we have

$$\begin{aligned}\dot{V}_\zeta(t) \leqslant & \dot{V}_\zeta^-[k_o] + D[k_o] + \left|\dot{V}_\zeta^-(t) - \dot{V}_\zeta^-[k_o]\right| + |D(t) - D[k_o]| \\ \leqslant & -c_\zeta \left\|s_\zeta[k_o]\right\|^2 + \alpha \left\|s_\zeta[k_o]\right\| \left\|\check{s}_\eta[k_o]\right\| + \kappa_{\zeta_1} N\delta \end{aligned} \tag{4.31}$$

where $\kappa_{\zeta_1} = (L_{s_\zeta} + L^D_{s_\zeta})M_{s_\zeta} + L^D_{s_\eta} M_{s_\eta} + L_{\hat{\vartheta}_\zeta} M_{\hat{\vartheta}_\zeta} + L_{\hat{\vartheta}_{\zeta^0}} M_{\hat{\vartheta}_{\zeta^0}}$ and $\alpha = \frac{2\alpha^*\left(\overline{F}+mg\right)}{m\lambda_\eta}$. Since $k_o = N\lfloor k_i/N \rfloor$, we also have

$$\left\|s_\zeta[k_i]\right\|^2 - \left\|s_\zeta[k_o]\right\|^2 \leqslant 2\sqrt{\rho_\zeta} M_{s_\zeta}(N-1)\delta \tag{4.32a}$$

$$\left\|s_\zeta[k_o]\right\| \left\|\check{s}_\eta[k_o]\right\| - \left\|s_\zeta[k_i]\right\| \left\|\check{s}_\eta[k_i]\right\| \leqslant \sqrt{\rho_\eta} M_{s_\zeta}(N-1)\delta + \sqrt{\rho_\zeta} M_{s_\eta}(N-1)\delta \tag{4.32b}$$

Finally, the upper bound of $\dot{V}_\zeta(t)$ for $t \in [k_i\delta, (k_i+1)\delta)$ is

$$\dot{V}_\zeta(t) \leqslant -c_\zeta \left\|s_\zeta[k_i]\right\|^2 + \alpha \left\|s_\zeta[k_i]\right\| \left\|\check{s}_\eta[k_i]\right\| + \kappa_{\zeta_1} N\delta + \kappa_{\zeta_2}(N-1)\delta + c_\zeta \kappa_{\zeta_3}(N-1)\delta \tag{4.33}$$

where $\kappa_{\zeta_2} = \alpha(\sqrt{\rho_\eta} M_{s_\zeta} + \sqrt{\rho_\zeta} M_{s_\eta})$ and $\kappa_{\zeta_3} = 2\sqrt{\rho_\zeta} M_{s_\zeta}$.

B. Lyapunov Function for Inner-loop

We now move on to the closed-loop rotational dynamics. Firstly, choose a candidate for the Lyapunov function

$$V_\eta = \frac{1}{2}\|\check{s}_\eta\|^2 + \frac{1}{2k_\eta \vartheta_\phi}\tilde{\vartheta}_\phi^2 + \frac{1}{2k_\eta \vartheta_\theta}\tilde{\vartheta}_\theta^2 + \frac{1}{2k_\eta \vartheta_\psi}\tilde{\vartheta}_\psi^2 + \frac{1}{2k_{\eta^0}}\|\tilde{\vartheta}_{\eta^0}\|^2 \tag{4.34}$$

Differentiating V_η along the rotation dynamics (4.7b), we have

$$\begin{aligned}\dot{V}_\eta = & \frac{1}{\varepsilon}\check{s}_\eta^\top \left(\lambda_\eta \check{e}_\omega + f_\eta(\check{\omega}) + \varepsilon^2\left(\Theta_\eta^{-1} u_\tau + \vartheta_{\eta^0} - \ddot{\eta}^r\right)\right) + \frac{1}{k_\eta \vartheta_\phi}\tilde{\vartheta}_\phi \dot{\hat{\vartheta}}_\phi + \frac{1}{k_\eta \vartheta_\theta}\tilde{\vartheta}_\theta \dot{\hat{\vartheta}}_\theta \\ & + \frac{1}{k_\eta \vartheta_\psi}\tilde{\vartheta}_\psi \dot{\hat{\vartheta}}_\psi + \frac{1}{k_{\eta^0}}\tilde{\vartheta}_{\eta^0}^\top \dot{\hat{\vartheta}}_{\eta^0} \end{aligned} \tag{4.35}$$

By substituting the adaptive laws (4.23), we have $\dot{V}_\eta(t)$ for $t \in [k_i\delta, (k_i+1)\delta)$ as

$$\dot{V}_\eta(t) = \dot{V}_\eta[k_i] + \left(\dot{V}_\eta(t) - \dot{V}_\eta[k_i]\right) \tag{4.36}$$

where

$$\begin{aligned}\dot{V}_\eta[k_i] =&\frac{1}{\varepsilon}\breve{s}_\eta^\top[k_i]\left(\lambda_\eta\breve{e}_\omega[k_i]+f_\eta(\breve{\omega}[k_i])+\varepsilon^2\left(\Theta_\eta^{-1}(t)u_\tau(t)+\vartheta_{\eta^0}(t)-\ddot{\eta}^r(t)\right)\right)+\varepsilon\breve{s}_\eta^\top[k_i]\tilde{\vartheta}_{\eta^0}[k_i]\\&-\varepsilon\breve{s}_\eta^\top[k_i]\Theta_\eta^{-1}(t)\tilde{\Theta}_\eta[k_i]\hat{\Theta}_\eta^{-1}[k_i]u_\tau(t)\\=&\frac{1}{\varepsilon}\breve{s}_\eta^\top[k_i]\left(\lambda_\eta\breve{e}_\omega[k_i]+f_\eta(\breve{\omega}[k_i])+\varepsilon^2\left(\hat{\Theta}_\eta^{-1}[k_i]u_\tau(t)+\hat{\vartheta}_{\eta^0}[k_i]-\ddot{\eta}^r(t)\right)\right)\end{aligned}$$

Then, we substitute the rotation control law (4.25) to $\dot{V}_\eta[k_i]$. It can be obtained that

$$\dot{V}_\eta[k_i]=-\frac{c_\eta}{\varepsilon}\left\|\breve{s}_\eta[k_i]\right\|^2 \tag{4.37}$$

Similarly, according to the Lipschitz continuity, we have

$$\begin{aligned}\left|\dot{V}_\eta^-(t)-\dot{V}_\eta^-[k_i]\right|\leqslant&L_{s_\eta}\|\breve{s}_\eta(t)-\breve{s}_\eta[k_i]\|+L_{\hat{\vartheta}_{\eta^0}}\|\hat{\vartheta}_{\eta^0}(t)-\hat{\vartheta}_{\eta^0}[k_i]\|+L_{\hat{\vartheta}_\phi}|\hat{\Theta}_\phi(t)-\hat{\vartheta}_\phi[k_i]|\\&+L_{\hat{\vartheta}_\theta}|\hat{\Theta}_\theta(t)-\hat{\vartheta}_\theta[k_i]|++L_{\hat{\vartheta}_\psi}|\hat{\Theta}_\psi(t)-\hat{\vartheta}_\psi[k_i]|\end{aligned} \tag{4.38a}$$

and

$$\|\breve{s}_\eta(t)-\breve{s}_\eta[k_i]\|\leqslant M_{s_\eta}\delta \tag{4.38b}$$

$$\|\hat{\vartheta}_{\eta^0}(t)-\hat{\vartheta}_{\eta^0}[k_i]\|\leqslant M_{\hat{\vartheta}_{\eta^0}}\delta \tag{4.38c}$$

$$|\hat{\vartheta}_\phi(t)-\hat{\vartheta}_\phi[k_i]|\leqslant M_{\hat{\vartheta}_\phi}\delta \tag{4.38d}$$

$$|\hat{\vartheta}_\theta(t)-\hat{\vartheta}_\theta[k_i]|\leqslant M_{\hat{\vartheta}_\theta}\delta \tag{4.38e}$$

$$|\hat{\vartheta}_\psi(t)-\hat{\vartheta}_\psi[k_i]|\leqslant M_{\hat{\vartheta}_\psi}\delta \tag{4.38f}$$

where L_{s_η}, $L_{\hat{\vartheta}_{\eta^0}}$, $L_{\hat{\vartheta}_{\tau m_{p,q,r}}}$, M_{s_η}, $M_{\hat{\vartheta}_{\eta^0}}$, and $M_{\hat{\vartheta}_{\tau m_{p,q,r}}}$ are positive Lipschitz constants.

Substituting (4.37) and (4.38) to (4.36) delivers

$$\dot{V}_\eta(t)\leqslant-\frac{c_\eta}{\varepsilon}\left\|\breve{s}_\eta[k_i]\right\|^2+\kappa_\eta\delta \tag{4.39}$$

where $\kappa_\eta=L_{s_\eta}M_{s_\eta}+L_{\hat{\vartheta}_{\eta^0}}M_{\hat{\vartheta}_{\eta^0}}+L_{\hat{\vartheta}_\phi}M_{\hat{\vartheta}_\phi}+L_{\hat{\vartheta}_\theta}M_{\hat{\vartheta}_\theta}+L_{\hat{\vartheta}_\psi}M_{\hat{\vartheta}_\psi}$.

C. Composite Lyapunov Function

By adding V_ζ and V_η up, we can construct a composite Lyapunov function to be

the Lyapunov function candidate for the full UAV dynamics

$$\begin{aligned} V =& V_\zeta + V_\eta \\ =& \frac{1}{2}\left\|s\right\|^2 + \frac{1}{2k_\zeta \vartheta_\zeta}\tilde{\vartheta}_\zeta^2 + \frac{1}{2k_{\zeta^0}}\|\tilde{\vartheta}_{\zeta^0}\|^2 + \frac{1}{2k_\eta \vartheta_\phi}\tilde{\vartheta}_\phi^2 + \frac{1}{2k_\eta \vartheta_\theta}\tilde{\vartheta}_\theta^2 \\ &+ \frac{1}{2k_\eta \vartheta_\psi}\tilde{\vartheta}_\psi^2 + \frac{1}{2k_{\eta^0}}\|\tilde{\vartheta}_{\eta^0}\|^2 \end{aligned} \tag{4.40}$$

where $s = \left[s_\zeta^\top \ \breve{s}_\eta^\top\right]^\top$ can be regarded as the full-state sliding mode tracking error.

The composite Lyapunov function V is in the positive-definite quadratic form of the tracking and estimation errors, so it can be upper bounded by

$$V \leqslant \frac{1}{2}\left\|\begin{bmatrix}\|s_\zeta\| \\ \|\breve{s}_\eta\|\end{bmatrix}\right\|^2 + \mu \tag{4.41}$$

where $\mu = \frac{1}{2k_{F)m}m}\rho_{\vartheta_\zeta} + \frac{3}{2\kappa_\eta \lambda_{\min}(I)}\rho_{\vartheta_\eta} + \frac{1}{2\kappa_{\zeta^0}}\rho_{\vartheta_{\zeta^0}} + \frac{1}{2\kappa_{\eta^0}}\rho_{\vartheta_{\eta^0}}$, in which ρ_{ϑ_ζ}, ρ_{ϑ_η}, $\rho_{\vartheta_{\zeta^0}}$ and $\rho_{\vartheta_{\eta^0}}$ are used to define the stability regions of the estimation errors as $\Omega_{\rho_{\vartheta_\zeta}} = \{\tilde{\vartheta}_\zeta | |\tilde{\vartheta}_\zeta|^2 \leqslant \rho_{\vartheta_\zeta}\}$, $\Omega_{\rho_{\vartheta_\eta}} = \{\tilde{\vartheta}_\eta | |\tilde{\vartheta}_\eta|^2 \leqslant \rho_{\vartheta_\eta}\}$, $\Omega_{\rho_{\vartheta_{\zeta^0}}} = \{\tilde{\vartheta}_{\zeta^0} | \|\tilde{\vartheta}_{\zeta^0}\|^2 \leqslant \rho_{\vartheta_{\zeta^0}}\}$, $\Omega_{\rho_{\vartheta_{\eta^0}}} = \{\tilde{\vartheta}_{\eta^0} | \|\tilde{\vartheta}_{\eta^0}\|^2 \leqslant \rho_{\vartheta_{\eta^0}}\}$.

4.4.2 Closed-loop Stability and Stability Conditions

With the composite Lyapunov function, we can prove the closed-loop stability of the proposed dual-loop and dual-rate hierarchical UAV control system using the Lyapunov stability theory. Theorem 1 below summarizes the conclusion that the closed-loop tracking errors are always bounded and ultimately converge to a small region around the origin with sufficient stability conditions.

Theorem 4.1. *Consider the closed-loop UAV dynamics (4.1a) and (4.1b), where the outer-loop controlled by the fault-tolerant MPC (4.14) with the adaptive law (4.13) and the inner-loop controlled by the rotation control law (4.22) with the adaptive estimating law (4.23). Given $\rho_s \in \mathbb{R}^+$ and $\epsilon \in \mathbb{R}^+$, if the selection of the inner-loop sampling period δ, the outer-loop and inner-loop sampling period ratio N, and the perturbation parameter ε satisfy the following conditions*

(C1) $\frac{1}{\varepsilon}c_\zeta c_\eta > \frac{\alpha^2}{4}$

(C2) $\frac{\epsilon}{\delta} + \delta\left(\kappa_{\zeta_1}N + \kappa_{\zeta_2}(N-1) + \kappa_\eta\right) \leqslant 2\lambda_{\min}(C)\left(\rho_s - \mu\right) - c_\zeta \kappa_{\zeta_3}(N-1)\delta$

then for all $t > 0$,

$$\limsup_{t\to\infty} \|s(t)\| \leqslant \sqrt{\rho_{\min}} \tag{4.42}$$

with $\rho_{\min} = \max\left\{ \|s(t)\|^2 \middle| V(t) \leqslant \rho_s \right\}$.

Proof. We firstly prove that under the developed control design, the composite tracking error of the UAV dynamics will converge to a small set Ω_{ρ_s} after a finite number of sampling periods. Following the obtained upper bounds of V_p and V_η given by the right-hand sides of (4.33) and (4.39), the time derivative of the composite Lyapunov function for all $t \in [k_i\delta, (k_i+1)\delta)$ is derived if $s_\zeta(0) \in \Omega_{\rho_\zeta}$ and $\breve{s}_\eta(0) \in \Omega_{\rho_\eta}$, as given below

$$\dot{V}(t) \leqslant -\begin{bmatrix} \|s_\zeta k_i]\| & \|\breve{s}_\eta[k_i]\| \end{bmatrix} C \begin{bmatrix} \|s_\zeta[k_i]\| \\ \|\breve{s}_\eta[k_i]\| \end{bmatrix} + \kappa_{\zeta_1} N\delta + \kappa_{\zeta_2}(N-1)\delta + c_\zeta \kappa_{\zeta_3}(N-1)\delta + \kappa_\eta \delta \tag{4.43}$$

where

$$C = \begin{bmatrix} c_\zeta & -\frac{\alpha}{2} \\ -\frac{\alpha}{2} & \frac{1}{\varepsilon} c_\eta \end{bmatrix}$$

The symmetric matrix C is positive definite if condition (C1) holds. In this case, we have

$$\dot{V}(t) \leqslant -\lambda_{\min}(C) \left\| \begin{bmatrix} \|s_p\zeta[k_i]\| \\ \|\breve{s}_\eta[k_i]\| \end{bmatrix} \right\|^2 + \kappa_{\zeta_1} N\delta + \kappa_{\zeta_2}(N-1)\delta + c_\zeta \kappa_{\zeta_3}(N-1)\delta + \kappa_\eta \delta \tag{4.44}$$

By recalling the upper bound of V given in (4.41), we have $\left\| \begin{bmatrix} \|s_\zeta\| & \|\breve{s}_\eta\| \end{bmatrix}^\top \right\|^2 \geqslant 2(V-\mu)$. Then, if condition (C2) is satisfied and $s[k_i] \in \left\{ s[k_i] \middle| \rho_s \leqslant V \leqslant \frac{1}{2}\rho_\zeta + \frac{1}{2}\rho_\eta + \mu \right\}$, then the following inequality holds for all $t \in [k_i\delta, (k_i+1)\delta)$

$$\dot{V}(t) \leqslant -\frac{\epsilon}{\delta} \tag{4.45}$$

By integrating it over $[k_i\delta, (k_i+1)\delta)$, we obtain $V(t) \leqslant V[k_i]$ for $t \in [k_i\delta, (k_i+1)\delta)$ and $V[k_i+1] \leqslant V[k_i] - \epsilon$. Therefore, it can be proven that the full-state tracking error vector s converges to Ω_{ρ_s} in a finite number of samplings without leaving the stability region. The set Ω_{ρ_s} is a level set of the Lyapunov function V, defined as

$\Omega_{\rho_s} = \{ s | \, V \leqslant \rho_s \}$.

Then, we prove that the closed-loop tracking error is ultimately bounded in the terminal set $\Omega_{\rho_{\min}}$. Because of the definition of $\rho_{\min}$, it can be concluded that once the composite tracking error s enters Ω_{ρ_s}, then it will remain inside $\Omega_{\rho_{\min}} = \{ s | \, V \leqslant \rho_{\min} \}$ for all time. Finally, we obtain (4.42) and complete the proof of Theorem 1. □

From the obtained stability conditions (C1) and (C2), it can be concluded that, given any $\rho_s \in \mathbb{R}^+$ (the size of the region that the tracking errors are expected to converge to) and any $\epsilon \in \mathbb{R}^+$ (the desired convergence step size), we can find appropriate combinations of sampling periods δ and $N\delta$, control gains c_ζ and c_η and singular perturbation parameters ε to guarantee the convergence of s_ζ and $\check{s}_\eta$. The establishment of Theorem 1 demonstrates the fulfillment of the sliding mode error convergence objectives (O1') and (O2'), and thus the tracking control objectives (O1) and (O2).

4.5 Simulation Study

To verify the effectiveness of the dual-loop dual-rate hierarchical control design proposed in this chapter, simulation studies are taken on a UAV model with the physical parameters listed in Table 4.1.

Table 4.1: Physical parameters of the UAV in simulations [1]

Parameters	Values
l — distance from rotors to the center	0.30486(m)
m — mass of the UAV	2.618(kg)
g — gravitational acceleration	9.81(m/s^2)
I_{xx} — moment of inertia around e_x^e	0.043467(kg·m^2)
I_{yy} — moment of inertia around e_y^e	0.043467(kg·m^2)
I_{zz} — moment of inertia around e_z^e	0.063267(kg·m^2)

4.5.1 Parameter Selection

In terms of the selection of user-defined parameters and sampling rates, the stability conditions (C1) and (C2) obtained in Section 4.4 play important roles. The following are the steps for parameter selection:

[1] UVic CfAR Tarot Quad Copter (NO. 0106-0000).

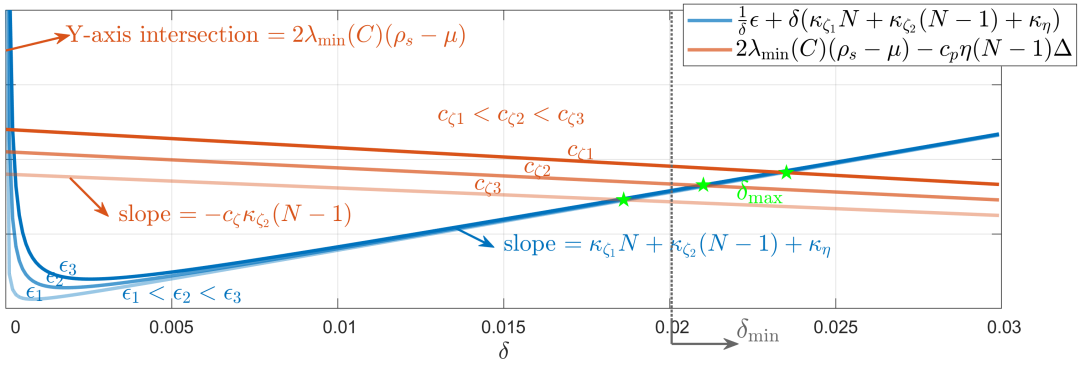

Figure 4.2: Graphical expression of stability condition (C2)

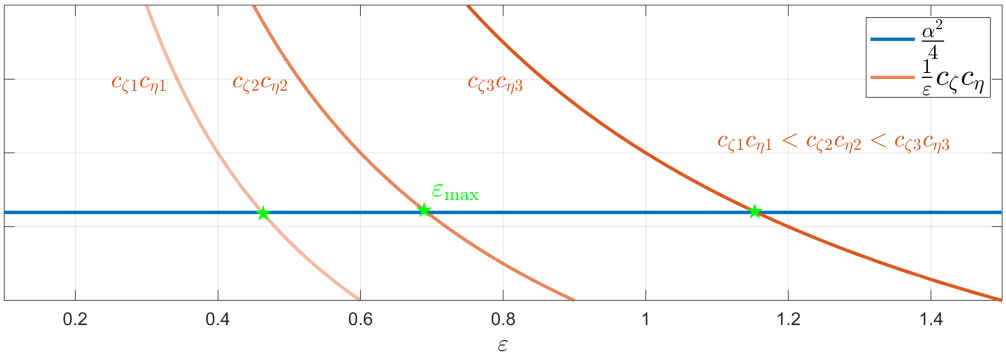

Figure 4.3: Graphical expression of stability condition (C1)

Step 1: Choose the stability regions for tracking errors and estimation errors. Determine the terminal region of the tracking error ρ_s and the desired convergence step size ϵ. Choose λ_ζ and λ_η for the Hurwitz polynomials. Choose the adaptive estimation gains k_ζ and k_η appropriately so that the bias μ can be small.

Step 2: Determine the minimum sampling period for the outer-loop according to the required computation time for solving the MPC problem and the scaling ratio N between the two sampling periods. Thereafter, pose a lower bound on the sampling period of the inner-loop, denoted as $\delta_{\min}$.

Step 3: Choose a candidate for the singular perturbation parameter ε. Calculate the Lipschitz constants and determine the values of κ_{ζ_1}, κ_{ζ_2}, κ_{ζ_3} and κ_η. By applying the graphical method shown in Fig. 4.2, we have that there away exists a maximal sampling period $\delta_{\max}$ for a given c_ζ such that (C2) is satisfied. Furthermore, the larger c_ζ is, the larger the value of $\delta_{\max}$. Finally,

ensuring $\delta_{\max} > \delta_{\min}$ delivers an admissible minimum of c_p.

Step 4: As c_ζ is selected within the admissible range and c_η is selected in the same order of c_ζ, the maximum of the singular perturbation parameter $\varepsilon_{\max}$ can be obtained by the graphic expression of (C1) as shown in Fig. 4.3. Evaluate whether the candidate of ε selected in **Step 3** is less than $\varepsilon_{\max}$. If so, the procedure is ended; otherwise, go back to **Step 3** and redo the selection.

Following the above steps, we choose $\rho_\zeta = 0.5$, $\rho_\eta = 0.01$, $\rho_{\vartheta_\zeta} = \rho_{\vartheta_{\zeta^0}} = \rho_{\vartheta_{\eta^0}} = 0.01$, $\rho_{\vartheta_\eta} = 0.0001$, $\rho_s = 0.4$, $\epsilon = 0.001$, $\lambda_\zeta = 0.5$, $\lambda_\eta = 1$, $k_\zeta = 0.1$, $k_{\zeta^0} = 1$, $k_\eta = 0.001$, $k_{\eta^0} = 5$. Then, $\mu \approx 0.06$. A feasible time interval for solving the MPC optimization problem is 0.1 s, and N is selected as 5. Thus, the sampling period of the inner-loop is selected as the minimal feasible value—that is, $\delta = 0.02$ s. The singular perturbation parameter is chosen as $\varepsilon = 0.2$. By calculating the Lipschitz constants, we obtain that $\alpha \approx 10.2$, $\kappa_{\zeta_1} \approx 22.99$, $\kappa_{\zeta_2} \approx 5.2$, $\kappa_{\zeta_3} \approx 1.54$, $\kappa_\eta \approx 19.52$. We then have that the control gain c_ζ should be at least 7 for ensuring sufficient stability conditions.

Remark 4.4. *Considering that the stability conditions we obtained in the closed-loop analysis are sufficient but not necessary, we still need to tune the control parameters further according to the actual response in simulations. The conservatism of the theoretical analysis mainly results from the calculation of Lipschitz constants for the nonlinear dynamics. The constants κ_{ζ_1}, κ_{ζ_2}, κ_{ζ_3}, and κ_η could be large since they are obtained along the boundaries of the stability regions, leading to an over large slope of the blue line in Fig. 4.2. Therefore, the actual admissible region of the control gain can be further expanded in practice.*

Remark 4.5. *Notice that the convergence regions we choose for calculating the Lipschitz constants are small, which intends to mitigate the conservatism of Lipschitz approximations. The actual regions of convergence could be much larger than the theoretical values, which will be investigated through simulation experiments later in Sections 6.2.1 and 6.2.2.*

4.5.2 Simulation Results

In the simulations, the desired flight trajectory to be tracked by the UAV is set as a downward spiral from the initial position to the base point on the ground ($[0\ 0\ 0]^\top$). The initial conditions of the UAV system are $\zeta(0) = [11\ 0\ 20]^\top$ and $\eta(0) = [0\ 0\ 0]^\top$,

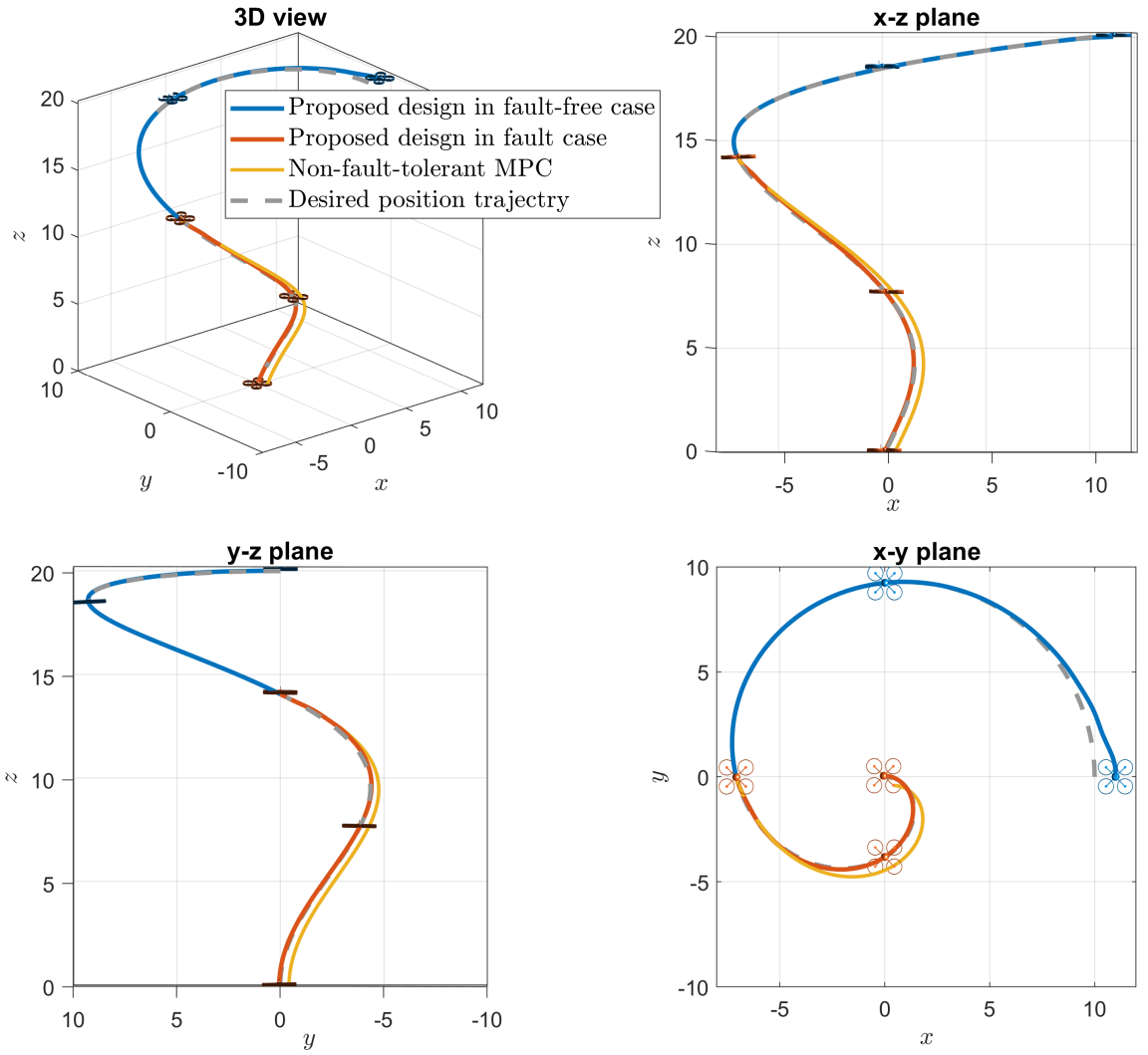

Figure 4.4: Translational motion of the UAV

$v(0) = [0\ 2\ 0]^\top$ and $\omega(0) = [0\ 0\ 0]^\top$. The initial values of the adaptive estimates are selected as $\hat{\vartheta}_\zeta(0) = m$, $\hat{\vartheta}_{\zeta^0}(0) = [0\ 0\ 0]^\top$, $\hat{\Theta}_\eta(0) = \text{diag}([1, 1, 1])$, $\hat{\vartheta}_{\eta^0}(0) = [0\ 0\ 0]^\top$. The input constraints are set as $\overline{F} = 2$ N and $\overline{\tau} = 2$ N·m.

The control parameters are chosen as: $\lambda_\zeta = 0.5$, $\lambda_\eta = 1$, $k_\zeta = 0.1$, $k_{\zeta^0} = 1$, $k_\eta = 0.001$, $k_{\eta^0} = 5$, $\varepsilon = 0.2$, $N = 5$, $c_\zeta = c_\eta = 7$. The prediction horizon of the MPC problem is selected as $N_p = 4$. The weighting matrices and scalar in the objective function are $Q = \text{diag}\{10, 10, 20\}$, $R_\eta = \text{diag}\{10, 10, 10\}$ and $R_F = 0.1$.

A. Fault Tolerance Capability

To manifest the fault-tolerant capability of the proposed design, we set up the

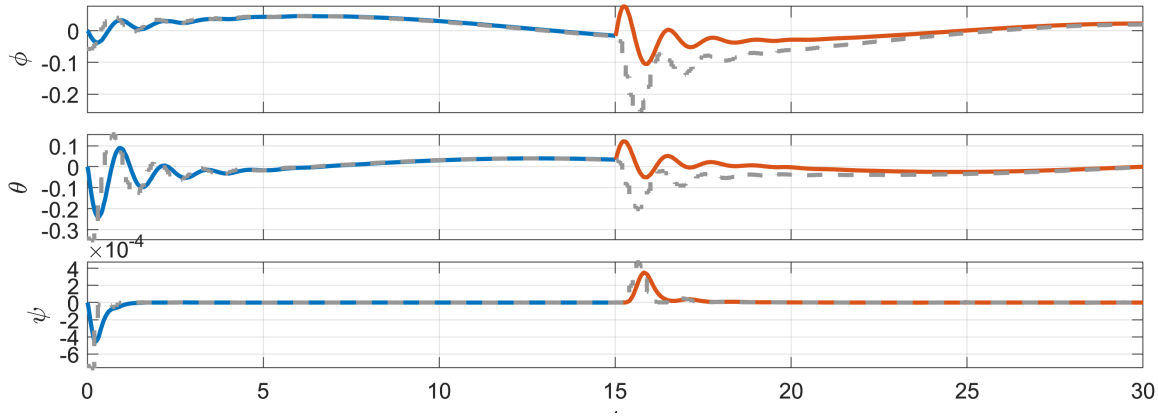

Figure 4.5: Rotational motion of the UAV

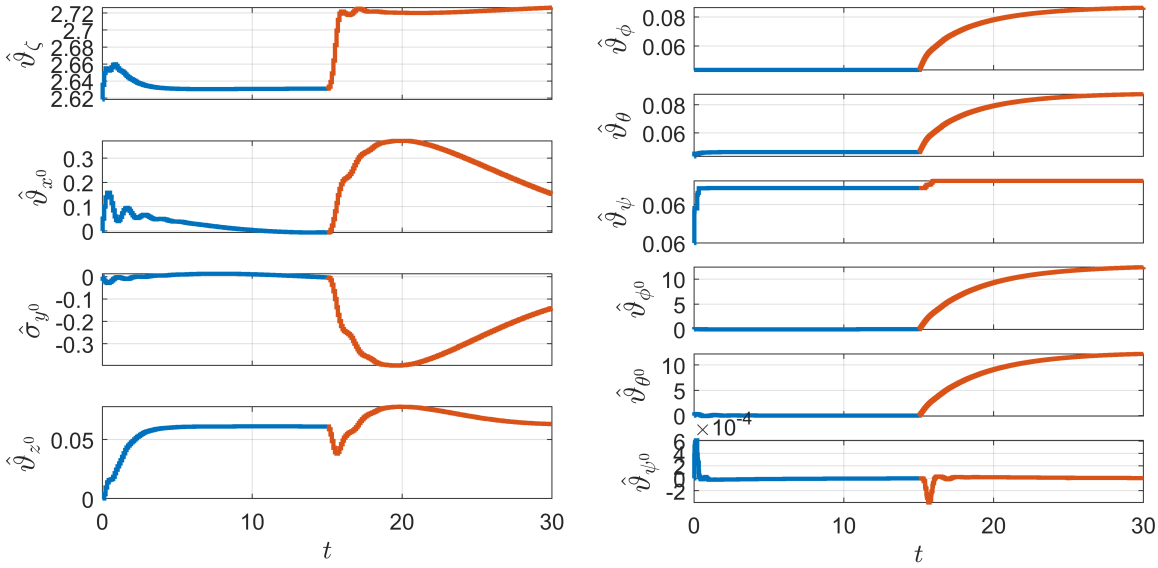

Figure 4.6: Online parameter estimation

simulation as

- *Fault-free case:* When 0 s $\leqslant t <$ 15 s, the UAV operates in a fault-free case with all the four rotors working properly.
- *Actuator fault case:* When 15 s $\leqslant t <$ 30 s, an over-voltage failure happens on two of the rotors, causing a sudden increase of 1 N on the rotor's thrust force F_1 and F_4. Also, a loss-of-effectiveness fault takes place, causing a 10% reduction in forces and torques.
- *Comparison method:* The performance of a control design without the adaptive fault parameter estimator is also tested for comparison.

