Multiple Model Based State Estimation and Trajectory Control for Micro Aerial Vehicles

by

© Mahmoud Abdelhakim Kamel Gomaa

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

Doctor of Philosophy

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

May 2021
St. John's, Newfoundland

Abstract

This thesis proposes the design of a multiple model state estimation and control scheme for micro aerial vehicles (MAVs) to cope with different flight conditions such as aggressive flights, hovering flights, and flights under high external disturbances. The work is divided into two main parts.

The first part of this thesis presents the design of an interacting multiple model (IMM) filter for visual-inertial navigation (VIN) of MAVs. VIN of MAVs in practice typically uses a single system model for its state estimator design. However, MAVs can operate in different scenarios requiring changes to the estimator model. This thesis proposes the use of a conventional VIN and a drag force VIN in an error-state IMM filtering framework to address the need for multiple models in the estimator. We use an epipolar geometry constraint for the design of the measurement model for both filters to realize computationally efficient state updates. Observability of the proposed modifications to VIN filters (drag force model, and epipolar measurement model) are analyzed, and observability-based consistency rules are derived for the two filters of the IMM. Monte Carlo numerical simulations validate the performance of the observability constrained IMM, which improved the accuracy and consistency of the VINS for changing flight conditions and external wind disturbance scenarios. Experimental validation is performed using the EuRoC dataset to evaluate the performance of the proposed IMM filter design.

The second part of this thesis presents the design of a multiple model controller for MAVs operating under different flight conditions. It presents the design of a stability-guaranteed nonlinear model predictive controller (NMPC) to operate robustly along fast trajectories. The NMPC considers system models with and without drag forces for the multiple model bank. The basic controller structure is designed without terminal conditions and therefore it is computationally less demanding and provides larger stability regions and better closed-loop performance than traditional nonlinear predictive control schemes. We perform a detailed stability analysis without terminal costs or constraints to prove the asymptotic stability and the necessary conditions for recursive feasibility of the controller. We derive the growth bound sequence that enables obtaining the shortest possible prediction horizon for stability. The proposed analysis provides the necessary conditions to implement the controller while using the shortest stabilizing prediction horizon when compared to the state-of-the-art model predictive control schemes reported in the literature. This particular feature enables the proposed controller to perform fast optimization and hence the capability to implement fast trajectories using feedback regularization. Several MATLAB simulations and lab experiments are conducted to demonstrate the validity of this new proposed control scheme. Combining these two key developments, the thesis presents the design of a multiple-NMPC controller, depending on the IMM filter, to detect the flight condition and then trigger the appropriate controller to improve the tracking performance. Monte Carlo numerical simulations validate the performance of the multiple-NMPC, which improved the tracking accuracy of the MAV for external wind disturbance scenarios.

To my wonderful parents and siblings

To my beloved wife and kids, Yusuf, Tamim, and Yazan

Acknowledgements

All praise and thanks are to Almighty Allah, most Gracious and most Merciful, for his blessing and guidance to complete this work.

Life is depicted by the people we encounter throughout the journey. I find myself incredibly lucky with the persons I had the opportunity to meet and interact with during these past four years spent at Memorial University. Therefore, I would like to express my deepest and sincere gratitude to my supervisors Dr. George Mann, Dr. Oscar De Silva, and Dr. Ray Gosine, for their amazing support, guidance, flexibility, motivating feedback, and encouragement. I would like to thank them for being such great supervisors who have stimulated me to achieve a considerable number of contributions towards my degree. It has been a great experience working with such a distinctive team.

I would like to acknowledge the financial support provided by my supervisors, Natural Sciences and Engineering Research Council of Canada (NSERC), and Memorial University of Newfoundland. I would like to thank the staff of the School of Graduate Studies. I thank also all of the ISLab colleagues for their assistance and support.

Last but not least, there are no words enough to thank my parents for their encouragement and moral support. Their love and prayers gave me the strength to reach this stage in my education. I owe them all the success in my life. I would also like to thank my siblings for always being next to me. An especial affectionate thought goes to my wife; her everlasting love, patience, and faith in me provided me the finest things I ever needed. Finally, I thank my kids for giving me so much purpose and hope in life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

C-NMPC	Conventional Visual-Inertial Navigation System.
C-VINS	Conventional Visual-Inertial Navigation System.
CGR	Cayley-Gibbs-Rodriguez.
COG	Center of Gravity.
DF-NMPC	Drag Force Visual-Inertial Navigation System.
DF-VINS	Drag Force Visual-Inertial Navigation System.
DOF	Degree of Freedom.
EKF	Extended Kalman Filter.
IMU	Inertial Measurement Unit.
INS	Inertial Navigation System.
MAV	Micro Aerial Vehicle.
MPC	Model Predictive Control.
MSCKF	Multi-State Constraint Kalman Filter.
NEES	Normalized Estimation Error Square.
OCP	Optimal Control Problem.
RMSE	Root Mean Square Error.
ROS	Robot Operating System.
VINS	Visual-Inertial Navigation System.

Chapter 1

Introduction

In this chapter, the research motivation and addressed challenges are discussed. Afterwards, the objectives and contributions of the thesis are formulated. Finally, the outline of the thesis is presented.

1.1 Research Motivation

Multi-rotor micro aerial vehicles (MAVs) are multi-rotor aircrafts that are designed to increase maneuverability in constrained environments while carrying small payloads. They can be used in a wide range of indoor or outdoor applications including, inspection and surveillance missions [1], package delivery [2], and aerial photography [3]. An MAV can be controlled remotely by a human operator or autonomously by its onboard controller [4]. Each rotor's speed can be controlled to generate the desired total thrust and torque. The use of multi-rotor MAVs, such as quadrotors, enables simple configurations, agile dynamics, limited maintenance requirements, and low safety concerns compared to fixed-wing aircrafts [5]. On the other hand, the autonomy tasks, including estimation and control, of quadrotor MAVs are quite challenging because the system is open-loop unstable, has 3D geometric system nonlinearities invoked by the rigid body dynamics, and is under-actuated, i.e., it has only four control actions (inputs) from the four rotors and six degrees of freedom states parameterizing the quadrotor pose in space. In addition, the payload constraints place considerable restrictions on available computational resources and sensing. As a result, quadrotor MAVs are attracting scholars to develop advanced estimation and control techniques to enhance the MAV performance and overcome the operational challenges.

One of the main challenges related to control and estimation of the MAVs is the changing dynamics[1] of the system under different behavior modes, e.g., external wind disturbance, ground effect[2], proximity to walls or objects in the environment, taking off, and landing [7]. Therefore, addressing different flight conditions using more representative system models would improve the estimation and control performance of MAVs. As a result, the main focus of this thesis is twofold: design (1) a multiple model state estimator and (2) a multiple model controller for different flight modes of the MAVs.

The design of a multiple model state estimator includes the design of a computationally efficient visual-inertial navigation system with improved consistency, and the design of an interacting multiple model system that is cable of switching among different filters in its bank to improve the estimation accuracy. Similarly, the design of a multiple model controller includes the design of a computationally efficient and highly stable nonlinear model predictive controller, and the design of a multiple model framework that supports operation under different flight conditions.

[1]The dynamic system model significantly changes due to external disturbances.

[2]Ground effect is the increase in the thrust of a rotor operating near surfaces compared to flight at a large distance from the ground [6].

1.2 Thesis Problem Statement

The problem statements related to the multiple model estimation and multiple model control problems are separately developed in the following two subsections.

1.2.1 Part1: Multiple Model State Estimation

Over the past few decades, inertial navigation systems (INS) [8] have been widely used for pose estimation of autonomous mobile vehicles, e.g., MAVs, in particular, in GPS-denied environments such as urban and indoor areas. Most INS rely on an inertial measurement unit (IMU) that measures the 3-axis linear acceleration and 3-axis angular velocity of the platform to which it is rigidly connected. With the recent advancements in micro-electronics and micro-machining technologies, low-cost light-weight micro-electro-mechanical (MEMS) IMUs have become compact and affordable [9], which enables high-accuracy localization for mobile vehicles [10]. However, such MEMS IMUs are subject to different level of systematic and stochastic errors, i.e., bias and noise. As a result, simple integration of these high-rate IMU measurements often leads to unreliable pose estimates for long-term navigation because the errors accumulate in the integration process and grow without bound in long runs [11]. Intriguingly, small, light-weight, and energy-efficient cameras provide rich information about the environment and can serve as an aiding source for INS, yielding a navigation system with drift-free velocity and attitude estimates [11].

Visual-Inertial navigation systems (VINS) are localization modules that can combine visual information from a camera and inertial information from an IMU for navigation purposes. Reported VINS designs primarily rely on either optimization-based approaches [12–14] or filtering-based approaches [11, 15, 16]. Filtering-based approaches are computationally more efficient than optimization approaches for the implementation of VINS where a marginal (<1% of traveled distance) decrease in accuracy is expected as compared in [17]. The number of camera poses stored in filtering-based VINS dictates the type of measurement model used for filter update. For two camera poses in the state vector, an epipolar constraint measurement model is preferred [11], for three poses a trifocal tensor geometry measurement model is preferred [16], while for any higher number of camera poses the Multi-State Constraint Kalman Filter (MSCKF) is utilized [15]. Incorporating more camera poses improves the estimation accuracy but with an increase in the computational resource requirement of the filter.

For improved performance of VINS on MAVs, the process model of the filter should be modified to address the aerodynamic rotor drag forces of the vehicle [18]. Recent studies showed performance improvement of VINS when the drag force model is used in the navigation filter [11, 19]. Performance improvements as a result of the drag force model are also reported in feedback controller design for aggressive flight of MAVs [20, 21]. However, under external disturbance and modeling errors such as wind and ground effects, it is necessary to fall back from the drag force models to the conventional models (kinematic models without the aerodynamic rotor drag forces) because the drag force models are invalid under external disturbance. This means that there should be a multiple model filtering technique in place to address these changing conditions i.e., transition between a conventional kinematic model VINS and a drag force model VINS to improve the estimation accuracy. Literature shows successful applications of such adaptive and hybrid estimation algorithms [22] in other domains, specifically the application of interacting multiple model (IMM) algorithm [23–27] that can adapt itself by updating the probability of each model to achieve improved estimation performance. However, implementing an IMM for VINS is not straight forward and has several challenges. This research work addresses the following three challenges related to IMM-VINS design.

First challenge: is related to the non-linearity while implementing the IMM for VINS. IMMs are generally applied to linear systems or system models with low order non-linearities. For instance, target tracking IMMs [28–30] and image tracking IMMs [31] deal with linear models in the multi-model filtering bank. Similarly, the ground vehicle models and steering geometry models used for IMM filters in [23,32,33] exhibit stable dynamics with non-linearities which are significantly different from spatial geometric and projective non-linearities present in VINS filters [15]. Furthermore, IMM is generally applied to filters that share similar state vectors and measurement models, whereas the conventional VINS [4] and drag force VINS [11] filters have a set of conceptually different state and measurement models. In order to address these drawbacks, this study performs, to the best of the author's knowledge, the first application of IMM for VINS filters on MAVs. This study develops an error-state implementation of the IMM algorithm incorporating geometrically consistent error definitions, which allows addressing non-linearities of VINS filters in the IMM algorithm. Moreover, methods available to address dissimilar states and maintaining consistent model probabilities are adopted in the algorithm to enable IMM on VINS filters [34, 35].

Second challenge: is addressing computational complexity related to maintaining multiple models. IMM requires multiple filter instances running in parallel, i.e., computational complexity is $O(n)$ in the number of filters of the IMM filter bank. Hence the underlying filters should be computationally less demanding to begin with to keep the IMM tractable. VINS filters with a large number of past camera poses have update steps that are computationally quite demanding for embedded processors [15,36]. Therefore, using traditional optimization-based VINS or generic MSCKF based filters in the IMM algorithm is deemed inefficient for solving the multi-model filtering problem related to VINS. In order to maintain a comparable estimation accuracy while significantly improving the computational efficiency, this study utilizes an epipolar constraint measurement model [11,37] in the VINS filters, which significantly lowers computational demand due to having only two camera poses in the state vector. Compared to MSCKF which maintains up to 30 camera poses and performs 3D point reconstruction and null space decorrelation steps, geometric constraint-based methods such as epipolar measurement models [11, 37] are highly efficient. Recent work [16] has found significant improvement in execution time while having a small decrease in accuracy when using such geometric constraints for VINS, which makes it preferable for IMM implementations over generic VINS filters.

Third challenge: is addressing the observability consistency of VINS filters in the IMM bank. IMM is generally applied to fully observable systems; hence observability consistency is not an issue. However, VINS models have an unobservable space spanning four degrees of freedom [38]. As a result, observability consistency and algorithm stability should be established and verified when using VINS filters in IMMs. The literature reports limited observability results related to the drag force VINS models. Furthermore, observability results related to the epipolar measurement model are non-existent. This study develops on the result of [38] and establishes observability conditions and consistency rules for both drag force and conventional VINS that use epipolar measurement models. To the best of the author's knowledge, IMM application for VINS on MAVs, and observability analysis of drag force and epipolar VINS filters are not reported elsewhere in the literature.

1.2.2 Part 2: Multiple Model Control

Many research work has been conducted on the control problem of MAVs to improve their stabilization and tracking performance, i.e., the stability characteristics of the vehicle under various operating speeds, tracking error, and computational time [4]. Initial developments in MAV state-space and nonlinear control has resulted in several advanced quadrotor MAV control algorithms such as, nonlinear PID control [39], high gain feedback control [40], robust control [41, 42], adaptive control [43, 44], and feedback linearization and backstepping control [45, 46]. Despite their promising results and performance, often these controllers require laborious tuning in order to accomplish stable performance, which is quite challenging. Moreover, majority of those controllers cannot guarantee input or state constraints and demands careful selection of feasible trajectories for stable maneuvering.

On the other hand, the control problem of the quadrotor MAV could be formulated as an optimization problem with a predefined objective function, where the minimization of this function over a finite time horizon leads to the optimal input values. Moreover, inputs and state constraints can be included directly in the optimization problem, which is crucial for real systems with physical constraints. One such optimization-based control algorithm is called Model Predictive Control (MPC) or Receding Horizon Control since the prediction horizon keeps shifting over the process time [47, 48]. Nonlinear Model Predictive Control (NMPC) is an MPC variant that uses the nonlinear system model in its prediction [49]. NMPC showed promising stabilization and tracking performance in ground and aerial vehicles in the literature [50–52] and robust collision and obstacle avoidance performance [53].

The NMPC requires to iteratively solve on-line a non-convex, nonlinear, dynamic optimal control problem on a finite prediction horizon and only the first element of the computed control sequence is applied to the system [49]. This process is then repeated at the subsequent time instant. Hence NMPC demands more computational time, implementing such a controller in a multiple model controller framework in real-time needs to address several challenges as explained below.

First challenge: is related to the closed-loop stability of the control system. Generally, closed-loop stability of receding horizon problems is not ensured except for special cases under certain conditions. For instance, closed-loop stability of NMPC scheme with infinite horizon (prediction horizon tends to infinity) can be ensured under certain controllability assumptions (e.g., there exists a suitable upper bound on the optimal value function) [54]. However, the infinite horizon control problem is not usually practical for online implementations mainly due to the computational complexity which is heavily governed by the length of the prediction horizon and the number of the manipulated variables. As a result, relevant concern has been given to the formulation of finite receding horizon problems whose solutions provide stabilizing control. Researchers were able to ensure the closed-loop stability by adding terminal cost, terminal constraint, terminal set, or a combination of them to the optimization problem [55] (for a thorough review see [47]). However, imposing terminal cost requires the solution of the algebraic Riccati equation to calculate the terminal weight [56]. Imposing terminal constraints makes the implementation of the control problem difficult since it is hard to obtain a satisfying solution of nonlinear equality constraints in a limited computational time frame. Under those conditions, an incomplete optimization during the limited sampling time may affect the system stability due to the errors caused by the force termination of the optimization [57]. On the other hand, designing the finite receding horizon control problem without stabilizing terminal costs or constraints [58–60] is computationally less demanding and provides larger stability regions and better closed-loop performance [49]. In this study, we design a novel NMPC controller with a tailored running cost to asymptotically stabilize the quadrotor MAV while adopting the techniques proposed in [58–60] to perform the stability analysis.

Second challenge: is addressing the computational complexity of the finite receding horizon problem. The MPC with stabilizing terminal cost can be used in the control of MAVs since it is easier in the design than a controller with terminal constraints, however, the prediction horizon must be large enough to achieve the closed-loop stability [52]. It was proven in [61] that extending the prediction horizon of the finite receding horizon problem leads to a good approximation to the optimal solution of the infinite receding horizon problem. However, extending the prediction horizon increases the complexity of the optimization problem and demands a higher computational time to obtain the optimal solution [62]. On the other hand, most small-sized aerial vehicles carry low-cost computer resources and demand fast control actions to maintain stable and smooth flights. Intriguingly, MPC without stabilizing terminal costs or constraints ensures asymptotic stability for much shorter prediction horizons [60] while yielding an approximately optimal infinite-horizon performance [48]. Therefore, this study derives a growth bound on the proposed NMPC value function (without terminal costs or constraints) that can be used to determine the minimal stabilizing prediction horizon for the finite optimization problem of the MAV that guarantees asymptotic stability.

Third challenge: is related to the recursive feasibility of the solution of the optimal control problem for any given constraints. The feasibility of the infinite horizon problem may turn into a critical concern while implementing the receding horizon schemes because of using a finite horizon. Finite receding horizon may result in loss of feasibility [63] or lead to an unsolvable infinite horizon optimal control problem [56]. The recursive feasibility is automatically ensured by imposing terminal costs or constraints into the optimization problem or by using a long enough prediction horizon [49, 56]. However, using terminal conditions reduces the region of attraction, i.e., reduces the feasible operating region of the MPC scheme. On the other hand, MPC schemes without terminal conditions can yield large (even unbounded) feasibility and stability regions for finite prediction horizon [49]. Grune et al., [49, 64] have showed that the recursive feasibility of an NMPC scheme without terminal conditions can be inherited from the closed-loop stability where the finite horizon cost acts as a Lyapunov function. Therefore, the calculation of the stabilizing prediction horizon from the derived growth bound can ensure both asymptotic stability and recursive feasibility.

Fourth challenge: is addressing the changing dynamics of the MAV in different environments or operating conditions. Incorporating the rotor drag force of the MAV in the system model aims at improving the controller performance at aggressive and agile maneuvers with high speeds and accelerations [20]. However, during periods of environmental changes, e.g., wind disturbance or ground effect, this controller is prone to failure because it is designed and tuned to work properly with aerodynamic disturbances and fast speed trajectories while ignoring the other environmental disturbances. Additionally, the effect of the wind disturbance may become quite significant and jeopardize the vehicle when operating in close proximity to obstacles or other aerial vehicles [65]. Therefore, switching to a conventional controller without drag force would be more reliable to guarantee a safe flight since the conventional controller is designed and tuned to work in slow flights and environments with expected disturbances. In order to address this issue, this work develops a multiple nonlinear model predictive controllers scheme to address the changing dynamics of the MAV at different flight modes and environments. The multiple model controller will make use of the interacting multiple model algorithm of the state estimation to help in detecting the flight mode and then triggering the appropriate controller.

1.3 Research Objective and Contributions

The main objective of this research is to develop an adaptive autonomous trajectory control system for an MAV to operate under various and changing operating conditions. This includes the development of an accurate state estimation system and a robust and fast trajectory control system. Under these conditions, the research objectives are categorized below with the expected contributions highlighted. The proposed research here has both theoretical and experimental objectives.

Objective 1: Design of a computationally efficient error-state visual inertial navigation system (VINS) with improved accuracy and consistency for MAVs.

- Designing two error-state VINS filters with epipolar constraints; (1) conventional VINS and (2) drag force VINS.
- Performing the observability analysis of the drag force VINS filters and the epipolar constraint measurement models.
- Development of observability-constrained VINS to maintain the consistency of the filter.

Objective 2: Design of a novel interacting multiple model for VINS (IMM-VINS) that supports operation during periods with aggressive flights and/or external disturbances (e.g., wind).

- Development of the error-state formulation of IMM incorporating geometric error definitions and optimal quaternion averaging to support VINS applications,
- First application and design of IMM for VINS (IMM-VINS) by using a drag force VINS and conventional VINS model to support operation during periods with external disturbances.

- Numerical and experimental validations of the proposed IMM-VINS using Matlab simulator and the EuRoC dataset.

Objective 3: Design of a novel computationally efficient and highly stable nonlinear model predictive controller (NMPC) without terminal conditions for the control of MAVs with/without drag force incorporated in the system model.

- Proposing the design of a novel computationally-efficient NMPC scheme (by tailoring an objective function) with improved stability characteristics for the control of quadrotor MAVs without the use of terminal costs or constraints while ensuring recursive feasibility. The design provides a unique analytical methodology that requires minimal tuning parameters compared to other controllers in the literature.

- Performing the stability analysis required to prove the asymptotic stability of the proposed controller by deriving a growth bound on the proposed MPC value function.

- Calculating the minimal stabilizing prediction horizon, which effectively minimizes the computational cost, and providing a formula for the estimation of the performance of the closed-loop scheme.

- Providing detailed numerical and experimental validations of the proposed controller at different initial conditions, system configurations, and various trajectories; and comparing its robustness against the traditional NMPC schemes in the literature.

Objective 4: Design of a novel computational-efficient multiple-NMPC scheme that supports various operation and flight conditions.

- Designing a multiple model control scheme, depending on the IMM filter, that can effectively recognize the flight mode and then trigger the appropriate NMPC from the controller's bank that includes a drag-force model based NMPC and a conventional-model based NMPC to support operation during periods with aggressive flights and/or external disturbances (e.g., wind).

- Conducting numerical validation of the proposed system.

1.4 Organization of the Thesis

This thesis is organized as follows.

- **Chapter 1 - Introduction**: Introduces the research topics and thesis motivation, and states the main problems and challenges that will be addressed in the thesis. Additionally, it summarizes the thesis objectives and expected contributions.

- **Chapter 2 Literature Review**: Reviews and discusses the related studies of the considered control/estimation problems and their drawbacks.

- **Chapter 3 - Visual-inertial Navigation System**: This chapter relates to objective 1 of the thesis. It presents the design of an error-state visual-inertial navigation system, the nonlinear observability analysis of the VINS filter, and the formulation of the observability-constrained VINS with and without drag force.