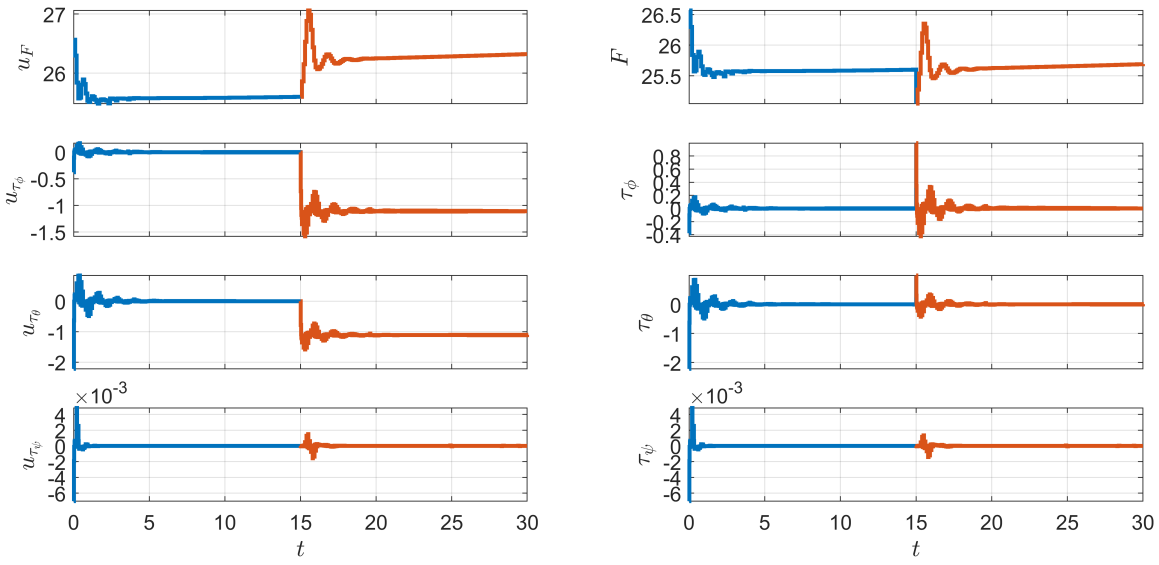

Figure 4.7: Control commands (left); Actually applied force and torques (right)

The simulation results are shown in Figs. 4.4-4.7, where system responses in fault-free and faulty cases are indicated by the blue and orange lines, respectively. Figs. 4.4 and 4.5 show the translational and rotational motions of the UAV. It can be seen that the UAV tracks the desired trajectory with high accuracy. At 15 s, when the actuator fault happens, the tracking performance is slightly degraded but then recovers in a short time. The compensation for the unexpected faults is achieved by the adaptive parameter estimation, as shown by Fig. 4.6. The left half of Fig. 4.7 shows the control commands generated by the controllers, and the right half shows the force and torques that are actually applied to the UAV. It is obvious that the input constraints are satisfied at all times. The tracking performance of the comparison method is indicated by the yellow line in Fig. 4.4. Without the adaptive estimator, the control performance obviously deteriorates after the fault occurs at 15 s and gets even worse over time. Therefore, by this simulation test, we can verify the effectiveness of the proposed design in terms of fault tolerance.

From the above simulation studies, we verify the fault-tolerant control performance of the proposed design in the presence of loss-of-effectiveness and additive faults. Furthermore, simulations with loss-of-effectiveness and additive faults in different degrees are conducted to investigate the maximum tolerable capability. As a result, the maximal tolerable multiplicative fault is up to 17%. The developed fault-tolerant scheme can compensate for, at most, a sudden change in the thrust force of 3.2 N or in the rotation torque of 1.6 N·m.

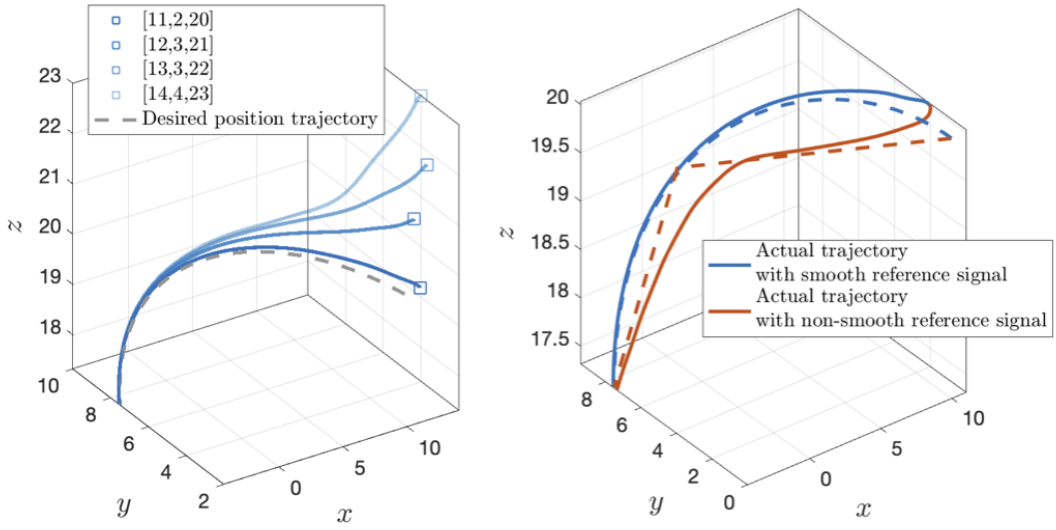

Figure 4.8: Tracking performance with different initial positions (left); Tracking performance with smooth and non-smooth reference trajectories (right)

B. Region of Convergence

In this test, trajectory tracking performances with different initial conditions are also evaluated. The obtained simulation results are given in the left half of Fig. 4.8. It can be seen that vehicles starting far always from the desired trajectory can still fulfill the tracking control objective, which implies that the actual regions of convergence are larger than those we selected for theoretical analysis.

Furthermore, we test the tracking performance for smooth and non-smooth reference signals, shown by the right half of Fig. 4.8. Both types of trajectories can be tracked under the control of the proposed design.

C. Adaptability to External Disturbances

The adaptability of the proposed design to external disturbances is also studied. A gust of wind is simulated and added to the UAV dynamics. The obtained result is shown in Fig. 4.9, in which the red line represents the tracking performance of a comparison method without the adaptive estimation mechanism. Fig. 4.9 shows the adaptability of our method for external disturbances.

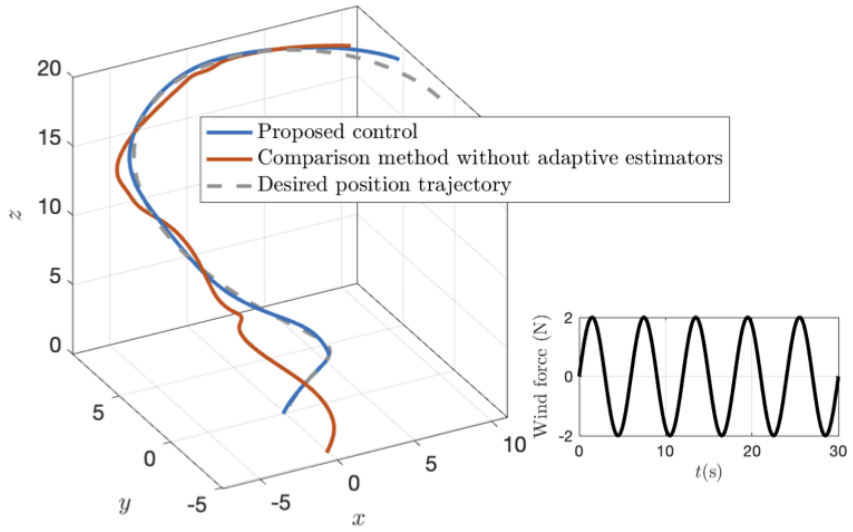

Figure 4.9: Tracking performance under wind gust

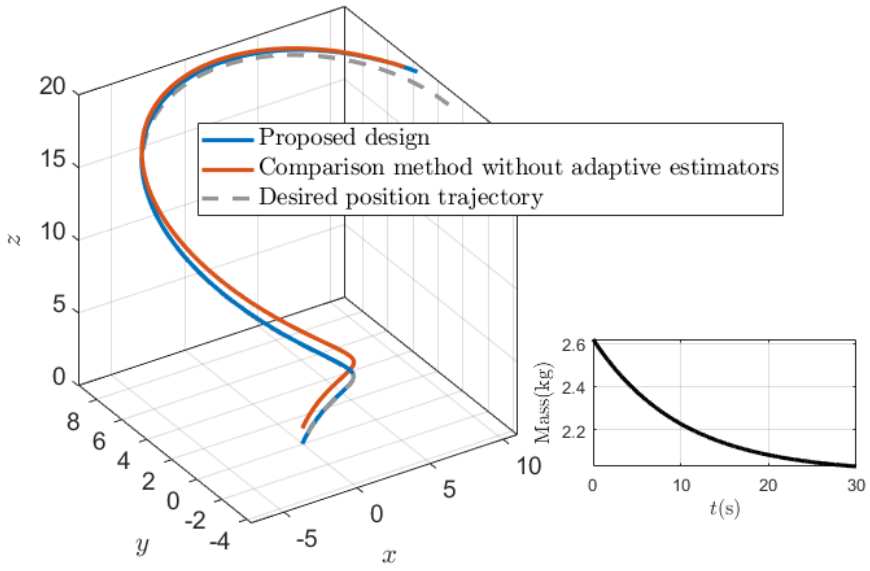

Figure 4.10: Tracking performance with mass changing

D. Adaptability to Parameter Changes

The proposed adaptive control scheme can also accommodate the changes in the vehicle's mass and the moment of inertia. These model parameters are included in the uncertain parameters ϑ_ζ, ϑ_{ζ^0}, Θ_η and ϑ_{η^0}, thereby being estimated by the developed adaptive laws. To simulate the practical situation of fuel consumption, tests are conducted when the mass of the vehicle keeps reducing during the flight. Results are

shown in Fig. 4.10, which verifies the adaptability to model uncertainties.

4.6 Conclusions

This chapter presents a dual-loop and dual-rate hierarchical control design for quadrotors subject to unexpected actuator faults. An adaptive parameter estimator and a Lyapunov-based MPC framework are integrated to develop the outer-loop translation controller. Singular perturbation theory is applied to legitimize the dual-loop design and explicitly characterize the high-gain property of the inner-loop rotation controller. The developed translation and rotation controllers are all implemented in a sample-and-hold fashion with different sampling periods. We provide a rigorous stability analysis of the sampled-data and dual-rate control systems with explicitly characterized stability conditions. Comprehensive simulation studies verify the effectiveness of the proposed dual-rate hierarchical fault-tolerant control design.

Chapter 5

Distributed Fault-tolerant Model Predictive Control of Multi-UAV Formation for Dynamic Leader Tracking with Actuator Faults

This chapter intends to extend the result from Chapter 4 to multi-UAV formation control in follower-leader tracking scenarios. This extension is nontrivial, given the distributed nature of control computation in multi-UAV systems and the partial availability of the leader's information. Hence, a novel distributed Lyapunov-based MPC strategy is developed in this chapter with reduced conservatism, lower computation amount, and moderate communication frequency.

5.1 Introduction

Multi-UAV formation, the most common and useful type of multi-UAV cooperation, involves a mix of several tasks [165], including controlling the positions and orientations of UAVs to attain a particular geometric configuration, as well as moving the whole formation as a rigid entity with or without a reference. While numerous studies have been made in the field of formation control [3,166–169], it is worth noting that a large portion of them consider linear dynamics like the double integrator and employ linear controllers for the sake of theoretical tractability. Additionally, many investigations do not account for input constraints and control optimality. However,

the incorporation of model nonlinearity and input constraints is imperative for an accurate formulation of real-world problems. The inclusion of optimality is also essential to fully exploit available control resources while satisfying input constraints.

An appealing framework for formation control is DMPC, which inherits the advantages of centralized MPC, including systematic handling of hard constraints, optimized control performance, inherent robustness, and the ability to cope with nonlinear multi-variable systems [170]. In addition, the distributed implementation fashion of DMPC effectively distributes the computation workload, further enhancing its appeal and practicality [58]. Numerous DMPC methods have been proposed, as summarized in review papers such as [59, 60]. However, existing DMPC results may encounter limitations when tackling the distinctive challenges posed by multi-UAV formation problems. First of all, the computation resources of vehicle-embedded hardware are limited, while the proposed DMPC methods with terminal constraints demand a sufficiently long prediction horizon and, thereby, a large computation amount to ensure feasibility. Secondly, the majority of DMPC methods are tailored to address the cooperative regulation problem that drives all agents toward a prior-known set point [73, 171]. These methods underlie an implicit assumption regarding the communication graph that each agent in the system is directly linked to the leader. Such an assumption is not true in the context of formation control, where the leader's information is often only available to a portion of the followers. A notable exception, proposed in [172], does not require globally known leader information but is only applicable to multi-vehicle platoon scenarios for tracking a constant-speed leader. Moreover, current existing DMPC algorithms usually entail a substantial amount of information exchange, iteratively [68–70] or sequentially [71, 72], to attain the most up-to-date optimization results from neighbors [73] for global stability. However, this causes a contradiction between the high communication workload and the limited communication bandwidth of wireless communication networks employed in multi-UAV systems.

In this chapter, a fault-tolerant DMPC framework is proposed for multi-UAV formation tracking control with input constraints and unexpected actuator faults. In the control system, each UAV is equipped with its own controller, adopting a hierarchical structure of three sequentially connected and separately designed control loops. The outer loop, responsible for translation control of formation tracking, is developed by integrating the Lyapunov-based MPC method with an adaptive parameter estimator. This loop aims to achieve optimal and fault-tolerant formation tracking performance.

The intermediate loop is developed locally to facilitate the convergence of actual rotation angles toward the desired ones calculated by the outer loop. Finally, the inner loop focuses on the determination of the torque control actions to ensure rapid convergence of the angular velocities. By using the Lyapunov stability theory, the closed-loop stability of the entire multi-UAV system is proven, and sufficient stability conditions are established. Simulation results are provided to substantiate the effectiveness of the proposed design in formation tracking and fault-tolerance. The main contributions of this work are as follows:

- More sophisticated UAV models containing both the translation and rotation motions are considered in this chapter. This deviates from the majority of existing formation control results that treat UAVs as first- or second-order integrators. The inclusion of both translation and rotation control is a nontrivial extension, as it involves the adoption of a triple-loop hierarchical control architecture, which is developed based on the multi-time-scale characteristic of the 6-DoF UAV dynamics.

- The proposed Lyapunov-based MPC design improves existing DMPC algorithms by relaxing the connectivity requirements of the multi-UAV system in two ways: (1) The vehicles are connected via a directed graph instead of an undirected one; (2) Instead of requiring a commonly known set-point, the leader's information is only required to be known by a portion of the followers.

- The proposed design also introduces several improvements over existing Lyapunov-based DMPC results in terms of communication and computation efficiency. First of all, it reduces the amount of inter-vehicle interaction by incorporating only one round of information flow along the directed communication graph per control update. In addition, the design utilizes predicted trajectories of neighboring vehicles instead of waiting for their current optimization results to formulate the MPC subproblems. This allows all vehicles to independently perform their calculations and simultaneously update their control actions, which enables faster control update rates.

- Through the integration of Lyapunov-based MPC and adaptive parameter estimation, the unexpected actuator faults are accommodated. Moreover, we treat the mismatch between the actual and assumed trajectories as an unknown parameter whose impacts can be mitigated by adaptive estimation as well.

- The closed-loop stability of the entire multi-UAV control system is evaluated rigorously. By taking account of the interconnections between each control loop and the uncontrolled inter-sample system evolution, we explicitly characterize how the sampling rate of the outer-loop impacts the closed-loop control performance. Based on the established stability conditions, this design allows for the joint tuning of control parameters and sampling periods instead of assuming a sufficiently fast control updating rate.

The rest of this chapter is organized as follows. We first formulate the multi-UAV formation tracking control problem in Section 5.2. The distributed control design is presented in Section 5.3. The closed-loop stability of the entire multi-UAV system is analyzed in Section 5.4. Simulations and conclusions are provided in Section 5.5 and 5.6.

5.2 Problem Formulation

This section provides a mathematical formulation of the multi-UAV formation tracking control problem. In 5.2.1, we present the dynamics model of individual UAVs and characterize the communication graph connecting them. In 5.2.2, we give the modeling of actuator faults. Finally, in 5.2.3, we introduce the virtual leader, specify the connectivity between the leader and the multi-UAV system, and define leader-follower tracking errors to formulate the formation tracking control objective.

5.2.1 Multi-UAV System

Consider a group of M independently actuated rotary-wing UAVs. The dynamics of each vehicle in the network can be described by the following differential equations:

$$\dot{\zeta}_i = v_i \tag{5.1a}$$

$$m_i \dot{v}_i = -m_i g + r(\eta_i) F_i \tag{5.1b}$$

$$\dot{\eta}_i = W(\eta_i)\omega_i \tag{5.1c}$$

$$J_i \dot{\omega}_i = -\omega_i \times J_i \omega_i + \tau_i \tag{5.1d}$$

with

$$r(\eta_i) = \begin{bmatrix} \sin\theta_i \cos\psi_i \cos\phi_i + \sin\psi_i \sin\phi_i \\ \sin\theta_i \sin\psi_i \cos\phi_i - \cos\psi_i \sin\phi_i \\ \cos\theta_i \cos\phi_i \end{bmatrix}$$

$$W(\eta_i) = \begin{bmatrix} 1 & \sin\phi_i \tan\theta_i & \cos\psi_i \tan\theta_i \\ 0 & \cos\phi_i & -\sin\phi_i \\ 0 & \sin\phi_i/\cos\theta_i & \cos\phi_i/\cos\theta_i \end{bmatrix}$$

where the subscript i indicates that the variable is associated with the ith UAV and $i \in \{1, 2, \cdots, M\}$; $\zeta_i = [x_i \ y_i \ z_i]^\top$ describes the position of UAV i by 3-dimensional coordinates of the center of mass in the earth-fixed frame; $\eta_i = [\phi_i \ \theta_i \ \psi_i]^\top$ describes the rotation of the body-fixed frame with respect to the earth-fixed frame by Euler angles; $v_i = [v_{x_i} \ v_{y_i} \ v_{z_i}]^\top$ and $\omega_i = [\omega_{p_i} \ \omega_{q_i} \ \omega_{r_i}]^\top$ represent the linear and angular velocities, respectively; m_i is the vehicle mass; $g = [0 \ 0 \ g^z]^\top$ and g^z is the gravitational acceleration; $J_i = \text{diag}(J_{xx_i}, J_{yy_i}, J_{zz_i})$ is a diagonal matrix of the moments of inertia.

The control inputs of the UAV dynamics are F_i and τ_i: F_i is the total upward-lifting force perpendicular to the plane of the vehicle body; $\tau_i = [\tau_{p_i} \ \tau_{q_i} \ \tau_{r_i}]^\top$ is a vector of the rotation torques. Given the inherent physical limitations of UAV rotors, their maximum rotation rates set explicit limitations on the control excitability, particularly affecting the lift force F_i. This limitation is formalized through the input constraint defined as

$$F_i \in \left\{ F_i \mid 0 \leqslant F_i \leqslant \overline{F}_i \right\} \tag{5.2}$$

where $\overline{F}_i \in \mathbb{R}^+$ represents the maximum achievable lift force the rotors can generate, which is greater than the force required to maintain equilibrium, $F_i^* = m_i g$.

The exchange of information among the M vehicles is captured by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as previously introduced in Section 2.1.2. Here, $\mathcal{V} = \{1, 2, \cdots, M\}$ is the set of nodes with each node representing a vehicle. $\mathcal{E} = \{(j, i) | i, j \in \mathcal{V}, i \neq j\}$ is the set of edges and $(j, i) \in \mathcal{E}$ means there is a communication link from UAV j to UAV i. $\mathcal{A} = [a_{ij}]$ for $i, j \in \mathcal{V}$ is the adjacency matrix, where $a_{ij} = 1$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. $\mathcal{L} = [l_{ij}]$ is the Laplacian matrix, where $l_{ii} = \sum_{j=1}^{M} a_{ij}$ and $l_{ij} = -a_{ij}$ for $i \neq j$.

To further clarify the directed information flow in $\mathcal{G}$, the concepts of "in-neighbors" and "out-neighbors" are introduced. The in-neighbor set of UAV i is defined as

$\mathcal{N}_i^- = \{j \in \mathcal{V} | (j,i) \in \mathcal{E}\}$, containing the neighboring nodes that have a edge pointing toward i for sending information to i. Similarly, the set of out-neighbors is defined as $\mathcal{N}_i^+ = \{l \in \mathcal{V} | (i,l) \in \mathcal{E}\}$ for UAV i, containing all the nodes that receive information from i.

5.2.2 Actuator Faults

Following the actuator fault modeling in (5.3), multiplicative or additive uncertainties are incorporated into the control channels as follow:

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \underbrace{\left[\begin{array}{c|ccc} \vartheta_{c_l} & 0 & 0 & 0 \\ \hline 0 & \vartheta_{c_l} & 0 & 0 \\ 0 & 0 & \vartheta_{c_l} & 0 \\ 0 & 0 & 0 & \vartheta_{c_r} \end{array}\right]}_{\Theta_\tau} \begin{bmatrix} u_F \\ u_\tau \end{bmatrix} + \underbrace{\begin{bmatrix} \vartheta_{F^0} \\ \vartheta_{\tau_\phi^0} \\ \vartheta_{\tau_\theta^0} \\ \vartheta_{\tau_\psi^0} \end{bmatrix}}_{\vartheta_{\tau 0}} \tag{5.3}$$

where u_{F_i} and $u_{\tau_i} = [u_{\tau_{\phi_i}} \; u_{\tau_{\theta_i}} \; u_{\tau_{\psi_i}}]^\top$ are the expected force and torque commands determined by the local controllers of UAV i. This relationship takes into consideration both the reduced efficiency of the actuators and the uncontrolled deviations in the PWM control signals.

Assumption 5.1. *We assume that all fault parameters in the actuator modeling–both the scaling factors and the additive deviations—are bounded. Additionally, it is assumed that these fault parameters evolve at slower rates compared to the dynamics of the vehicle.*

To ensure the satisfaction of (5.2) under various fault conditions, it is necessary to further restrict the force command u_{F_i} to a tighter range:

$$u_{F_i} \in \Omega_{F_i} \triangleq \left\{ u_{F_i} | \; \overline{\vartheta}_{F_i^0} \leqslant F_i \leqslant \overline{F}_i - \overline{\vartheta}_{F_i^0} \right\} \tag{5.4}$$

where $\overline{\vartheta}_{F_i^0} = \max_{t \geqslant 0}(\vartheta_{F_i^0}(t))$, quantifying the maximum deviation in the lift force induced by faults.

5.2.3 Formation Tracking Control Objective

This paper aims to develop a distributed control system to enforce the N UAVs to track a virtual leader in a prescribed formation. The virtual leader to be tracked is labeled as node 0, whose trajectory is time-varying and updated by a second-order reference generator:

$$\dot{\zeta}_0 = \upsilon_0 \tag{5.5a}$$

$$\dot{\upsilon}_0 = f_0(t) \tag{5.5b}$$

in which $\zeta_0 = [x_0 \ y_0 \ z_0]^\top$ and $\upsilon_0 = [\upsilon_{x_0} \ \upsilon_{y_0} \ \upsilon_{z_0}]^\top$ are the position and linear velocity vectors of the virtual leader; $f_0(t) : [0, \infty) \to \mathbb{R}^3$ is Lipschitz continuous in t.

Note that not all UAVs in the system have direct access to the virtual leader. Instead, only a subset of them can receive the real-time values of ζ_0, υ_0, and f_0. To describe the connectivity between the leader and the multi-UAV system, we introduce a pinning matrix $\mathcal{B} = \text{diag}(b_1, b_2, \cdots, b_N)$, where $b_i > 0$ if the ith UAV receives the leader's information directly and $b_i = 0$ otherwise.

To quantify the formation tracking control performance, we define concatenated vectors $\tilde{\zeta} = [\tilde{\zeta}_1^\top \ \tilde{\zeta}_2^\top \ \cdots \ \tilde{\zeta}_N^\top]^\top$ and $\tilde{\upsilon} = [\tilde{\upsilon}_1^\top \ \tilde{\upsilon}_2^\top \ \cdots \ \tilde{\upsilon}_N^\top]^\top$ as the leader-follower tracking errors of the N UAVs, in which the ith elements are defined for UAV i as

$$\tilde{\zeta}_i = \zeta_i - \zeta_0 - \Delta_{i0} \tag{5.6a}$$

$$\tilde{\upsilon}_i = \upsilon_i - \upsilon_0 \tag{5.6b}$$

where $\Delta_{i0} \in \mathbb{R}^3$ is the desired formation displacement of UAV i from the virtual leader 0. With the defined leader-follower formation error, the control objective of this paper is to achieve convergence of $\tilde{\zeta}$ and $\tilde{\upsilon}$ to a small neighborhood around zero in the presence of the unexpected actuator faults (5.3).

Assumption 5.2. *In the directed graph $\mathcal{G}$, each node is either part of a spanning tree with the root node connected to the virtual leader, or is a standalone node that is directly connected to the virtual leader.*

Remark 5.1. *Assumption 5.2 is a general connectivity assumption for multi-agent system, which is to assume that all UAVs in the system have either direct or indirect access to the virtual leader. UAVs that are located at the root nodes or standalone nodes receive the leader's information directly, while other UAVs are connected to the*

leader node indirectly through a chain of directed communication edges.

Lemma 5.1. *Under Assumption 5.2, the matrices $\mathcal{D}+\mathcal{B}$ and $\mathcal{L}+\mathcal{B}$ are nonsingular and positive-definite.*

Proof. Without loss of generality, we define a subset of nodes containing all the nodes that have direct access to the virtual leader as $\mathcal{V}' \triangleq \{r \in \mathcal{V} | b_r > 0\}$. For a node $i \notin \mathcal{V}'$, one has that $d_i > 0$. As a result, all the diagonal elements of $\mathcal{D}+\mathcal{B}$ are positive. This implies the nonsingularity and the positive-definiteness of $\mathcal{D}+\mathcal{B}$. For the matrix $\mathcal{L}+\mathcal{B}$, $|l_{rr}+b_r| > \sum_{j=1,j\neq r}^{N} l_{rj} > 0$ for $r \in \mathcal{V}'$ and $|l_{ii}+b_i| \geqslant \sum_{j=1,j\neq i}^{N} l_{ij} > 0$ for arbitrary $i \in \mathcal{V}$. Using the Theorem proposed in [173], we can then conclude that $\mathcal{L}+\mathcal{B}$ is a nonsingular and positive-definite matrix. □

5.3 Distributed Triple-Layer Control Design

In this section, the distributed control design for formation tracking is provided. As illustrated in Fig. 5.1, the control system for UAV i consists of three control layers, which is developed based on the model decomposition in 5.3.1. The detailed development of the three control layers are elaborated in 5.3.2, 5.3.3, and 5.3.4, respectively.

5.3.1 Model Decomposition

In this work, we propose a new triple-layer hierarchical control architecture based on the model decomposition of the translation model (5.1a-5.1b), the rotation angle model (5.1c) and the angular velocity model (5.1d), which is legitimized by a three-time-scale separation: the rotation angular velocity dynamics evolve in the fastest time-scale, while the rotation angle dynamics in the medium time-scale and the position dynamics in the slowest time-scale. In this case, η_i in (5.1b) and ω_i in (5.1c), whose transients are negligible in slower time-scales, can be replaced by their desired value denoted as η_i^d and ω_i^d. As a result, the decomposed model is obtained as

$$\dot{\zeta}_i = \upsilon_i \tag{5.7a}$$

$$m_i \dot{\upsilon}_i = -m_i g + r(\eta_i^d) F_i \tag{5.7b}$$

$$\dot{\eta}_i = W(\eta_i)\omega_i^d \tag{5.7c}$$

$$J_i \dot{\omega}_i = -\omega_i \times J_i \omega_i + \tau_i \tag{5.7d}$$

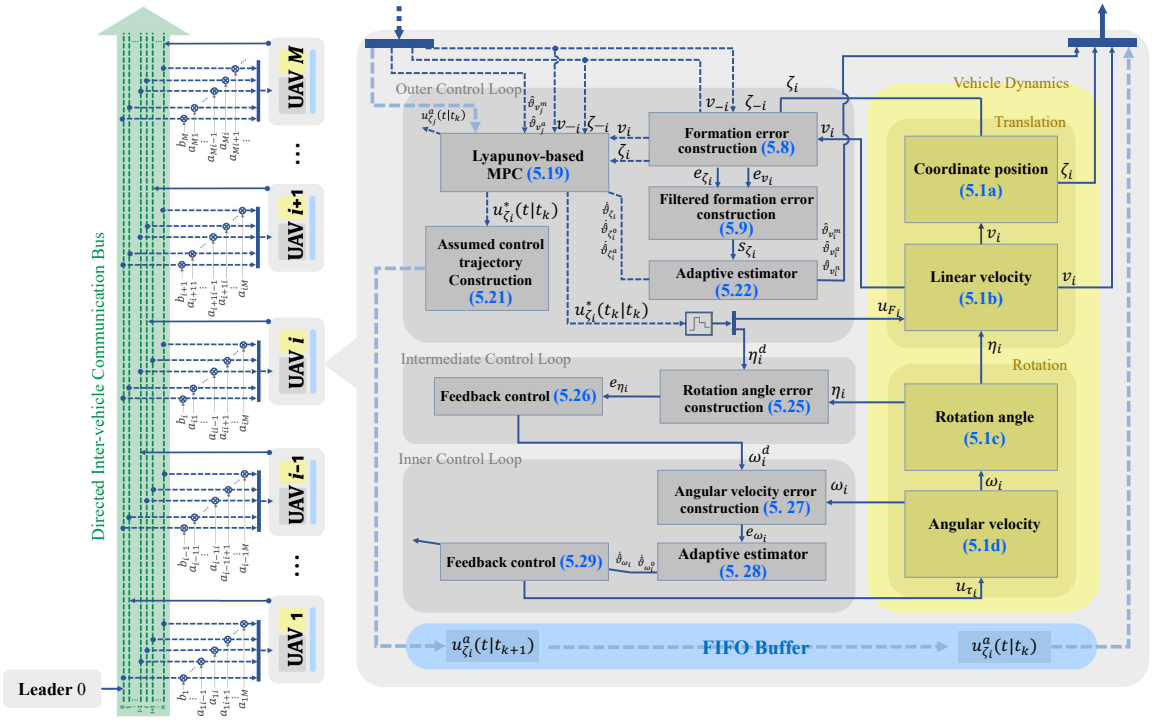

Figure 5.1: Block diagram of the distributed formation control system

In the rest of this section, the three control layers of UAV i are designed separately by studying (5.7a-5.7b), (5.7c) and (5.7d), respectively. As depicted in Fig. 5.1, the outer control layer exchanges information with its neighbors, determines the lift force control signal u_{F_i} and the desired rotation η_i^d, and sends η_i^d to the intermediate layer as the reference signal. The intermediate layer ensures that η_i converges to η_i^d by designing the desired rotation angular velocity ω_i^d. The inner control layer is responsible for angular velocity control, enforcing the rapid convergence of ω_i to ω_i^d.

5.3.2 Outer Translation Control Layer

A. Auxiliary Controller and Lyapunov Stability

In order to establish the stability constraint for MPC, we must first define the stability criteria for the closed-loop control system. This involves deriving an auxiliary control law and elucidating the underlying Lyapunov stability principle that supports this design.

We start by defining the neighborhood formation tracking error for the local control system. For a random UAV i, the formation discrepancy can be described by the

differences between its current state and the states of its in-neighbors:

$$e_{\zeta_i} = \sum_{j\in\mathcal{N}_i^-} a_{ij}\left(\zeta_i - \zeta_j + \Delta_{ij}\right) + b_i\left(\zeta_i - \zeta_0 - \Delta_{i0}\right) \tag{5.8a}$$

$$e_{\upsilon_i} = \sum_{j\in\mathcal{N}_i^-} a_{ij}\left(\upsilon_i - \upsilon_j\right) + b_i\left(\upsilon_i - \upsilon_0\right) \tag{5.8b}$$

where $\Delta_{ij} = \Delta_{i0} - \Delta_{j0}$ denotes the desired relative displacement between UAVs i and j, which is available to UAV i if $j \in \mathcal{N}_i^-$. A filtered local formation error can then be defined as

$$s_{\zeta_i} = \lambda_i e_{\zeta_i} + e_{\upsilon_i} \tag{5.9}$$

where $\lambda_i \in \mathbb{R}^+$ is the boundary layer gain. $s_{\zeta_i} = 0$ defines a sliding surface on which the formation errors e_{ζ_i} and e_{υ_i} asymptotically converges to zero. s_{ζ_i}, e_{ζ_i} and e_{υ_i} are the errors available for the developing the local distributed control system for UAV i.

In order to find a stabilizing control solution and establish the desired stability characteristics, we choose a Lyapunov function candidate as follows

$$V_{\zeta_i} = \frac{1}{2} s_{\zeta_i}^\top s_{\zeta_i} \tag{5.10}$$

Taking the derivative of $V_{s_{\zeta_i}}$ along (5.7a-5.7b) and substituting in the actuator model (5.3), we have

$$\dot{V}'_{\zeta_i} = s_{\zeta_i}^\top \left(\lambda_i e_{\upsilon_i} + \left(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i \right) \left(-g + m_i^{-1} r(\eta_i^d) \left(\vartheta_{c_{l_i}} u_{F_i} + \vartheta_{F_i}^0 \right) \right) - \sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{\upsilon}_j - b_i f_0 \right) \tag{5.11}$$

Here, the prime symbol is used to indicate that this derivative is calculated along the decomposed reduced-order translation dynamics instead of the original full-order ones.