- **Chapter 4 - Multiple Model State Estimation**: This chapter relates to objective 2 of the thesis. It presents the methodology proposed for the design of the multiple model state estimation. The overall system is validated using several numerical and experimental studies.

- **Chapter 5 - Nonlinear Model Predictive Control**: This chapter relates to objective 3 of the thesis. It presents the design of a nonlinear model predictive controller without terminal costs or constraints incorporated in the value function. Also, presents the stability analysis of the proposed controller and provides multiple numerical studies and lab experiments to validate the control algorithm.

- **Chapter 6 - Multiple Model Control**: This chapter relates to objective 4 of the thesis. It presents the design of a multiple model controller, where two nonlinear model predictive controllers are used in the controller/model bank and the interacting multiple model proposed in Chapter 4 is used in the switching module to trigger the appropriate controller at different operating conditions.

- **Chapter 7 - Summary and Future Work**: This chapter concludes the thesis and discusses the practicality of the proposed methods. Also, it presents the resulting publications of this work and the possible future research directions.

Chapter 2

Literature Review

In this chapter, we briefly discuss the related studies to the estimation and control of quadrotor MAVs and their drawbacks. Then, in the following chapters, we discuss how our proposed research can overcome these drawbacks and provide better performance.

2.1 Visual-Inertial Navigation Systems

VINS has been developed as an indoor equivalent for GPS inertial navigation to estimate the vehicle pose based on the fusion of onboard sensors and visual information [15, 38]. Optimization-based VINS approaches solve a nonlinear optimization problem over a set of measurements considering a moving window [66]. Optimization-based VINS has improved estimation accuracy and solution robustness but demands considerable computational power [66]. On the other hand, filtering-based VINS approaches are more computationally efficient because they follow a simplified gain correction step while marginalizing all past measurements using a handful of states corresponding to past camera poses of the platform [15]. Recent work in [17] comparatively evaluates the performance of popular VINS algorithms with a detailed analysis of the computational resource requirement of each algorithm. The work identifies filtering-based approaches as a good compromise considering computational demand and accuracy of the algorithm for resource-constrained vehicles like MAVs [17]. Other work has also been conducted to implement the visual measurement model to incorporate geometric constraints without having 3D feature point reconstruction in the methodology as in [11, 16, 37].

Many VINS filter formulations [15, 16, 36] have ignored aerodynamics effects while modeling VINS for MAVs. This practice is common in MAV controller design as well because stability at hovering is the main design objective [67]. While the MAV is in hover, the velocity is too small compared to other dynamics terms that can be neglected, and these simplistic assumptions may be valid. However, at fast or aggressive maneuvers, the ignored aerodynamics could significantly affect the MAV controller performance [18]. Rotor drag is the main aerodynamics effect that highly influences tracking errors of controllers [20]. Rotor drag forces introduce dynamic constraints to the estimator models, which can be exploited to improve state estimation capability of the filter [68]. Work in [69] used the rotor drag model of MAVs to produce accurate velocity estimates from accelerometer measurements, [70] used rotor drag model for improved UWB localization of MAVs, and [11] extended the method to VINS. However, the reported dynamic models only provide better estimates under certain conditions, i.e., under low external disturbance and aggressive flights of MAVs [11, 20]. Therefore, IMM filtering can be proposed as a framework to overcome those drawbacks by incorporating additional supporting filters in the framework [71, 72].

2.2 Interacting Multiple Model

IMM is an efficient estimation technique that has been developed to estimate the states of dynamic systems with different behavior modes, and it ensures better performance than individual filters [33,73]. The IMM algorithm can switch from one filter to another based on a posterior defined Markov transition probability for switching between models [71,72]. IMM has been used to address the maneuvering target tracking problem in different applications including ground, marine, and aerial vehicles [23–27], where IMM estimator had significant improvement in model estimation and tracking accuracy than stand-alone estimators. IMM was demonstrated to be one of the simplest and cost-effective solutions to handle mode changes in dynamic systems [74]. In [73], a Kalman filter failed to reduce the noises during non-maneuvering intervals because all the filter parameters were previously tuned for maneuvering intervals; however, the IMM performance was stable, and the error was almost constant during both intervals by setting-up two filters with adequate parameters for each interval. In [75] authors showed that multiple model algorithm ensures satisfactory performance than single-modeled filters in tracking applications when the target undergoes turns or maneuvers. IMM based estimation using nonlinear models was recently employed in automobile navigation systems, and the results showed superior ability of IMM to produce consistent location estimates, even under conditions where GPS is producing spurious measurements. Past work in [33, 76] showed applications of IMM for GPS navigation, and [77] has shown the use of Chi-squared tests to dynamically select the best sensors for the filter update. Work in [78] reports an application of the methods to MAVs where individual motor failure is found using a bank of models. Further improvements to IMM were proposed in [79], where selective re-initialization and state augmentation strategies were used to minimize the number of models used in fault detection.

IMM design and implementation for MAV VINS is a challenging open research field. It involves identification of models having complementary features such as accuracy, stability, robustness, and self-calibration. Hence the design requires a firm knowledge base of the underlying state estimators and strategies for computationally efficient implementation.

2.3 Observability-Constrained VINS

Nonlinear observability analysis establishes the locally weakly observable subspace of a nonlinear system model description. Lie derivative based observability rank condition criteria proposed in [80] is primarily employed for this analysis task. The method has been used in [81] for the analysis of simultaneous localization and mapping (SLAM), in [82] it was used for inter-robot localization, and [83] analyses relative localization with platform velocity constraints. Work in [84] showed that the observability properties of linearized systems used for the design of error-state filters do not match the observability of the true nonlinear system. The authors proved that the unobservable subspace of the linearized system is lower than the actual system, resulting in increased filter inconsistency due to the reduction of covariance estimates in the unobservable direction of the true system. The study has resulted in observability-constrained filter design rules for robotic navigation [81, 84]. Hesch et al [38, 85] have developed an observability-constrained estimator for VINS to enforce the unobservable directions of the system by modifying the system and measurement Jacobians to improve the system consistency. Results showed that standard VINS pseudo information was inserted into the filter through the unobservable directions that violate the observability properties of the system, while the observability-constrained VINS remains consistent with better position and orientation accuracy. The observability analysis of [38] used a basis function based approach over the more popular observability rank condition analysis [80]. The latter requires an exhaustive analysis of the VINS model. A recent development in [86] follows an invariant filter design approach to implicitly enforce observability consistency rules for VINS. Observability-constrained design is considered an important characteristic of a filter; however, the literature does not present consistent filter designs for drag force based VINS nor epipolar measurement models. Hence an IMM that incorporates rotor drag modeling and epipolar constraints should first establish the observability consistency rules for the VINS filter design.

2.4 Stability Analysis of Nonlinear Model Predictive Control

Pioneering work reported in [87, 88] introduced a novel method to design a finite optimal control problem for MPC without using the stabilizing terminal costs or constraints. This method adopts a Lyapunov function to represent the finite horizon value function without terminal conditions, which allows the stability to be implied by the monotonicity of the value function [88]. Grimm et al. [88] showed that the asymptotic stability of the closed-loop system can be guaranteed for a sufficiently long prediction horizon. However, this initial study did not provide explicit bounds for the length of the prediction horizon. A series of recent studies [58–60] has ensured the closed-loop asymptotic stability relying on the concept of cost controllability, i.e., local controllability condition (see [89,90] for thorough details). This was achieved through computing a growth function that bounds the MPC value function. This growth function can then be used to calculate a performance measure of the finite horizon scheme, i.e., the proximity to the infinite horizon problem. This in turn provided a way to estimate a stabilizing prediction horizon that guarantees a monotonically decreasing value function and closed-loop asymptotic stability. This particular feature allows the NMPC to be implemented at a faster rate for nonlinear robotic systems while ensuring closed-loop stability [49, Sec. 7.4].

Continued with this pioneering work, research work in [50, 91] effectively demonstrated a real-time implementation of this method to stabilize a ground robotic system to operate fast on a given trajectory. This ground robot uses the holonomic/non-holonomic kinematic model, which is generally open-loop stable. This ground robotic application constructed specific open-loop trajectories to derive the growth bounds on the proposed value function and then to verify the cost controllability condition. The proposed scheme was implemented successfully on a ground robot for a point stabilization problem. However, the method was not tested on trajectory tracking problems.

An early study in [92] has proposed the first application of NMPC for the position and heading tracking control of an aerial-type vehicle, e.g., rotorcraft-based unmanned aerial vehicle. This traditional NMPC demonstrated in this work was then adopted by many researchers and those research were able to demonstrate successful trajectory control of quadrotor MAV [52, 93, 94]. However, this traditional NMPC has incorporated terminal costs in the MPC objective function which caused those controllers to adopt a longer prediction horizon to achieve the expected stability of the system. In absence of a proper stability analysis, these methods require to adopt trial and error approach in order to obtain a prediction horizon value for satisfactory performance. This particular drawback generally prohibits the traditional NMPC based MAVs to operate at a higher frequency owing to its higher computational cost, as we have demonstrated in our experimental results in this work. A recent work in [95] has proposed a robust NMPC scheme for the visual servoing of quadrotors MAVs subject to external disturbances. The developed NMPC scheme incorporates terminal costs and constraints, which reduce the feasibility region of the closed-loop system, as shown in the recursive feasibility and stability analysis in [95]. The system was proven to be recursive feasible and stable only under some derived conditions, e.g., the system is locally Lipschitz continuous with a Lipschitz constant bounded by the maximum eigenvalues of the terminal weight matrix. The proposed controller was numerically able to stabilize the system within a small region around the desired position. However, in the experimental validation, it was only able to steer the quadrotor to the neighborhood of the desired position. This shortcoming might have happened due to excluding the terminal constraint from the optimization problem in the real-time implementation of the experiment. Another recent work in [96] explains an NMPC scheme without terminal constraints. However, the controller contained a feedback linearized system that requires the vehicle to be at an initial position closer to an equilibrium point in order for the system to be stable. This will result in a narrow stability region for the vehicle and make the flying envelope smaller. The control bounds are chosen based on this linearized model, and as a result, more tuning parameters are required to be adjusted for better performance. Additionally, the prediction length is quite high for stability, which makes the implementation of this controller quite difficult.

2.5 Multiple Model Control

Multiple model predictive control has been primarily proposed in the literature for process and medical applications with nonlinear systems or systems with changing dynamics. Some pioneering studies [97, 98] used a set of linear model/MPC pairs to handle all possible operating regions or conditions, where a recursive Bayes theorem was used to weight the outputs of the controllers in the controller bank based on their residual and probability of representing the plant at each iteration sample, as shown in Fig. 2.1. Similarly, the work in [99] proposed a switching function to switch among the MPCs in the controller bank using the Hotelling's T^2 statistic of

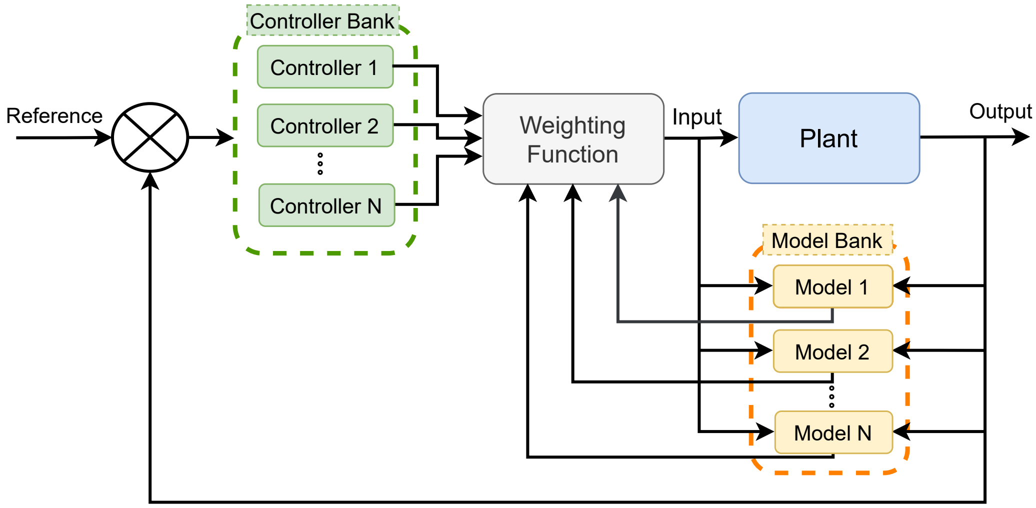


Fig. 2.1: Schematic of the multiple-MPC strategy with weighting function.

the measured output variables of each local MPC, as shown in Fig. 2.2. The results of these reported studies showed better performance compared to a single controller. However, this technique is not efficient for robotics applications because it requires all the controllers to be active and run simultaneously at the same time, which is computationally inefficient due to the limited computation resources of mobile robots and the necessitate of fast actions.

Alternatively, other reported studies [100–102] proposed the use of a single constrained linear MPC, for the sake of computational complexity reduction, and a weighted model bank as a prediction model, as shown in Fig. 2.3. The model bank includes multiple linearized models at different equilibrium points while the Bayesian weighting approach has been employed to weight the models' predictions. The more models incorporated in the model bank, the more accuracy of nonlinear system approximation. However, determining the optimal number of models that encompass the plant behavior is far from trivial. Also, tuning the single-MPC parameters to perform robustly in the possible range of prediction models stemming from the model bank is very difficult.

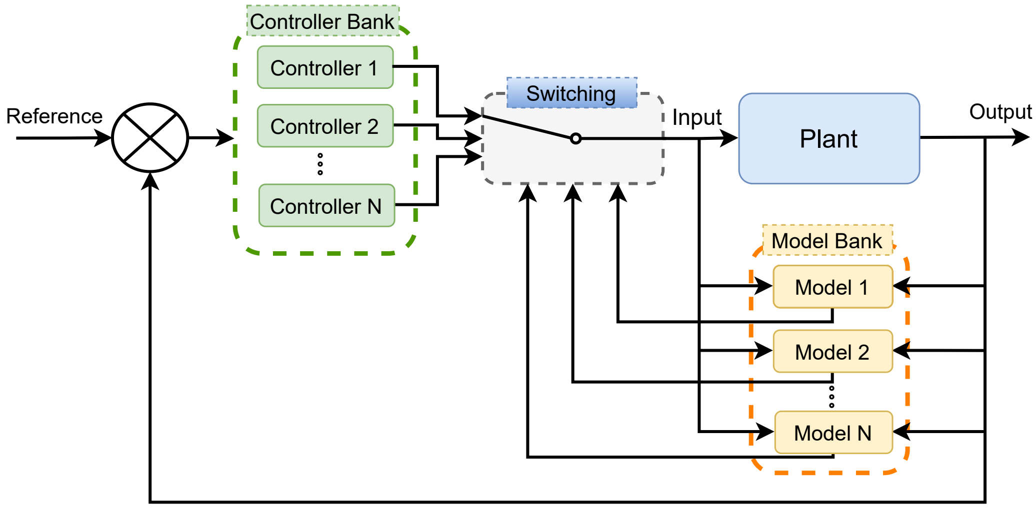


Fig. 2.2: Schematic of the multiple-MPC strategy with switching function.

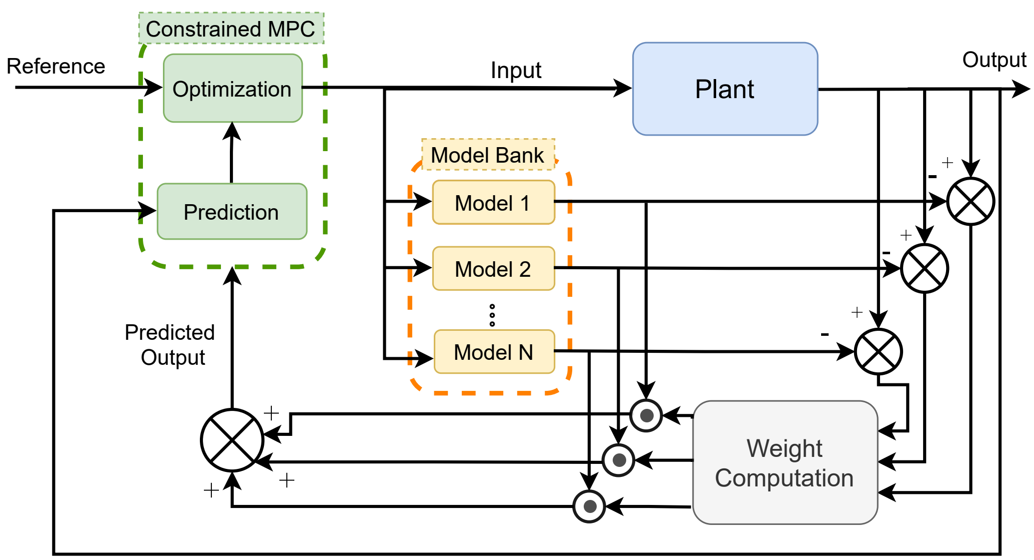


Fig. 2.3: Schematic of the multiple-MPC strategy with one constrained MPC.

The work in [103] has proposed a switching multiple linear model predictive attitude controller for the control of MAVs, where the switching is carried out using basic rules according to the current regime of the state vector. The drawbacks of the previous method still exist, where the selection of the optimal number of models and appropriate linearization operating points are still challenging.

Therefore, using a computationally efficient NMPC would be more efficient since it is rich enough to represent the nonlinear behavior of the MAV. Additionally, making use of the IMM algorithm in the switching function of the multiple model control algorithm (multiple-NMPC) would be computationally efficient since it requires only one active controller at a time.

Chapter 3

Visual-inertial Navigation System

In this chapter, we are designing a visual-inertial navigation system for the MAV. The design includes two different models, the first model uses the conventional kinematic model of the platform (INS mechanization equations) as proposed in [15], and the second model uses dynamic constraints that incorporate the drag force as proposed in [11]. Finally, we perform the observability analysis for both models and develop the observability-constrained VINS to maintain the consistency of the filter.

3.1 Design of VINS Filter

3.1.1 System description

The position and orientation of an MAV can be defined relative to the body frame $\{B\}$ attached to its center of gravity, and the global inertial frame $\{G\}$. The MAV is equipped with an IMU located at coordinate frame $\{I\}$ and a forward-facing monocular camera located at coordinate frame $\{C\}$, as shown in Fig 3.1. For simplicity of derivation, we assume that the IMU and body frames are aligned, which will be relaxed later in the next chapter when validating the filter for experimental data (Eu-

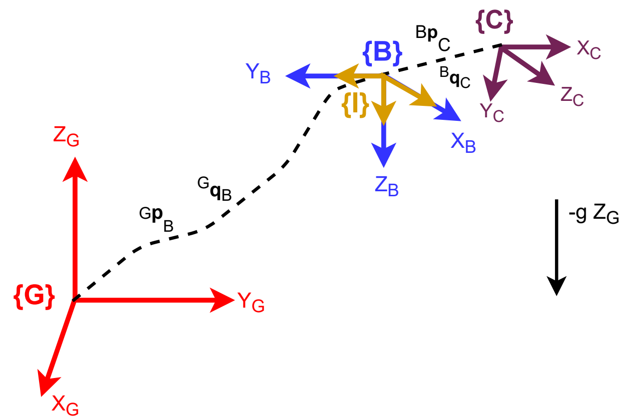


Fig. 3.1: Coordination systems related to VINS on MAV. $\{G\}$ is the global frame, $\{B\}$ is the body frame located at the center of gravity which is the same as the IMU frame $\{I\}$, and $\{C\}$ is the camera frame.

roC [104]). The intrinsic and extrinsic parameters are assumed to be known following an IMU-camera calibration procedure [105].

The nonlinear state space model of the system is given as,

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}, \mathbf{n}_w) \\ \mathbf{y} &= h(\mathbf{x}, \mathbf{n}_\nu) \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is the system states vector, $\mathbf{u}$ is the input vector, $\mathbf{n}_w$ is the process noise vector which is assumed zero-mean Gaussian, $\mathbf{y}$ is the measurement vector, and $\mathbf{n}_\nu$ is the measurements noise vector also assumed zero-mean Gaussian. The state vector of the system has a dimension of 23 and is defined as follows,

$$\mathbf{x} = \begin{bmatrix} {}^G\mathbf{p}_B{}^T & {}^B\mathbf{q}_G{}^T & {}^B\mathbf{v}^T & \mathbf{b}_a{}^T & \mathbf{b}_g{}^T & {}^G\acute{\mathbf{p}}_B{}^T & {}^B\acute{\mathbf{q}}_G{}^T \end{bmatrix}^T .$$

The state vector includes the position of the MAV with respect to the global frame ${}^G\mathbf{p}_B$, unit quaternion ${}^B\mathbf{q}_G$ corresponding to the MAV rotation from $\{G\}$ to $\{B\}$, MAV velocity vector ${}^B\mathbf{v}$ of $\{B\}$ relative to $\{G\}$ expressed in $\{B\}$, accelerometer bias $\mathbf{b}_a$, and gyroscope bias $\mathbf{b}_g$ expressed in $\{B\}$. This work uses a keyframe-based approach to handle visual measurements similar to [11, 13]. As a result, the position and orientation of the MAV corresponding to the previous keyframe are also stored in the state vector, which are denoted using ${}^G\acute{\mathbf{p}}_B$ and ${}^B\acute{\mathbf{q}}_G$, respectively. The error-state vector corresponds to the geometric difference between the true state vector $\mathbf{x}$ and the estimated state vector $\hat{\mathbf{x}}$. In this thesis, we are using $\hat{}$ and $\tilde{}$ above variables to denote estimated and error variables, respectively. The error-state is defined as, $\tilde{\mathbf{x}} = \mathbf{x} \ominus \hat{\mathbf{x}}$, where the inverse mapping $\ominus$ is used to capture geometrically consistent error terms similar to work in [86]. This inverse mapping is the same as standard subtraction in case of position, velocity, and bias states, but is different for quaternions. The quaternion error-state is computed as, $\tilde{\mathbf{q}} = \mathbf{q} * \hat{\mathbf{q}}^{-1} \xrightarrow{\text{Linearize}} \delta\mathbf{q} = \begin{pmatrix} 1 & \frac{1}{2}\delta\boldsymbol{\theta}^T \end{pmatrix}^T$, where $\delta\boldsymbol{\theta}$ is a small angle approximation of rotation and $*$ denotes the quaternion multiplication.