An auxiliary controller can be designed as

$$u_{\zeta_i} \triangleq r(\eta_i^d) u_{F_i} = \vartheta_{\zeta_i} \left(\left(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i \right)^{-1} \left(-c_{\zeta_i} s_{\zeta_i} - \lambda_i e_{\upsilon_i} + \sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{\upsilon}_j + b_i f_0 \right) + g - \vartheta_{\zeta_i^0} \right) \tag{5.12}$$

where $c_{\zeta_i} \in \mathbb{R}^+$ is the user-determined positive control gain; $\vartheta_{\zeta_i} = m_i \vartheta_{c_{l_i}}^{-1}$ and $\vartheta_{\zeta_i^0} = m_i^{-1} r(\eta_i) \vartheta_{F_i}^0$ are collections of uncertain system parameters. Substituting u_{ζ_i} into

(5.11) leads to

$$\dot{V}'_{\zeta_i} = -c_{\zeta_i} s_{\zeta_i}^\top s_{\zeta_i} \tag{5.13}$$

which implies that (5.12) provides a stabilizing solution as it ensures the non-increasing of the local Lyapunov function.

B. Adaptive Parameter Estimator

However, the auxiliary control law as designed in (5.12) relies on information that is not known a prior. Specifically, the parameters $\vartheta_{\zeta_i^m}$ and $\vartheta_{\zeta_i^a}$ are unknown. Additionally, the real-time value of $\dot{v}_j$ for each $j \in \mathcal{N}_i^-$ is not accessible to UAV i. Instead of actual data, an assumed value, denoted as $\dot{v}_j^a$, can be constructed from the previous optimization result of j and made available to i at every instant. Further details will be provided later in the MPC formation section. To account for the resultant discrepancies, we introduce another parameter as $\vartheta_{\zeta_i^a} = \sum_{j\in\mathcal{N}_i^-} a_{ij}\left(\dot{v}_j^a - \dot{v}_j\right)$. The auxiliary controller is thereby modified as

$$u_{\zeta_i} = \hat{\vartheta}_{\zeta_i}\left(\left(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\right)^{-1}\left(-c_{\zeta_i} s_{\zeta_i} - \lambda_i e_{v_i} + \sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{v}_j^a - \hat{\vartheta}_{\zeta_i^a} + b_i f_0\right) + g - \hat{\vartheta}_{\zeta_i^0}\right) \tag{5.14}$$

with $\hat{\vartheta}_{\zeta_i}$, $\hat{\vartheta}_{\zeta_i^0}$ and $\hat{\vartheta}_{\zeta_i^a}$ being the estimates of ϑ_{ζ_i}, $\vartheta_{\zeta_i^0}$ and $\vartheta_{\zeta_i^a}$ respectively. They are updated online via the following adaptive laws

$$\dot{\hat{\vartheta}}_{\zeta_i} = -k_{\zeta_i}\left(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\right)\hat{\vartheta}_{\zeta_i}^{-1} s_{\zeta_i}^\top r(\eta_i^d) u_{F_i} \tag{5.15a}$$

$$\dot{\hat{\vartheta}}_{\zeta_i^0} = k_{\zeta_i^0}\left(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\right) s_{\zeta_i} \tag{5.15b}$$

$$\dot{\hat{\vartheta}}_{\zeta_i^a} = k_{\zeta_i^a} s_{\zeta_i} \tag{5.15c}$$

with $k_{\zeta_i}, k_{\zeta_i^0}, k_{\zeta_i^a} \in \mathbb{R}^+$ are the adaptive gains.

Correspondingly, the Lyapunov function in (5.10) is redefined as

$$V_{\zeta_i} = \frac{1}{2} s_{\zeta_i}^\top s_{\zeta_i} + \frac{1}{2k_{\zeta_i}\vartheta_{\zeta_i}}\tilde{\vartheta}_{\zeta_i}^2 + \frac{1}{2k_{\zeta_i^0}}\tilde{\vartheta}_{\zeta_i^0}^\top\tilde{\vartheta}_{\zeta_i^0} + \frac{1}{2k_{\zeta_i^a}}\tilde{\vartheta}_{\zeta_i^a}^\top\tilde{\vartheta}_{\zeta_i^a} \tag{5.16}$$

where $\tilde{\vartheta}_{\zeta_i} = \hat{\vartheta}_{\zeta_i} - \vartheta_{\zeta_i}$, $\tilde{\vartheta}_{\zeta_i^0} = \hat{\vartheta}_{\zeta_i^0} - \vartheta_{\zeta_i^0}$ and $\tilde{\vartheta}_{\zeta_i^a} = \hat{\vartheta}_{\zeta_i^a} - \vartheta_{\zeta_i^a}$, denoting the estimation

error. Substituting the adaptive laws into the time derivative of V_{ζ_i} yields

$$\begin{aligned}\dot{V}'_{\zeta_i} =& s_{\zeta_i}^\top\Bigg(\lambda_i e_{v_i}+\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij}+b_i\Big)\Big(-g+\frac{\hat{\vartheta}_{\zeta_i}}{\vartheta_{\zeta_i}}\hat{\vartheta}_{\zeta_i}^{-1}r(\eta_i^d)u_{F_i}+\vartheta_{\zeta_i^0}\Big)-\sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{v}_j^a+\vartheta_{\zeta_i^a}-b_i f_0\Bigg)\\ &- s_{\zeta_i}^\top\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij}+b_i\Big)\frac{\tilde{\vartheta}_{\zeta_i}}{\vartheta_{\zeta_i}}\hat{\vartheta}_{\zeta_i}^{-1}r(\eta_i^d)u_{F_i}+s_{\zeta_i}^\top\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij}+b_i\Big)\tilde{\vartheta}_{\zeta_i^0}+s_{\zeta_i}^\top\tilde{\vartheta}_{\zeta_i^a}\\ =& s_{\zeta_i}^\top\Bigg(\lambda_i e_{v_i}+\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij}+b_i\Big)\Big(-g+\hat{\vartheta}_{\zeta_i}^{-1}r(\eta_i^d)u_{F_i}+\hat{\vartheta}_{\zeta_i^0}\Big)-\sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{v}_j^a+\hat{\vartheta}_{\zeta_i^a}-b_i f_0\Bigg)\end{aligned}\tag{5.17}$$

Similarly, by applying the auxiliary control law (5.15) to the closed-loop system, the Lyapunov function remains non-increasing, as demonstrated below:

$$\dot{V}'_{\zeta_i}=-c_{\zeta_i}s_{\zeta_i}^\top s_{\zeta_i}\tag{5.18}$$

It is important to note that the auxiliary control law is not an ideal solution. Firstly, it necessitates continuous updates of the states of neighboring vehicles, yet information exchange through the inter-vehicle communication network occurs at discrete time intervals. Secondly, it is not optimal with respect to input constraints. Nevertheless, this auxiliary design offers a good choice of the Lyapunov function and its corresponding minimum decay rate, which inspires the subsequent formulation of the Lyapunov-based MPC problem.

C. Lyapunov-based Adaptive MPC Problem Formulation

MPC, the controller actually employed in the outer layer, operates in a discrete, sample-and-hold fashion, initiating a new calculation cycle upon receiving new information from neighboring systems. The frequency of these outer-layer control updates is synchronized with the transmission frequency of the inter-vehicle communication network. The sequence of control update instants is denoted as $\{t_k = k\delta | k \in 0, 1, 2, \cdots\}$, with $\delta \in \mathbb{R}^+$ representing the sampling period of the outer layer.

At the time instant t_k, an MPC optimization problem is solved over a forward-looking prediction horizon $[t_k, t_k + T]$ with $T > \delta$. This finite-horizon optimization utilizes the measured current states of the UAV itself, $\zeta_i(t_k)$ and $v_i(t_k)$, as well as the received states of its in-neighbors, $\zeta_j(t_k)$ and $\zeta_j(t_k)$ for $j \in \mathcal{N}_i^-$, as the initial conditions. If UAV i has direct access to the virtual leader, it also obtains $\zeta_0(t|t_k)$ and $v_0(t|t_k)$ for all $t \in [t_k, t_k + T]$, along with $f_0(t_k)$. The MPC problem at t_k is

formulated as follows:

$$\min_{u^p_{F_i},\eta^{d_p}_i} \int_{t_k}^{t_k+T} \left(\|s^p_{\zeta_i}(t|t_k)\|^2_{Q_i} + |u^p_{F_i}(t|t_k) - F^*_i|^2_{R_{F_i}} + \|\eta^{d_p}_i(t|t_k)\|^2_{R_{\eta_i}} \right) \mathrm{d}t \tag{5.19a}$$

with $R_{F_i} \in \mathbb{R}^+$, $Q_i, R_{\eta_i} \in \mathbb{R}^{3\times3}$ and $s^p_{\zeta_i} = \lambda_i\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij}(\zeta^p_i - \zeta^a_j + \Delta_{ij}) + b_i\left(\zeta^p_i - \zeta_0 - \Delta_{i0}\right)\Big) + \sum_{j\in\mathcal{N}_i^-} a_{ij}\left(\upsilon^p_i - \upsilon^a_j\right) + b_i\left(\upsilon^p_i - \upsilon_0\right)$, and subject to

$$\begin{cases} \dot{\zeta}^p_i(t|t_k) &= \upsilon^p_i(t|t_k), \\ \dot{\upsilon}^p_i(t|t_k) &= -g + \hat{\vartheta}^{-1}_{\zeta_i}(t_k) r(\eta^{d_p}_i(t|t_k)) u^p_{F_i}(t|t_k) + \hat{\vartheta}_{\zeta^0_i}(t_k), \end{cases} \tag{5.19b}$$

$$\begin{cases} \dot{\zeta}^a_j(t|t_k) &= \upsilon^a_j(t|t_k), \\ \dot{\upsilon}^a_j(t|t_k) &= -g + \hat{\vartheta}^{-1}_{\zeta_j}(t_k) r(\eta^{d^a}_j(t|t_k)) u^a_{F_j}(t|t_k) + \hat{\vartheta}_{\zeta^0_j}(t_k), \end{cases} \tag{5.19c}$$

$$\begin{cases} \zeta^p_i(t_k|t_k) &= \zeta_i(t_k), \\ \upsilon^p_i(t_k|t_k) &= \upsilon_i(t_k), \end{cases} \tag{5.19d}$$

$$\begin{cases} \zeta^a_j(t_k|t_k) &= \zeta_j(t_k), \\ \upsilon^a_j(t_k|t_k) &= \upsilon_j(t_k), \end{cases} \tag{5.19e}$$

$$\begin{cases} u^p_{F_i}(t|t_k) &\in \Omega_{F_i}, \\ \eta^{d_p}_i(t|t_k) &\in \Omega_{\eta_i}, \end{cases} \tag{5.19f}$$

$$\begin{aligned} & s^{p\top}_{\zeta_i}(t_k|t_k)\left(\lambda_i e^p_{\upsilon_i}(t_k|t_k) + \left(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\right)\dot{\upsilon}^p_i(t_k|t_k) - \sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{\upsilon}^a_j(t_k|t_k) + \hat{\vartheta}_{\zeta^a_i}(t_k) - b_i f_0(t_k)\right) \\ & \leqslant -c_{\zeta_i} s^{p\top}_{\zeta_i}(t_k|t_k) s^p_{\zeta_i}(t_k|t_k) \end{aligned} \tag{5.19g}$$

for $t \in [t_k, t_k + T]$.

The feasibility of this Lyapunov-based MPC optimization problem can be guaranteed by the existence of the auxiliary controller, upon which the Lyapunov stability constraint (5.19g) is formulated. A comprehensive proof of this is provided in [174]. By solving this optimization problem, optimized control sequences, $u^o_{F_i}(t|t_k)$ and $\eta^{d^o}(t|t_k)$ for $t \in [t_k, t_k + T]$, can be obtained. MPC employs a receding horizon implementation manner, meaning that the optimization results obtained at t_k are applied as actual control actions only for the current time interval $t \in [t_k, t_{k+1})$, as detailed below

$$u_{F_i}(t) = u^o_{\zeta_i}(t_k|t_k) \tag{5.20a}$$

$$\eta^d(t) = \eta^{d^o}(t_k|t_k) \tag{5.20b}$$

At the next time instant t_{k+1}, the optimization problem will be solved again with updated initial conditions and this iterative process is repeated.

In the optimization problem, (5.19b) and (5.19c) serve as the prediction models. (5.19b) is to predict the future evolution of UAV i itself, with related variables marked by a superscript p. In contrast, (5.19c) calculates the assumed trajectories of neighboring vehicles j over the prediction horizon, with variables marked by a superscript a. The calculation governed by (5.19c) is driven by assumed control sequences of j, which can be derived from j's previous optimization result. For example, $u^a_{F_j}(t|t_k)$ and $\eta^{d^a}_j(t|t_k)$ utilized in (5.19c) are constructed based on the optimization result of j at t_{k-1}:

$$u^a_{F_j}(t|t_k) = \begin{cases} u^o_{F_j}(t|t_{k-1}), & t \in [t_k, t_{k-1}+T] \\ u^o_{F_j}(t_{k-1}+T|t_{k-1}), & t \in (t_{k-1}+T, t_k+T] \end{cases} \tag{5.21a}$$

$$\eta^{d^a}_j(t|t_k) = \begin{cases} \eta^{d^o}_{F_j}(t|t_{k-1}), & t \in [t_k, t_{k-1}+T] \\ \eta^{d^o}_{F_j}(t_{k-1}+T|t_{k-1}), & t \in (t_{k-1}+T, t_k+T] \end{cases} \tag{5.21b}$$

They are stored in a buffer integrated with the local control system of j and broadcast to all the out-neighboring vehicles in $\mathcal{N}_j^+$ together with the updated state measurements at t_k.

Remark 5.2. *An exception of the assumed trajectory construction is at the initial time t_0 when no previous optimization result is available. In this case, $u^a_{F_j}$ remains equal to F_i^*, the required thrust force at the equilibrium point. $\eta^{d^a}_j$ is set to zero.*

The adaptive laws for updating $\hat{\vartheta}_{\zeta_i}$, $\hat{\vartheta}_{\zeta_i^0}$ and $\hat{\vartheta}_{\zeta_i^a}$ in the MPC problem (5.19) follow the form of (5.15) but but are implemented in a sample-and-hold fashion:

$$\dot{\hat{\vartheta}}_{\zeta_i}(t) = -k_{\zeta_i}\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\Big)\hat{\vartheta}^{-1}_{\zeta_i}(t_k)s^\top_{\zeta_i}(t_k)r(\eta^d_i(t_k))u_{F_i}(t_k) \tag{5.22a}$$

$$\dot{\hat{\vartheta}}_{\zeta_i^0}(t) = k_{\zeta_i^0}\Big(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\Big)s_{\zeta_i}(t_k) \tag{5.22b}$$

$$\dot{\hat{\vartheta}}_{\zeta_i^a}(t) = k_{\zeta_i^a}s_{\zeta_i}(t_k) \tag{5.22c}$$

for $t \in [t_k, t_{k+1})$.

The time derivative of the Lyapunov function, as defined in (5.16), at the time

instant t_k is given by

$$\begin{aligned}\dot{V}'_{\zeta_i}(t_k) =& s_{\zeta_i}^{\top}(t_k)\Bigg(\lambda_i e_{v_i}(t_k) + \Big(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\Big)\Big(-g + \vartheta_{\zeta_i}^{-1}(t_k) r(\eta_i^d(t_k)) u_{F_i}(t_k) + \vartheta_{\zeta_i^0}(t_k)\Big) \\ & - \sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{v}_j^a(t_k) + \vartheta_{\zeta_i^a}(t_k) - b_i f_0(t_k)\Bigg) + \frac{1}{k_{\zeta_i}\vartheta_{\zeta_i}(t_k)}\tilde{\vartheta}_{\zeta_i}(t_k)\dot{\hat{\vartheta}}_{\zeta_i}(t_k) \\ & + \frac{1}{k_{\zeta_i^0}}\tilde{\vartheta}_{\zeta_i^0}^{\top}(t_k)\dot{\hat{\vartheta}}_{\zeta_i^0}(t_k) + \frac{1}{k_{\zeta_i^a}}\tilde{\vartheta}_{\zeta_i^a}^{\top}(t_k)\dot{\hat{\vartheta}}_{\zeta_i^a}(t_k) \end{aligned} \tag{5.23}$$

Substituting the stability constraint (5.19g) of MPC and the adaptive laws (5.22) into it, we have

$$\begin{aligned}\dot{V}'_{\zeta_i}(t_k) =& s_{\zeta_i}^{\top}(t_k)\Bigg(\lambda_i e_{v_i}(t_k) + \Big(\sum_{j\in\mathcal{N}_i^-} a_{ij} + b_i\Big)\Big(-g + \hat{\vartheta}_{\zeta_i}^{-1}(t_k) r(\eta_i^d(t_k)) u_{F_i}(t_k) + \hat{\vartheta}_{\zeta_i^0}(t_k)\Big) \\ & - \sum_{j\in\mathcal{N}_i^-} a_{ij}\dot{v}_j^a(t_k) + \hat{\vartheta}_{\zeta_i^a}(t_k) - b_i f_0(t_k)\Bigg) \\ \leqslant & - c_{\zeta_i} s_{\zeta_i}^{\top}(t_k) s_{\zeta_i}(t_k) \end{aligned} \tag{5.24}$$

which demonstrates the decay of the local Lyapunov function at each time instant t_k. This important conclusion will be called back later in the stability analysis of the overall multi-UAV system.

5.3.3 Intermediate Rotation Angle Control Layer

Unlike the outer layer, both the intermediate and inner layers for rotation control utilize only locally measured real-time information, thereby continuously updating themselves.

With the desired rotation angle determined by the outer control layer, the intermediate control layer is responsible for controlling the rotation angle. We first define the error of rotation angle tracking as:

$$e_{\eta_i} = \eta_i - \eta_i^d \tag{5.25}$$

In order to eliminate this error, the desired angular velocity is determined by applying feedback control

$$\omega_i^d = W^{-1}(\eta_i)(-c_{\eta_i} e_{\eta_i} + \dot{\eta}_i^d) \tag{5.26}$$

where $c_{\eta_i} \in \mathbb{R}^+$ is the rotation angle control gain.

5.3.4 Inner Angular Velocity Control Layer

Given the desired angular velocity from the intermediate layer, the inner layer is developed for angular velocity control. The angular velocity tracking error is defined as

$$e_{\omega_i} = \omega_i - \omega_i^d \tag{5.27}$$

where ω_i^d is determined by (5.26).

An adaptive control law for determining the rotation torque control command is developed as

$$u_{\tau_i} = \text{diag}(\hat{\vartheta}_{\omega_i}) \left(-c_{\omega_i} e_{\omega_i} + f_\omega(\omega_i) - \hat{\vartheta}_{\omega_i^0} + \dot{\omega}_i^d \right) \tag{5.28}$$

where $f_\omega(\omega_i) = J_i^{-1} (\omega_i \times J_i \omega_i)$; $c_{\omega_i} \in \mathbb{R}^+$ is the angular velocity control gain; $\hat{\vartheta}_{\omega_i}$ and $\hat{\vartheta}_{\omega_i^0}$ are estimates of ϑ_{ω_i} and $\vartheta_{\omega_i^0}$, defined as $\vartheta_{\omega_i} = [J_{xx_i}\vartheta_{c_{l_i}} \;\; J_{yy_i}\vartheta_{c_{l_i}} \;\; J_{zz_i}\vartheta_{c_{r_i}}]^\top$ and $\vartheta_{\omega_i^0} = [J_{xx_i}^{-1}\vartheta^0_{\tau_{\phi_i}} \;\; J_{yy_i}^{-1}\vartheta^0_{\tau_{\theta_i}} \;\; J_{zz_i}^{-1}\vartheta^0_{\tau_{\psi_i}}]^\top$. The adaptive laws for updating $\hat{\vartheta}_{\omega_i}$ and $\hat{\vartheta}_{\omega_i^0}$ are designed as

$$\dot{\hat{\vartheta}}_{\omega_i} = -k_{\omega_i} \text{diag}^{-1}(\hat{\vartheta}_{\omega_i} u_{\tau_i}) e_{\omega_i} \tag{5.29a}$$

$$\dot{\hat{\vartheta}}_{\omega_i^0} = -k_{\omega_i^a} e_{\omega_i} \tag{5.29b}$$

where k_{ω_i}, $k_{\omega_i^0} \in \mathbb{R}^+$ are positive adaptive gains.

5.4 Stability Analysis

This section evaluates the closed-loop formation tracking performance of the entire multi-UAV system with the proposed distributed control design. A global Lyapunov function of the entire multi-UAV system is defined and its time derivative is derived in Section 5.4.1. In Section 5.4.2, the main result of the closed-loop analysis is concluded by a theorem with explicitly characterized stability conditions.

5.4.1 Global Lyapunov Function

The global Lyapunov function for the overall N-UAV system is defined as the superposition of Lyapunov functions for three control layers:

$$V = V_\zeta + V_\eta + V_\omega \tag{5.30}$$

where V_ζ is the outer-layer Lyapunov function, extended from (5.16) to include all N UAVs:

$$\begin{aligned} V_\zeta =& \sum_{i=1}^{N} V_{\zeta_i} = \frac{1}{2} s_\zeta^\top s_\zeta + \frac{1}{2} \tilde{\vartheta}_\zeta^\top K_\zeta^{-1} \mathrm{diag}(\vartheta_\zeta)^{-1} \tilde{\vartheta}_\zeta + \frac{1}{2} \tilde{\vartheta}_{\zeta^0}^\top \left(K_{\zeta^0} \otimes b I_3\right)^{-1} \tilde{\vartheta}_{\zeta^0} \\ &+ \frac{1}{2} \tilde{\vartheta}_{\zeta^a}^\top \left(K_{\zeta^a} \otimes I_3\right)^{-1} \tilde{\vartheta}_{\zeta^a} \end{aligned} \tag{5.31}$$

Similarly, V_η and V_ω are the intermediate- and inner-layer Lyapunov functions, chosen as

$$V_\eta = \frac{1}{2} e_\eta^\top e_\eta \tag{5.32}$$

$$V_\omega = \frac{1}{2} e_\omega^\top e_\omega + \frac{1}{2} \tilde{\vartheta}_\omega^\top \left(K_\omega \otimes I_3\right)^{-1} \mathrm{diag}(\vartheta_\omega)^{-1} \tilde{\vartheta}_\omega + \frac{1}{2} \tilde{\vartheta}_{\omega^0}^\top \left(K_{\omega^0} \otimes I_3\right)^{-1} \tilde{\vartheta}_{\omega^0} \tag{5.33}$$

The involved concatenated errors and gain matrices are defined as $s_\zeta = [s_{\eta_1}^\top \; s_{\eta_2}^\top \cdots s_{\eta_N}^\top]^\top$, $e_{\eta,\omega} = [e_{\eta,\omega_1}^\top \; e_{\eta,\omega_2}^\top \cdots e_{\eta,\omega_N}^\top]^\top$, $\tilde{\vartheta}_\zeta = \hat{\vartheta}_\zeta - \vartheta_\zeta = [\hat{\vartheta}_{\zeta_1} \; \hat{\vartheta}_{\zeta_2} \cdots \hat{\vartheta}_{\zeta_N}]^\top - [\vartheta_{\zeta_1} \; \vartheta_{\zeta_2} \cdots \vartheta_{\zeta_N^m}]^\top$, $\tilde{\vartheta}_{\zeta^{0,a}} = \hat{\vartheta}_{\zeta^{0,a}} - \vartheta_{\zeta^{0,a}} = [\hat{\vartheta}_{\zeta_1^{0,a}}^\top \; \hat{\vartheta}_{\zeta_2^{0,a}}^\top \cdots \hat{\vartheta}_{\zeta_N^{0,a}}^\top]^\top - [\vartheta_{\zeta_1^{0,a}}^\top \; \vartheta_{\zeta_2^{0,a}}^\top \cdots \vartheta_{\zeta_N^{0,a}}^\top]^\top$, $\tilde{\vartheta}_\omega = \hat{\vartheta}_\omega - \vartheta_\omega = [\hat{\vartheta}_{\omega_1}^\top \; \hat{\vartheta}_{\omega_2}^\top \cdots \hat{\vartheta}_{\omega_N}^\top]^\top - [\vartheta_{\omega_1}^\top \; \vartheta_{\omega_2}^\top \cdots \vartheta_{\omega_N}^\top]^\top$, $\tilde{\vartheta}_{\omega^0} = \hat{\vartheta}_{\omega^0} - \vartheta_{\omega^0} = [\hat{\vartheta}_{\omega_1^0}^\top \; \hat{\vartheta}_{\omega_2^0}^\top \cdots \hat{\vartheta}_{\omega_N^0}^\top]^\top - [\vartheta_{\omega_1^0}^\top \; \vartheta_{\omega_2^0}^\top \cdots \vartheta_{\omega_N^0}^\top]^\top$, $K_{\zeta,\omega} = \mathrm{diag}\big(k_{\zeta,\omega_1}, k_{\zeta,\omega_2}, \cdots, k_{\zeta,\omega_N}\big)$, $K_{\zeta^{0,a}} = \mathrm{diag}\big(k_{\zeta_1^{0,a}}, k_{\zeta_2^{0,a}}, \cdots, k_{\zeta_N^{0,a}}\big)$, and $K_{\omega^0} = \mathrm{diag}\big(k_{\omega_1^0}, k_{\omega_2^0}, \cdots, k_{\omega_N^0}\big)$.

Taking the derivative of V_ζ along the full-order translation dynamics (5.1a)-(5.1b) gives

$$\dot{V} = \dot{V}_\zeta' + D_\zeta + \dot{V}_\eta' + D_\eta + V_\omega \tag{5.34}$$

where

$$\dot{V}'_{\zeta} = s_{\zeta}^{\top}\Big((\Lambda \otimes I_3)\, e_{\upsilon} + \Big((\mathcal{D}+\mathcal{B}) \otimes I_3\Big)\Big(-\underline{g}+\mathrm{diag}(\vartheta_{\zeta})^{-1}\underline{R}\Big(\eta^{d}\Big)u_F+\vartheta_{\zeta^0}\Big) - (\mathcal{A} \otimes I_3)\,\dot{\upsilon}^{a} + \vartheta_{\zeta^a} - \Big(\mathcal{B} \otimes I_3\Big)\underline{f_0}\Big) \tag{5.35a}$$

$$D_{\zeta} = s_{\zeta}^{\top}\Big(\Big((\mathcal{L}+\mathcal{B}) \otimes I_3\Big)\mathrm{diag}^{-1}(\vartheta_{\zeta}\Big(\Big(\underline{R}\Big(\eta\Big) - \underline{R}\Big(\eta^{d}\Big)\Big)u_F\Big) \tag{5.35b}$$

$$\dot{V}'_{\eta} = e_{\eta}^{\top}\left(\underline{W}\Big(\eta\Big)\omega^{d} - \dot{\eta}^{d}\right) \tag{5.35c}$$

$$D_{\eta} = e_{\eta}^{\top}\underline{W}\Big(\eta\Big)\left(\omega - \omega^{d}\right) \tag{5.35d}$$

$$\dot{V}_{\omega} = e_{\omega}^{\top}\Big(\underline{f_{\omega}}(\omega) + \mathrm{diag}^{-1}(\vartheta_{\omega^m})u_{\tau}(t) + \vartheta_{\omega^a} - \dot{\omega}^{d}\Big) \tag{5.35e}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, $e_{\zeta,\upsilon} = [e_{\zeta,\upsilon_1}^{\top}\ e_{\zeta,\upsilon_2}^{\top}\ \cdots\ e_{\zeta,\upsilon_N}^{\top}]^{\top}$, $\underline{f_0}(t) = [f_0^{\top}(t)\cdots f_0^{\top}(t)]^{\top}$, $\underline{g} = [g^{z^{\top}}\cdots g^{z^{\top}}]^{\top}$, $\eta^{d} = \left[\eta_1^{d\top}\ \eta_2^{d\top}\cdots\eta_N^{d\top}\right]^{\top}$, $\underline{R}(\eta) = \mathrm{diag}\left(r\left(\eta_1\right), r\left(\eta_2\right), \cdots, r\left(\eta_N\right)\right)$, $u_F = \left[u_{F_1}\ u_{F_2}\ \cdots u_{F_N}\right]^{\top}$; $\underline{W}\left(\eta\right) = \mathrm{blkdiag}\left(W\left(\eta_1\right), W\left(\eta_2\right), \cdots,\ W\left(\eta_N\right)\right)$; $\underline{f_{\omega}}(\omega) = [f_{\omega}^{\top}(\omega_1)\ f_{\omega}^{\top}(\omega_2)\ \cdots f_{\omega}^{\top}(\omega_N)]^{\top}$; $u_{\tau} = [u_{\tau_1}^{\top}\ u_{\tau_2}^{\top}\cdots u_{\tau_N}^{\top}]^{\top}$. Here, derivatives marked with prime symbols are calculated based on the decoupled dynamics. The terms D_{ζ} and D_{η} are used to denote the deviations of the actual derivatives of V_{ζ} and V_{η} from $\dot{V}'_{\zeta}$ and $\dot{V}'_{\eta}$. They arise due to the transient dynamics of η and ω, which are neglected in the control development.

Recalling (5.24), we can have that

$$\dot{V}'_{\zeta}(t_k) = \sum_{i=1}^{N}\dot{V}'_{\zeta_i}(t_k) \leqslant -s_{\zeta}^{\top}(t_k)\,(C_{\zeta} \otimes I_3)\,s_{\zeta}(t_k) \tag{5.36}$$

where $C_{\zeta} = \mathrm{diag}\left(c_{\zeta_1}, c_{\zeta_2}, \cdots, c_{\zeta_N}\right)$. From (5.36), the decay of V'_{ζ} at each time outer-layer control update instant t_k can be ensured. However, there is no guarantee of convergence between two control update instants. To manage the system's uncontrolled inter-sample behavior, it is essential to establish a bound on its progression. The concept of Lipschitz continuity can be used to quantify the system's rate of change. Given the boundedness of the control inputs and the continuously differentiable property of the Lyapunov function, there exist positive Lipschitz constants $L_{s_{\zeta}}$, $L_{e_{\zeta}}$, $L_{\vartheta_{\upsilon}^{n}}$, $L_{\vartheta_{\upsilon}^{a}}$, $L_{\vartheta_{\upsilon}^{m}}$, $M_{s_{\zeta}}$, $M_{e_{\zeta}}$, $M_{\vartheta_{\upsilon}^{n}}$, $M_{\vartheta_{\upsilon}^{a}}$, $M_{\vartheta_{\upsilon}^{m}}$ such that the following inequalities

hold

$$\begin{aligned}\left|\dot{V}_\zeta'(t)-\dot{V}_\zeta'(t_k)\right| \leqslant & L_{s_\zeta}\left\|s_\zeta(t)-s_\zeta(t_k)\right\|+L_{e_\zeta}\left\|e_\zeta(t)-e_\zeta(t_k)\right\|+L_{\vartheta_\zeta}\left\|\hat{\vartheta}_\zeta(t)-\hat{\vartheta}_\zeta(t_k)\right\| \\ & +L_{\vartheta_\zeta^0}\left\|\hat{\vartheta}_{\zeta^0}(t)-\hat{\vartheta}_{\zeta^0}(t_k)\right\|+L_{\vartheta_\zeta^a}\left\|\hat{\vartheta}_{v^a}(t)-\hat{\vartheta}_{v^a}(t_k)\right\| \qquad (5.37a)\end{aligned}$$

$$\left|s_\zeta^\top(t)s_\zeta(t)-s_\zeta^\top(t_k)s_\zeta(t_k)\right| \leqslant L_\zeta\left\|s_\zeta(t)-s_\zeta(t_k)\right\| \qquad (5.37b)$$

$$\left\|s_\zeta(t)-s_\zeta(t_k)\right\| \leqslant M_{s_\zeta}\delta \qquad (5.37c)$$

$$\left\|e_\zeta(t)-e_\zeta(t_k)\right\| \leqslant M_{e_\zeta}\delta \qquad (5.37d)$$

$$\left\|\hat{\vartheta}_\zeta(t)-\hat{\vartheta}_\zeta(t_k)\right\| \leqslant M_{\vartheta_\zeta}\delta \qquad (5.37e)$$

$$\left\|\hat{\vartheta}_{\zeta^0}(t)-\hat{\vartheta}_{\zeta^0}(t_k)\right\| \leqslant M_{\vartheta_\zeta^0}\delta \qquad (5.37f)$$

$$\left\|\hat{\vartheta}_{\zeta^a}(t)-\hat{\vartheta}_{\zeta^a}(t_k)\right\| \leqslant M_{\vartheta_\zeta^a}\delta \qquad (5.37g)$$

where $t\in[t_k,t_{k+1})$.

With $\left\|\partial\underline{R}\left(\eta\right)/\partial\eta\right\|\leqslant 1$, the following equality also hold

$$\begin{aligned}D_\zeta &\leqslant \left|s_\zeta^\top\left(\left(\left(\mathcal{L}+\mathcal{B}\right)\otimes I_3\right)\mathrm{diag}^{-1}(\vartheta_\zeta)\left(\underline{R}(\eta)-\underline{R}(\eta^d)\right)u_F\right)\right| \\ &\leqslant \kappa_1\lambda_{\max}(\mathcal{L}+\mathcal{B})\|s_\zeta\|\|e_\eta\| \qquad (5.38)\end{aligned}$$

in which $\kappa_1=\max_{i=1}^n(\overline{F}_i)/\min_{i=1}^n(m_i)+g$.