First order linearization of the continuous error-state dynamics $\dot{\tilde{\mathbf{x}}}$ results in,

$$\dot{\tilde{\mathbf{x}}} \xrightarrow{\text{Linearize}} \delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{G}_w\delta\mathbf{n}_w \tag{3.2}$$

where $\mathbf{F}$ and $\mathbf{G}_w$ are the process model and noise Jacobians and $\delta\mathbf{n}_w$ is the process noise vector. Mathematical descriptions corresponding to the two models considered for the IMM in this study are introduced in the following subsections.

3.1.2 Mathematical model of the conventional VINS

The conventional VINS (C-VINS) process model [15] corresponds to the INS mechanization equations given by,

$$
\begin{aligned}
{}^{G}\dot{\mathbf{p}}_B &= R({}^{B}\mathbf{q}_G)^T \; {}^{B}\mathbf{v} \\
{}^{B}\dot{\mathbf{q}}_G &= -\frac{1}{2} \; {}^{B}\boldsymbol{\omega} * {}^{B}\mathbf{q}_G \\
{}^{B}\dot{\mathbf{v}} &= R({}^{B}\mathbf{q}_G)^{G} g \bar{\mathbf{e}}_{\mathbf{3}} + {}^{B}\mathbf{a} - {}^{B}\boldsymbol{\omega} \times {}^{B}\mathbf{v} \\
{}^{G}\dot{\mathbf{p}}_B &= \mathbf{0}_{3\times1} \quad , \quad {}^{B}\dot{\mathbf{q}}_G = \mathbf{0}_{3\times1}
\end{aligned}
\tag{3.3}
$$

where the platform angular velocity input ${}^{B}\boldsymbol{\omega}$ and acceleration input ${}^{B}\mathbf{a}$ are driven using measurements from an inertial measurement unit given by,

$$
\begin{aligned}
\boldsymbol{\omega}_m &= ({}^{B}\boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g) \\
\mathbf{a}_m &= ({}^{B}\mathbf{a} + \mathbf{b}_a + \mathbf{n}_a) \\
\dot{\mathbf{b}}_a &= \mathbf{n}_{ba} \quad , \quad \dot{\mathbf{b}}_g = \mathbf{n}_{bg}.
\end{aligned}
\tag{3.4}
$$

In (3.3,3.4), $R({}^{B}\mathbf{q}_G)$ is the rotation matrix from frame $\{G\}$ to $\{B\}$, ${}^{G}g$ is the gravitational acceleration expressed in $\{G\}$, the standard basis $\bar{\mathbf{e}}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, $\boldsymbol{\omega}_m$ is the gyroscope measurements vector, and $\mathbf{a}_m$ is the accelerometer measurements vector. $\mathbf{n}_a$, $\mathbf{n}_g$, $\mathbf{n}_{ba}$, $\mathbf{n}_{bg}$ are stochastic Gaussian noise variables of the accelerometer measurement, gyroscope measurement, accelerometer bias random walk, and gyroscope bias random walk, respectively. The process noise vector given in (3.2) is defined as $\mathbf{n}_w = \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_a^T & \mathbf{n}_{ba}^T & \mathbf{n}_{bg}^T \end{bmatrix}^T$. The pose stored for the previous keyframe has zero dynamics since it does not change with time. The $\mathbf{F}$ and $\mathbf{G}_w$ matrices corresponding to this model can be found as,

$$
\mathbf{F} = \begin{pmatrix}
\mathbf{0}_3 & -R(\hat{\mathbf{q}})^T \left[{}^{B}\hat{\mathbf{v}}\right]_\times & R(\hat{\mathbf{q}})^T & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\
\mathbf{0}_3 & -\left[\boldsymbol{\omega}\right]_\times & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_{3\times6} \\
\mathbf{0}_3 & \left[R(\hat{\mathbf{q}}) \; {}^{G}g\bar{\mathbf{e}}_{\mathbf{3}}\right]_\times & -\left[\boldsymbol{\omega}\right]_\times & -\mathbf{I}_3 & -\left[{}^{B}\hat{\mathbf{v}}\right]_\times & \mathbf{0}_{3\times6} \\
\mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times6}
\end{pmatrix}
$$

$$
\mathbf{G}_w = \begin{pmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\ -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\ -\left[{}^{B}\hat{\mathbf{v}}\right]_\times & -\mathbf{I}_3 & \mathbf{0}_{3\times6} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{I}_6 \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_6 \end{pmatrix}
$$

where, $[\cdot]_\times$ is the skew symmetric matrix operator. For the sake of brevity, $R(\hat{\mathbf{q}})$ is the same as $R({}^{B}\hat{\mathbf{q}}_G)$, $\mathbf{I}_i$ is an $i \times i$ identity matrix, $\mathbf{0}_i$ is an $i \times i$ zero matrix, $\mathbf{0}_{i\times j}$ is an $i \times j$ zero matrix, and $\boldsymbol{\omega} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g$.

We consider visual measurements from one forward-facing monocular camera. Corresponding features between each pair of consecutive images are extracted using features detection and features matching techniques. These matched features are used to construct the visual measurement residual using the epipolar geometry constraints as follows,

$$
\begin{aligned}
\tilde{\mathbf{y}}_v = \tilde{h}(\mathbf{x}, \mathbf{p}_i, \acute{\mathbf{p}}_i) = (\mathbf{p}_i)^T \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \acute{\mathbf{p}}_i \\
\mathbf{E} = R({}^{C}\mathbf{q}_B) R({}^{B}\mathbf{q}_G) \left[{}^{G}\acute{\mathbf{p}}_C - {}^{G}\mathbf{p}_C\right]_\times R({}^{B}\acute{\mathbf{q}}_G)^T R({}^{C}\mathbf{q}_B)^T
\end{aligned} \tag{3.5}
$$

where $\tilde{\mathbf{y}}_v$ is the visual measurement residual for one pair of corresponding features, $\mathbf{p}_i$ is the image coordinates corresponding to feature point i of the current image, $\acute{\mathbf{p}}_i$ is the image coordinates corresponding to feature point i of the previous image, $\mathbf{E}$ is the essential matrix, $R({}^{C}\mathbf{q}_B)$ is the rotation matrix from frame $\{B\}$ to $\{C\}$, $\mathbf{K}$ is the camera intrinsic matrix, and ${}^{G}\mathbf{p}_C$ and ${}^{G}\acute{\mathbf{p}}_C$ are the positions of $\{C\}$ with respect to $\{G\}$ of the current and previous poses, respectively.

The linearized error-state visual measurement model is formulated as, $\delta\mathbf{y}_v = \mathbf{H}_v\delta\mathbf{x} + \mathbf{G}_v\delta\mathbf{n}_v$, where $\mathbf{H}_v$ and $\mathbf{G}_v$ are the Jacobian measurement and noise matrices of the visual measurement model given in (3.5). $\delta\mathbf{n}_v = \begin{pmatrix} \delta\mathbf{p}_i & \delta\acute{\mathbf{p}}_i \end{pmatrix}^T$ is the camera measurement noise vector in pixels, where $\delta\mathbf{p}_i$ and $\delta\acute{\mathbf{p}}_i$ are the camera noise vectors in pixels of feature points in the current and previous keyframes, respectively. The matrices $\mathbf{H}_v$ and $\mathbf{G}_v$ are defined as,

$$\mathbf{H}_v = \begin{pmatrix} \mathbf{p}_i^T \mathbf{A} R({}^B\hat{\mathbf{q}}_G)[R({}^B\hat{\hat{\mathbf{q}}}_G)^T \mathbf{B}\ \acute{\mathbf{p}}_i]_\times \\ \mathbf{H}_{v2} \\ \mathbf{0}_{9\times 1} \\ -\mathbf{p}_i^T \mathbf{A} R({}^B\hat{\mathbf{q}}_G)[R({}^B\hat{\hat{\mathbf{q}}}_G)^T \mathbf{B}\ \acute{\mathbf{p}}_i]_\times \\ \mathbf{H}_{v7} \end{pmatrix}^T \tag{3.6}$$

$$\mathbf{G}_v = \begin{pmatrix} \acute{\mathbf{p}}_i^T \mathbf{B}^T R({}^B\hat{\hat{\mathbf{q}}}_G)[\mathbf{C}]_\times^T\ R({}^B\hat{\mathbf{q}}_G)^T \mathbf{A}^T \\ \mathbf{p}_i^T \mathbf{A} R({}^B\hat{\mathbf{q}}_G)[\mathbf{C}]_\times\ R({}^B\hat{\hat{\mathbf{q}}}_G)^T \mathbf{B} \end{pmatrix}^T$$

where $\mathbf{H}_{v2}$, $\mathbf{H}_{v7}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are defined as follows.

$$\begin{aligned} \mathbf{H}_{v2} = -\mathbf{p}_i^T \mathbf{A} R({}^B\hat{\mathbf{q}}_G)[R({}^B\hat{\hat{\mathbf{q}}}_G)^T \mathbf{B}\ \acute{\mathbf{p}}_i]_\times R({}^B\hat{\mathbf{q}}_G)^T [{}^B\mathbf{p}_C]_\times - \\ \acute{\mathbf{p}}_i^T \mathbf{B}^T R({}^B\hat{\hat{\mathbf{q}}}_G)[\mathbf{C}]_\times^T R({}^B\hat{\mathbf{q}}_G)^T [\mathbf{A}^T\ \mathbf{p}_i]_\times \end{aligned}$$

$$\begin{aligned} \mathbf{H}_{v7} = \mathbf{p}_i^T \mathbf{A} R({}^B\hat{\mathbf{q}}_G)[R({}^B\hat{\hat{\mathbf{q}}}_G)^T\ \mathbf{B}\ \acute{\mathbf{p}}_i]_\times R({}^B\hat{\hat{\mathbf{q}}}_G)^T [{}^B\mathbf{p}_C]_\times - \\ \mathbf{p}_i^T \mathbf{A} R({}^B\hat{\mathbf{q}}_G)[\mathbf{C}]_\times R({}^B\hat{\hat{\mathbf{q}}}_G)^T [\mathbf{B}\ \acute{\mathbf{p}}_i]_\times \end{aligned}$$

$$\mathbf{A} = \mathbf{K}^{-T} R({}^C\mathbf{q}_B) \quad , \quad \mathbf{B} = R({}^C\mathbf{q}_B)^T \mathbf{K}^{-1}$$

$$\mathbf{C} = {}^G\hat{\hat{\mathbf{p}}}_B - {}^G\hat{\mathbf{p}}_B + \left(R({}^B\hat{\hat{\mathbf{q}}}_G)^T - R({}^B\hat{\mathbf{q}}_G)^T\right)\ {}^B\mathbf{p}_C.$$