Furthermore, substituting the intermediate-layer control law (5.26) into (5.35c) leads to

$$\dot{V}_\eta'=-e_\eta^\top\left(C_\eta\otimes I_3\right)e_\eta \qquad (5.39)$$

where $C_\eta=\mathrm{diag}\left(c_{\eta_1},c_{\eta_2},\cdots,c_{\eta_N}\right)$. Also, given $\|\underline{W}(\eta)\|\leqslant 1$, it can be obtained that

$$D_\eta \leqslant \left|e_\eta^\top\underline{W}\left(\eta\right)\left(\omega-\omega^d\right)\right| \leqslant \|e_\eta\|\|e_\omega\| \qquad (5.40)$$

Moreover, substituting the inner-layer control law and the adaptive laws into (5.35e) gives

$$\dot{V}_\omega' \leqslant -e_\omega^\top\left(C_\omega\otimes I_3\right)e_\omega \qquad (5.41)$$

where $C_\omega=\mathrm{diag}\left(c_{\omega_1},c_{\omega_2},\cdots,c_{\omega_N}\right)$.

Substituting simultaneously (5.36), (5.37), (5.38), (5.39), (5.40) and (5.41) to

(5.34), we can obtain

$$\begin{aligned}\dot{V} \leqslant & -\lambda_{\min}(C_\zeta)\|s_\zeta\|^2 + \kappa_1\lambda_{\max}(\mathcal{L}+\mathcal{B})\|s_\zeta\|\|e_\eta\| - \lambda_{\min}(C_\eta)\|e_\eta\|^2 + \|e_\zeta\|\|e_\omega\| \\ & - \lambda_{\min}(C_\omega)\|e_\omega\|^2 + (\kappa_2 + \kappa_3\lambda_{\max}(C_\zeta))\delta \end{aligned} \tag{5.42}$$

where $\kappa_2 = L_{s_\zeta}M_{s_\zeta} + L_{e_\zeta}M_{e_\zeta} + L_{\vartheta_\zeta}M_{\vartheta_\zeta} + L_{\vartheta_\zeta^0}M_{\vartheta_\zeta^0} + L_{\vartheta_\zeta^a}M_{\vartheta_\zeta^a}$ and $\kappa_3 = L_{s_\zeta}M_{s_\zeta}$.

Remark 5.3. *In the right-hand side of (5.42), κ_1 quantifies the interconnection between the outer and intermediate layer, which is overlooked during the control development, is now being reconsidered for a comprehensive evaluation of the multi-layer control strategy. $(\kappa_2 + \kappa_3\lambda_{\max}(C_\zeta))\delta$ establishes a constrain on the system's evolution between two consecutive control updates, which is directly proportional to the sampling period and the changing rate of the system. Through this characterization, we can bridge the gap between discrete-time control actions and continuous-time system dynamics, ensuring a coherent and comprehensive understanding of the sampled data system's behavior over time.*

5.4.2 Closed-loop Performance and Stability Conditions

With the composite Lyapunov function, we can then prove the closed-loop stability and establish the stability conditions of the entire multi-UAV control system using the Lyapunov stability theory.

First of all, we have the following lemma elucidating the relationship between the neighborhood formation errors and the global leader-follower tracking errors:

Lemma 5.2. *With the directed communication graph $\mathcal{G}$ satisfying Assumption 5.2, the following relationships hold*

$$\|\tilde{\zeta}\| \leqslant \frac{1}{\lambda_{\min}(\mathcal{L}+\mathcal{B})}\|e_\zeta\| \tag{5.43a}$$

$$\|\tilde{v}\| \leqslant \frac{1}{\lambda_{\min}(\mathcal{L}+\mathcal{B})}\|e_v\| \tag{5.43b}$$

Proof. From the error definitions of e_ζ, e_v, $\tilde{\zeta}$ and $\tilde{v}$, we can have

$$e_\zeta = ((\mathcal{L}+\mathcal{B}) \otimes I_3)\,\tilde{\zeta} \tag{5.44a}$$

$$e_v = ((\mathcal{L}+\mathcal{B}) \otimes I_3)\,\tilde{v} \tag{5.44b}$$

Recalling Lemma 5.1, $\mathcal{L}+\mathcal{B}$ is nonsingular and positive-definite. It can then be obtained from (5.44) that

$$\tilde{\zeta} = \left((\mathcal{L}+\mathcal{B})^{-1} \otimes I_3\right) e_\zeta \tag{5.45a}$$

$$\tilde{\upsilon} = \left((\mathcal{L}+\mathcal{B})^{-1} \otimes I_3\right) e_\upsilon \tag{5.45b}$$

We can finally derive

$$\|\tilde{\zeta}\| \leqslant \| \left((\mathcal{L}+\mathcal{B})^{-1} \otimes I_3\right) e_\zeta\| \leqslant \frac{1}{\lambda_{\min}(\mathcal{L}+\mathcal{B})}\|e_\zeta\| \tag{5.46a}$$

$$\|\tilde{\upsilon}\| \leqslant \| \left((\mathcal{L}+\mathcal{B})^{-1} \otimes I_3\right) e_\upsilon\| \leqslant \frac{1}{\lambda_{\min}(\mathcal{L}+\mathcal{B})}\|e_\upsilon\| \tag{5.46b}$$

□

In order to formulate the global formation tracking performance of the entire N-UAV system, we introduce a scaled state vector $z = \left[\|s_\zeta\| \ \frac{1}{\varepsilon_\eta}\|e_\eta\| \ \frac{1}{\varepsilon_\omega}\|e_\omega\|\right]^\top$, in which $\varepsilon_\eta, \varepsilon_\omega \in \mathbb{R}^+$ are scaling factors. With respect to z, the Lyapunov function V can be rewritten as

$$\begin{aligned} V =& \frac{1}{2}z^\top S z + \frac{1}{2}\tilde{\vartheta}_\zeta^\top K_\zeta^{-1}\mathrm{diag}(\vartheta_\zeta)^{-1}\tilde{\vartheta}_\zeta + \frac{1}{2}\tilde{\vartheta}_{\zeta^0}^\top \left(K_{\zeta^0}\otimes I_3\right)^{-1}\tilde{\vartheta}_{\zeta^0} + \frac{1}{2}\tilde{\vartheta}_{\zeta^a}^\top \left(K_{\zeta^a}\otimes I_3\right)^{-1}\tilde{\vartheta}_{\zeta^a} \\ &+ \frac{1}{2}\tilde{\vartheta}_\omega^\top \left(K_\omega \otimes I_3\right)^{-1}\mathrm{diag}(\vartheta_\omega)^{-1}\tilde{\vartheta}_\omega + \frac{1}{2}\tilde{\vartheta}_{\omega^0}^\top \left(K_{\omega^0}\otimes I_3\right)^{-1}\tilde{\vartheta}_{\omega^0} \end{aligned} \tag{5.47}$$

where

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \varepsilon_\eta^2 & 0 \\ 0 & 0 & \varepsilon_\omega^2 \end{bmatrix}$$

The Lyapunov function is in the positive-definite quadratic form of z and estimation errors, thus it can be upper bounded by

$$V \leqslant \frac{1}{2}\lambda_{\max}(S)\|z\|^2 + \mu \tag{5.48}$$

where $\mu = \frac{\overline{k}_\zeta}{2\underline{\vartheta}_\zeta}\rho_{\vartheta_\zeta} + \frac{\overline{k}_{\zeta^0}}{2}\rho_{\vartheta_\zeta^0} + \frac{\overline{k}_{\zeta^a}}{2}\rho_{\vartheta_\zeta^a} + \frac{\overline{k}_\omega}{2\underline{\vartheta}_\omega}\rho_{\vartheta_\omega} + \frac{\overline{k}_{\omega^0}}{2}\rho_{\vartheta_\omega^0}$ with $\overline{k}_\zeta$, $\overline{k}_{\zeta^0}$, $\overline{k}_{\zeta^a}$, $\overline{k}_\omega$ and $\overline{k}_{\omega^0}$ being the maximal adaptive gains, $\underline{\vartheta}_\zeta$ and $\underline{\vartheta}_\omega$ being the lower bounds of $\|\vartheta_\zeta\|$ and $\|\vartheta_\omega\|$, ρ_{ϑ_ζ}, $\rho_{\vartheta_\zeta^0}$, $\rho_{\vartheta_\zeta^a}$, ρ_{ϑ_ω} and $\rho_{\vartheta_\omega^0}$ being the level values of the stability regions for

$\|\vartheta_\zeta\|^2$, $\|\vartheta_{\zeta^0}\|^2$, $\|\vartheta_{\zeta^a}\|^2$, $\|\vartheta_\omega\|^2$ and $\|\vartheta_{\omega^0}\|^2$.

Theorem 5.1 below summarizes the main result of the closed-loop analysis, concluding that the proposed distributed triple-layer control design can achieve the formation tracking objective with certain sufficient stability conditions.

Theorem 5.1. *Consider a group of N UAVs connected via the directed graph $\mathcal{G}$ with Assumption 5.2 satisfied for a formation tracking task toward a virtual leader. Each UAV i in $\mathcal{V}$ is modeled by (5.1) subject to unexpected actuator faults (5.3), and manipulated by a distributed controller, consisting of an MPC-based outer translation control layer with the MPC optimization problem formulated as (5.19) and the adaptive parameter estimation laws designed as (5.15), an intermediate rotation angle control layer with the control law developed as (5.26), and an inner angular velocity control layer with the control law (5.28) and adaptive laws (5.29). The control gains c_{ζ_i}, c_{η_i} and c_{ω_i} of three control layers for $i \in \mathcal{V}$ are selected in proportion to the scales of their respective motions, such that a positive constant c exists satisfying the following rule*

$$c = \lambda_{\min}(C_\zeta) = \varepsilon_\eta^2 \lambda_{\min}(C_\eta) = \varepsilon_\omega^2 \lambda_{\min}(C_\omega) \tag{5.49}$$

For $z(0) \in \Omega_{\rho_0} \triangleq \{z | V \leqslant \rho_0\}$ with V being a Lyapunov function defined later in the proof, if the following stability conditions are satisfied by c and the outer-layer control update period δ:

$$\text{(C1)} \quad c^2 > \frac{\varepsilon_\eta^2 \kappa_1^2 \lambda_{\max}^2(\mathcal{L}+\mathcal{B})}{4} + \frac{\varepsilon_\eta^2 \varepsilon_\omega^2}{4}$$

$$\text{(C2)} \quad -\frac{2\left(c - \frac{\varepsilon_\eta \sqrt{\varepsilon_\omega^2 + \kappa_1^2 \lambda_{\max}(\mathcal{L}+\mathcal{B})^2}}{2}\right)(\rho_z - \mu)}{\max\left(1, \epsilon_\eta^2, \epsilon_\omega^2\right)} + \left(\kappa_2 + \kappa_3 \lambda_{\max}(C_\zeta)\right)\delta \leqslant 0$$

then $z(t)$ remains within Ω_{ρ_0} for all $t > 0$ and converges to $\Omega_{\rho_z} \triangleq \{z | V \leqslant \rho_z\}$ eventually. This further implies the ultimate boundedness of $\|s_\zeta\|$, $\|e_\eta\|$ and $\|e_\omega\|$. Additionally, the neighborhood tracking errors $\|e_\zeta\|$, $\|e_v\|$ and the leader tracking errors $\|\tilde{\zeta}\|$, $\|\tilde{v}\|$ are also ultimately bounded, which means that the formation tracking control objective is achieved.

Proof. We start by proving that the closed-loop trajectory of z, originating from an initial state in Ω_{ρ_0}, remains in Ω_{ρ_0} for all $t > 0$ and eventually converges to the set Ω_{ρ_z} as $t \to \infty$.

If the user-determined control parameters are chosen following the rule (5.49), we

can rewrite (5.42) as

$$\dot{V} \leqslant -z^{\top} H z + \left(\kappa_2 + \kappa_3 \lambda_{\max}(C_\zeta)\right) \delta \tag{5.50}$$

where

$$H = \begin{bmatrix} c & -\frac{\varepsilon_\eta \kappa_1 \lambda_{\max}(\mathcal{L}+\mathcal{B})}{2} & 0 \\ -\frac{\varepsilon_\eta \kappa_1 \lambda_{\max}(\mathcal{L}+\mathcal{B})}{2} & c & -\frac{\varepsilon_\eta \varepsilon_\omega}{2} \\ 0 & -\frac{\varepsilon_\eta \varepsilon_\omega}{2} & c \end{bmatrix}$$

If condition (C1) holds the matrix H is positive-definite. In this case, we have

$$\dot{V} \leqslant -\lambda_{\min}(H) \|z\|^2 + \left(\kappa_2 + \kappa_3 \lambda_{\max}(C_\zeta)\right) \delta \tag{5.51}$$

where $\lambda_{\min}(H) = c - \frac{\varepsilon_\eta \sqrt{\varepsilon_\omega^2 + \kappa_1^2 \lambda_{\max}(\mathcal{L}+\mathcal{B})^2}}{2}$. Since the Lyapunov function V defines the level sets Ω_{ρ_0} and Ω_{ρ_z}, hence it is direct to derive $\dot{V} \leqslant 0$ for V large enough. Combining (5.48) and (5.51), one has

$$\dot{V} \leqslant -\frac{2\lambda_{\min}(H)}{\lambda_{\max}(S)}(V - \mu) + \left(\kappa_2 + \kappa_3 \lambda_{\max}(C_\zeta)\right) \delta \tag{5.52}$$

where $\lambda_{\min}(S) = \min\left(1, \varepsilon_\eta^2, \varepsilon_\omega^2\right)$. If condition (C2) is satisfied, then it can be derived that $\dot{V} < 0$ for all $z \in \{z | \rho_z < V \leqslant \rho_0\}$ and $\dot{V} = 0$ for $V = \rho_z$. Therefore, it can be proven that z converges to Ω_{ρ_z} as t approaches ∞ without leaving the stability region Ω_{ρ_0}.

The boundedness of the collected state z further implies the ultimate boundedness of $\|s_\zeta\|$, $\|e_\eta\|$ and $\|e_\omega\|$. Since s_ζ defines the sliding surface for e_ζ, we can thereby further conclude the boundedness of $\|e_\zeta\|$ and $\|e_v\|$.

Finally, with Lemma 5.2, we can conclude that the practical stability of the leader-follower tracking errors $\|\tilde{\zeta}\|$ and $\|\tilde{v}\|$. This completes the proof of Theorem 5.1. □

Remark 5.4. *The sufficient stability conditions (C1) and (C2) provided in Theorem 5.1 guarantees the formation tracking performance of the developed distributed control system, on account of the translation updating period fixed as δ and the control parameters selected following the rule (5.49). Notice that parameter selection rules are not unique. Here, we just provide one solution for parameter selection that allocates the gains in proportion to the scales of their respective motions.*

5.5 Simulation Study

To verify the effectiveness of the proposed control design, we consider a 4-UAV system connected via a directed graph for numerical simulations to perform a formation tracking task toward a virtual leader. The selection of system and user-defined control parameters are given in 5.5.1. The fault-tolerant formation tracking performance in the presence of actuator faults is shown in 5.5.2.

5.5.1 Parameter Selection

The model parameters of the 4 vehicles are listed in TABLE 5.1. The communication topology among the 4 UAVs and the virtual leader is illustrated in Fig. 5.2. The adjacency matrix, in-degree matrix, Laplacian matrix, and pinning matrix are

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

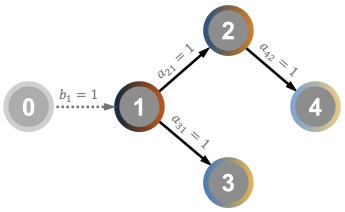

Figure 5.2: Digraph of 4 agents with leader node 0

Table 5.1: Model parameters in simulations

Model parameters		Symbols	Values	Units	Model parameters		Symbols	Values	Units
UAV 1	Vehicle mass	m_1	2.62	kg	UAV 3	Vehicle mass	m_3	2.68	kg
	Inertia moments	J_{xx_1}	0.043	kg·m^2		Inertia moments	J_{xx_3}	0.049	kg·m^2
		J_{yy_1}	0.043	kg·m^2			J_{yy_3}	0.049	kg·m^2
		J_{zz_1}	0.063	kg·m^2			J_{zz_3}	0.069	kg·m^2
UAV 2	Vehicle mass	m_2	2.65	kg	UAV 4	Vehicle mass	m_4	2.7	kg
	Inertia moments	J_{xx_2}	0.046	kg·m^2		Inertia moments	J_{xx_4}	0.051	kg·m^2
		J_{yy_2}	0.046	kg·m^2			J_{yy_4}	0.051	kg·m^2
		J_{zz_2}	0.066	kg·m^2			J_{zz_4}	0.071	kg·m^2

Table 5.2: User-determined adaptive parameters in simulations

Adaptive gains	Symbols	Values	Adaptive gains	Symbols	Values
Estimation gains	$K_{v^a_{1,2,3,4}}$	$\mathrm{diag}\,(0.1, 0.1, 0.1)$	Convergence gains	$\Gamma_{v^a_{1,2,3,4}}$	$\mathrm{diag}\,(30, 30, 30)$
	$K_{v^n_{1,2,3,4}}$	$\mathrm{diag}\,(0.0001, 0.0001, 0.0001)$		$\Gamma_{v^n_{1,2,3,4}}$	$\mathrm{diag}\,(3\times10^4, 3\times10^4, 3\times10^4)$
	$k_{v^m_{1,2,3,4}}$	0.05		$\gamma_{v^m_{1,2,3,4}}$	60
	$K_{\omega^a_{1,2,3,4}}$	$\mathrm{diag}\,(1, 1, 1)$		$\Gamma_{\omega^a_{1,2,3,4}}$	$\mathrm{diag}\,(3, 3, 3)$
	$K_{\omega^m_{1,2,3,4}}$	$\mathrm{diag}\,(0.002, 0.002, 0.002)$		$\Gamma_{\omega^m_{1,2,3,4}}$	$\mathrm{diag}\,(1.5\times10^3, 1.5\times10^3, 1.5\times10^3)$

The desired formation displacement vectors of the 4 UAVs with respect to the leader are set as $\Delta_{10} = [-2\ 0\ 0]^\top$, $\Delta_{20} = [0\ 2\ 0]^\top$, $\Delta_{30} = [0\ -2\ 0]^\top$, $\Delta_{40} = [2\ 0\ 0]^\top$. The initial positions of the 4 followers are $\zeta_1(0) = [10\ 2\ 21]^\top$, $\zeta_2(0) = [10\ 0\ 20.5]^\top$, $\zeta_3(0) = [10\ 0\ 19.5]^\top$, $\zeta_4(0) = [10\ 0\ 20]^\top$. The linear velocities are the same at the initial point as $v_1(0) = v_2(0) = v_3(0) = v_4(0) = [0\ 2\ 0]^\top$. The initial rotation angles and the angular velocities are $\eta_1(0) = \eta_2(0) = \eta_3(0) = \eta_4(0) = [0\ 0\ 0]^\top$ and $\omega_1(0) = \omega_2(0) = \omega_3(0) = \omega_4(0) = [0\ 0\ 0]^\top$. The initial values of the adaptive estimates are selected as $\hat{\vartheta}_{\zeta_1,p_2,p_3,p_4}(0) = \hat{\vartheta}_{\eta_1,q_2,q_3,q_4}(0) = [0\ 0\ 0]^\top$.

Table 5.3: User-determined control parameters in simulations

Control parameters		Symbols	Values	Control parameters		Symbols	Values
UAV 1	Boundary loop gain	λ_1	0.2	UAV 3	Boundary loop gain	λ_3	0.2
	Control gains	c_{ζ_1}	5		Control gains	c_{ζ_3}	5.2
		c_{η_1}	20			c_{η_3}	20
		c_{ω_1}	35			c_{ω_3}	35
	MPC cost weights	Q_1	$\mathrm{diag}\,(2,2,5)$		MPC cost weights	Q_3	$\mathrm{diag}\,(2,2,5)$
		R_{F_1}	0.1			R_{F_3}	0.1
		R_{η_1}	$\mathrm{diag}\,(10,10,10)$			R_{η_3}	$\mathrm{diag}\,(10,10,10)$
UAV 2	Boundary loop gain	λ_2	0.2	UAV 4	Boundary loop gain	λ_4	0.2
	Control gains	c_{ζ_2}	5.1		Control gains	c_{ζ_4}	5
		c_{η_2}	20.2			c_{η_4}	21
		c_{ω_2}	35.1			c_{ω_4}	36
	MPC cost weights	Q_2	$\mathrm{diag}\,(2,2,5)$		MPC cost weights	Q_4	$\mathrm{diag}\,(2,2,5)$
		R_{F_2}	0.1			R_{F_4}	0.1
		R_{η_2}	$\mathrm{diag}\,(10,10,10)$			R_{η_4}	$\mathrm{diag}\,(10,10,10)$

In simulations, the user-determined control parameters and the adaptive parameters are selected by following the rules and the stability conditions presented in Theorem 5.1. The Lipschitz constants κ_1, κ_2 and κ_3 can be obtained by evaluating the bounds of dynamics within the stability region and the input constraints, whose values are approximated as $\kappa_1 \approx 11$, $\kappa_1 \approx 6$, $\kappa_1 \approx 2$. Considering the limited computation and communication capabilities in practice, the minimum acceptable control update period for the outer-loop loop is 0.1 s. For selecting the appropriate control gains and parameters for this control system set up, we first select the scaling factors to align with the scales of system variables as $\varepsilon_\eta = 0.5$, $\varepsilon_\omega = 0.378$, $\varepsilon_{\vartheta_v^a} = 0.3162$, $\varepsilon_{\vartheta_v^n} = 0.01$, $\varepsilon_{\vartheta_v^m} = 0.3661$, $\varepsilon_{\vartheta_\omega^a} = 1$, $\varepsilon_{\vartheta_\omega^m} = 0.0011$. The adaptive estimation gains are selected to make $\underline{k}$ and $\overline{k}$ near 1, and the adaptive convergence gains are selected based on the rule with $\gamma = 3$, as given in TABLE 5.2. According to the stability conditions (C1) and (C2), c is chosen as 5. The control gains are then selected based on the rule (5.49) as given in TABLE 5.3.

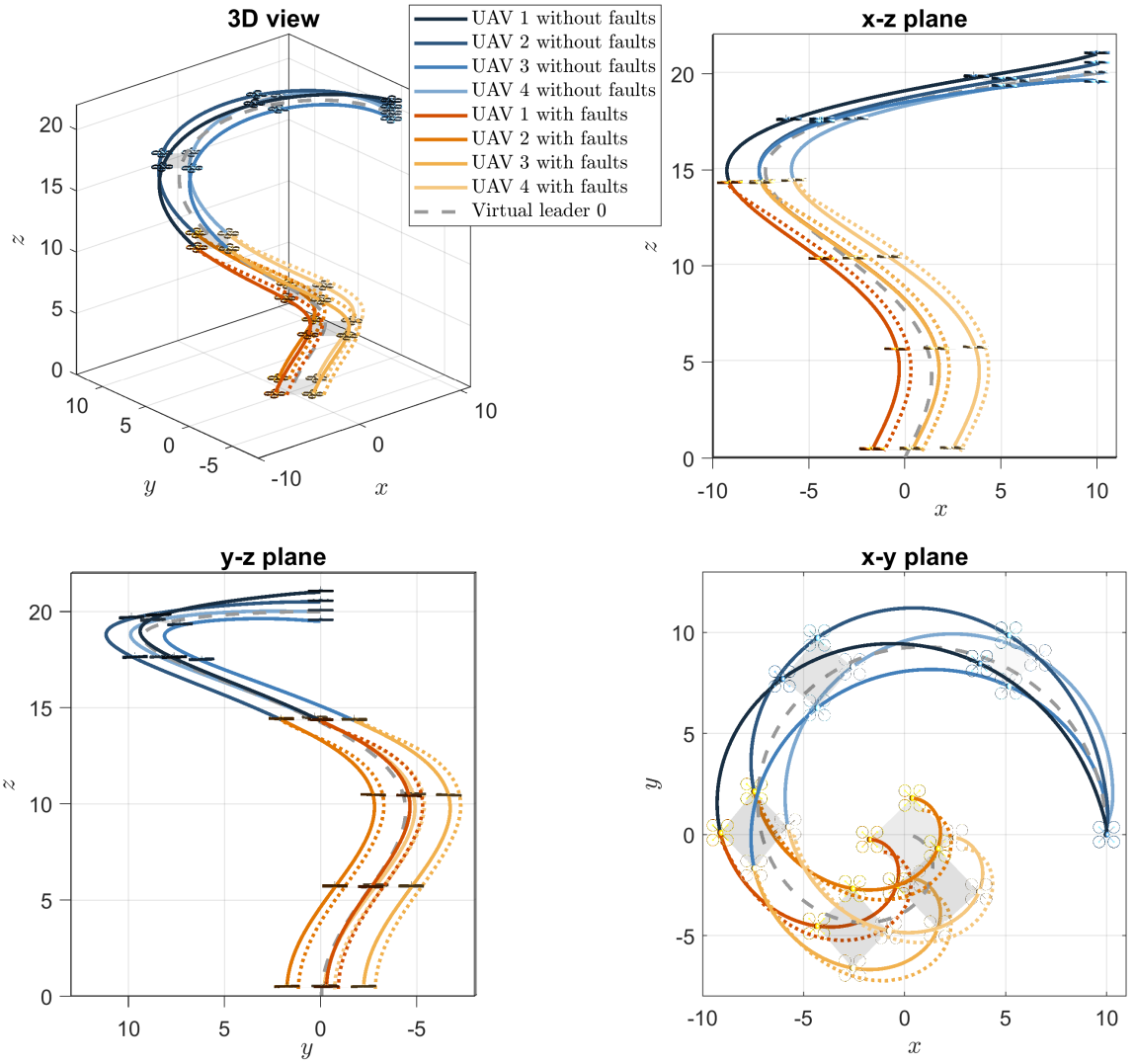

Figure 5.3: Formation tracking of the 4-UAV system with a virtual leader

5.5.2 Fault-tolerant Formation Tracking Performance

The virtual leader to be tracked follows a downward spiral trajectory, as given below for $t \in [0, 30]$

$$\dot{\zeta}_0(t) = v_0(t) \tag{5.53a}$$

$$\dot{v}_0(t) = \begin{bmatrix} \frac{\pi^2}{45}\sin(\frac{\pi}{15}t)\sin(\frac{\pi}{60}t) - \frac{17\pi^2}{360}\cos(\frac{\pi}{15}t)\cos(\frac{\pi}{60}t) \\ -\frac{\pi^2}{45}\cos(\frac{\pi}{15}t)\sin(\frac{\pi}{60}t) - \frac{17\pi^2}{360}\sin(\frac{\pi}{15}t)\cos(\frac{\pi}{60}t) \\ -\frac{\pi^2}{180}\cos(\frac{\pi}{60}t) \end{bmatrix} \tag{5.53b}$$

To manifest the fault-tolerant capability of the proposed design, we set up the

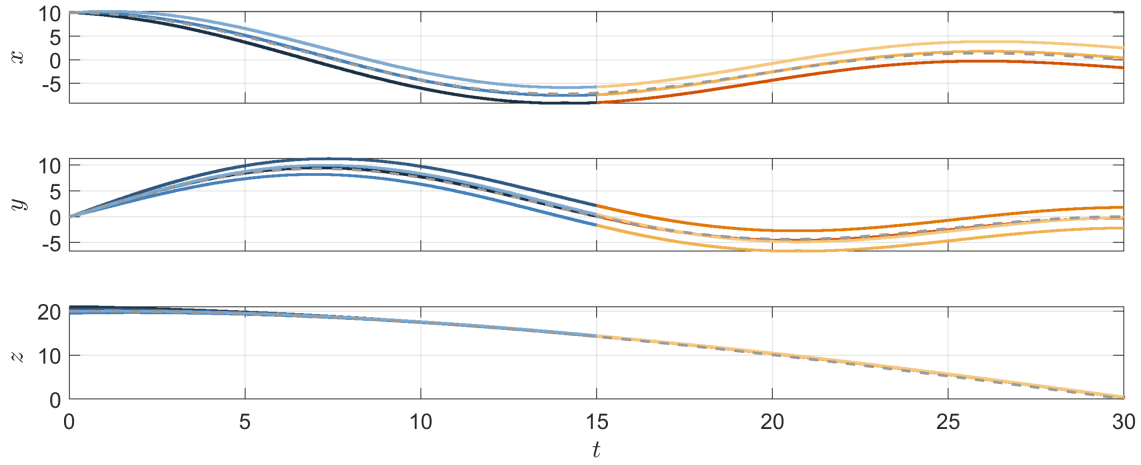

Figure 5.4: Translation motions of the 4 UAVs

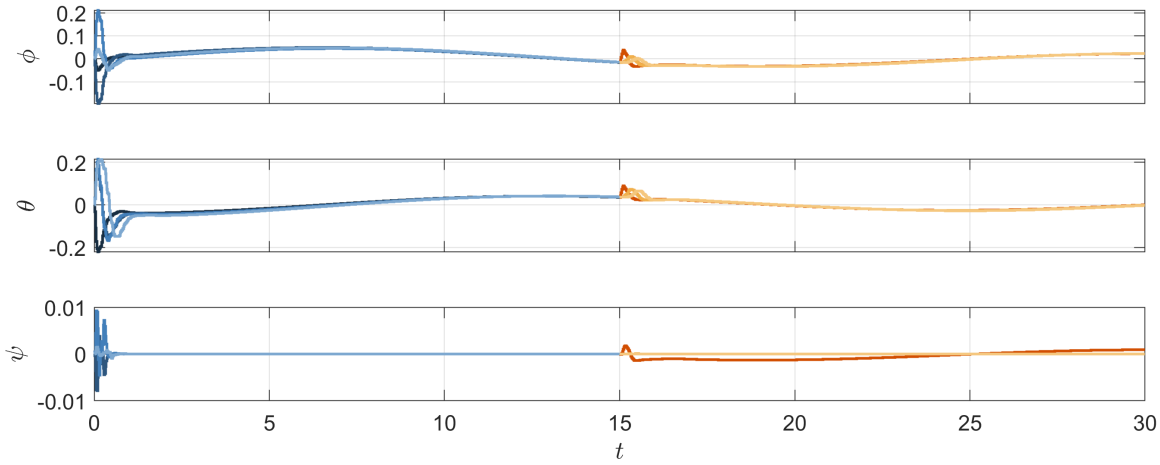

Figure 5.5: Rotation motions of the quadrotor

simulation as

- *Fault-free case:* When 0 s $\leqslant t <$ 15 s, all the 4 UAVs work properly in fault-free case.
- *Actuator fault case:* When 15 s $\leqslant t <$ 30 s, an over-voltage failure happens on two of the rotors of UAV 1, causing a sudden increase of 1 N on the rotor's thrust force T_1 and T_4. Also, a loss-of-effectiveness fault takes place, causing a 10% reduction in forces and torques.
- *Comparison method:* A MPC design without adaptive fault estimation is also tested for comparison.

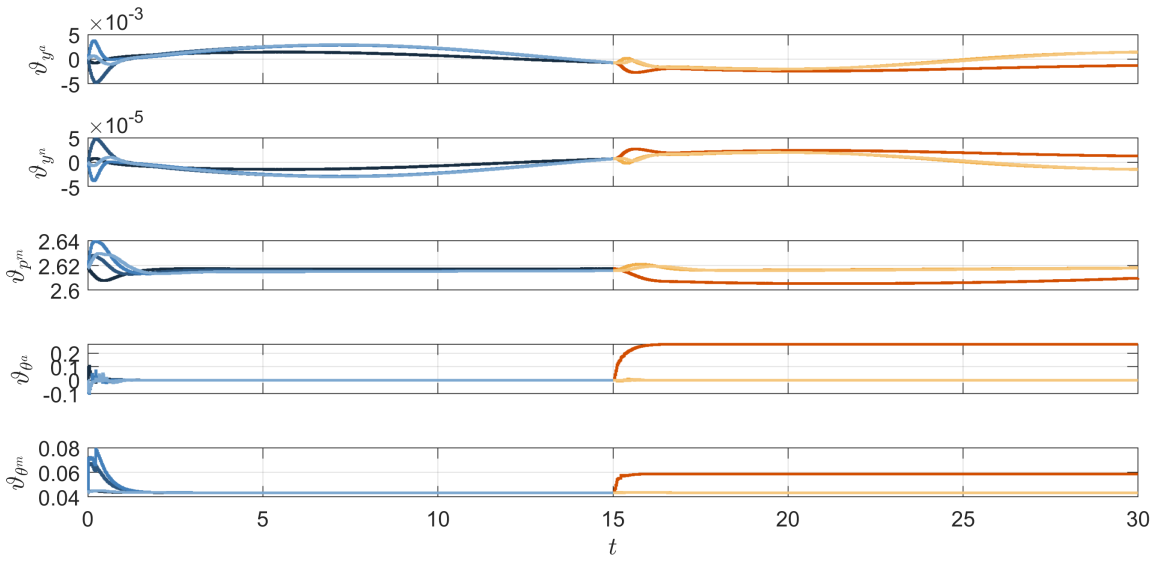

Figure 5.6: Online parameter estimations

The simulation results are shown in Figs. 5.3-5.7, where system responses in fault-free and faulty cases are indicated by the blue and orange lines, respectively. Figs. 5.4 and 5.5 show the translational and rotational motions of the four UAVs. It can be seen that UAVs track the virtual leader in the desired square shape with high accuracy. At 15 s, when the actuator fault happens on UAV 1, the formation tracking performance is slightly degraded but then recovers in a short time. The compensation for the unexpected faults is achieved by the adaptive parameter estimation, as shown by Fig. 5.6. The left half of Fig. 5.7 shows the control commands generated by the controllers and the right half shows the forces and torques that are actually applied to UAVs. It is obvious that the input constraints are satisfied at all times. The resultant formation tracking performance of the comparison method is indicated by the dotted lines in Fig. 5.3. Without the adaptive estimator, the performance obviously deteriorates after the fault occurs at 15 s and gets even worse over time. Therefore, by this simulation test, we can verify the effectiveness of the proposed design in terms of fault tolerance.

5.6 Conclusions

This chapter presents a new control framework for multi-UAV formation tracking toward a dynamic virtual leader in the presence of unexpected actuator faults. The proposed design employs a distributed control structure, where each agent is con-

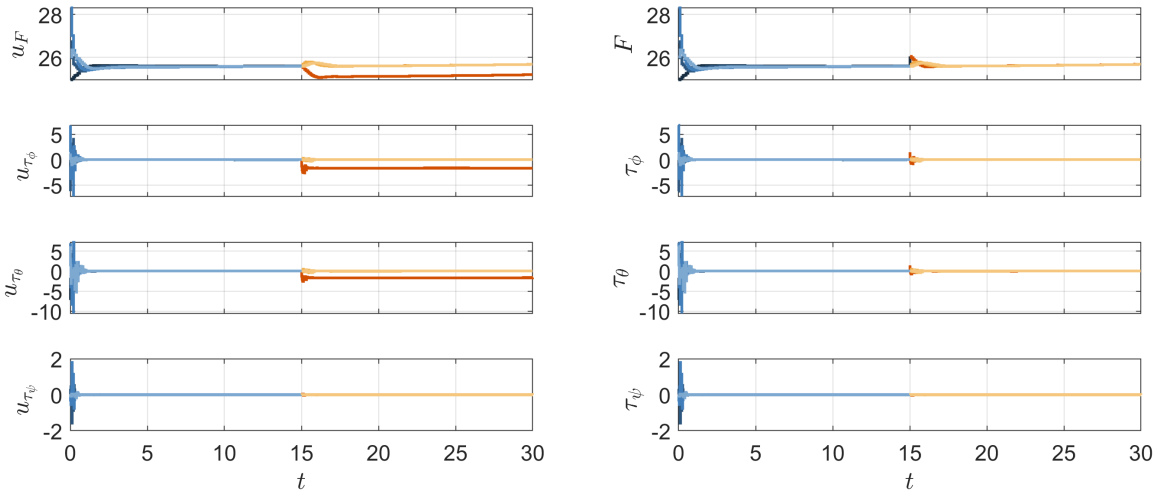

Figure 5.7: Control commands (left); Actually applied force and torques (right)

nected through a directed communication graph. Within each local control system of this distributed control network, a triple-loop hierarchical control architecture is proposed to handle complex UAV dynamics. The proposed Lyapunov-based MPC design improves upon existing algorithms by relaxing requirements on leader dynamics and vehicle connectivity, enhancing communication and computation efficiency. This chapter also addresses the hybrid communication problem that arises with the hierarchical control architecture by incorporating the sample-and-hold implementation of the outer control loop at both the control development and closed-loop analysis levels. The integration of Lyapunov-based MPC and adaptive parameter estimation provides an adaptive solution for handling unknown faults with less conservatism. The closed-loop stability of the overall system is proven using Lyapunov stability theory and graph theory. Numerical simulations demonstrate the performance of formation tracking and fault tolerance for the proposed design.

Chapter 6

Adaptive Distributed Observer-based Model Predictive Control for Multi-agent Formation with Resilience to Communication Link Faults

Chapter 5 proposes a distributed fault-tolerant model predictive control (FTMPC) design for multi-UAV systems to maintain formation tracking in the presence of actuator faults. In addition to the faults that affect individual local agents, the communication network is also susceptible to potential faults or external attacks. In this chapter, we aim to develop a novel distributed FTMPC method for enhancing the resilience of a class of nonlinear multi-agent systems against communication link faults.

6.1 Introduction

In a multi-agent system (MAS), the communication network plays a crucial role in enabling interactions and facilitating cooperative behaviors among agents. However, the inclusion of communication networks introduces additional vulnerabilities to the control system, particularly when facing unexpected events such as cyber-attacks and channel fading. Communication faults between agents can pose major threats

to multi-agent control systems, potentially deteriorating control performance or even overall system stability. Therefore, it is critical to develop resilient distributed control strategies to mitigate the negative effects of these communication faults. Motivated by this need, this chapter focuses on the development of a novel resilient distributed model predictive control (DMPC) method to achieve formation tracking of nonlinear MASs despite the presence of communication faults.

Attacks and fading within communication networks can be modeled as uncertainties in the communication links. Recent studies in [175–177] explore the consensus of MASs with stochastic uncertain communication networks. In [178,179], deterministic network uncertainties are examined within the context of MASs with single integrator agents. In [136], a distributed state observer-based adaptive control protocol is designed to address the leader-follower consensus for linear MASs with communication link faults. This study demonstrates that the distributed leader state observer network is resilient to communication link faults. However, it requires that all following agents know the leader dynamics. As an extension of this result, [180] proposes adaptive distributed leader state/dynamics observers and control protocols, offering a completely distributed solution for synchronizing linear MASs with time-varying edge weights without the need for global knowledge of the leader dynamics. Most existing research on resilience cooperative control in the presence of communication uncertainties is directed towards unconstrained MASs with linear dynamics. Moreover, to the best of our knowledge, fully distributed control for formation tracking under communication link faults has not yet received significant research attention.

Motivated by the aforementioned investigations, this chapter develops a novel adaptive distributed observer-based DMPC method for nonlinear MASs in the presence of input constraints and communication link faults. To achieve the formation tracking objective without relying on global access to the leader's information, adaptive distributed observers are developed for all local control systems, estimating online the leader's state, dynamics, and the desired relative position with respect to the leader. With information estimated by these observers, distributed MPC controllers are independently developed to manipulate each agent toward a predetermined formation relative to the estimated leader while adhering to input constraints. The asymptotic convergence of the observation process is demonstrated, which in turn proves the closed-loop control performance of the overall system. To validate the efficacy of the proposed design, simulations are conducted using both a numerical example and a practical 5-UAV system. The key contributions of this research work

include:

- In contrast to prior works such as [136, 178–180] that focus on the consensus problem in unconstrained, linear MASs, this study explores the formation tracking control problem in MASs with both input constraints and nonlinear dynamics. Adaptive distributed observers are utilized not only to estimate the leader's state and dynamics but also the desired formation displacements of each agent relative to the leader. With the estimated real-time information as the reference, MPC is employed for the local controller design to achieve optimized control performance subject to the input constraints.

- By locally estimating tracking references through corresponding adaptive observers, the distributed formation tracking control task can be decoupled into several fully distributed tracking control problems at the local level. This facilitates the development of local controllers. Therefore, the integration of adaptive observers can significantly reduce the complexity of the distributed MPC formulation compared to other designs proposed in [44, 46, 73, 181, 182].

The organization of the remainder of this chapter is structured as follows: Section 6.2 provides the mathematical formulation of the control problem and objective; Section 6.3 elaborates on the distributed control design, presenting the adaptive observer and the MPC-based controller; Section 6.4 conducts the closed-loop analysis, evaluating the convergence of the estimation and the stability of the control system; Section 6.5 offers two simulation examples to validate the effectiveness of the proposed design; Finally, Section 6.6 summarizes this chapter.

6.2 Problem Formulation

This section outlines the mathematical formulation of the control problem to tackle: multi-agent formation tracking control in the presence of communication faults. Firstly, we detail the dynamics models of individual agents and the virtual leader and describe their intercommunication through a directed weighted graph. Subsequently, the modeling of communication faults is presented. Finally, we introduce leader-follower tracking errors to formulate the control objective for formation tracking.

6.2.1 Multi-agent System

Consider a multi-agent system comprising M followers and one virtual leader. The dynamics of both followers and the leader are detailed below, while their interactions are modeled using a weighted directed graph.

A. Follower Dynamics and Intercommunication

The dynamics of the ith follower can be described by the following higher-order MIMO nonlinear model:

$$\begin{cases} \dot{x}_{i,1} &= x_{i,2} \\ &\vdots \\ \dot{x}_{i,r-1} &= x_{i,r} \\ \dot{x}_{i,r} &= f_i(x_i) + G_i(x_i)u_i \\ y_i &= x_{i,1} \end{cases} \tag{6.1}$$

where $x_i = \begin{bmatrix} x_{i,1}^\top & x_{i,2}^\top & \cdots & x_{i,r}^\top \end{bmatrix}^\top \in \mathbb{R}^{rn}$ is the system state vector with each segment $x_i^l \in \mathbb{R}^n$ for $l = 1, 2, \cdots, r$; $u_i = [u_{i,1}\ u_{i,2}\ \cdots\ u_{i,n}] \in \mathbb{R}^n$ and $y_i = [y_{i,1}\ y_{i,2}\ \cdots\ y_{i,n}] \in \mathbb{R}^n$ are the control input and system output, respectively; $f_i(x_i) = [f_{i,1}(x_i)\ f_{i,1}(x_i)\ \cdots\ f_{i,n}(x_i)]^\top : \mathbb{R}^{rn} \to \mathbb{R}^n$ is a vector function, and $G_i(x_i) = [g_{i,1}(x_i)\ g_{i,2}(x_i)\ \cdots g_{i,n}(x_i)] : \mathbb{R}^{rn} \to \mathbb{R}^{n\times n}$ is a square matrix function. To ensure that the system's behavior is predictable and well-behave around the origin, the following assumption is necessary and commonly employed

Assumption 6.1. *Al entries of $f_i(x_i)$ and $G_i(x_i)$ are sufficiently smooth and locally Lipschitz function of x_i and satisfy $f_i(0) = 0$ and $G_i(0) \neq 0$.*

The communication among these M followers can be described using a directed weighted graph. As previously introduced in Section 2.1.2, such graph can be represented by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. In this representation, $\mathcal{V} = \{1, 2, \cdots, M\}$ is the set of nodes, with each node corresponding to a follower agent. $\mathcal{E} = \{(j, i) | i, j \in \mathcal{V}, i \neq j\}$ is the set of edges and $(j, i) \in \mathcal{E}$ means there is a communication link from agent j to agent i. Associated with this graph are two critical matrices. The adjacency matrix $\mathcal{A} = [a_{ij}]$ is defined such that $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The Laplacian matrix $\mathcal{L} = [l_{ij}]$ is defined with $l_{ii} = \sum_{j=1}^{M} a_{ij}$ capturing the in-degree of node i and $l_{ij} = -a_{ij}$ for $i \neq j$.

B. Leader Dynamics and Connectivity

In addition to the follower agents, the system includes a virtual leader whose role is to guide the overall behavior of the MAS. The dynamics of this virtual leader can be governed by:

$$\dot{\xi}_0 = S_0 \xi_0 \tag{6.2}$$

where $\xi_0 = \left[\xi_{0,1}^\top \ \xi_{0,2}^\top \ \cdots \ \xi_{0,r}^\top\right]^\top \in \mathbb{R}^{rn}$ represents the state vector of the leader; $S_0 \in \mathcal{R}^{rn \times rn}$ denotes the system dynamics matrix.

Remark 6.1. *It is imperative that the leader's state vector ξ_0 is of equivalent dimensionality to the followers' dynamics, ensuring it can serve as a reference for the followers' outputs. For instance, the lth segment of ξ_0 serves as the reference for $y_i^{(l)}$ of follower i.*

Note that the state vector ξ_0 and the dynamics matrix S_0 of the leader are only accessible to certain followers. The leader can be labeled as node 0, and the connections from this leader to the followers, labeled $1, 2, \cdots, M$, can be defined by a set of pinning edges $\mathcal{E}^0 = \{(0, i) | i \in \mathcal{V}\}$. An edge $(0, i) \in \mathcal{E}^0$ indicates that follower i has direct access to the leader's state and dynamics. Additionally, we introduce a pinning matrix $\mathcal{B} = \text{diag}(b_1, b_2, \cdots, b_M)$, where $b_i > 0$ if $(0, i) \in \mathcal{E}^0$ and $b_i = 0$ otherwise. This matrix effectively quantifies the influence of the leader on each follower.

6.2.2 Input Constraint and Communication Fault

In this work, we address both the input constraints of individual follower agents and unknown faults that may occur within the communication network. These considerations are crucial for ensuring the robustness and reliability of the system under various operational conditions. The mathematical models that incorporate these elements are provided below.

Firstly, considering the limitations on excitable control actions, the control input of the ith follower is restricted to a nonempty compact convex set, as defined by

$$u_i \in \Omega_{u_i} \triangleq \{u_i \in \mathbb{R}^n | \ -\underline{u}_i \leqslant u_i \leqslant \overline{u}_i\} \tag{6.3}$$

where $\underline{u}_i = [\underline{u}_{i,1} \ \underline{u}_{i,2} \ \cdots \ \underline{u}_{i,n}]^\top \in \mathbb{R}^{n+}$ and $\overline{u}_i = [\overline{u}_{i,1} \ \overline{u}_{i,2} \ \cdots \ \overline{u}_{i,n}]^\top \in \mathbb{R}^{n+}$.

Recalling the modeling given previously in 2.11, communication faults can be

modeled as time-varying uncertainties affecting the graph edges:

$$a_{ij}^f(t) = a_{ij} + \vartheta_{ij}^a(t) \tag{6.4a}$$

$$b_i^f(t) = b_i + \vartheta_i^b(t) \tag{6.4b}$$

where a_{ij} and b_i are the idea weights of general and pinning edges, and ϑ_{ij}^a and ϑ_i^b denote corrupted weights caused by communication faults.

The existence of ϑ_{ij}^a and ϑ_i^b introduces time-variation and uncertainty into the weights of the communication links. Consequently, in the event of communication link failures as specified in (6.4), both the Laplacian matrix and the pinning matrix of the directed graph $\mathcal{G}$ undergo modifications. Specifically, the Laplacian matrix is redefined as $\mathcal{L}^f(t) = [l_{ij}^f(t)]$, where $l_{ii}^f(t) = \sum_{j=1}^M a_{ij}^f(t)$ for the diagonal elements, and $l_{ij}^f(t) = -a_{ij}^f(t)$ for off-diagonal elements with $i \neq j$. Similarly, the pinning matrix is revised to $\mathcal{B}^f(t) = \text{diag}(b_1^f(t), b_2^f(t), \ldots, b_M^f(t))$.

Assumption 6.2. *The communication link faults $\vartheta_{ij}^a(t)$ and $\vartheta_i^b(t)$ in the directed graph, as well as their derivatives, are bounded. In addition, the signs of $a_{ij}^f(t)$ and $b_i^f(t)$ are the same to that of a_{ij} and b_i.*

Remark 6.2. *Assumption 6.2, as also utilized in [136, 175], ensures the boundedness of communication faults and maintains the invariance of the network structure despite these faults. The modeling of communication link faults in (6.4) under Assumption 6.2 can cover various types of communication faults and cyber attacks with bounded derivatives, such as bias attacks and fading channels.*

6.2.3 Formation Tracking Control Objective

Having modeled the MAS, taking into account input constraints and communication faults, we now proceed to formulate the formation tracking control objective. Formation refers to a specific spatial shape maintained by the followers, which is typically defined prior to executing any formation control. To delineate a formation task, we assign each follower in the system a specific formation displacement relative to the virtual leader, denoted as Δ_i for $i = 1, 2, \cdots, M$. Furthermore, let $x = \begin{bmatrix} x_1^\top & x_2^\top & \cdots & x_M^\top \end{bmatrix}^\top \in \mathbb{R}^{rnM}$ represent the collective state vector of all the followers. The state of the leader 0 is extended correspondingly as $\xi = 1_M \otimes \xi_0 \in \mathbb{R}^{rnM}$. Then,

a global formation tracking error can be defined

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1^\top & \tilde{x}_2^\top & \cdots & \tilde{x}_M^\top \end{bmatrix}^\top = x - \xi - \Delta \tag{6.5}$$

where $\Delta = \begin{bmatrix} \Delta_1^\top & \Delta_2^\top & \cdots & \Delta_M^\top \end{bmatrix}^\top \in \mathbb{R}^{rnM}$ is the collective formation displacement vector.

Assumption 6.3. *To define a practical formation task, the displacement vector Δ_i, which encodes the desired offset between $x_i = \begin{bmatrix} y_i^\top & \dot{y}_i^\top & \cdots & y_i^{(r-1)\top} \end{bmatrix}^\top$ and ξ_0, is structured as $\begin{bmatrix} \Delta_{y_i}^\top & \dot{\Delta}_{y_i}^\top & \cdots & \Delta_{y_i}^{(r-1)\top} \end{bmatrix}^\top$. A default assumption is that $\Delta_{y_i} \in \mathbb{R}^n$ should be at least $(r-1)$-times differentiable, and the rth derivative, $\Delta_{y_i}^{(r)}$, is considered to be zero.*

Remark 6.3. *Given the previously defined directed communication topology, the knowledge of the leader's state and dynamics, as well as the desired formation displacement information, is not required to be globally known across the MAS. Only agents directly connected to the virtual leader have access to the real-time values of ξ_0 and the respective Δ_i. Agents that do not have a direct communication link with the virtual leader are only required to store and transmit the displacement vector relative to their out-neighbors, defined as $\Delta_{ij} = \Delta_i - \Delta_j$, along with their state measurement x_j, to their designated out-neighbor node $j \in \mathcal{N}_i^+$ via the communication links.*

The control objective of this study is to develop a distributed control strategy that utilizes solely locally available neighborhood information for effective formation tracking of a MAS composed of M followers (6.1) and a virtual leader (6.2). The primary goal is to ensure that the global formation error $\tilde{x}$ not only converges to, but also remains within a small region near the origin. To ensure such formation tracking control objective is achievable, the following assumption of the graph topology holds throughout this paper.

Assumption 6.4. *In the directed graph $\mathcal{G}$, each node is either part of a spanning tree with the root node connected to the virtual leader or a standalone node directly connected to the virtual leader.*

Remark 6.4. *The above assumption ensures that there is either direct or indirect connectivity between each follower and the leader, providing a directed path from the leader to all the followers, as stated in Lemma 2.4. This network structure is critical for achieving synchronized behaviors among the agents, enabling the distributed control strategies to eliminate the formation error across the system effectively.*

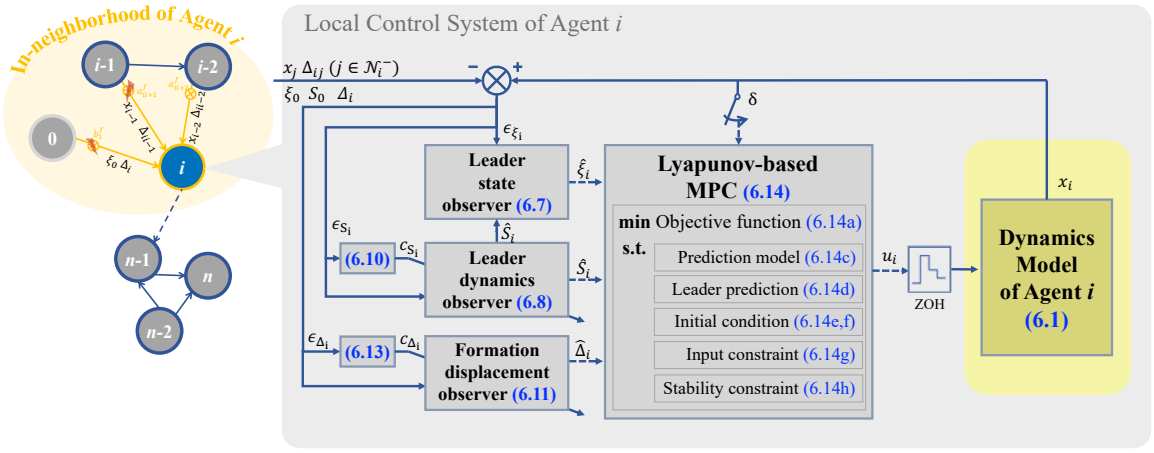

Figure 6.1: Detailed view of agent i's local control system in the distributed network

6.3 Distributed Control Design

In this section, we present the design of an adaptive distributed MPC framework for addressing the formation tracking control problem with input constraints and communication faults. This framework integrates state observers with MPC controllers via a distributed structure. As illustrated in Figure 6.1, each follower's local control system operates independently and relies exclusively on locally available information, consisting of an adaptive observer for estimating the leader information with resilience to communication link faults and an MPC-based controller for determining optimal formation tracking control actions online based on the local estimation of leader's state, dynamics matrix and desired displacement vector. Let us clarify and elaborate on each component in the subsequent subsections.

6.3.1 Adaptive Leader Observer

Given the limitations on the availability of direct, real-time access to the state and dynamics of the virtual leader among all followers in the network, it becomes essential to develop an adaptive distributed observer within each local control system. This observer is responsible for estimating the leader's information and the formation displacement, which are critical components for effective formation tracking controller design.

The locally estimated leader state for follower i is denoted as $\hat{\xi}_i$. We can then

define a leader state estimation error as

$$\epsilon_{\xi_i} = \sum_{j=1}^{M} a_{ij}(t)\left(\hat{\xi}_i - \hat{\xi}_j\right) + b_i(t)\left(\hat{\xi}_i - \xi_0\right) \tag{6.6}$$

which is available for the local control system of follower i. The distributed adaptive leader state observer is then designed as

$$\dot{\hat{\xi}}_i = \hat{S}_i\hat{\xi}_i - c_{\xi_i}\epsilon_{\xi_i} \tag{6.7}$$

where c_{ξ_i} is a user-designed positive observation gain.

In (6.7), $\hat{S}_i$ is the estimate of the leader's dynamics matrix S_0, updated following the following estimating law

$$\dot{\hat{S}}_i = -\left(c_{S_i} + \dot{c}_{S_i}\right)\epsilon_{S_i} \tag{6.8}$$

where ϵ_{S_i} is the local estimation error for S_0, defined as

$$\epsilon_{S_i} = \sum_{j=1}^{M} a_{ij}(t)\left(\hat{S}_i - \hat{S}_j\right) + b_i(t)\left(\hat{S}_i - S_0\right) \tag{6.9}$$

In (6.8), c_{S_i} satisfying $c_{S_i}(0) \geqslant 1$, is updated by

$$\dot{c}_{S_i} = \vec{\epsilon}_{S_i}^{\top}\vec{\epsilon}_{S_i} \tag{6.10}$$

with $\vec{\epsilon}_{S_i} = \mathrm{vec}(\epsilon_{S_i})$ being the vector form of the matrix ϵ_{S_i}. The operation $\mathrm{vec}(\cdot)$ rearranges the matrix segments into a column vector.

Similarly, let $\hat{\Delta}_i$ denote the estimate of the desired formation displacement Δ_i. Its estimating law is

$$\dot{\hat{\Delta}}_i = -\left(c_{\Delta_i} + \dot{c}_{\Delta_i}\right)\epsilon_{\Delta_i} \tag{6.11}$$

where ϵ_{Δ_i} is the local estimation errors for Δ_i, defined as

$$\epsilon_{\Delta_i} = \sum_{j=1}^{M} a_{ij}(t)\left(\hat{\Delta}_i - \hat{\Delta}_j - \Delta_{ij}\right) + b_i(t)\left(\hat{\Delta}_i - \Delta_i\right) \tag{6.12}$$

with $\Delta_{ij} = \Delta_i - \Delta_j$ being the desired relative displacement between follower i and j.

In (6.11), c_{Δ_i} satisfies $c_{\Delta_i} \geqslant 1$ and follows the following adaptive law:

$$\dot{c}_{\Delta_i} = \epsilon_{\Delta_i}^{\top} \epsilon_{\Delta_i} \tag{6.13}$$

6.3.2 MPC-based Formation Tracking Controller

With the local estimation of the leader's state $\hat{\xi}_i$, the leader's dynamics matrix $\hat{S}_i$, and the desired formation displacement vector $\hat{\Delta}_i$, we can move on to the development of the formation tracking controller.

By applying model predictive control, a finite-horizon constrained optimization problem for a forward-looking prediction horizon $T \in \mathbb{R}^+$ is solved at each control update instant. The optimization solution is then implemented to the plant system recedingly, under a sampled-and-hold manner for execution. The control update instant sequence is $\{t_k = \delta k | k \in \{0, 1, 2, \cdots\}\}$, where $\delta < T \in \mathbb{R}^+$ represents the sampling period.

At the time instant t_k, the MPC optimization problem is formulated as

$$\min_{u_i^p(t|t_k)} \int_{t_k}^{t_k+T} \left(\|s_i^p(t|t_k)\|_{Q_i}^2 + \|u_i^p(t|t_k)\|_{R_i}^2 \right) \mathrm{d}t \tag{6.14a}$$

with

$$s_i^p(t|t_k) = \sum_{l=0}^{r-2} \lambda_{i,l} \left(x_{i,l+1}^p(t|t_k) - \xi_{0,l+1}^p(t|t_k) - \hat{\Delta}_{i,l+1}(t_k) \right) + \left(x_{i,r}^p(t|t_k) - \xi_{0,r}^p(t|t_k) - \hat{\Delta}_{i,r}(t_k) \right) \tag{6.14b}$$

https://www.dropbox.com/sh/uh4k7nn7zkpw9vo/AACDnq1gCyDnWqLKYaGkIMBHa?dl=0

subject to

$$\begin{cases} \dot{x}^p_{i,1}(t|t_k) = x^p_{i,2}(t|t_k) \\ \qquad\qquad \vdots \\ \dot{x}^p_{i,r-1}(t|t_k) = x^p_{i,r}(t|t_k) \\ \dot{x}^p_{i,r}(t|t_k) = f_i(x^p_i(t|t_k)) + G_i(x^p_i(t|t_k))u^p_i(t|t_k) \end{cases} \tag{6.14c}$$

$$\dot{\xi}^p_0(t|t_k) = \hat{S}_i(t_k)\xi^p_0(t|t_k) \tag{6.14d}$$

$$x^p_i(t_k|t_k) = x_i(t_k) \tag{6.14e}$$

$$\xi^p_0(t_k|t_k) = \xi_i(t_k) \tag{6.14f}$$

$$u^p_i(t|t_k) \in \Omega_u \tag{6.14g}$$

$$s^{p\top}_i(t_k|t_k)\Big(\sum_{l=0}^{r-2}\lambda_{i,l}\left(x^p_{i,l+2}(t_k|t_k) - \xi^p_{0,l+2}(t_k|t_k) - \hat{\Delta}_{i,l+1}(t_k)\right) + f_i(x^p_i(t_k|t_k)) \\ + G_i(x^p_i(t_k|t_k))u^p_i(t_k|t_k) - \dot{\xi}^p_{0,r}(t_k|t_k)\Big) \leqslant -c_i\|s^p_i(t_k|t_k)\|^2 \tag{6.14h}$$

where $t \in [t_k, t_k + T]$, and the internal variables are denoted by a superscript p to distinguish them from the actual system signals. In the optimization problem (6.14), constraint (6.14c) serves as the prediction model to predict the future evolution of the follower itself, while (6.14d) is to predict the leader's behavior by making use of $\hat{S}_i$ estimated by the observer. Constraints (6.14e) and (6.14f) specify the initial conditions of the prediction models (6.14c) and (6.14d) , respectively. Compliance with the input constraint is ensured by (6.14g). The Lyapunov-based stability constraint (6.14h) is designed to enforce the decay of the Lyapunov function at the current instant t_k. The rationale behind such stability constraint design was thoroughly introduced in Chapter 3. Hence, a detailed discussion is not reiterated here.

Lemma 6.1. *There always exists a feasible solution to the optimization problem*

(3.17), constructed as

$$u_i^0(t|t_k) = \text{sat}\left(v_i(t|t_k), \underline{u}_i, \overline{u}_i\right) \tag{6.15a}$$

$$v_i(t|t_k) = G_i^{-1}(x_i^p(t|t_k))\left(v_i^a(t|t_k) + v_i^d(t|t_k)\right) \tag{6.15b}$$

$$\begin{aligned} v_i^a(t|t_k) = & -c_i s_i^p(t|t_k) - \sum_{l=0}^{r-2} \lambda_{i,l}\Big(x_{i,l+2}^p(t|t_k) - \xi_{0,l+2}^p(t|t_k) - \hat{\Delta}_{i,l+1}(t_k)\Big) - f_i(x_i^p(t|t_k)) \\ & + \dot{\xi}_{0,r}^p(t|t_k) \end{aligned} \tag{6.15c}$$

$$v_i^d(t|t_k) = -k_{s_i}\underline{\chi}_i^{-1}\text{sgn}\left(s_i^p(t|t_k)\right) \tag{6.15d}$$

for $t \in [t_k, t_k + T]$. In (6.15a), $\text{sat}\left(v_i(t), \underline{u}_i, \overline{u}_i\right)$ is a saturation function, equaling to $\chi_i v_i$ with χ_i defined in Lemma 3.1. Here, $\underline{\chi}_i$ is the lower bound of χ_i. Additionally, k_{s_i} is a positive constant gain, chosen such that $k_{s_i} \geqslant \|(\chi_i - I_n)\, v_i^a\|_1$.

Proof. The proof of Lemma 6.1 is similar to the proof of Lemma 3.1 and is therefore omitted here for brevity. □

Given the feasibility of the optimization problem (6.14), an optimal control profile $u_i^*(t|t_k)$ for $t \in [t_k, t_k + T]$ can always be found by solving it at t_k. The found optimal solution is then implemented in a receding horizon manner. In this regard, $u_i^*(t|t_k)$ is applied to the ith follower until the next measurement is available, so the actual control command $u_i(t)$ for $t \in [t_k, t_{k+1})$ is

$$u_i(t) = u_i^*(t_k|t_k) \tag{6.16}$$

When the new measurement is updated at t_{k+1}, the optimization problem (6.14) will be solved again with t_k replaced by t_{k+1}, and a new optimal control profile $u_i^*(t|t_{k+1})$ for $t \in [t_{k+1}, t_{k+1} + T]$ will be found. In turn, the newly found optimal control profile updates the actual control command $u_i(t)$ for $t \in [t_{k+1}, t_{k+2})$.

Finally, we can have that with the initial conditions (6.14e and 6.14f) specified and the stability condition (6.14h) satisfied, the following inequality holds for $t \in [t_k, t_{k+1})$

$$\begin{aligned} & s_i^\top(t_k)\left(\sum_{l=0}^{r-2} \lambda_l \left(x_{i,l+2}(t_k) - \hat{\xi}_{i,l+2}(t_k)\right) + f_i(x_i(t_k)) + G_i(x(t_k))u_i(t) - \dot{\hat{\xi}}_{i,r}(t_k)\right) \\ & \leqslant -c_i \left\|s_i(t_k)\right\|^2 \end{aligned} \tag{6.17}$$

6.4 Closed-loop Stability Analysis

Given that the local control system includes two decoupled components—an adaptive observer and an MPC-based controller—we can perform the closed-loop stability analysis in a two-step manner. This approach allows us to access the stability contributions of the observer and the controller separately. In this section, we first examine the closed-loop performance of the adaptive observer network to prove that global estimation errors converge asymptotically. Subsequently, we evaluate the MPC-based controller, verifying its ability to maintain system stability and formation tracking control performance based on the observer's estimations.

6.4.1 Convergence of Estimation

First of all, let us focus on the estimation performance of the observers. We start by denoting collective vectors of $\hat{\xi}_i$, $\hat{S}_i$ and $\hat{\Delta}_i$ for $i = 1, 2, \cdots, M$ as $\hat{\xi} = \begin{bmatrix}\hat{\xi}_1^\top & \hat{\xi}_2^\top & \cdots & \hat{\xi}_M^\top\end{bmatrix}^\top$, $\hat{S} = \text{diag}(\hat{S}_1, \hat{S}_2, \cdots, \hat{S}_M)$ and $\hat{\Delta} = \begin{bmatrix}\hat{\Delta}_1^\top & \hat{\Delta}_2^\top & \cdots & \hat{\Delta}_M^\top\end{bmatrix}^\top$. Then, we can define the following collective estimation errors:

$$\tilde{\xi} = \begin{bmatrix}\tilde{\xi}_1^\top & \tilde{\xi}_2^\top & \cdots & \tilde{\xi}_M^\top\end{bmatrix}^\top = \hat{\xi} - \xi \tag{6.18}$$

$$\tilde{S} = \text{diag}(\tilde{S}_1, \tilde{S}_2, \cdots, \tilde{S}_M) = \hat{S} - (I_M \otimes S_0) \tag{6.19}$$

$$\tilde{\Delta} = \begin{bmatrix}\tilde{\Delta}_1^\top & \tilde{\Delta}_2^\top & \cdots & \tilde{\Delta}_M^\top\end{bmatrix}^\top = \hat{\Delta} - \Delta \tag{6.20}$$

We can also define a vector form of the leader matrix estimation error as

$$\tilde{\vec{S}} = \begin{bmatrix}\tilde{\vec{S}}_1^\top & \tilde{\vec{S}}_2^\top & \cdots & \tilde{\vec{S}}_M^\top\end{bmatrix}^\top = \begin{bmatrix}\text{vec}(\hat{S}_1)^\top & \text{vec}(\hat{S}_2)^\top & \cdots & \text{vec}(\hat{S}_M)^\top\end{bmatrix}^\top - (1_M \otimes \text{vec}(S_0)) \tag{6.21}$$

By recalling the adaptive distributed observer design in (6.7), (6.8), and (6.11), we can have the dynamics of these global estimation errors as follows

$$\dot{\tilde{\xi}} = \left(I_M \otimes S_0 + \tilde{S}\right)\tilde{\xi} - (C_x \otimes I_{rn})\epsilon_\xi + \tilde{S}\xi \tag{6.22}$$

$$\dot{\tilde{S}} = -\left(\left(C_S + \dot{C}_S\right) \otimes I_{rn}\right)\epsilon_S \tag{6.23}$$

$$\dot{\tilde{\vec{S}}} = -\left(\left(C_S + \dot{C}_S\right) \otimes I_{(rn)^2}\right)\vec{\epsilon}_S \tag{6.24}$$

$$\dot{\tilde{\Delta}} = -\left(\left(C_\Delta + \dot{C}_\Delta\right) \otimes I_{rn}\right)\epsilon_\Delta \tag{6.25}$$

where $C_x = \text{diag}(c_{\xi_1}, c_{\xi_2}, \cdots, c_{\xi_M})$, $C_S = \text{diag}(c_{S_1}, c_{S_2}, \cdots, c_{S_M})$, $C_\Delta = \text{diag}(c_{\Delta_1}, c_{\Delta_2}, \cdots, c_{\Delta_M})$, $\epsilon_\xi = \begin{bmatrix} \epsilon_{\xi_1}^\top & \epsilon_{\xi_2}^\top & \cdots & \epsilon_{\xi_M}^\top \end{bmatrix}^\top$, $\epsilon_S = \text{diag}(\epsilon_{S_1}, \epsilon_{S_2}, \cdots, \epsilon_{S_M})$, $\vec{\epsilon}_S = \begin{bmatrix} \vec{\epsilon}_{S_1}^\top & \vec{\epsilon}_{S_2}^\top & \cdots & \vec{\epsilon}_{S_M}^\top \end{bmatrix}^\top$, $\epsilon_\Delta = \begin{bmatrix} \epsilon_{\Delta_1}^\top & \epsilon_{\Delta_2}^\top & \cdots & \epsilon_{\Delta_M}^\top \end{bmatrix}^\top$.