3.1.3 Mathematical model of the Drag-force VINS

The dynamic model of the Drag-force VINS (DF-VINS) includes the forces applied on the MAV, including the total thrust, the weight, and the aerodynamic rotor drag force. The continuous dynamics of DF-VINS process model is given by,

$$
\begin{aligned}
{}^{G}\dot{\mathbf{p}}_{B} &= R({}^{G}\mathbf{q}_{B})^{T}\ {}^{B}\mathbf{v} \\
{}^{B}\dot{\mathbf{q}}_{G} &= -\frac{1}{2}\ (\boldsymbol{\omega}_{m} - \mathbf{b}_{g} + \mathbf{n}_{g}) * {}^{B}\mathbf{q}_{G} \\
{}^{B}\dot{\mathbf{v}} &= R({}^{G}\mathbf{q}_{B})^{G} g\bar{\mathbf{e}}_{\mathbf{3}} - \bar{\mathbf{D}}_{L}{}^{B}\mathbf{v} + \mathbf{f}_{ip} + \mathbf{n}_{m} \\
\dot{\mathbf{b}}_{a} = \mathbf{n}_{ba}\ ,\ \dot{\mathbf{b}}_{g} &= \mathbf{n}_{bg}\ ,\ {}^{G}\dot{\mathbf{p}}_{B} = \mathbf{0}_{3\times 1}\ ,\ {}^{G}\dot{\mathbf{q}}_{B} = \mathbf{0}_{3\times 1}
\end{aligned}
\tag{3.7}
$$

where $\mathbf{f}_{ip}$ and $\bar{\mathbf{D}}_{L}$ are defined as follows,

$$
\begin{aligned}
\mathbf{f}_{ip} &= \bar{T}_{L}\bar{\mathbf{e}}_{3} - (\boldsymbol{\omega}_{m} - \mathbf{b}_{g} + \mathbf{n}_{g}) \times {}^{B}\mathbf{v} \\
\bar{T}_{L} &= a_{mz} - b_{az},\ \bar{\mathbf{D}}_{L} = diag(k_{1_x},\ k_{1_y},\ k_{1_z}).
\end{aligned}
$$

The random Gaussian noise in the MAV drag force model is denoted by $\mathbf{n}_{m}$, a_{mz} is the accelerometer measurement in z-direction, and b_{az} is the accelerometer bias in z-direction. The mass normalized thrust is denoted by $\bar{T}_{L}$. The mass normalized drag parameters matrix $\bar{\mathbf{D}}_{L}$ is a diagonal matrix with elements $k_{1_x}, k_{1_y}, k_{1_z}$ and is crucial for estimation accuracy of the DF-VINS. The drag parameters in x-axis and y-axis can be estimated following a least squared optimization procedure while it is reasonable to assume that the drag parameter in z-axis to be zero similar to work in [20]. The system model noise vector given in (3.2) is defined as $\mathbf{n}_{w} = \begin{bmatrix}\mathbf{n}_{g}^{T} & \mathbf{n}_{m}^{T} & \mathbf{n}_{ba}^{T} & \mathbf{n}_{bg}^{T}\end{bmatrix}^{T}$. The filtering matrices $\mathbf{F}$ and $\mathbf{G}_{w}$ corresponding to the DF-VINS are defined as,

$$
\mathbf{F} = \begin{pmatrix}
\mathbf{0}_{3} & -R(\hat{\mathbf{q}})^{T}\left[{}^{B}\hat{\mathbf{v}}\right]_{\times} & R(\hat{\mathbf{q}})^{T} & \mathbf{0}_{3} & \mathbf{0}_{3} & \mathbf{0}_{3\times 6} \\
\mathbf{0}_{3} & -\left[\boldsymbol{\omega}\right]_{\times} & \mathbf{0}_{3} & \mathbf{0}_{3} & -\mathbf{I}_{3} & \mathbf{0}_{3\times 6} \\
\mathbf{0}_{3} & \left[R(\hat{\mathbf{q}})\ {}^{G}g\bar{\mathbf{e}}_{\mathbf{3}}\right]_{\times} & -\bar{D}_{L} - \left[\boldsymbol{\omega}\right]_{\times} & -\bar{\mathbf{e}}_{\mathbf{3}}\bar{\mathbf{e}}_{\mathbf{3}}^{T} & -\left[{}^{B}\hat{\mathbf{v}}\right]_{\times} & \mathbf{0}_{3\times 6} \\
\mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 6}
\end{pmatrix}
$$

$$
\mathbf{G}_w = \begin{pmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\ -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\ -\left[{}^B\hat{\mathbf{v}}\right]_\times & \mathbf{I}_3 & \mathbf{0}_{3\times6} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{I}_6 \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_6 \end{pmatrix}
$$

As demonstrated later in sections 3.2.1.2, the accelerometer bias ($\mathbf{b}_a$) of the DF-VINS state vector will be updated to include a bias term that captures the thrust of the MAV. This modification simplifies filter design when body frame and the IMU frame are not aligned, and also simplifies the observability analysis shown in Section 3.2.1.2. As a result of these modifications, the state vectors of C-VINS and DF-VINS have dissimilar definitions.

Two measurement models are considered for the DF-VINS model; the first one is the inertial measurement model that contains the accelerometer measurements along x and y directions.

$$
\mathbf{h}_a = \Upsilon(-\bar{\mathbf{D}}_L\ {}^B\mathbf{v} + \mathbf{b}_a + \mathbf{n}_a) \tag{3.8}
$$

where $\Upsilon = \begin{bmatrix} \bar{\mathbf{e}}_1^T \ ; \ \bar{\mathbf{e}}_2^T \end{bmatrix}$ is a 2×3 matrix used to extract the first two rows, and $\bar{\mathbf{e}}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ and $\bar{\mathbf{e}}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$ are the first and second standard basis vectors. The linearized residual measurement model is given by, $\delta\mathbf{y}_i = \mathbf{H}_i\delta\mathbf{x} + \mathbf{G}_i\delta\mathbf{n}_i$, where $\mathbf{H}_i$ and $\mathbf{G}_i$ are the measurement and noise Jacobian matrices of the inertial measurement model given in (3.8), and $\mathbf{n}_i$ is the accelerometer measurement noise vector along x-axis and y-axis and defined as, $\mathbf{n}_i = \begin{pmatrix} n_{ax} & n_{ay} \end{pmatrix}^T$. The filtering matrices $\mathbf{H}_i$ and $\mathbf{G}_i$ are defined as,

$$
\mathbf{H}_i = \begin{pmatrix} \mathbf{0}_{1\times6} & -k_{1_x}\bar{\mathbf{e}}_1^T & \bar{\mathbf{e}}_1^T & \mathbf{0}_{1\times9} \\ \mathbf{0}_{1\times6} & -k_{1_y}\bar{\mathbf{e}}_2^T & \bar{\mathbf{e}}_2^T & \mathbf{0}_{1\times9} \end{pmatrix} \ , \quad \mathbf{G}_i = \mathbf{I}_2 \tag{3.9}
$$

The second measurement model for the DFVIS is identical to the visual measurement model of C-VINS given in (3.5).

3.1.4 Error-state Kalman filtering

The filter propagation and update follow an error-state filtering formulation as presented in *Algorithm 1- The general continuous-discrete EKF* of [86]. The implementation of the discrete filter prediction and update is introduced here.

3.1.4.1 Filter prediction

For digital implementation, the process model in (3.2) has been discretized after setting the system and measurement noises to their expected value of zero. After a new IMU reading is received, the estimated state vector $\hat{\mathbf{x}}$ of both filters is propagated using a 4^{th} order Runge-Kutta numerical integration of (3.3) and (3.7). The discrete-time state transition matrix and covariance prediction are implemented as,

$$\begin{aligned} \mathbf{\Phi}_k &= e^{\left(\int_0^{\Delta T}\mathbf{F}(\tau)d\tau\right)} \approx e^{\mathbf{F}\Delta T} \\ \mathbf{P}_{k+1|k} &= \mathbf{\Phi}_k \mathbf{P}_{k|k}\ {\mathbf{\Phi}_k}^T + \mathbf{Q}_k \\ \mathbf{Q}_k = \mathbf{G}_k\mathbf{Q}_w{\mathbf{G}_k}^T, \mathbf{G}_k &= \int_0^{\Delta T} e^{\mathbf{F}(\Delta T-\tau)}\mathbf{G}_w\ d\tau \end{aligned} \tag{3.10}$$

where ΔT is the sampling time, $\mathbf{Q}_k$ is the discrete-time system noise covariance matrix, $\mathbf{Q}_w = \mathbb{E}\left(\mathbf{n}_w\mathbf{n}_w^T\right)$ is the the continuous-time system noise covariance matrix, and k is the time index.

3.1.4.2 Filter update mechanism

The DF-VINS filter has two types of measurement updates, (1) inertial measurement update that happens at IMU measurement rate, e.g., 200 Hz for the EuRoC dataset, and (2) visual measurement update that triggers at image acquisition rate, e.g., 20 Hz for the EuRoC dataset. The C-VINS filter only uses the visual measurement update. The execution diagram of both filters is illustrated in Fig. 3.2. The visual update is only executed at each new keyframe registered when there is sufficient feature disparity between two images (assuming static world). The use of keyframes avoids drifting of the filter when the MAV is stationary because when the camera is stationary the pixel locations of corresponding points do not carry information to estimate an essential matrix, i.e., the information related to the camera poses up to scale is not present. As a result, the epipolar constraints do not contain any useful information about the estimated state vector $\hat{\mathbf{x}}$ [11]. Therefore, turning the visual update off minimizes the risk of inconsistent updates leading to divergence. The feature disparity f_d between two corresponding sets of features is calculated as proposed in [11] as follows,

$$f_d = \frac{1}{n_f} \sum_{i=1}^{n_f} \|\mathbf{p}_i - \acute{\mathbf{p}}_i\| \tag{3.11}$$

where n_f is the number of matched features between the two images. When f_d exceeds the set threshold, a new keyframe is registered then a Kalman correction step is performed as, $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} \oplus \mathbf{K}_k \tilde{\mathbf{y}}$, where $\hat{\mathbf{x}}_{k+1}$ is the posterior estimate, $\hat{\mathbf{x}}_{k+1|k}$ is the prior estimate, $\mathbf{K}_k$ is the Kalman gain, and $\tilde{\mathbf{y}}$ is the measurement residual. The operation $\oplus$ defines retraction used for geometrically consistent mapping of corrections $\mathbf{K}_k\tilde{\mathbf{y}}$ to state estimates $\hat{\mathbf{x}}$ as in [86]. The operator $\oplus$ is the standard vector addition in case of position, velocity, and bias states but is different for quaternions which is

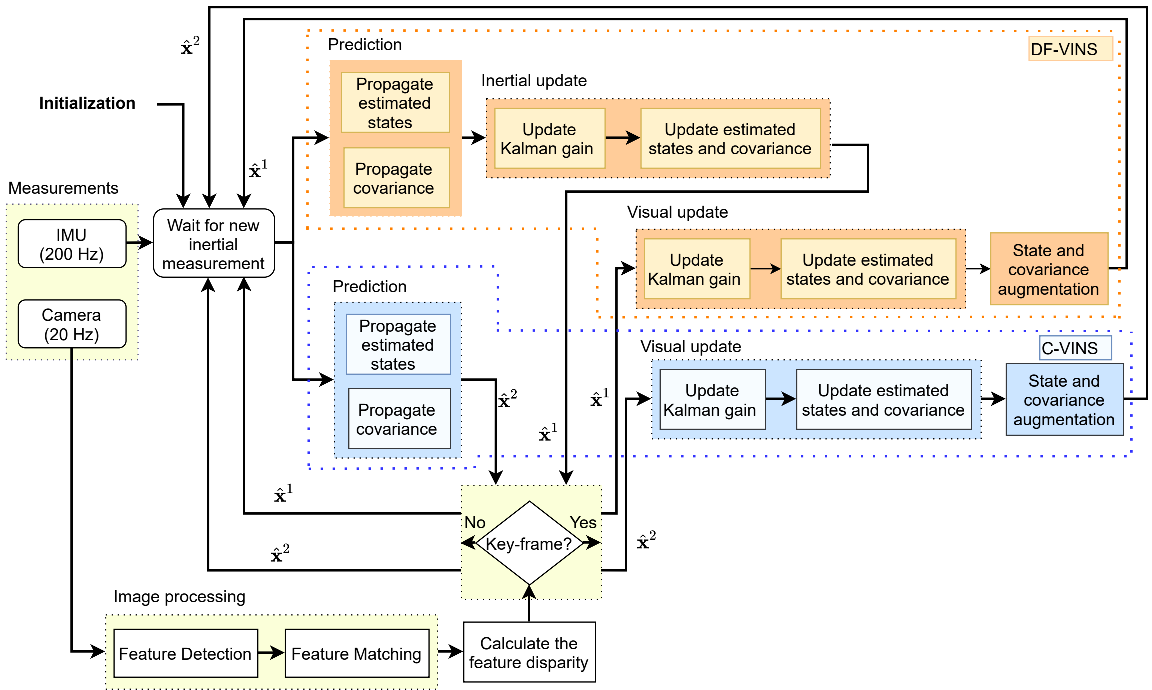


Fig. 3.2: Kalman filter architecture for the proposed two filters, DF-VINS in orange and C-VINS in blue, that will be used in the IMM filtering bank. The block diagram illustrates the prediction and update steps for each filter.

defined such that,

$$
{}^{B}\mathbf{q}_G = \begin{pmatrix} cos\frac{|\delta\boldsymbol{\theta}|}{2} \\ \frac{\delta\boldsymbol{\theta}}{|\delta\boldsymbol{\theta}|} sin\frac{|\delta\boldsymbol{\theta}|}{2} \end{pmatrix} * {}^{B}\hat{\mathbf{q}}_G \quad , \quad \delta\boldsymbol{\theta} = \mathbf{K}_q \tilde{\mathbf{y}} \tag{3.12}
$$

where $\mathbf{K}_q$ is the Kalman gain matrix associated with the orientation state. The measurement noise covariance matrices of the filter $\mathbf{R}_i$ and $\mathbf{R}_v$ are given as,

$$
\mathbf{R}_{i_k} = \mathbf{G}_i \mathbf{R}_i \mathbf{G}_i{}^T \quad , \quad \mathbf{R}_{v_k} = \mathbf{G}_v \mathbf{R}_v \mathbf{G}_v{}^T \tag{3.13}
$$

where $\mathbf{R}_i = \mathbb{E}\left(\mathbf{n}_i \mathbf{n}_i^T\right)$ is the inertial measurement noise covariance matrix and $\mathbf{R}_v = \mathbb{E}\left(\mathbf{n}_v \mathbf{n}_v^T\right)$ is the visual measurement noise covariance matrix, both are assumed to be known from sensor specifications or a calibration procedure.

3.1.4.3 State and covariance augmentation

After the measurement correction step is implemented, the state vector and covariance matrices are augmented to update the previous pose information. The state augmentation is simply implemented by storing the current position and orientation of the MAV in the previous pose, i.e., ${}^G\dot{\hat{\grave{\mathbf{P}}}}_B = {}^G\dot{\hat{\mathbf{P}}}_B$ and ${}^B\dot{\hat{\grave{\mathbf{q}}}}_G = {}^B\dot{\hat{\mathbf{q}}}_G$. The covariance augmentation is implemented as,

$$\mathbf{P}^+ = \begin{pmatrix} \mathbf{I}_{15\times 21} \\ \mathbf{J} \end{pmatrix} \mathbf{P}^- \begin{pmatrix} \mathbf{I}_{15\times 21} \\ \mathbf{J} \end{pmatrix}^T, \quad \mathbf{J} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 15} \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 15} \end{pmatrix}$$

where $\mathbf{P}^-$ and $\mathbf{P}^+$ are the covariance matrices before and after the augmentation, respectively, and $\mathbf{I}_{15\times 21}$ is a 15×21 identity matrix.

3.2 Design of Observability-constrained VINS

Due to errors in the estimated state of a filter, the evaluated process and measurements Jacobians during filter operation can cause the filter to update along unobservable directions [38, 84]. As a result, the filter operates in violation of the observability properties corresponding to the true nonlinear system, consequently, causing inconsistent estimates and large estimation errors. To improve the consistency of the filter, the observability-constrained VINS can be used to preserve the observability properties of the filter to match with the nullspace of the true nonlinear system.

In this section, we analyze the observability of the C-VINS and DF-VINS models and deduce the observability based consistency rules for the filters. For this purpose, we first perform a nonlinear observability analysis for the C-VINS and DF-VINS system models considering the epipolar constraint measurement model. The analysis allows identifying the unobservable subspace corresponding to the nonlinear systems, which are needed to design observability-constrained consistent filters for VINS.

3.2.1 Nonlinear observability analysis

Nonlinear observability analysis is typically performed by evaluating Lie derivatives for the system and constructing the observability matrix $\mathcal{O}$. In this work we use the basis function approach [38] which does not need exhaustive evaluation of Lie derivatives to establish the observability properties of the system. In the basis function approach, first a nonlinear equivalent of the observable canonical form is realized for the state-space system. This is performed by defining a new set of basis $\boldsymbol{\beta}_i(\mathbf{x})$ which are functions of state $\mathbf{x}$, following the procedure given in (*Theorem 4.1*) of [38]. As proved in the theorem, if we can show that the resulting dynamical system on $\boldsymbol{\beta}$ is observable (i.e., its observability matrix $\boldsymbol{\Xi}$ is of full column rank), the observability properties of the system under study (with observability matrix $\mathcal{O}$) can be conveniently found using the gradient matrix of the basis set $\mathbf{B} \triangleq \frac{\partial \boldsymbol{\beta}(\mathbf{x})}{\partial \mathbf{x}}$, where $\mathcal{O} = \boldsymbol{\Xi} \cdot \mathbf{B}$ and $\boldsymbol{\beta}(\mathbf{x}) = [\boldsymbol{\beta}_1(\mathbf{x})^T \cdots \boldsymbol{\beta}_m(\mathbf{x})^T]^T$ is the vector of basis set with m basis elements. Therefore, the unobservable directions of the system can be found using the null space of $\mathbf{B}$, i.e., $null(\mathcal{O}) = null(\mathbf{B})$.

Observability of VINS is studied in detail in [38] using a VINS system model which includes a 3D feature point in its state vector, and a measurement model which includes a 3D feature point visual observation, i.e., a visual SLAM model. In this work, we will instead consider a VINS which includes a previous camera pose in the state vector and consider a measurement model that uses the epipolar constraint. We separately study both conventional and drag force VINS introduced in Sections 3.1.2 and 3.1.3. We will use the basis functions approach introduced in [38] to study system observability, find the nullspace, and determine the unobservable subspace of both filters.

Therefore the following simplifications are performed to find an equivalent visual measurement model for the epipolar constraints to make the observability analysis easier. First, we assume that the camera frame and body frame are coincident. Second, we assume that we can find the essential matrix $\mathbf{E}$ using the measurements. It is possible to compute the essential matrix given at least 8 point correspondences between two images [106]. As a result, the essential matrix given in (3.5) can be simplified as follows,

$$
\begin{aligned}
&\mathbf{E} = R({}^{B}\mathbf{q}_{G}) \left[{}^{G}\acute{\mathbf{p}}_{B} - {}^{G}\mathbf{p}_{B}\right]_{\times} R({}^{B}\acute{\mathbf{q}}_{G})^{T} \\
\implies &\mathbf{E} = \left[R({}^{B}\mathbf{q}_{G})({}^{G}\acute{\mathbf{p}}_{B} - {}^{G}\mathbf{p}_{B})\right]_{\times} R({}^{B}\mathbf{q}_{G}) R({}^{B}\acute{\mathbf{q}}_{G})^{T} = \left[{}^{B}\mathbf{p}_{B\acute{B}}\right]_{\times} R({}^{B}\mathbf{q}_{\acute{B}})
\end{aligned}
\tag{3.14}
$$

where ${}^{B}\mathbf{p}_{B\acute{B}}$ is the translation between the current and previous body (or camera) frames expressed in the current body frame and $R({}^{B}\mathbf{q}_{\acute{B}})$ is the rotation from the previous frame to the current frame. Using factorization of (3.14) to an orthogonal and skew-symmetric matrix, the translation ${}^{B}\mathbf{p}_{B\acute{B}}$ up to a scale, and the rotation ${}^{B}\mathbf{q}_{\acute{B}}$ can be extracted resulting in four possible solutions [106]. As the third simplifications, we assume that the ambiguity related to this factorization is resolved by considering the solution where the reconstructed point is in front of both cameras. Following these simplifications, an equivalent visual measurement model for (3.5) can be found as,

$$
\mathbf{h}_{v} = \begin{bmatrix} \mathbf{h}_{v1} \\ \hline \mathbf{h}_{v2} \end{bmatrix} = \begin{bmatrix} p_x/p_z \\ p_y/p_z \\ \hline R({}^{B}\mathbf{q}_{G}) R({}^{B}\acute{\mathbf{q}}_{G})^{T} \bar{\mathbf{e}}_{i} \end{bmatrix} = \begin{bmatrix} p_x/p_z \\ p_y/p_z \\ \hline R({}^{B}\mathbf{q}_{G}) R({}^{B}\acute{\mathbf{q}}_{G})^{T} \bar{\mathbf{e}}_{1} \\ R({}^{B}\mathbf{q}_{G}) R({}^{B}\acute{\mathbf{q}}_{G})^{T} \bar{\mathbf{e}}_{2} \end{bmatrix}
\tag{3.15}
$$

where,

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = {}^{B}\mathbf{p}_{B\acute{B}} = R({}^{B}\mathbf{q}_{G})\left({}^{G}\acute{\mathbf{p}}_{B} - {}^{G}\mathbf{p}_{B}\right) \tag{3.16}$$

Vector ${}^{B}\mathbf{p}_{B\acute{B}}$ is the translation between the current and previous body frames. In order to capture the orientation information $R({}^{B}\mathbf{q}_{\acute{B}})$ between the two poses, equivalent measurements related to two reference vector observations are used in $\mathbf{h}_{v2}$ of (3.15) where $\bar{\mathbf{e}}_i$ is the standard basis vector with $i = 1, 2$.

We express the MAV orientation using the Cayley-Gibbs-Rodriguez (CGR) parameterization [107] as suggested in [38] to assist with the observability analysis. The orientation of $\{G\}$ with respect to $\{B\}$ is represented by ${}^{B}\mathbf{s}_{G}$, where $\mathbf{s}$ is the 3×1 CGR parameter.