We introduce a new notation, $\mathcal{L}_\mathcal{B}^f(t) = \mathcal{L}^f(t) + \mathcal{B}^f(t)$, to represent the superposition of the Laplacian matrix and the pinning matrix. This notation allows us to elucidate the relationships between global estimation errors and collective local estimation errors in the presence of communication link faults, as demonstrated below:

$$\epsilon_\xi = \left(\mathcal{L}_\mathcal{B}^f(t) \otimes I_{rn}\right) \tilde{\xi} \tag{6.26}$$

$$\epsilon_S = \left(\mathcal{L}_\mathcal{B}^f(t) \otimes I_{rn}\right) \tilde{S} \tag{6.27}$$

$$\vec{\epsilon}_S = \left(\mathcal{L}_\mathcal{B}^f(t) \otimes I_{(rn)^2}\right) \tilde{\vec{S}} \tag{6.28}$$

$$\epsilon_\Delta = \left(\mathcal{L}_\mathcal{B}^f(t) \otimes I_{rn}\right) \tilde{\Delta} \tag{6.29}$$

Subsequently, we can derive the dynamics of the local estimation errors, which are outlined below

$$\begin{aligned} \dot{\epsilon}_\xi &= \left(\mathcal{L}_\mathcal{B}^f(t) \otimes I_{rn}\right) \dot{\tilde{\xi}} + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{rn}\right) \tilde{\xi} \\ &= \left(I_M \otimes S_0 + \tilde{S} - \mathcal{L}_\mathcal{B}(t) C_\xi \otimes I_{rn}\right) \epsilon_\xi + \epsilon_S \xi + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{rn}\right) \tilde{\xi} \end{aligned} \tag{6.30}$$

$$\begin{aligned} \dot{\epsilon}_S &= \left(\mathcal{L}_\mathcal{B}(t) \otimes I_{rn}\right) \dot{\tilde{S}} + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{rn}\right) \tilde{S} \\ &= -\left(\mathcal{L}_\mathcal{B}^f(t) \left(C_S + \dot{C}_S\right) \otimes I_{rn}\right) \epsilon_S + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{rn}\right) \tilde{S} \end{aligned} \tag{6.31}$$

$$\dot{\vec{\epsilon}}_S = -\left(\mathcal{L}_\mathcal{B}^f(t) \left(C_S + \dot{C}_S\right) \otimes I_{(rn)^2}\right) \vec{\epsilon}_S + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{(rn)^2}\right) \tilde{\vec{S}} \tag{6.32}$$

$$\begin{aligned} \dot{\epsilon}_\Delta &= \left(\mathcal{L}_\mathcal{B}^f(t) \otimes I_{rn}\right) \dot{\tilde{\Delta}} + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{rn}\right) \tilde{\Delta} \\ &= -\left(\mathcal{L}_\mathcal{B}^f(t) \left(C_\Delta + \dot{C}_\Delta\right) \otimes I_{rn}\right) \epsilon_\Delta + \left(\dot{\mathcal{L}}_\mathcal{B}(t) \otimes I_{rn}\right) \tilde{\Delta} \end{aligned} \tag{6.33}$$

Having derived the dynamics of the error, we can then move on to formulate the first theorem regarding the convergence of distributed adaptive estimation. Before proceeding, however, it is essential to establish several foundational lemmas. These lemmas are building blocks for the proof of the main theorem.

Lemma 6.2. *Under Assumptions 6.2 and 6.4, the following relationships hold*

$$\left\|\tilde{\xi}\right\| \leqslant \frac{\|\epsilon_\xi\|}{\lambda_{\min}(\mathcal{L}_{\mathcal{B}}^f(t))} \tag{6.34}$$

$$\left\|\tilde{\vec{S}}\right\| \leqslant \frac{\|\vec{\epsilon}_S\|}{\lambda_{\min}(\mathcal{L}_{\mathcal{B}}^f(t))} \tag{6.35}$$

$$\left\|\tilde{\Delta}\right\| \leqslant \frac{\|\epsilon_\Delta\|}{\lambda_{\min}(\mathcal{L}_{\mathcal{B}}^f(t))} \tag{6.36}$$

Proof. Referring back to Lemma 2.5 in Chapter 2, it can be obtained that the matrix $\mathcal{L}_{\mathcal{B}}^f(t)$ is nonsingular and positive-definite. This guarantees that $\left(\mathcal{L}_{\mathcal{B}}^f(t)\right)^{-1}$ exists, and is nonnegative. Then, it follows from (6.26) - (6.29) that

$$\tilde{\xi} = \left(\mathcal{L}_{\mathcal{B}}^{f^{-1}}(t) \otimes I_{rn}\right) \epsilon_\xi \tag{6.37}$$

$$\tilde{\vec{S}} = \left(\mathcal{L}_{\mathcal{B}}^{f^{-1}}(t) \otimes I_{(rn)^2}\right) \vec{\epsilon}_S \tag{6.38}$$

$$\tilde{\Delta} = \left(\mathcal{L}_{\mathcal{B}}^{f^{-1}}(t) \otimes I_{rn}\right) \epsilon_\Delta \tag{6.39}$$

we can then have that

$$\left\|\tilde{\xi}\right\| \leqslant \left\|\left(\mathcal{L}_{\mathcal{B}}^{f^{-1}}(t) \otimes I_{rn}\right) \epsilon_\xi\right\| \leqslant \frac{\|\epsilon_\xi\|}{\lambda_{\min}(\mathcal{L}_{\mathcal{B}}^f(t))} \tag{6.40}$$

$$\left\|\tilde{\vec{S}}\right\| \leqslant \left\|\left(\mathcal{L}_{\mathcal{B}}^{f^{-1}}(t) \otimes I_{(rn)^2}\right) \vec{\epsilon}_S\right\| \leqslant \frac{\|\vec{\epsilon}_S\|}{\lambda_{\min}(\mathcal{L}_{\mathcal{B}}^f(t))} \tag{6.41}$$

$$\left\|\tilde{\Delta}\right\| \leqslant \left\|\left(\mathcal{L}_{\mathcal{B}}^{f^{-1}}(t) \otimes I_{rn}\right) \epsilon_\Delta\right\| \leqslant \frac{\|\epsilon_\Delta\|}{\lambda_{\min}(\mathcal{L}_{\mathcal{B}}^f(t))} \tag{6.42}$$

which prove Lemma 6.2. □

Lemma 6.3. *Define a diagonal matrix* $P(t) = \operatorname{diag}(p_1(t), p_2(t), \cdots, p_M(t))$ *with* $[p_1(t)\ p_2(t)\ \cdots\ p_M(t)]^\top = \left(\mathcal{L}_{\mathcal{B}}^f(t)\right)^{-1} 1_M$. *If Assumptions 6.2 and 6.4 hold, we can have that* $P(t)$ *is positive-definite. Furthermore, the symmetric matrix* $Q(t)$ *defined by*

$$Q(t) = P(t)\mathcal{L}_{\mathcal{B}}^f(t) + \mathcal{L}_{\mathcal{B}}^{f^\top}(t)P(t) \tag{6.43}$$

is positive-definite as well. Additionally, both $Q(t)$ *and its time derivative are bounded.*

Proof. The proof of Lemma 6.3 can be found under Lemmas 1 and 2 in [136], and is therefore omitted here for brevity. □

Now, we present our first main result of the closed-loop analysis by the following theorem.

Theorem 6.1. *Suppose that Assumptions 6.2 and 6.4 hold. Consider the M-agent system with the virtual leader (6.2), interconnected via the weighted directed graph $\mathcal{G}$. Implement the leader dynamics (6.2), the distributed leader state observer (6.7), the leader dynamics observer (6.8) and the formation displacement observer (6.11) for $i = 1, 2, \cdots, M$. If the leader state observer gain c_{ξ_i} for $i = 1, 2, \cdots, M$ are selected such that the following condition is satisfied*

$$\lambda_{\min}(C_\xi) > 1 + \frac{\kappa^*}{\kappa_0} \tag{6.44}$$

where $\kappa^* = \frac{5\kappa_2}{4\kappa_0} + \frac{5\kappa_3}{\kappa_0} + \frac{5\kappa_4}{\kappa_0} + \frac{5\kappa_5}{\kappa_0}$ with $\kappa_0 = \min_{\forall t \geqslant 0} \kappa_{\min}(Q(t))$, $\kappa_1 = \max_{\forall t \geqslant 0} \lambda_{\max}(P^2(t))$, $\kappa_2 = \max_{\forall t \geqslant 0} \lambda_{\max}\left(\dot{P}^2(t)\right)$, $\kappa_3 = \max_{\forall t \geqslant 0} \lambda_{\max}\left(P^2(t)\mathcal{L}_{\mathcal{B}}^{f^{-\top}}(t)\dot{\mathcal{L}}_{\mathcal{B}}^{f^{\top}}(t)\dot{\mathcal{L}}_{\mathcal{B}}^{f}(t)\mathcal{L}_{\mathcal{B}}^{f}(t)\right)$, $\kappa_4 = \max_{\forall t \geqslant 0} \lambda_{\max}\left(P^2(t) \otimes S_0^{\top} S_0\right)$, and $\kappa_5 = \max_{\forall t \geqslant 0} \lambda_{\max}\left(\tilde{S}^{\top}\left(P^2(t) \otimes I_{rn}\right)\tilde{S}\right)$, then all signals within the observer network are globally bounded. Moreover, all the estimated errors, $\tilde{\xi}$, $\tilde{S}$, $\tilde{\vec{S}}$ and $\tilde{\Delta}$, asymptotically converge to the origin.

Proof. To prove the convergence of the observer network, we divide the proof into three parts. Firstly, in Part 1 and Part 2, the convergence of the displacement estimation error and the dynamics matrix estimation error are proven, respectively. Finally, in Part 3, we can prove that the leader state estimation error asymptotically converges to the origin.

Part 1: To demonstrate the convergence of the formation displacement estimation error, we begin by examining its corresponding local estimation error $\epsilon_{!\Delta}$. A Lyapunov function candidate can be selected as follows

$$\begin{aligned} V_\Delta =& \sum_{i=1}^{m} \left(2c_{\Delta_i} + \dot{c}_{\Delta_i}\right) p_i(t) \epsilon_{\Delta_i}^{\top} \epsilon_{\Delta_i} + \sum_{i=1}^{m} \left(c_{\Delta_i} - \alpha_\Delta\right)^2 \\ =& \epsilon_\Delta^{\top} \left(\left(2C_\Delta + \dot{C}_\Delta\right) P(t) \otimes I_{rn}\right) \epsilon_\Delta + \operatorname{tr}\left(\left(C_\Delta - \alpha_\Delta I_M\right)^2\right) \end{aligned} \tag{6.45}$$

where α_Δ is a positive constant to be determined later; $P(t)$ is defined in Lemma 6.3.

Taking the time derivative of V_Δ gives

$$\begin{aligned}\dot{V}_\Delta =&4\sum_{i=1}^{m}\left(c_{\Delta_i}+\dot{c}_{\Delta_i}\right)p_i(t)\epsilon_{\Delta_i}^\top\dot{\epsilon}_{\Delta_i}+2\sum_{i}^{m}\dot{c}_{\Delta_i}p_i(t)\epsilon_{\Delta_i}^\top\epsilon_{\Delta_i}+\sum_{i=1}^{m}\left(2c_{\Delta_i}+\dot{c}_{\Delta_i}\right)\dot{p}_i(t)\epsilon_{\Delta_i}^\top\epsilon_{\Delta_i}\\&+2\sum_{i=1}^{m}\left(c_{\Delta_i}-\alpha_\Delta\right)\dot{c}_{\Delta_i}\\=&4\epsilon_\Delta^\top\left(\left(C_\Delta+\dot{C}_\Delta\right)P(t)\otimes I_{rn}\right)\dot{\epsilon}_\Delta+2\epsilon_\Delta^\top\left(\dot{C}_\Delta P(t)\otimes I_{rn}\right)\epsilon_\Delta+\epsilon_\Delta^\top\left(\left(2C_\Delta+\dot{C}_\Delta\right)\dot{P}(t)\right.\\&\left.\otimes I_{rn}\right)\epsilon_\Delta+2\epsilon_\Delta^\top\left(C_\Delta\otimes I_{rn}\right)\epsilon_\Delta-2\alpha_\Delta\epsilon_\Delta^\top\epsilon_\Delta\end{aligned}\tag{6.46}$$

We have $\dot{c}_{\Delta_i}\geqslant 0$ and $c_{\Delta_i}(t)\geqslant 1$. Substituting (6.33) into (6.46) gives

$$\begin{aligned}\dot{V}_\Delta\leqslant&-2\kappa_0\epsilon_\Delta^\top\left(\left(C_\Delta+\dot{C}_\Delta\right)^2\otimes I_{rn}\right)\epsilon_\Delta+2\epsilon_\Delta^\top\left(\dot{C}_\Delta P(t)\otimes I_{rn}\right)\epsilon_\Delta+\epsilon_\Delta^\top\left(\left(2C_\Delta+\dot{C}_\Delta\right)\dot{P}(t)\right.\\&\left.\otimes I_{rn}\right)\epsilon_\Delta+2\epsilon_\Delta^\top\left(C_\Delta\otimes I_{rn}\right)\epsilon_\Delta+4\epsilon_\Delta^\top\left(\left(C_\Delta+\dot{C}_\Delta\right)P(t)\dot{\mathcal{L}}_{\mathcal{B}}^f(t)\otimes I_{rn}\right)\tilde{\Delta}-2\alpha_\Delta\epsilon_\Delta^\top\epsilon_\Delta\end{aligned}\tag{6.47}$$

where $\kappa_0=\min_{\forall t\geqslant 0}\kappa_{\min}(Q(t))$.

Applying Young's inequality, one has

$$2\epsilon_\Delta^\top\left(\dot{C}_\Delta P(t)\otimes I_{rn}\right)\epsilon_\Delta\leqslant\frac{\kappa_0}{2}\epsilon_\Delta^\top\left(\dot{C}_\Delta^2\otimes I_{rn}\right)\epsilon_\Delta+\frac{2}{\kappa_0}\epsilon_\Delta^\top\left(P^2(t)\otimes I_{rn}\right)\epsilon_\Delta\tag{6.48a}$$

$$\epsilon_\Delta^\top\left(\left(2C_\Delta+\dot{C}_\Delta\right)\dot{P}(t)\otimes I_{rn}\right)\epsilon_\Delta\leqslant\frac{\kappa_0}{4}\epsilon_\Delta^\top\left(\left(C_\Delta^2+\dot{C}_\Delta^2\right)\otimes I_{rn}\right)\epsilon_\Delta+\frac{5}{\kappa_0}\epsilon_\Delta^\top\left(\dot{P}^2(t)\otimes I_{rn}\right)\epsilon_\Delta\tag{6.48b}$$

$$2\epsilon_\Delta^\top\left(C_\Delta\otimes I_{rn}\right)\epsilon_\Delta\leqslant\frac{\kappa_0}{2}\epsilon_\Delta^\top\left(C_\Delta^2\otimes I_{rn}\right)\epsilon_\Delta+\frac{2}{\kappa_0}\epsilon_\Delta^\top\epsilon_\Delta\tag{6.48c}$$

$$\begin{aligned}4\epsilon_\Delta^\top\left(\left(C_\Delta+\dot{C}_\Delta\right)P(t)\dot{\mathcal{L}}_{\mathcal{B}}^f(t)\otimes I_{rn}\right)\tilde{\Delta}\leqslant&\frac{\kappa_0}{4}\epsilon_\Delta^\top\left(\left(C_\Delta+\dot{C}_\Delta\right)^2\otimes I_{rn}\right)\epsilon_\Delta\\&+\frac{16}{\kappa_0}\epsilon_\Delta^\top\left(P^2(t)\mathcal{L}_{\mathcal{B}}^{f^{-\top}}(t)\dot{\mathcal{L}}_{\mathcal{B}}^{f^\top}(t)\dot{\mathcal{L}}_{\mathcal{B}}^f(t)\mathcal{L}_{\mathcal{B}}^f(t)\otimes I_{rn}\right)\epsilon_\Delta\end{aligned}\tag{6.48d}$$

We define $\kappa_1=\max_{\forall t\geqslant 0}\lambda_{\max}\left(P^2(t)\right)$, $\kappa_2=\max_{\forall t\geqslant 0}\lambda_{\max}\left(\dot{P}^2(t)\right)$, and $\kappa_3=\max_{\forall t\geqslant 0}\lambda_{\max}\left(P^2(t)\mathcal{L}_{\mathcal{B}}^{f^{-\top}}(t)\dot{\mathcal{L}}_{\mathcal{B}}^{f^\top}(t)\dot{\mathcal{L}}_{\mathcal{B}}^f(t)\mathcal{L}_{\mathcal{B}}^f(t)\right)$. Then, substituting the above inequalities into $\dot{V}_\Delta$ yields

$$\dot{V}_\Delta\leqslant-\lambda_0\epsilon_\Delta^\top\left(\left(C_\Delta+\dot{C}_\Delta\right)^2\otimes I_{rn}\right)\epsilon_\Delta+\kappa\epsilon_\Delta^\top\epsilon_\Delta-2\alpha_\Delta\epsilon_\Delta^\top\epsilon_\Delta\tag{6.49}$$

where $\kappa=\frac{2\kappa_1}{\kappa_0}+\frac{5\kappa_2}{\kappa_0}+\frac{2}{\kappa_0}+\frac{16\kappa_3}{\kappa_0}$. There exists a bounded constant α_Δ satisfying $\alpha_\Delta\geqslant\frac{\kappa}{2}$

such that

$$\dot{V}_{\Delta} \leqslant -\epsilon_{\Delta}^{\top}\left(\left(C_{\Delta}+\dot{C}_{\Delta}\right)^{2}\otimes I_{rn}\right)\epsilon_{\Delta} \tag{6.50}$$

which implies that all signals in the developed observer including ϵ_{Δ}, C_{Δ} and $\dot{C}_{\Delta}$ are globally bounded under communication link faults.

To get the convergence of ϵ_{Δ}, we solve (6.50) as

$$V_{\Delta}(t)-V_{\Delta}(0) \leqslant -\lambda_{0}\int_{0}^{t}\epsilon_{\Delta}^{\top}\left(\left(C_{\Delta}+\dot{C}_{\Delta}\right)^{2}\otimes I_{rn}\right)\epsilon_{\Delta}\mathrm{d}\tau \tag{6.51}$$

With the fact that $\dot{c}_{\Delta_i} \geqslant 0$ and $c_{\Delta_i} \geqslant 1$, one has

$$\begin{aligned} 0 \leqslant V_{\Delta}(t) &\leqslant -\lambda_{0}\int_{0}^{t}\epsilon_{\Delta}^{\top}\left(C_{\Delta}\otimes I_{rn}\right)\epsilon_{\Delta}\mathrm{d}\tau + V_{\Delta}(0) \\ &\leqslant -\kappa_{0}\int_{0}^{t}\epsilon_{\Delta}^{\top}\epsilon_{\Delta}\mathrm{d}\tau + V_{\Delta}(0) \end{aligned} \tag{6.52}$$

or equivalently

$$0 \leqslant \int_{0}^{t}\epsilon_{\Delta}^{\top}\epsilon_{\Delta}\mathrm{d}\tau \leqslant \frac{V_{\Delta}(0)}{\kappa_{0}} \tag{6.53}$$

This demonstrates that $\int_0^t \epsilon_{\Delta}^{\top}\epsilon_{\Delta}\mathrm{d}\tau$ is bounded. By applying Barbalat's Lemma (Lemma 2.2), we establish that $\epsilon_{!\Delta}$ converges to zero. Furthermore, according to Lemma 6.2, it can be deduced that the estimation error $\tilde{\Delta}$ also globally converges to zero, even in scenarios involving communication link faults.

Part 2: Similarly, the leader state estimation error convergence can be proven. We firstly select a candidate Lyapunov function as

$$\begin{aligned} V_{S} =& \sum_{i=1}^{m}\left(2c_{S_i}+\dot{c}_{S_i}\right)p_{i}(t)\vec{\epsilon}_{S_i}^{\top}\vec{\epsilon}_{S_i} + \sum_{i=1}^{m}\left(c_{S_i}-\alpha_{S}\right)^{2} \\ =& \vec{\epsilon}_{S}^{\top}\left(\left(2C_{S}+\dot{C}_{S}\right)P(t)\otimes I_{(rn)^2}\right)\vec{\epsilon}_{S} + \mathrm{tr}\left(\left(C_{S}-\alpha_{S}I_{(rn)^2}\right)^{2}\right) \end{aligned} \tag{6.54}$$

Following the same procedure, the derivative of V_S can be bounded by

$$\dot{V}_{S} \leqslant -\lambda_{0}\vec{\epsilon}_{S}^{\top}\left(\left(C_{S}+\dot{C}_{S}\right)^{2}\otimes I_{(rn)^2}\right)\vec{\epsilon}_{S} \tag{6.55}$$

with α_S appropriately chosen such that $\alpha_S \geqslant \frac{\kappa}{2}$.

We can further have that

$$0 \leqslant \int_0^t \vec{\epsilon}_S^{\top} \vec{\epsilon}_S \mathrm{d}\tau \leqslant \frac{V_S(0)}{\kappa_0} \tag{6.56}$$

This implies that $\int_0^t \vec{\epsilon}_S^{\top} \vec{\epsilon}_S \mathrm{d}\tau$ is bounded. Applying Barbalat's Lemma, we can conclude that $\vec{\epsilon}_S$ converges to zero. Furthermore, based on Lemma 6.2, it can be concluded that the estimation errors ϵ_S, $\tilde{\vec{S}}$, and $\tilde{S}$ also converge to zero.

Part 3: Finally, to demonstrate the stability and convergence of the leader state estimation, we select the following candidate for a Lyapunov function:

$$V_\xi = \sum_{i=1}^{m} p_i(t) \epsilon_{\xi_i}^{\top} \epsilon_{\xi_i} = \epsilon_\xi^{\top} \left(P(t) \otimes I_{rn}\right) \epsilon_\xi \tag{6.57}$$

By taking the time derivative of V_ξ, we can have

$$\dot{V}_\xi = 2\epsilon_\xi^{\top} \left(P(t) \otimes I_{rn}\right) \dot{\epsilon}_\xi + \epsilon_\xi^{\top} \left(\dot{P}(t) \otimes I_{rn}\right) \epsilon_\xi \tag{6.58}$$

Substituting (6.30) into it gives

$$\begin{aligned} \dot{V}_\xi \leqslant & -\kappa_0 \lambda_{\min}(C_\xi) \epsilon_\xi^{\top} \epsilon_\xi + \epsilon_\xi^{\top} \left(\dot{P}(t) \otimes I_{rn}\right) \epsilon_\xi + 2\epsilon_\xi^{\top} \left(P(t) \dot{\mathcal{L}}_{\mathcal{B}}(t) \otimes I_{rn}\right) \tilde{\xi} + 2\epsilon_x^{\top} \left(P(t) \otimes S_0\right) \epsilon_\xi \\ & + 2\epsilon_\xi^{\top} \left(P(t) \otimes I_{rn}\right) \tilde{S} \epsilon_\xi + 2\epsilon_\xi^{\top} \left(P(t) \otimes I_{rn}\right) \epsilon_S \xi \end{aligned} \tag{6.59}$$

where $\kappa_0 = \min_{\forall t \geqslant 0} \lambda_{\min}(Q(t))$.

We define $\varsigma = \epsilon_S \xi$ as a new variable that converges to zero given the convergence of ϵ_S. Applying Young's inequality, one has

$$\epsilon_\xi^{\top} \left(\dot{P}(t) \otimes I_{rn}\right) \epsilon_\xi \leqslant \frac{\kappa_0}{5} \epsilon_\xi^{\top} \epsilon_\xi + \frac{5}{4\kappa_0} \epsilon_\xi^{\top} \left(\dot{P}^2(t) \otimes I_{rn}\right) \epsilon_\xi \tag{6.60a}$$

$$2\epsilon_\xi^{\top} \left(P(t) \dot{\mathcal{L}}_{\mathcal{B}}^{f}(t) \otimes I_{rn}\right) \tilde{\xi} \leqslant \frac{\kappa_0}{5} \epsilon_\xi^{\top} \epsilon_\xi + \frac{5}{\kappa_0} \epsilon_\xi^{\top} \left(P^2(t) \mathcal{L}_{\mathcal{B}}^{f^{-\top}}(t) \dot{\mathcal{L}}_{\mathcal{B}}^{f^{\top}}(t) \dot{\mathcal{L}}_{\mathcal{B}}^{f}(t) \mathcal{L}_{\mathcal{B}}^{f}(t) \otimes I_{rn}\right) \epsilon_\xi \tag{6.60b}$$

$$2\epsilon_\xi^{\top} \left(P(t) \otimes S_0\right) \epsilon_\xi \leqslant \frac{\kappa_0}{5} \epsilon_\xi^{\top} \epsilon_\xi + \frac{5}{\kappa_0} \epsilon_\xi^{\top} \left(P^2(t) \otimes S_0^{\top} S_0\right) \epsilon_\xi \tag{6.60c}$$

$$2\epsilon_\xi^{\top} \left(P(t) \otimes I_{rn}\right) \tilde{S} \epsilon_\xi \leqslant \frac{\kappa_0}{5} \epsilon_\xi^{\top} \epsilon_\xi + \frac{5}{\kappa_0} \epsilon_\xi^{\top} \tilde{S}^{\top} \left(P^2(t) \otimes I_{rn}\right) \tilde{S} \epsilon_\xi \tag{6.60d}$$

$$2\epsilon_\xi^{\top} \left(P(t) \otimes I_{rn}\right) \varsigma \leqslant \frac{\kappa_0}{5} \epsilon_\xi^{\top} \epsilon_\xi + \frac{5}{\kappa_0} \varsigma^{\top} \left(P^2(t) \otimes I_{rn}\right) \varsigma \tag{6.60e}$$

Given the boundedness of $\tilde{S}$, we can define new constants $\kappa_4 = \max_{\forall t \geqslant 0} \lambda_{\max} \left(P^2(t) \otimes S_0^{\top} S_0\right)$ and $\kappa_5 = \max_{\forall t \geqslant 0} \lambda_{\max} \left(\tilde{S}^{\top} \left(P^2(t) \otimes I_{rn}\right) \tilde{S}\right)$, and recall the definitions of

$\kappa_1 = \max_{\forall t \geqslant 0} \lambda_{\max}\left(P^2(t)\right)$, $\kappa_2 = \max_{\forall t \geqslant 0} \lambda_{\max}\left(\dot{P}^2(t)\right)$, $\kappa_3 = \max_{\forall t \geqslant 0} \lambda_{\max}\Big(P^2(t)\mathcal{L}_{\mathcal{B}}^{f^{-\top}}(t)$ $\dot{\mathcal{L}}_{\mathcal{B}}^{f^{\top}}(t)\dot{\mathcal{L}}_{\mathcal{B}}^{f}(t)\mathcal{L}_{\mathcal{B}}^{f}(t)\Big)$. Substituting the above inequalities into $\dot{V}_\xi$ yields

$$\dot{V}_\xi \leqslant -\left(\kappa_0 \lambda_{\min}(C_\xi) - \kappa_0 - \kappa^*\right)\epsilon_\xi^\top \epsilon_\xi + \frac{5\kappa_1}{\kappa_0}\varsigma^\top \varsigma \tag{6.61}$$

where $\kappa^* = \frac{5\kappa_2}{4\kappa_0} + \frac{5\kappa_3}{\kappa_0} + \frac{5\kappa_4}{\kappa_0} + \frac{5\kappa_5}{\kappa_0}$.

Therefore, when the user-designated gains c_{ξ_i} for $i = 1, 2, \cdots, M$ are appropriately chosen to satisfy the stability condition (6.61), a positive constant α_ξ must exist such that $0 < \alpha_\xi \leqslant \kappa_0 \lambda_{\min}(C_\xi) - \kappa_0 - \kappa^*$. As a result, one has

$$\dot{V}_\xi \leqslant -\alpha_\xi \epsilon_\xi^\top \epsilon_\xi + \frac{5\kappa_1}{\kappa_0}\varsigma^\top \varsigma \tag{6.62}$$

which means that the dynamics of ϵ_ξ is robust input-to-state stable with ς as a disturbances input. Integrating the above inequalities over $[0, t]$ gives

$$V_\xi(t) - V_\xi(0) \leqslant -\alpha_\xi \int_0^t \epsilon_\xi^\top \epsilon_\xi \mathrm{d}\tau + \frac{5\kappa_1}{\kappa_0}\int_0^t \varsigma^\top \varsigma \mathrm{d}\tau \tag{6.63}$$

which further yields

$$\alpha_\xi \int_0^t \epsilon_\xi^\top \epsilon_\xi \mathrm{d}\tau \leqslant \frac{5\kappa_1}{\kappa_0}\int_0^t \varsigma^\top \varsigma \mathrm{d}\tau \tag{6.64}$$

From Part 2, we have proven the convergence of ϵ_S, which also implies the convergence of ς to zero. Hence, it can be concluded that $\int_0^t \varsigma^\top \varsigma \mathrm{d}\tau$ is bounded. From the above inequality, we can prove the boundedness of $\int_0^t \epsilon_\xi^\top \epsilon_\xi \mathrm{d}\tau$. By using Barbalat's Lemma, ϵ_ξ can be proven to converge to zero as well. From Lemma 6.2, it can be obtained that $\tilde{\xi}$ also asymptotically converges to the origin. □

6.4.2 Stability of Control

Following the establishment of the asymptotic convergence of the estimation errors as demonstrated in Theorem 6.1, we can now present our second main result of this chapter, which summarizes the convergence of the system's actual state to the locally estimated leader state under the proposed Lyapunov-based MPC framework.

We start by defining a sliding mode tracking control error for the follower i as

$$s_i = \sum_{l=0}^{r-2} \lambda_{i,l} \left(x_{i,l+1} - \hat{\xi}_{i,l+1} - \hat{\Delta}_{i,l+1} \right) + \left(x_{i,r} - \hat{\xi}_{i,r} - \hat{\Delta}_{i,r} \right) \tag{6.65}$$

Then, the following Lyapunov function candidate for follower i can be considered

$$V_i = \frac{1}{2} \|s_i\|^2 \tag{6.66}$$

Differentiating V along the closed-loop dynamics yields

$$\dot{V}_i = s_i^\top \left(\sum_{l=0}^{r-2} \lambda_l \left(x_{i,l+2} - \hat{\xi}_{i,l+2} - \hat{\Delta}_{i,l+2} \right) + f_i(x_i) + G_i(x_i) u_i - \dot{\hat{\xi}}_{i,r} \right) \tag{6.67}$$

Recalling the inequality (6.17), adding and subtracting the right-hand side of it to and from (6.67), we can rewrite $\dot{V}_i(t)$ over the interval $t \in [t_k, t_{k+1})$ as follows:

$$\begin{aligned} \dot{V}_i(t) \leqslant & - c_i \|s_i(t)\|^2 + c_i \left(\|s_i(t)\|^2 - \|s_i(t_k)\|^2 \right) + s_i^\top(t) \Bigg(\sum_{l=0}^{r-2} \lambda_l \Big(x_{i,l+2}(t) - \hat{\xi}_{i,l+2}(t) \\ & - \hat{\Delta}_{i,l+2}(t) \Big) + f_i(x_i(t)) + G_i(x_i(t)) u_i(t) - \dot{\hat{\xi}}_{i,r}(t) \Bigg) - s_i^\top(t_k) \Bigg(\sum_{l=0}^{r-2} \lambda_l \left(x_{i,l+2}(t_k) \right. \\ & \left. - \hat{\xi}_{i,l+2}(t_k) - \hat{\Delta}_{i,l+2}(t_k) \right) + f_i(x_i(t_k)) + G_i(x(t_k)) u_i(t) - \dot{\hat{\xi}}_{i,r}(t_k) \Bigg) \end{aligned} \tag{6.68}$$

By invoking Lipschitz continuity and under Assumption 6.1, there must exist positive Lipschitz constants $L_{s_i^2}$, L_{x_i} and L_{ξ_i} such that

$$\left| \|s_i(t)\|^2 - \|s_i(t_k)\|^2 \right| \leqslant L_{s_i^2} \|x_i(t) - x_i(t_k)\| \tag{6.69a}$$

$$\begin{aligned} & \Bigg| s_i^\top(t) \Bigg(\sum_{l=0}^{r-2} \lambda_l \left(x_{i,l+2}(t) - \hat{\xi}_{i,l+2}(t) - \hat{\Delta}_{i,l+2}(t) \right) + f_i(x_i(t)) + G_i(x_i(t)) u_i(t) - \dot{\hat{\xi}}_{i,r}(t) \Bigg) \\ & - s_i^\top(t_k) \Bigg(\sum_{l=0}^{r-2} \lambda_l \Big(x_{i,l+2}(t_k) - \hat{\xi}_{i,l+2}(t_k) - \hat{\Delta}_{i,l+2}(t_k) \Big) + f_i(x_i(t_k)) + G_i(x(t_k)) u_i(t) - \dot{\hat{\xi}}_{i,r}(t_k) \Bigg) \Bigg| \\ & \leqslant L_{x_i} \|x_i(t) - x_i(t_k)\| + L_{\xi_i} \left\| \hat{\xi}_i(t) - \hat{\xi}_i(t_k) \right\| \end{aligned} \tag{6.69b}$$

We also have positive constants M_{x_i} and M_{ξ_i} satisfying

$$\|x_i(t) - x_i(t_k)\| \leqslant M_{x_i}\delta \tag{6.69c}$$

$$\left\|\hat{\xi}_i(t) - \hat{\xi}_i(t_k)\right\| \leqslant M_{\xi_i}\delta \tag{6.69d}$$

By substituting (6.69) into (6.68), we have that

$$\dot{V}_i(t) \leqslant -c_i\|s_i(t)\|^2 + (\kappa_{i_1} + c_i\kappa_{i_2})\delta \tag{6.70}$$

where $\kappa_{i1} = L_{x_i}M_{x_i} + L_{\xi_i}M_{\xi_i}$ and $\kappa_{i2} = L_{s_i^2}M_{x_i}$.

We define a collective sliding mode tracking control error as $s = \begin{bmatrix} s_1^\top & s_2^\top & \cdots & s_M^\top \end{bmatrix}^\top$, and Theorem 6.2 below encapsulates the second main result of this chapter

Theorem 6.2. *Suppose Assumptions 6.1-6.4 hold. Consider the MASs with M followers (6.1) and a virtual leader 6.2 in closed loop under the developed adaptive distributed observer-based Lyapunov-based MPC framework with the leader observer (6.7)-(6.11) and the MPC problem (6.14). If* $s(0) \in \Omega_{\rho_s^0} \triangleq \{s|V \leqslant \rho_s^0\}$ *and the following stability condition is satisfied by choosing appropriate control gains* c_i *for* $i = 1, 2, \cdots, M$,

$$-2\rho_s\lambda_{\min}(C) + (\kappa_1 + \lambda_{\max}(C)\kappa_2)\,\delta \leqslant 0 \tag{6.71}$$

where $\rho_s \leqslant \rho_s^0$, $C = \mathrm{diag}(c_1, c_2, \cdots, c_M)$, $\kappa_1 = \max(\kappa_{1_1}, \kappa_{2_1}, \cdots, \kappa_{M_1})$ *and* $\kappa_2 = \max(\kappa_{1_2}, \kappa_{2_2}, \cdots, \kappa_{M_2})$, *then, the sliding mode error* s *of the closed-loop system is always bounded and ultimately converges to* $\Omega_{\rho_s} \triangleq \{s|V \leqslant \rho_s\}$.

Proof. A global Lyapunov function for the entire MAS can be selected as

$$V = \frac{1}{2}\|s\|^2 = \sum_{i=1}^{M}\frac{1}{2}\|s_i\|^2 \tag{6.72}$$

Recalling (6.70), we can obtain that

$$\dot{V} \leqslant -\lambda_{\min}(C)\|s(t)\|^2 + (\kappa_1 + \lambda_{\max}(C)\kappa_2)\delta \tag{6.73}$$

where $C = \mathrm{diag}(c_1, c_2, \cdots, c_M)$, $\kappa_1 = \max(\kappa_{1_1}, \kappa_{2_1}, \cdots, \kappa_{M_1})$ and $\kappa_2 = \max(\kappa_{1_2}, \kappa_{2_2}, \cdots, \kappa_{M_2})$.

From the definition of V, we further have

$$\dot{V} \leqslant -\lambda_{\min}(C)V + (\kappa_1 + \lambda_{\max}(C)\kappa_2)\delta \tag{6.74}$$

If the condition (6.71) is satisfied, then it can be derived that $\dot{V} < 0$ for all $s \in \{s|\rho_s^0 < V \leqslant \rho_s\}$ and $\dot{V} = 0$ for $V = \rho_s$. Therefore, it implies that s converges to Ω_{ρ_s} without leaving the stability region $\Omega_{\rho_s^0}$ as t approaches ∞ □

Based on Theorem 6.2 and reflecting on Proposition 2.2, we can conclude that the state variable x converges towards $\hat{\xi} + \hat{\Delta}$. Given the previously validated asymptotic convergence of $\hat{\xi}$ to ξ and $\hat{\Delta}$ to Δ in Theorem 6.1, we can conclusively demonstrate the ultimate boundedness and convergence of the global formation tracking error $\tilde{x}$, as defined in (6.5). This conclusion is drawn by combining the results from both Theorem 6.1 and Theorem 6.2.

Remark 6.5. *It is crucial to recognize that the analytical convergence error arises from the sampled-and-hold implementation of the employed MPC. The asymptotically stable nature of the distributed observer network does not compromise the ultimate accuracy of formation tracking by providing sufficiently accurate estimations to the controller.*

6.5 Simulation Study

Simulation studies on two different examples are carried out to evaluate the performance of the proposed method in achieving formation tracking control under input constraints and communication link faults. This section provides the simulation results and

6.5.1 Example 1: A Numerical Multi-agent System

We first consider a numerical example—a nonlinear MAS with 3 followers and a leader node 0. The 3 followers can be described by third-order nonlinear systems as

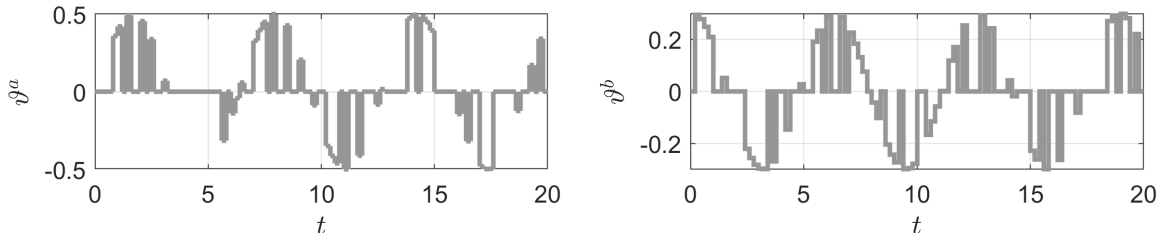

Figure 6.2: Communication link faults

follows

$$\begin{cases} \dot{x}_{1,1} &= x_{1,2} \\ \dot{x}_{1,2} &= x_{1,3} \\ \dot{x}_{1,3} &= x_{1,1}x_{1,2} + x_{1,3} - x_{1,1}^3 + u_1 \\ y_1 &= x_{1,1} \end{cases} \tag{6.75a}$$

$$\begin{cases} \dot{x}_{2,1} &= x_{2,2} \\ \dot{x}_{2,2} &= x_{2,3} \\ \dot{x}_{2,3} &= x_{2,1}\sin(x_{2,2}) + \cos^2(x_{2,3}) + u_2 \\ y_2 &= x_{2,1} \end{cases} \tag{6.75b}$$

$$\begin{cases} \dot{x}_{3,1} &= x_{2,2} \\ \dot{x}_{3,2} &= x_{2,3} \\ \dot{x}_{3,3} &= -\frac{1}{2}(x_{3,1} + x_{3,2} - 1)^2(x_{3,3} - 1) + u_3 \\ y_3 &= x_{3,1} \end{cases} \tag{6.75c}$$

with the input constraint defined as

$$\Omega_{u_i} = \{u_i | -3 \leqslant u_i \leqslant 3\} \tag{6.76}$$

for $i = 1, 2, 3$. The initial conditions of the 3 followers are $x_1(0) = [1.3\ 0\ 0]^\top$, $x_2(0) = [0.5\ 0\ 0]^\top$, $x_3(0) = [0\ 0\ 0]^\top$.

Let the dynamics of the leader node be

$$\begin{cases} \dot{\xi}_{0,1} &= \xi_{0,2} \\ \dot{\xi}_{0,2} &= \xi_{0,3} \\ \dot{\xi}_{0,3} &= -\xi_{0,1} - 1.16\xi_{0,2} - 2\xi_{0,3} \end{cases} \tag{6.77}$$

with $\xi_0(0) = [1\ 0\ 0]^\top$.

In simulations, the desired formation displacement vectors of the 3 followers with

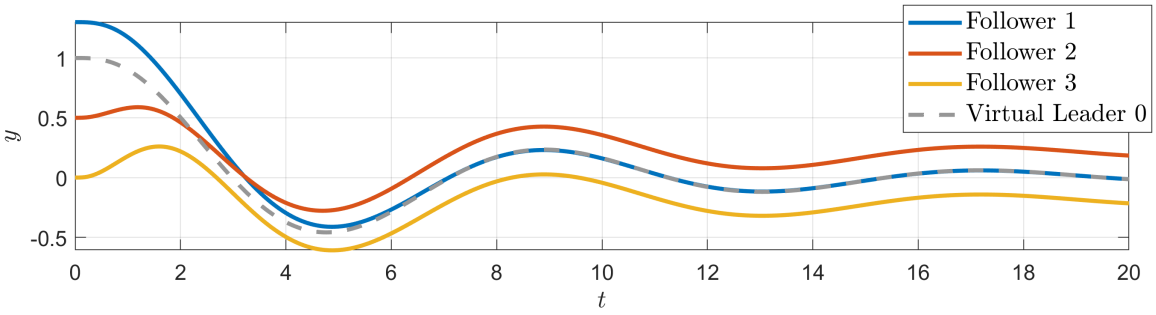

Figure 6.3: Formation tracking performance

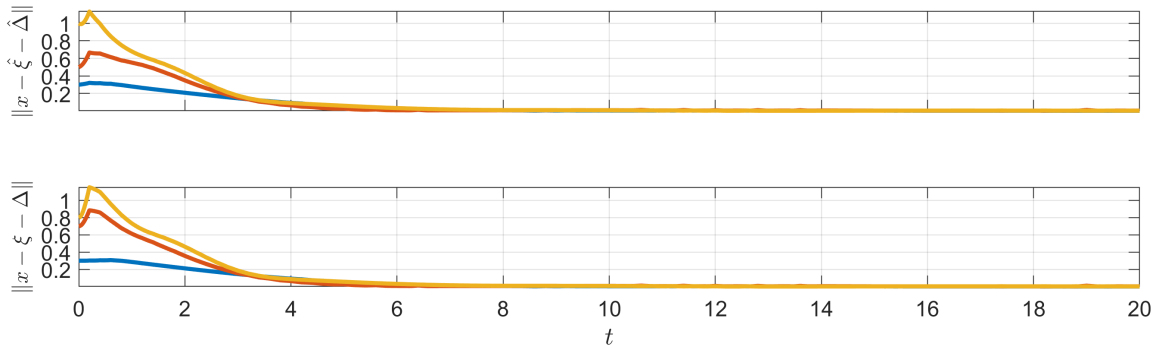

Figure 6.4: Norms of formation tracking errors

respect to the leader 0 are set as $\Delta_{10} = [0\ 0\ 0]^\top$, $\Delta_{20} = [0.2\ 0\ 0]^\top$, $\Delta_{30} = [-0.2\ 0\ 0]^\top$, $\Delta_{40} = [2\ 0\ 0]^\top$. Time-varying edge weights, including the adjacency matrix and pinning gains, are designed to mimic faults in the communication network. In particular, the adjacency matrix and the pinning matrix are

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 + 0.5\sin(t) * \text{rand}([0,1]) & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{6.78}$$

$$\mathcal{B} = \begin{bmatrix} 1 + 0.3\sin(t) * \text{rand}([0,1]) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.79}$$

where rand$([0,1])$ is a random signal chosen from the interval $[0,1]$.

The control parameters are selected following the obtained stability conditions. The sampling period for updating the control actions is set as 0.2s. The leader state observation gains are chosen as $c_{\xi_1} = c_{\xi_2} = c_{\xi_3} = 2$. In the definition of the sliding mode tracking error, $\lambda_{1,0} = \lambda_{2,0} = \lambda_{3,0} = 1$ and $\lambda_{1,1} = \lambda_{2,1} = \lambda_{3,1} = 2$. The control

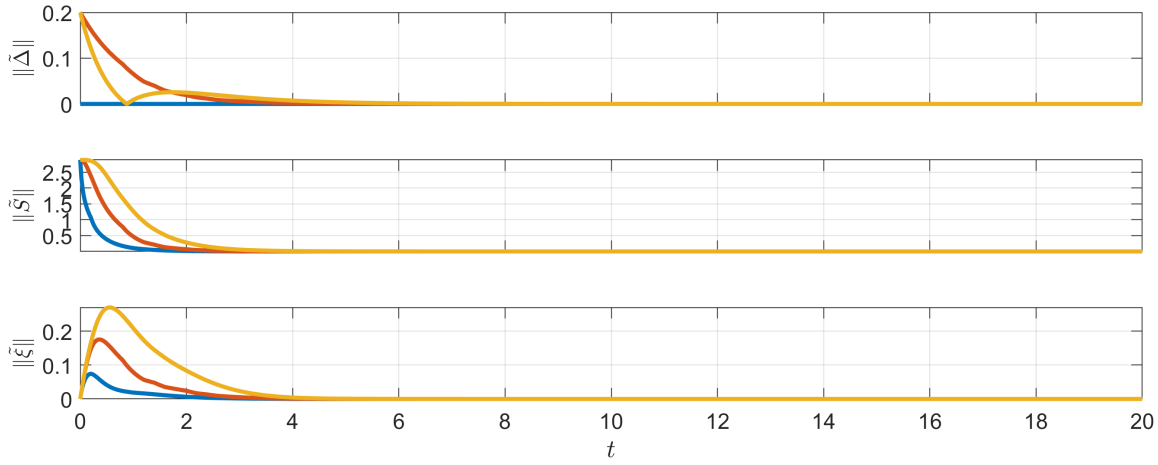

Figure 6.5: Norms of estimation errors

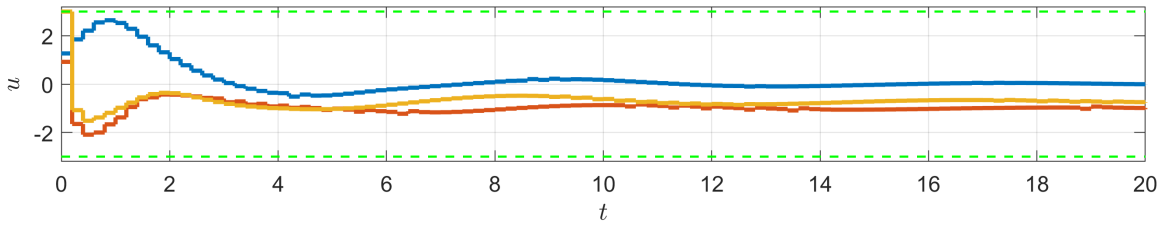

Figure 6.6: Control commands

gains are $c_1 = c_2 = c_3 = 2$. In the MPC problems, the prediction horizon is 0.8s, $Q_1 = Q_2 = Q_3 = 10$ and $R_1 = R_2 = R_3 = 0.1$.

The time-varying communication fault parameters introduced to the network are depicted in Figure 6.2. The simulation results, as shown in Figures 6.3-6.6, illustrate the responses of the three followers with solid lines in blue, orange, and yellow, while the virtual leader's responses are represented with gray dashed lines. Specifically, Figure 6.3 illustrates the output trajectories of the followers, demonstrating that the formation tracking objective has been successfully achieved. The norms of the tracking errors, relative to both the estimated and actual leader states, are displayed in Figure 6.4. Additionally, Figure 6.5 presents the estimation errors of the relative position displacement, the leader's dynamics, and the leader state. Figure 6.6 illustrates the control commands applied to the followers, showing that the input constraint (6.76) is satisfied.

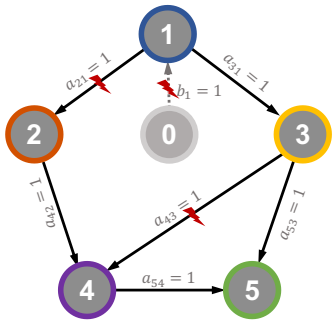

Figure 6.7: Formation shape and communication graph of the 5-UAV system

6.5.2 Example 2: A Multi-UAV System

Next, we consider applying the proposed adaptive distributed control strategy to the outer-loop translation control of a group of UAVs. This control problem has been formulated and extensively investigated in Chapter 5. The networked UAV system comprises 5 UAVs, with their translational motions described by

$$\begin{cases} \dot{\zeta}_i &= v_i \\ \dot{v}_i &= -g + \frac{1}{m_i} r(u_{\eta_i}) u_{F_i} \end{cases} \tag{6.80a}$$

where $i = 1, 2, \cdots, 5$; $g = [0\ 0\ 9.81]^\top$ and $m_i = 2.618$ kg; $\zeta_i \in \mathbb{R}^3$ and $v_i \in \mathbb{R}^3$ are position and velocity vectors of the 5 UAVs; $u_{\eta_i} = [u_{\phi_i}\ u_{\theta_i}\ u_{\psi_i}]^\top$ and $u_{F_i} \in \mathbb{R}$ are control inputs of the translational subsystem, representing the desired rotation angles and the total thrust force, respectively. The control inputs suffer from the following input constraints:

$$\Omega_{u_{\eta_i}} = \left\{ u_{\eta_i} | [-0.5\ \ -0.5\ \ -0.1]^\top \leqslant u_{\eta_i} \leqslant [0.5\ 0.5\ 0.1]^\top \right\} \tag{6.81}$$

$$\Omega_{u_{F_i}} = \{ u_{F_i} | -4 \leqslant u_{F_i} \leqslant 4 \} \tag{6.82}$$

The initial positions of the 5 UAVs are $\zeta_1(0) = [10\ 0\ 0]^\top$, $x_2(0) = [7\ 0\ 0]^\top$, $x_3(0) = [13\ 0\ 0]^\top$, $x_4(0) = [8.5\ 0\ 0]^\top$, $x_5(0) = [11.5\ 0\ 0]^\top$. Their initial linear velocities are all zero.

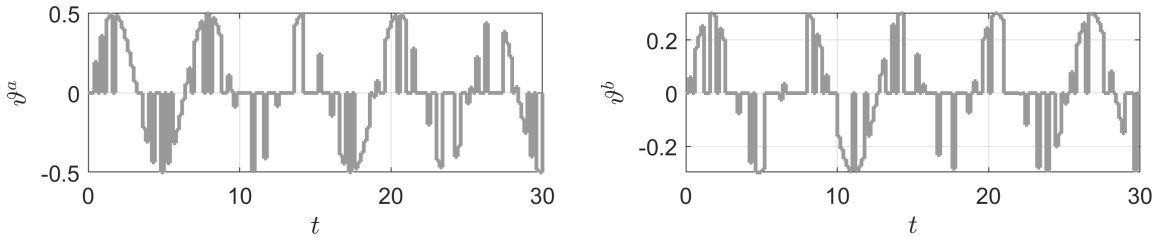

Figure 6.8: Communication link faults in the 5-UAV system

Let the dynamics of the leader node be

$$\dot{\xi}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -0.0676 & 0 & 0 & -0.1040 & 0 & 0 \\ 0 & -0.0676 & 0 & 0 & -0.1040 & 0 \\ 0 & 0 & -0.0025 & 0 & 0 & -0.02 \end{bmatrix} \xi_0 \tag{6.83}$$

with $\xi_0(0) = [10\ 0\ 0\ 0\ 3\ 1.2]^\top$.

The prescribed formation geometric shape and directed communication graph of the 5-UAV systems are illustrated in Figure 6.7. The desired displacement vectors of the followers with respect to the leader 0 are set as $\Delta_{10} = [0\ 1.1\ 0\ 0\ 0\ 0]^\top$, $\Delta_{20} = [-1.5\ 0\ 0\ 0\ 0\ 0]^\top$, $\Delta_{30} = [1.5\ 0\ 0\ 0\ 0\ 0]^\top$, $\Delta_{40} = [-0.95\ \ -1.8\ 0\ 0\ 0\ 0]^\top$, $\Delta_{50} = [0.95\ \ -1.8\ 0\ 0\ 0\ 0]^\top$. Time-varying edge weights, including the adjacency matrix and pinning gains, are designed to mimic faults in the communication network. In particular, the adjacency matrix and the pinning matrix are

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1+0.5\sin(t)*\text{rand}([0,1]) & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1+0.5\sin(t)*\text{rand}([0,1]) & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{6.84}$$

$$\mathcal{B} = \begin{bmatrix} 1+0.3\sin(t)*\text{rand}([0,1]) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.85}$$

where rand$([0,1])$ is a random signal chosen from the interval $[0,1]$. The time-varying fault communication parameters added to the network are illustrated in Figure 6.2.

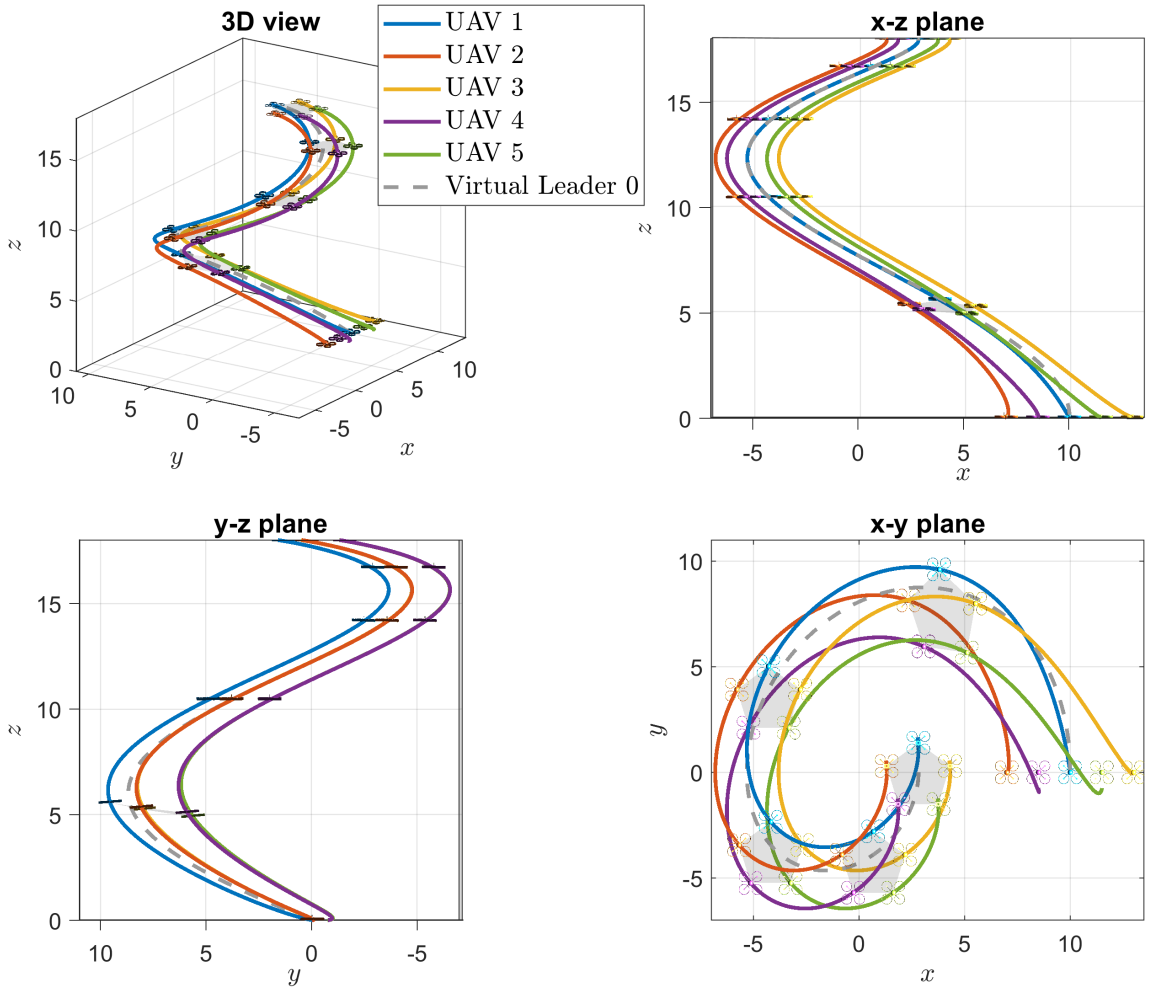

Figure 6.9: Formation tracking performance of the 5 UAVs

The control parameters are selected following the previously established stability conditions. The sampling period for updating the control actions is set as 0.2s. The leader state observation gains are chosen as $c_{\xi_1} = c_{\xi_2} = c_{\xi_3} = c_{\xi_4} = c_{\xi_5} = 1.2$. In the definition of the sliding mode tracking error, $\lambda_{1,0} = \lambda_{2,0} = \lambda_{3,0} = \lambda_{4,0} = \lambda_{5,0} = 1$. The control gains are $c_1 = c_2 = c_3 = c_4 = c_5 = 2$. In the MPC problems, the prediction horizon is 0.8s, $Q_1 = Q_2 = Q_3 = Q_4 = Q_5 = \text{diag}(2, 2, 5)$ and $R_1 = R_2 = R_3 = R_4 = R_5 = \text{diag}(0.1, 10, 10, 10)$.

The simulation results are illustrated in Figures 6.9-6.11, where the responses of the 5 UAVs are depicted with solid lines in blue, orange, yellow, purple, and green, and the virtual leader's responses are depicted with gray dashed lines. Figure 6.9 illustrates the formation tracking performance of the 5-UAV system in 3D and three-plane views. The norms of the estimation and tracking errors of the 5 UAVs are

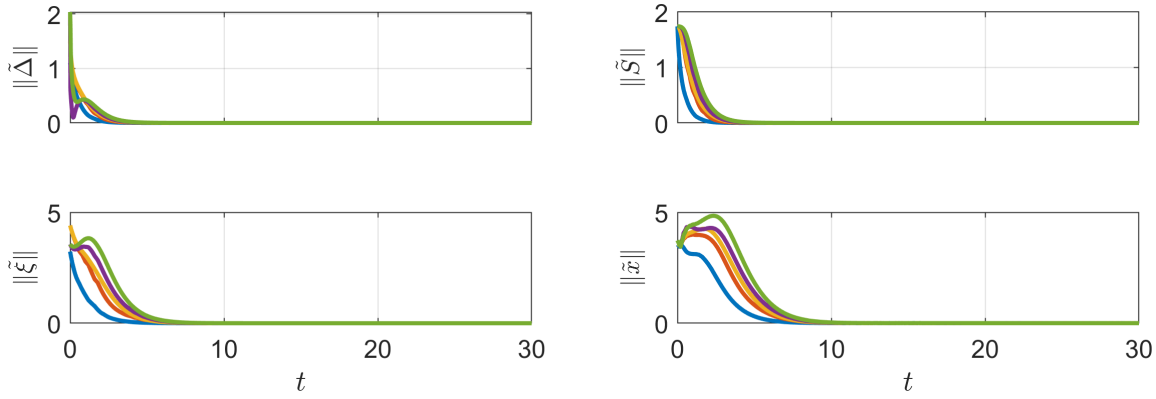

Figure 6.10: Norms of estimation and tracking errors of the 5 UAVs

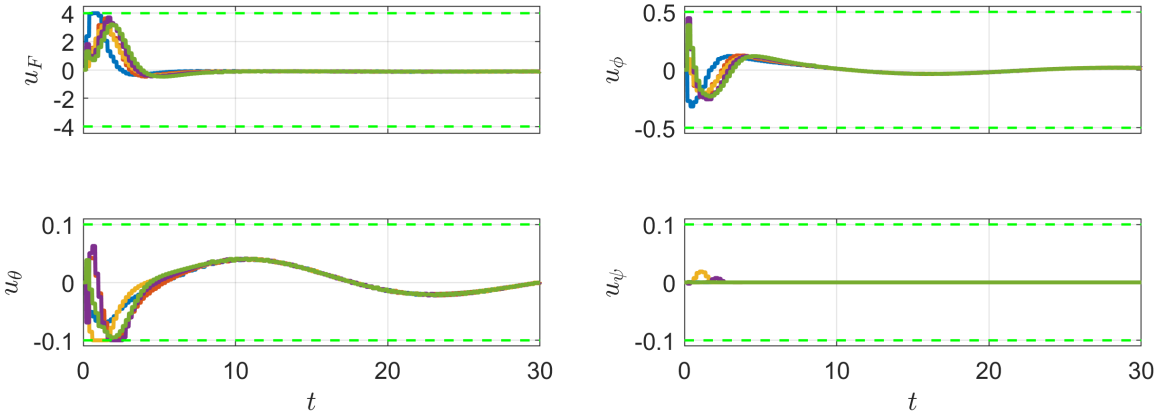

Figure 6.11: Control commands of the 5 UAVs

displayed in Figure 6.10, respectively. Figure 6.11 shows the control commands of the 5 UAVs. It can be seen that their input constraints are all satisfied.

6.6 Conclusions

A novel adaptive distributed observer-based DMPC method has been introduced in this chapter, which is developed for nonlinear multi-agent formation tracking with input constraints and unknown communication faults. The method utilizes adaptive distributed observers in each local control system to estimate the state, dynamics, and relative position of the leader, enabling each agent to independently achieve formation tracking without direct access to the leader's information. The designed distributed MPC controllers use the estimated information to manipulate agents into a predefined formation while respecting input constraints. This research employs

adaptive observers to reduce the complexity in the DMPC design, allowing for effective local controller formulation and resilient distributed formation tracking.

Chapter 7

Conclusions and Future Directions

Conclusions of this dissertation are made in this chapter, followed by a discussion of potential future research directions.

7.1 Conclusions

In this dissertation, the integration of model predictive control (MPC) with fault-tolerant control (FTC) is investigated. A unified fault-tolerant model predictive control (FTMPC) framework is established, aimed at enhancing the efficiency and reliability of UAV systems. Both primary theoretical obstacles and application concerns are thoroughly addressed.

The key research outcomes of this dissertation are presented in Chapters 3-6. More precisely, Chapter 3 develops a novel adaptive fault-tolerant MPC method for addressing fault-tolerant tracking control of constrained nonlinear systems. This design integrates an adaptive fault estimator into the Lyapunov-based MPC framework, thereby ensuring closed-loop control performance and system stability in the presence of actuator faults with reduced computational complexity. Chapter 4 extends the FTMPC framework developed in Chapter 3 by applying it to the trajectory tracking control problem of UAVs with input constraints and actuator faults. It presents the design and stability analysis of a dual-loop, dual-rate hierarchical UAV control system. By implementing MPC only to the outer-loop at a slower sampling rate, this design significantly reduces the computational demands of solving the MPC problem while maintaining the rapid response capabilities of the inner loop. Furthermore, the dual-sampling-rate issue is rigorously evaluated in the closed-loop analysis

using singular perturbation theory, providing vital guidelines for selecting control parameters based on the sampling frequency. Chapter 5 tackles the fault-tolerant formation tracking control problem of a multi-UAV system interconnected through a directed communication graph. With the developed adaptive distributed Lyapunov-based MPC method, the formation tracking control objective is achieved with partially known leader information and unexpected actuator faults. This design also significantly reduces communication and computational burdens by requiring only a single round of calculation and communication per control update. Chapter 6 further addresses unknown communication faults between agents in a nonlinear multi-agent system, instead of only considering the actuator faults that only affect individual local agents in Chapter 5. A novel adaptive distributed observer-based DMPC method is developed, enhancing the resilience of distributed formation tracking against communication faults. Additionally, this strategy is able to simplify the complexity of local MPC design by decomposing the original formation tracking control problem into several fully localized tracking control problems.

Chapters 3-6 are interconnected yet distinct in addressing specific problems. Their connections and differences are summarized as follows:

- **Theoretical & application studies:** The development of novel FTMPC strategies for general nonlinear systems is explored in Chapters 3 and 6 with a focus on single- and multi-agent scenarios, respectively. Practical challenges and application issues of these FTMPC strategies to UAV flight control are further investigated in Chapters 4 and 5. This dissertation consistently addresses control problems of MIMO nonlinear systems, with the consideration of input constraints and higher-order dynamics. Comprehensive stability analyses are conducted, and sufficient stability conditions are derived across all chapters.

- **Centralized & distributed control:** Chapters 3 and 4 are dedicated to addressing centralized tracking control problems, focusing on individual system performance and optimization. In contrast, Chapters 5 and 6 explore distributed control approaches for multi-agent systems (MASs), emphasizing cooperative behavior and network-involved system-wide stability.

- **Individual & network faults:** Chapters 3, 4, and 5 address actuator faults that affect individual systems by using adaptive parameter estimators. In contrast, Chapter 6 considers communication faults on the links between agents

in MASs and employs adaptive state observers to enhance the resilience of the distributed control system.

- **Single-loop & multi-loop structures:** Chapters 3 and 6 utilize a single-loop structure, whereas Chapters 4 and 5 employ a multi-loop structure to tackle the under-actuated control challenge of UAVs.
- **Sampled-data control implementation:** All chapters consider the sample-and-hold nature of MPC. Notably, Chapter 4 explicitly addresses the multi-sampling-rate issue inherent in the multi-loop control structure.

7.2 Future Work

The research conducted in this dissertation lays a foundational framework for further exploration and development in the FTMPC. Future studies could extend this work in several promising directions:

- **Experiment verification:** Future research should include real-world experimental verification of the FTMPC strategies developed in this dissertation. This could involve flight testing on UAVs or other relevant platforms to validate the effectiveness and robustness of the control strategies under various operational conditions and fault scenarios. These experiments could help bridge the gap between theoretical advancements and practical applications.
- **Event-triggered FTMPC:** This approach would update the control inputs only when specific events occur, thereby reducing the computational load and enhancing the system's efficiency. Additionally, predictions generated by the MPC controller could potentially facilitate the development of proactive, forward-looking event-triggered criteria, further optimizing system responsiveness and resource utilization.
- **MHE-based FTMPC:** Incorporating moving horizon estimation (MHE) into the FTMPC framework could enhance fault detection and isolation capabilities. MHE could provide a more accurate estimation of the system states and fault dynamics over a moving horizon, thereby improving the predictive accuracy and reliability of the control system. Future research could focus on the development and integration of robust MHE algorithms tailored for fault-tolerant

applications. This enhancement would ensure that the control system not only accurately predicts and compensates for potential faults but also maintains operational integrity under faulty conditions.

- **Complete corruption of communication links:** Addressing scenarios involving complete corruption of communication links in MASs is crucial, especially in critical applications. Future work could focus on developing resilient control strategies to ensure system resilience and continuous operation, even when standard communication channels fail entirely. This could involve leveraging the predictive capabilities of MPC to provide alternative information about neighboring agents in the event of a complete communication breakdown.

Bibliography

[1] Hassan Shraim, Ali Awada, and Rafic Youness. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerospace and Electronic Systems Magazine*, 33(7):14–33, 2018.

[2] Minh-Duc Hua, Tarek Hamel, Pascal Morin, and Claude Samson. Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Control Systems Magazine*, 33(1):61–75, 2013.

[3] Jiahu Qin, Qichao Ma, Yang Shi, and Long Wang. Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Transactions on Industrial Electronics*, 64(6):4972–4983, 2016.

[4] Georgy Skorobogatov, Cristina Barrado, and Esther Salamí. Multiple UAV systems: A survey. *Unmanned Systems*, 8(02):149–169, 2020.

[5] Steven X Ding. *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer Science & Business Media, 2008.

[6] JW Grizzle and PV Kokotovic. Feedback linearization of sampled-data systems. *IEEE Transactions on Automatic Control*, 33(9):857–859, 1988.

[7] Luca Zaccarian, Andrew R Teel, and Dragan Nešić. On finite gain Lp stability of nonlinear sampled-data systems. *Systems & Control Letters*, 49(3):201–212, 2003.

[8] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice - A survey. *Automatica*, 25(3):335–348, 1989.

[9] David W Clarke, Coorous Mohtadi, and PS Tuffs. Generalized predictive control - Part I. The basic algorithm. *Automatica*, 23(2):137–148, 1987.

[10] Ján Drgoňa, Javier Arroyo, Iago Cupeiro Figueroa, David Blum, Krzysztof Arendt, Donghun Kim, Enric Perarnau Ollé, Juraj Oravec, Michael Wetter, Draguna L Vrabie, et al. All you need to know about model predictive control for buildings. *Annual Reviews in Control*, 50:190–232, 2020.

[11] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[12] Sergio Vazquez, Jose I Leon, Leopoldo G Franquelo, Jose Rodriguez, Hector A Young, Abraham Marquez, and Pericle Zanchetta. Model predictive control: A review of its applications in power electronics. *IEEE Industrial Electronics Magazine*, 8(1):16–31, 2014.

[13] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[14] Eduardo F Camacho and Carlos Bordons. *Model Predictive Control: Classical, Robust and Stochastic*. Springer International Publishing, 2016.

[15] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[16] Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[17] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

[18] Donald Chmielewski and V Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29(3):121–129, 1996.

[19] Giuseppe De Nicolao, Lalo Magni, and Riccardo Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 43(7):1030–1036, 1998.

[20] Lalo Magni, Giuseppe De Nicolao, Lorenza Magnani, and Riccardo Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37(9):1351–1362, 2001.

[21] Lalo Magni and Riccardo Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49(6):900–906, 2004.

[22] Fernando ACC Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42(2):127–143, 2001.

[23] Nael H El-Farra, Prashant Mhaskar, and Panagiotis D Christofides. Uniting bounded control and MPC for stabilization of constrained linear systems. *Automatica*, 40(1):101–110, 2004.

[24] Éva Gyurkovics and Ahmed M Elaiw. Stabilization of sampled-data nonlinear systems by receding horizon control via discrete-time approximations. *Automatica*, 40(12):2017–2028, 2004.

[25] Gene Grimm, Michael J Messina, Sezai Emre Tuna, and Andrew R Teel. Model predictive control: For want of a local control Lyapunov function, all is not lost. *IEEE Transactions on Automatic Control*, 50(5):546–558, 2005.

[26] Prashant Mhaskar, Nael H El-Farra, and Panagiotis D Christofides. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50(11):1670–1680, 2005.

[27] Prashant Mhaskar, Nael H El-Farra, and Panagiotis D Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Systems & Control Letters*, 55(8):650–659, 2006.

[28] James A Primbs, Vesna Nevistic, and John C Doyle. A receding horizon generalization of pointwise min-norm controllers. *IEEE Transactions on Automatic Control*, 45(5):898–909, 2000.

[29] Simone Loureiro de Oliveira Kothare and Manfred Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, 2000.

[30] Chao Shen, Yang Shi, and Brad Buckham. Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. *IEEE Transactions on Industrial Electronics*, 65(7):5796–5805, 2017.

[31] Prashant Mhaskar. Robust model predictive control design for fault-tolerant control of process systems. *Industrial & Engineering Chemistry Research*, 45(25):8565–8574, 2006.

[32] Rui Huang, Lorenz T Biegler, and Sachin C Patwardhan. Fast offset-free nonlinear model predictive control based on moving horizon estimation. *Industrial & Engineering Chemistry Research*, 49(17):7882–7890, 2010.

[33] Buddhadeva Das and Prashant Mhaskar. Lyapunov-based offset-free model predictive control of nonlinear process systems. *The Canadian Journal of Chemical Engineering*, 93(3):471–478, 2015.

[34] Buddhadeva Das and Prashant Mhaskar. Adaptive output-feedback Lyapunov-based model predictive control of nonlinear process systems. *International Journal of Robust and Nonlinear Control*, 28(5):1597–1609, 2018.

[35] Maaz Mahmood and Prashant Mhaskar. Lyapunov-based model predictive control of stochastic nonlinear systems. *Automatica*, 48(9):2271–2276, 2012.

[36] Prashant Mhaskar, Adiwinata Gani, and Panagiotis D Christofides. Fault-tolerant control of nonlinear processes: Performance-based reconfiguration and robustness. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 16(3):91–111, 2006.

[37] Liangfeng Lao, Matthew Ellis, and Panagiotis D Christofides. Proactive fault-tolerant model predictive control. *AIChE Journal*, 59(8):2810–2820, 2013.

[38] David Muñoz de la Peña and Panagiotis D Christofides. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Transactions on Automatic Control*, 53(9):2076–2089, 2008.

[39] Jinfeng Liu, David Munoz de la Pena, Panagiotis D Christofides, and James F Davis. Lyapunov-based model predictive control of nonlinear systems subject to time-varying measurement delays. *International Journal of Adaptive Control and Signal Processing*, 23(8):788–807, 2009.

[40] Jinfeng Liu, David Muñoz de la Peña, and Panagiotis D Christofides. Distributed model predictive control of nonlinear process systems. *AIChE Journal*, 55(5):1171–1184, 2009.

[41] Jinfeng Liu, Xianzhong Chen, David Muñoz de la Peña, and Panagiotis D Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE Journal*, 56(8):2137–2149, 2010.

[42] David Chilin, Jinfeng Liu, David Muñoz de la Peña, Panagiotis D Christofides, and James F Davis. Detection, isolation and handling of actuator faults in distributed model predictive control systems. *Journal of Process Control*, 20(9):1059–1075, 2010.

[43] Jinfeng Liu, Xianzhong Chen, David Muñoz Muñoz de la Pena, and Panagiotis D Christofides. Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements. *IEEE Transactions on Automatic Control*, 57(2):528–534, 2011.

[44] Henglai Wei, Chao Shen, and Yang Shi. Distributed Lyapunov-based model predictive formation tracking control for autonomous underwater vehicles subject to disturbances. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(8):5198–5208, 2019.

[45] Chao Shen and Yang Shi. Distributed implementation of nonlinear model predictive control for AUV trajectory tracking. *Automatica*, 115:108863, 2020.

[46] Henglai Wei, Qi Sun, Jicheng Chen, and Yang Shi. Robust distributed model predictive platooning control for heterogeneous autonomous surface vehicles. *Control Engineering Practice*, 107:104655, 2021.

[47] Guilherme V Raffo, Manuel G Ortega, and Francisco R Rubio. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, 2010.

[48] Dailiang Ma, Yuanqing Xia, Tianya Li, and Kai Chang. Active disturbance rejection and predictive control strategy for a quadrotor helicopter. *IET Control Theory & Applications*, 10(17):2213–2222, 2016.

[49] Ahmed T Hafez, Anthony J Marasco, Sidney N Givigi, Mohamad Iskandarani, Shahram Yousefi, and Camille Alain Rabbath. Solving multi-UAV dynamic encirclement via model predictive control. *IEEE Transactions on Control Systems Technology*, 23(6):2251–2265, 2015.

[50] Ngoc Thinh Nguyen, Ionela Prodan, and Laurent Lefèvre. Stability guarantees for translational thrust-propelled vehicles dynamics through NMPC designs. *IEEE Transactions on Control Systems Technology*, 29(1):207–219, 2020.

[51] Yeonsik Kang and J Karl Hedrick. Linear tracking for a fixed-wing UAV using nonlinear model predictive control. *IEEE Transactions on Control Systems Technology*, 17(5):1202–1210, 2009.

[52] Kostas Alexis, George Nikolakopoulos, and Anthony Tzes. Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10):1195–1207, 2011.

[53] Kostas Alexis, George Nikolakopoulos, and Anthony Tzes. Model predictive quadrotor control: Attitude, altitude and position experimental studies. *IET Control Theory & Applications*, 6(12):1812–1827, 2012.

[54] Abolfazl Eskandarpour and Inna Sharf. A constrained error-based MPC for path following of quadrotor with stability analysis. *Nonlinear Dynamics*, 99(2):899–918, 2020.

[55] Dong Wang, Quan Pan, Yang Shi, Jinwen Hu, and Chunhui Zhao. Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment. *IEEE Transactions on Cybernetics*, 2021.

[56] Yoshiro Hamada, Taro Tsukamoto, and Shinji Ishimoto. Receding horizon guidance of a small unmanned aerial vehicle for planar reference path following. *Aerospace Science and Technology*, 77:129–137, 2018.

[57] Hai-Tao Zhang, Michael ZhiQiang Chen, Guy-Bart Stan, Tao Zhou, and Jan M Maciejowski. Collective behavior coordination with predictive mechanisms. *IEEE Circuits and Systems Magazine*, 8(3):67–85, 2008.

[58] Tamás Keviczky, Francesco Borrelli, and Gary J Balas. A study on decentralized receding horizon control for decoupled systems. In *Proceedings of the 2004 American Control Conference*, volume 6, pages 4921–4926, Boston, MA, USA, May 2004.

[59] Panagiotis D Christofides, Riccardo Scattolini, David Munoz de la Pena, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[60] Rudy R Negenborn and Jose Maria Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4):87–97, 2014.

[61] Eduardo Camponogara, Dong Jia, Bruce H Krogh, and Sarosh Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.

[62] Nader Motee and Bijan Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the 2003 American Control Conference*, volume 6, pages 5300–5305, Denver, CO, USA, June 2003.

[63] Jingyuan Zhan and Xiang Li. Flocking of multi-agent systems via model predictive control based on position-only measurements. *IEEE Transactions on Industrial Informatics*, 9(1):377–385, 2012.

[64] Jingyuan Zhan and Xiang Li. Consensus of sampled-data multi-agent networking systems via model predictive control. *Automatica*, 49(8):2502–2507, 2013.

[65] Hai-Tao Zhang, Zhaomeng Cheng, Guanrong Chen, and Chunguang Li. Model predictive flocking control for second-order multi-agent systems with input constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(6):1599–1606, 2015.

[66] Pierre OM Scokaert and David Q Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[67] Dong Jia and Bruce Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the 2002 American Control Conference*, volume 6, pages 4507–4512, Anchorage, AK, USA, May 2002.

[68] Mehmet Mercangöz and Francis J Doyle III. Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17(3):297–308, 2007.

[69] Brett T Stewart, Aswin N Venkat, James B Rawlings, Stephen J Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, 2010.

[70] Aswin N Venkat, James B Rawlings, and Stephen J Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6680–6685, Seville, Spain, December 2005.

[71] Arthur Richards and Jonathan How. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the 2004 American Control Conference*, volume 5, pages 4261–4266, Boston, MA, USA, June 2004.

[72] Arthur Richards and Jonathan P How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[73] William B Dunbar and Richard M Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[74] Tamás Keviczky, Francesco Borrelli, and Gary J Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

[75] Jingyuan Zhan, Zhong-Ping Jiang, Yebin Wang, and Xiang Li. Distributed model predictive consensus with self-triggered mechanism in general linear multiagent systems. *IEEE Transactions on Industrial Informatics*, 15(7):3987–3997, 2018.

[76] Jin Jiang. Fault-tolerant control systems - An introductory overview. *ACTA Automatica Sinica*, 31(1):161–174, 2005.

[77] Df D Šiljak. Reliable control using multiple control systems. *International Journal of Control*, 31(2):303–329, 1980.

[78] Robert J Veillette, Jure V Medanic, and William R Perkins. Design of reliable control systems. In *Proceedings of 29th IEEE Conference on Decision and Control*, pages 1131–1136, Honolulu, HI, USA, December 1990.

[79] Robert J Veillette. Reliable linear-quadratic state-feedback control. *Automatica*, 31(1):137–143, 1995.

[80] Guang-Hong Yang, Si-Yang Zhang, James Lam, and Jianliang Wang. Reliable control using redundant controllers. *IEEE Transactions on Automatic Control*, 43(11):1588–1593, 1998.

[81] G-H Yang, Jian Liang Wang, and Yeng Chai Soh. Reliable LQG control with sensor failures. *IEE Proceedings-Control Theory and Applications*, 147(4):433–439, 2000.

[82] Yew-Wen Liang, Der-Cheng Liaw, and Ti-Chung Lee. Reliable control of nonlinear systems. *IEEE Transactions on Automatic Control*, 45(4):706–710, 2000.

[83] Chien-Shu Hsieh. Performance gain margins of the two-stage LQ reliable control. *Automatica*, 38(11):1985–1990, 2002.

[84] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229–252, 2008.

[85] JM Maciejowski. Modelling and predictive control: Enabling technologies for reconfiguration. *Annual Reviews in Control*, 23:13–23, 1999.

[86] Mostafa Abdel-Geliel, Essameddin Badreddin, and Adrian Gambier. Application of model predictive control for fault tolerant system using dynamic safety margin. In *Proceedings of the 2006 American Control Conference*, pages 5493–5498, Minneapolis, Minnesota, USA, June 2006.

[87] Bin Yu, Youmin Zhang, Ismael Minchala, and Yaohong Qu. Fault-tolerant control with linear quadratic and model predictive control techniques against actuator faults in a quadrotor UAV. In *Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems*, pages 661–666, Nice, France, October 2013.

[88] Reza Sheikhbahaei, Aria Alasty, and Gholamreza Vossoughi. Robust fault tolerant explicit model predictive control. *Automatica*, 97:248–253, 2018.

[89] Ridong Zhang, Jingyi Lu, Hongyi Qu, and Furong Gao. State space model predictive fault-tolerant control for batch processes with partial actuator failure. *Journal of Process Control*, 24(5):613–620, 2014.

[90] DK Kufoalor and Tor Arne Johansen. Reconfigurable fault tolerant flight control based on nonlinear model predictive control. In *Proceedings of 2013 American Control Conference*, pages 5128–5133, Washington, DC, USA, June 2013.

[91] Eduardo F Camacho, Teodoro Alamo, and D Muñoz de la Peña. Fault-tolerant model predictive control. In *Proceedings of 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation*, Bilbao, Spain, November 2010.

[92] Sachin C Patwardhan, Seema Manuja, Shankar Narasimhan, and Sirish L Shah. From data to diagnosis and control using generalized orthonormal basis filters - Part II: Model predictive and fault tolerant control. *Journal of Process Control*, 16(2):157–175, 2006.

[93] J Prakash, Shankar Narasimhan, and Sachin C Patwardhan. Integrating model based fault diagnosis with model predictive control. *Industrial & Engineering Chemistry Research*, 44(12):4344–4360, 2005.

[94] Fabio A De Almeida and Dirk Leissling. Fault-tolerant flight control system using model predictive control. In *Proceedings of the 2009 Brazilian Symposium on Aerospace Eng. & Applications*, São Paulo, Brazil, September 2009.

[95] Feng Xu, Vicenç Puig, Carlos Ocampo-Martinez, and Xueqian Wang. Set-valued observer-based active fault-tolerant model predictive control. *Optimal Control Applications and Methods*, 38(5):683–708, 2017.

[96] Xiaoke Yang and Jan Maciejowski. Fault tolerant control using Gaussian processes and model predictive control. *International Journal of Applied Mathematics and Computer Science*, 25(1):133–148, 2015.

[97] Hojjat A Izadi, Youmin Zhang, and Brandon W Gordon. Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation. In *Proceedings of the the 18th World CongressThe International Federation of Automatic Control*, volume 44(1), pages 6343–6348, Milano, Italy, August 2011.

[98] Anjali P Deshpande, Sachin C Patwardhan, and Shankar S Narasimhan. Intelligent state estimation for fault tolerant nonlinear predictive control. *Journal of Process Control*, 19(2):187–204, 2009.

[99] Tong Wang, Huijun Gao, and Jianbin Qiu. A combined fault-tolerant and predictive control for network-based industrial processes. *IEEE Transactions on Industrial Electronics*, 63(4):2529–2536, 2016.

[100] Jan M Maciejowski and Xiaoke Yang. Fault tolerant control using Gaussian processes and model predictive control. In *Proceedings of 2013 Conference on Control and Fault-Tolerant Systems*, pages 1–12, Nice, France, October 2013.

[101] S Vahid Naghavi, Ali Akbar Safavi, and M Kazerooni. Decentralized fault tolerant model predictive control of discrete-time interconnected nonlinear systems. *Journal of the Franklin Institute*, 351(3):1644–1656, 2014.

[102] Kamel Menighed, Christophe Aubrun, and Joseph-Julien Yamé. Distributed state estimation and model predictive control: Application to fault tolerant control. In *Proceedings of 2009 IEEE International Conference on Control and Automation*, pages 936–941, Christchurch, New Zealand, December 2009.

[103] Fabio A De Almeida and Dirk Leißling. Fault-tolerant model predictive control with flight-test results. *Journal of Guidance, Control, and Dynamics*, 33(2):363–375, 2010.

[104] Hojjat A Izadi, Brandon W Gordon, and Youmin Zhang. A data-driven fault tolerant model predictive control with fault identification. In *Proceedings of 2010 Conference on Control and Fault-Tolerant Systems*, pages 732–737, Nice, France, October 2010.

[105] Chao Huang, Fazel Naghdy, and Haiping Du. Delta operator-based fault estimation and fault-tolerant model predictive control for steer-by-wire systems. *IEEE Transactions on Control Systems Technology*, 26(5):1810–1817, 2017.

[106] Tobias Miksch, Adrian Gambier, and Essameddin Badreddin. Real-time implementation of fault-tolerant control using model predictive control. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, pages 11136–11141, Seoul, Korea, July 2008.

[107] DA Joosten, Ton JJ van den Boom, and TJJ Lombaerts. Fault-tolerant control using dynamic inversion and model-predictive control applied to an aerospace benchmark. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, volume 41(2), pages 12030–12035, Seoul, Korea, July 2008.

[108] Krzysztof Patan and Józef Korbicz. Nonlinear model predictive control of a boiler unit: A fault tolerant control study. *International Journal of Applied Mathematics and Computer Science*, 22:225–237, 2012.

[109] Xiaoke Yang and Jan M Maciejowski. Fault-tolerant model predictive control of a wind turbine benchmark. In *Proceedings of the 8th IFAC Symposium on Fault Detection,Supervision and Safety of Technical Processes*, pages 337–342, Mexico City, Mexico, August 2012.

[110] Alexandar Ichtev, J Hellendoom, Robert Babuska, and Stanimir Mollov. Fault-tolerant model-based predictive control using multiple Takagi-Sugeno fuzzy models. In *Proceedings of 2002 IEEE World Congress on Computational Intelligence*, volume 1, pages 346–351, Honolulu, HI, USA, May 2002.

[111] J Prakash, Sachin C Patwardhan, and Sirish L Shah. Design and implementation fault tolerant model predictive control scheme on a simulated model of a three-tank hybrid system. In *Proceedings of 2010 Conference on Control and Fault-Tolerant Systems*, pages 173–178, Nice, France, October 2010.

[112] Anas Alanqar, Helen Durand, and Panagiotis D Christofides. Fault-tolerant economic model predictive control using error-triggered online model identification. *Industrial & Engineering Chemistry Research*, 56(19):5652–5667, 2017.

[113] Roger Keller, Steven X Ding, M Müller, and D Stolten. Fault-tolerant model predictive control of a direct methanol-fuel cell system with actuator faults. *Control Engineering Practice*, 66:99–115, 2017.

[114] Laura Ferranti, Yiming Wan, and Tamas Keviczky. Fault-tolerant reference generation for model predictive control with active diagnosis of elevator jamming faults. *International Journal of Robust and Nonlinear Control*, 29(16):5412–5428, 2019.

[115] Carlos Ocampo-Martinez and Vicenç Puig. Fault-tolerant model predictive control within the hybrid systems framework: Application to sewer networks. *International Journal of Adaptive Control and Signal Processing*, 23(8):757–787, 2009.

[116] Davide M Raimondo, G Roberto Marseglia, Richard D Braatz, and Joseph K Scott. Fault-tolerant model predictive control with active fault isolation. In

Proceedings of 2013 Conference on Control and Fault-Tolerant Systems, pages 444–449, Nice, France, October 2013.

[117] M. Gopinathan, J.D. Boskovic, R.K. Mehra, and C. Rago. A multiple model predictive scheme for fault-tolerant flight control design. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 2, pages 1376–1381, Tampa, FL, USA, December 1998.

[118] Mouhacine Benosman and Kai-Yew Lum. Online references reshaping and control reallocation for nonlinear fault tolerant control. *IEEE Transactions on Control Systems Technology*, 17(2):366–379, 2009.

[119] Xiaoxiang Hu, Hamid Reza Karimi, Ligang Wu, and Yang Guo. Model predictive control-based non-linear fault tolerant control for air-breathing hypersonic vehicles. *IET Control Theory & Applications*, 8(13):1147–1153, 2014.

[120] Alexey Zakharov, Elena Zattoni, Miao Yu, and Sirkka-Liisa Jämsä-Jounela. A performance optimization algorithm for controller reconfiguration in fault tolerant distributed model predictive control. *Journal of Process Control*, 34:56–69, 2015.

[121] Guannan Xiao and Fei Liu. Observer-based cooperative distributed fault-tolerant model predictive control with imperfect network communication and asynchronous measurements. *International Journal of Robust and Nonlinear Control*, 30(12):4531–4549, 2020.

[122] Ionela Prodan, Enrico Zio, and Florin Stoican. Fault tolerant predictive control design for reliable microgrid energy management under uncertainties. *Energy*, 91:20–34, 2015.

[123] Qingfang Teng, Jianyong Bai, Jianguo Zhu, and Yunxia Sun. Fault tolerant model predictive control of three-phase permanent magnet synchronous motors. *WSEAS Transactions on systems*, 12(8):385–397, 2013.

[124] Tao Tao, Wenxiang Zhao, Yuxuan Du, Yu Cheng, and Jihong Zhu. Simplified fault-tolerant model predictive control for a five-phase permanent-magnet motor with reduced computation burden. *IEEE Transactions on Power Electronics*, 35(4):3850–3858, 2019.

[125] Deneb Robles, Vicenc Puig, Carlos Ocampo-Martinez, and Luis E Garza-Castanon. Reliable fault-tolerant model predictive control of drinking water transport networks. *Control Engineering Practice*, 55:197–211, 2016.

[126] Jan M Maciejowski and Colin N Jones. MPC fault-tolerant flight control case study: Flight 1862. In *Proceedings of IFAC Fault Delection, Supervision and Safety of Technical Processes*, volume 36(5), pages 119–124, Washington, D.C., USA, 2003.

[127] Carlos Ocampo-Martínez, Vicenç Puig, Joseba Quevedo, and Ari Ingimundarson. Fault tolerant model predictive control applied on the barcelona sewer network. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 1349–1354, Seville, Spain, December 2005.

[128] Johan Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Prooceedings of 2004 IEEE International Conference on Robotics and Automation*, pages 284–289, Taipei, Taiwan, September 2004.

[129] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

[130] H. K. Khalil. *Nonlinear Systems (1st edition)*. Macmillan, 1992.

[131] Samir Bouabdallah. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, 2007.

[132] Wei Ren and Yongcan Cao. *Distributed Coordination of Multi-Agent Networks: Emergent Problems, Models, and Issues*, volume 1. Springer, 2011.

[133] Qinmin Yang, Shuzhi Sam Ge, and Youxian Sun. Adaptive actuator fault tolerant control for uncertain nonlinear systems with multiple actuators. *Automatica*, 60:92–99, 2015.

[134] Xiao-Jian Li and Guang-Hong Yang. Adaptive fault-tolerant synchronization control of a class of complex dynamical networks with general input distribution matrices and actuator faults. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):559–569, 2015.

[135] Qikun Shen, Bin Jiang, and Peng Shi. Adaptive fault tolerant control against actuator faults. *International Journal of Adaptive Control and Signal Processing*, 31(2):147–162, 2017.

[136] Ci Chen, Kan Xie, Frank L Lewis, Shengli Xie, and Rafael Fierro. Adaptive synchronization of multi-agent systems with resilience to communication link faults. *Automatica*, 111:108636, 2020.

[137] Zhihua Qu. *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. Springer Science & Business Media, 2009.

[138] Shuzhi Sam Ge, Chang C Hang, Tong H Lee, and Tao Zhang. *Stable Adaptive Neural Network Control*, volume 13. Springer Science & Business Media, 2013.

[139] Zhiyong Chen and Jie Huang. Stabilization and regulation of nonlinear systems. *Cham, Switzerland: Springer*, 2015.

[140] Rafig Agaev and Pavel Chebotarev. The matrix of maximum out forests of a digraph and its applications. *arXiv Preprint Math/0602059*, 2006.

[141] Wei Ren and Randal W Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*, volume 27(2). Springer, 2008.

[142] Jan M Maciejowski. The implicit daisy-chaining property of constrained predictive control. *Applied Mathematics and Computer Science*, 8:695–712, 1998.

[143] David Q Mayne and H Michalska. Adaptive receding horizon control for constrained nonlinear systems. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 1286–1291, San Antonio, TX, USA, August 1993.

[144] Manoj Shouche, Hasmet Genceli, Vuthandam Premkiran, and Michael Nikolaou. Simultaneous constrained model predictive control and identification of DARX processes. *Automatica*, 34(12):1521–1530, 1998.

[145] Danielle Dougherty and Doug Cooper. A practical multiple model adaptive strategy for single-loop MPC. *Control Engineering Practice*, 11(2):141–159, 2003.

[146] Hiroaki Fukushima, Tae-Hyoung Kim, and Toshiharu Sugie. Adaptive model predictive control for a class of constrained linear systems based on the comparison model. *Automatica*, 43(2):301–308, 2007.

[147] T-H Kim and Toshiharu Sugie. Adaptive receding horizon predictive control for constrained discrete-time linear systems with parameter uncertainties. *International Journal of Control*, 81(1):62–73, 2008.

[148] Veronica Adetola, Darryl DeHaan, and Martin Guay. Adaptive model predictive control for constrained nonlinear systems. *Systems & Control Letters*, 58(5):320–326, 2009.

[149] Bing Zhu and Xiaohua Xia. Lyapunov-based adaptive model predictive control for unconstrained non-linear systems with parametric uncertainties. *IET Control Theory & Applications*, 10(15):1937–1943, 2016.

[150] Samir Bouabdallah, Andre Noth, and Roland Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2451–2456, Sendai, Japan, September 2004.

[151] Agus Budiyono and Singgih S Wibowo. Optimal tracking controller design for a small scale helicopter. *Journal of Bionic Engineering*, 4(4):271–280, 2007.

[152] John Hauser, Shankar Sastry, and George Meyer. Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft. *Automatica*, 28(4):665–679, 1992.

[153] Daewon Lee, H Jin Kim, and Shankar Sastry. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of Control, Automation and Systems*, 7(3):419–428, 2009.

[154] Tarek Madani and Abdelaziz Benallegue. Control of a quadrotor mini-helicopter via full state backstepping technique. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1515–1520, San Diego, CA, USA, December 2006.

[155] Fuyang Chen, Rongqiang Jiang, Kangkang Zhang, Bin Jiang, and Gang Tao. Robust backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV. *IEEE Transactions on Industrial Electronics*, 63(8):5044–5056, 2016.

[156] Hadi Razmi and Sima Afshinfar. Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV. *Aerospace Science and Technology*, 91:12–27, 2019.

[157] Moussa Labbadi and Mohamed Cherkaoui. Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV. *Aerospace Science and Technology*, 93:105306, 2019.

[158] Kunwu Zhang, Yang Shi, and Huaiyuan Sheng. Robust nonlinear model predictive control based visual servoing of quadrotor UAVs. *IEEE/ASME Transactions on Mechatronics*, 26(2):700–708, 2021.

[159] Petar Kokotović, Hassan K Khalil, and John O'reilly. *Singular Perturbation Methods in Control: Analysis and Design*. SIAM, 1999.

[160] S Esteban, F Gordillo, and J Aracil. Three-time scale singular perturbation control and stability analysis for an autonomous helicopter on a platform. *International Journal of Robust and Nonlinear Control*, 23(12):1360–1392, 2013.

[161] Sylvain Bertrand, Nicolas Guénard, Tarek Hamel, Hélène Piet-Lahanier, and Laurent Eck. A hierarchical controller for miniature VTOL UAVs: Design and stability analysis using singular perturbation theory. *Control Engineering Practice*, 19(10):1099–1108, 2011.

[162] Ricardo Pérez-Alcocer, Javier Moreno-Valenzuela, and Roger Miranda-Colorado. A robust approach for trajectory tracking control of a quadrotor with experimental validation. *ISA Transactions*, 65:262–274, 2016.

[163] Huiping Li and Yang Shi. *Robust Receding Horizon Control for Networked and Distributed Nonlinear Systems*. Springer, 2017.

[164] Holger Voos. Nonlinear control of a quadrotor micro-UAV using feedback-linearization. In *Proceedings of 2009 IEEE International Conference on Mechatronics*, pages 315–320, Malaga, Spain, April 2009.

[165] Brian DO Anderson, Barış Fidan, Changbin Yu, and Dirk Walle. *UAV Formation Control: Theory and Application*. Springer London, 2008.

[166] Xiwang Dong, Bocheng Yu, Zongying Shi, and Yisheng Zhong. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Transactions on Control Systems Technology*, 23(1):340–348, 2014.

[167] Xiwang Dong, Yan Zhou, Zhang Ren, and Yisheng Zhong. Time-varying formation control for unmanned aerial vehicles with switching interaction topologies. *Control Engineering Practice*, 46:26–36, 2016.

[168] Kexin Guo, Xiuxian Li, and Lihua Xie. Ultra-wideband and odometry-based cooperative relative localization with application to multi-UAV formation control. *IEEE Transactions on Cybernetics*, 50(6):2590–2603, 2019.

[169] Fei Liu, Yongzhao Hua, Xiwang Dong, Qingdong Li, and Zhang Ren. Adaptive fault-tolerant time-varying formation tracking for multi-agent systems under actuator failure and input saturation. *ISA Transactions*, 104:145–153, 2020.

[170] Yang Shi and Kunwu Zhang. Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives. *Annual Reviews in Control*, 52:170–196, 2021.

[171] Huiping Li and Yang Shi. Robust distributed model predictive control of constrained continuous-time nonlinear systems: A robustness constraint approach. *IEEE Transactions on Automatic Control*, 59(6):1673–1678, 2013.

[172] Yang Zheng, Shengbo Eben Li, Keqiang Li, Francesco Borrelli, and J Karl Hedrick. Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Transactions on Control Systems Technology*, 25(3):899–910, 2016.

[173] PN Shivakumar and Kim Ho Chew. A sufficient condition for nonvanishing of determinants. *Proceedings of the American Mathematical Society*, pages 63–66, 1974.

[174] Binyan Xu, Afzal Suleman, and Yang Shi. A multi-rate hierarchical fault-tolerant adaptive model predictive control framework: Theory and design for quadrotors. *Automatica*, 153:111015, 2023.

[175] Zhongkui Li and Jie Chen. Robust consensus for multi-agent systems communicating over stochastic uncertain networks. *SIAM Journal on Control and Optimization*, 57(5):3553–3570, 2019.

[176] Tao Li, Fuke Wu, and Ji-Feng Zhang. Multi-agent consensus with relative-state-dependent measurement noises. *IEEE Transactions on Automatic Control*, 59(9):2463–2468, 2014.

[177] Xu Ma and Nicola Elia. Mean square performance and robust yet fragile nature of torus networked average consensus. *IEEE Transactions on Control of Network Systems*, 2(3):216–225, 2015.

[178] Jing Wang and Nicola Elia. Consensus over networks with dynamic channels. *International Journal of Systems, Control and Communications*, 2(1-3):275–297, 2010.

[179] Daniel Zelazo and Mathias Bürger. On the robustness of uncertain consensus networks. *IEEE Transactions on Control of Network Systems*, 4(2):170–178, 2015.

[180] Qiyu Yang, Yi Lyu, Xiaolei Li, Ci Chen, and Frank L Lewis. Adaptive distributed synchronization of heterogeneous multi-agent systems over directed graphs with time-varying edge weights. *Journal of the Franklin Institute*, 358(4):2434–2452, 2021.

[181] William B Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, 2007.

[182] Henglai Wei, Changxin Liu, and Yang Shi. A robust distributed MPC framework for multi-agent consensus with communication delays. *IEEE Transactions on Automatic Control*, 2024.

Appendix A Publications

- **Published journal paper:**

J1. **Binyan Xu**, Afzal Suleman, and Yang Shi, "A Multi-rate hierarchical fault-tolerant adaptive model predictive control framework: Theory and design for quadrotors", *Automatica*, 153: 111015, 2023. [Full Paper]

- **Under revision journal paper:**

J2. **Binyan Xu**, Yufan Dai, Afzal Suleman, and Yang Shi, "Distributed fault-tolerant control of multi-UAV formation for dynamic leader tracking: A Lyapunov-based MPC framework", *Automatica*, under major revision, 2024.

- **In preparation journal paper:**

J3. **Binyan Xu**, Yufan Dai, Afzal Suleman, and Yang Shi, "Adaptive distributed observer-based MPC for multi-agent formation with resilience to communication link faults", in preparation.

- **Accepted conference papers:**

C1. **Binyan Xu**, Yufan Dai, Afzal Suleman, and Yang Shi, "Adaptive distributed model predictive control for multi-UAV formation tracking with weighted directed graphs", *the 18th IEEE International Conference on Control & Automation (ICCA 2024)*, Reykjavík, Iceland, June 18–21, 2024.

C2. **Binyan Xu**, Afzal Suleman, and Yang Shi, "Distributed fault-tolerant model predictive control for multi-UAV formation", *the 22nd World Congress of the International Federation of Automatic Control (IFAC 2023)*, Yokohama, Japan, Jul 9-14, 2023.

C3. **Binyan Xu**, Yang Shi, and Afzal Suleman, "Hierarchical adaptive fault-tolerant model predictive control of a quadrotor", *International Symposium on Unmanned Systems and Defense Industry 2021 (ISUDEF 2021)*, Washington DC, USA, Oct 26-28, 2021.