3.2.1.1 C-VINS observability analysis

The C-VINS model in (3.3) is expressed in terms of CGR parameters and input-affine form as,

$$
\begin{bmatrix} {}^G\dot{\mathbf{p}}_B \\ {}^B\dot{\mathbf{s}}_G \\ {}^B\dot{\mathbf{v}} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{b}}_g \\ {}^G\dot{\acute{\mathbf{p}}}_B \\ {}^B\dot{\acute{\mathbf{s}}}_G \end{bmatrix} = \underbrace{\begin{bmatrix} R({}^B\mathbf{q}_G)^T\ \mathbf{v}_B \\ -\mathbf{D}\mathbf{b}_g \\ R({}^B\mathbf{q}_G)^G g\bar{\mathbf{e}}_{\mathbf{3}} - \mathbf{b}_a + [\mathbf{b}_g]_\times\ {}^B\mathbf{v} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix}}_{\mathbf{f}_0} + \underbrace{\begin{bmatrix} \mathbf{0}_3 \\ \mathbf{D} \\ [{}^B\mathbf{v}]_\times \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix}}_{\mathbf{f}_1} \boldsymbol{\omega}_m + \underbrace{\begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{I}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix}}_{\mathbf{f}_2} \mathbf{a}_m \tag{3.17}
$$

$$
\mathbf{h} = \mathbf{h}_v = \begin{bmatrix} \mathbf{h}_{v1} \\ \hline \mathbf{h}_{v2} \end{bmatrix} = \begin{bmatrix} p_x/p_z \\ p_y/p_z \\ \hline R({}^B\mathbf{q}_G)R({}^B\acute{\mathbf{q}}_G)^T\bar{\mathbf{e}}_i \end{bmatrix},
$$

where $\mathbf{D} = \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}} = \frac{1}{2}\left(\mathbf{I}_3 + [\mathbf{s}]_\times + \mathbf{s}\mathbf{s}^T\right)$. The basis functions set for the C-VINS model that satisfies conditions stated in (*Theorem 4.1*) in [38] is as follows.

$$
\boldsymbol{\beta}_1 = \mathbf{h}_{v1} = \begin{bmatrix} p_x/p_z \\ p_y/p_z \end{bmatrix} \quad , \quad \beta_2 = 1/p_z
$$

$$
\boldsymbol{\beta}_3 = {}^B\mathbf{v} \quad , \quad \boldsymbol{\beta}_4 = \mathbf{b}_g
$$

$$
\boldsymbol{\beta}_5 = R({}^B\mathbf{q}_G)^G g\bar{\mathbf{e}}_{\mathbf{3}} \quad , \quad \boldsymbol{\beta}_6 = \mathbf{b}_a
$$

$$
\boldsymbol{\beta}_7 = \mathbf{h}_{v2}|_{i=1} = R({}^B\mathbf{q}_G)R({}^B\acute{\mathbf{q}}_G)^T\mathbf{e}_1
$$

$$
\boldsymbol{\beta}_8 = \mathbf{h}_{v2}|_{i=2} = R({}^B\mathbf{q}_G)R({}^B\acute{\mathbf{q}}_G)^T\mathbf{e}_2
$$

The resulting realization of the system model expressed using the basis functions is,

$$\begin{bmatrix} \dot{\boldsymbol{\beta}}_1 \\ \dot{\beta}_2 \\ \dot{\boldsymbol{\beta}}_3 \\ \dot{\boldsymbol{\beta}}_4 \\ \dot{\boldsymbol{\beta}}_5 \\ \dot{\boldsymbol{\beta}}_6 \\ \dot{\boldsymbol{\beta}}_7 \\ \dot{\boldsymbol{\beta}}_8 \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{\bar{\boldsymbol{\beta}}}_1 \left(- \left[\bar{\boldsymbol{\beta}}_1 \right]_\times \boldsymbol{\beta}_4 - \boldsymbol{\beta}_2 \boldsymbol{\beta}_3 \right) \\ \boldsymbol{\beta}_2 \bar{\mathbf{e}}_3^T \left(\left[\bar{\boldsymbol{\beta}}_1 \right]_\times \boldsymbol{\beta}_4 - \boldsymbol{\beta}_2 \boldsymbol{\beta}_3 \right) \\ - \left[\boldsymbol{\beta}_3 \right]_\times \boldsymbol{\beta}_4 + \boldsymbol{\beta}_5 - \boldsymbol{\beta}_6 \\ \mathbf{0}_{3\times 1} \\ - \left[\boldsymbol{\beta}_5 \right]_\times \boldsymbol{\beta}_4 \\ \mathbf{0}_{3\times 1} \\ - \left[\boldsymbol{\beta}_7 \right]_\times \boldsymbol{\beta}_4 \\ - \left[\boldsymbol{\beta}_8 \right]_\times \boldsymbol{\beta}_4 \end{bmatrix}}_{\mathbf{g}_0} + \underbrace{\begin{bmatrix} \bar{\bar{\boldsymbol{\beta}}}_1 \left[\boldsymbol{\beta}_1 \right]_\times \\ \boldsymbol{\beta}_2 \bar{\mathbf{e}}_3^T \left[\boldsymbol{\beta}_1 \right]_\times \\ \left[\boldsymbol{\beta}_3 \right]_\times \\ \mathbf{0}_3 \\ \left[\boldsymbol{\beta}_5 \right]_\times \\ \mathbf{0}_3 \\ \left[\boldsymbol{\beta}_7 \right]_\times \\ \left[\boldsymbol{\beta}_8 \right]_\times \end{bmatrix}}_{\mathbf{g}_1} \boldsymbol{\omega}_m + \underbrace{\begin{bmatrix} \mathbf{0}_{2\times 3} \\ \mathbf{0}_{1\times 3} \\ \mathbf{I}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix}}_{\mathbf{g}_2} \mathbf{a}_m \tag{3.18}$$

$$\mathbf{h} = \begin{bmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_7 & \boldsymbol{\beta}_8 \end{bmatrix}^T$$

where $\bar{\boldsymbol{\beta}}_1 = \begin{bmatrix} \boldsymbol{\beta}_1 & 1 \end{bmatrix}^T$, $\bar{\bar{\boldsymbol{\beta}}}_1 = \begin{bmatrix} \mathbf{I}_2 & -\boldsymbol{\beta}_1 \end{bmatrix}$, and $\mathbf{I}_2$ is a 2×2 identity matrix. The measurement model $\mathbf{h}_{v2}$ does not introduce any additional observable directions for the C-VINS filter model, since the spans of the bases $\boldsymbol{\beta}_7$ and $\boldsymbol{\beta}_8$ corresponding to $\mathbf{h}_{v2}$ are functions of the remaining basis set ($\boldsymbol{\beta}_7$ ... $\boldsymbol{\beta}_8$) and hence do not introduce any new observable directions to the system.

In order to compute the null space of the system (3.17), we have to construct the $\boldsymbol{\Xi}$ matrix using a suitable subset of Lie derivatives [38] of the system (3.18) and then prove that this matrix is of full column rank. Matrix $\boldsymbol{\Xi}$ is selected as follows,

$$\boldsymbol{\Xi}' = \begin{bmatrix} \frac{\partial \mathcal{L}^0 \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_{13} \mathbf{g}_{21}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^1_{\mathbf{g}_0} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_{13} \mathbf{g}_{13}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_0 \mathbf{g}_{21}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^2_{\mathbf{g}_0 \mathbf{g}_0} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_0 \mathbf{g}_{13}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_0 \mathbf{g}_0} \mathbf{h}}{\partial \beta} \end{bmatrix}^T$$

where $\mathbf{g}_{ij}$ is the column j of matrix $\mathbf{g}_i$ of system (3.17). The matrix $\boldsymbol{\Xi}'$ was found to be of full column rank, therefore, the observability properties of the system were found using matrix $\mathbf{B}$ that contains the gradients of the basis set. The system has four unobservable directions as indicated by the columns of the nullspace matrix in (3.19), where $\acute{\mathbf{D}} = \frac{\partial \acute{\mathbf{s}}}{\partial \acute{\boldsymbol{\theta}}}$. The first three unobservable directions are exposed by the first block column of (3.19) that corresponds to the three degrees of freedom global translation of the current and previous positions pair together, while the fourth condition is exposed by the second column which indicates that the filter does not gain information for the global rotation about the gravity axis.

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_3 & -\left[{}^G\mathbf{p}_B\right]_\times {}^G g\bar{\mathbf{e}}_\mathbf{3} \\ \mathbf{0}_3 & \mathbf{D}R({}^B\mathbf{q}_G)\ {}^G g\bar{\mathbf{e}}_\mathbf{3} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{0}_3 & \mathbf{0}_{3\times 1} \\ \mathbf{I}_3 & -\left[{}^G\acute{\mathbf{p}}_B\right]_\times {}^G g\bar{\mathbf{e}}_\mathbf{3} \\ \mathbf{0}_3 & R({}^B\acute{\mathbf{q}}_G)\acute{\mathbf{D}}\ {}^G g\bar{\mathbf{e}}_\mathbf{3} \end{bmatrix} \tag{3.19}$$

3.2.1.2 DF-VINS Observabiliy

For this analysis, the DF-VINS model in (3.7) has been modified by combining the bias of the accelerometer $\mathbf{b}_a$ state and the mass normalized thrust $\bar{T}_L\mathbf{e}_3$. For MAVs in flight, $\bar{T}_L\mathbf{e}_3$ is much larger than b_{az}. As a result the z-axis bias state can be redefined as follows, $b_{az} \rightarrow b_{az} + \bar{T}_L\mathbf{e}_3 \approx \bar{T}_L\mathbf{e}_3$. This modified state currently includes b_{ax}, b_{ay}, and (time-varying) thrust, which makes the analysis tractable and does not affect the observability performance of the filter as will be shown in this section. We denote this modified bias state as $\bar{\mathbf{b}}_a$. As a result, the inertial measurement model $\mathbf{h}_a$ now includes all three measurements of the accelerometer. The resulting DF-VINS model expressed in terms of the CGR parameter in the input-affine form is as follows,

$$\begin{bmatrix} {}^G\dot{\mathbf{p}}_B \\ {}^B\dot{\mathbf{s}}_G \\ {}^B\dot{\mathbf{v}} \\ \dot{\bar{\mathbf{b}}}_a \\ \dot{\mathbf{b}}_g \\ {}^G\dot{\acute{\mathbf{p}}}_B \\ {}^B\dot{\acute{\mathbf{s}}}_G \end{bmatrix} = \underbrace{\begin{bmatrix} R({}^B\mathbf{q}_G)^T \ \mathbf{v}_B \\ -\mathbf{D}\mathbf{b}_g \\ R({}^B\mathbf{q}_G)^G g\bar{\mathbf{e}}_\mathbf{3} - \bar{\mathbf{D}}_L{}^B\mathbf{v} + \bar{b}_{az}\bar{\mathbf{e}}_3 + [\mathbf{b}_g]_\times \ {}^B\mathbf{v} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix}}_{\mathbf{f}_0} + \underbrace{\begin{bmatrix} \mathbf{0}_3 \\ \mathbf{D} \\ [{}^B\mathbf{v}]_\times \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \end{bmatrix}}_{\mathbf{f}_1} \boldsymbol{\omega}_m \tag{3.20}$$

$$\mathbf{h} = \left[\begin{array}{c} \mathbf{h}_v \\ \hline \mathbf{h}_a \end{array}\right] = \left[\begin{array}{c} \mathbf{h}_{v1} \\ \mathbf{h}_{v2} \\ \hline \mathbf{h}_a \end{array}\right] = \left[\begin{array}{c} p_x/p_z \\ p_y/p_z \\ R({}^B\mathbf{q}_G)R({}^B\acute{\mathbf{q}}_G)^T\bar{\mathbf{e}}_i \\ \hline -\bar{\mathbf{D}}_L{}^B\mathbf{v} + \bar{\mathbf{b}}_a \end{array}\right].$$

The basis functions set for the DF-VINS model is,

$$\boldsymbol{\beta}_1 = \mathbf{h}_{v1} = \begin{bmatrix} p_x/p_z \\ p_y/p_z \end{bmatrix} \quad , \quad \beta_2 = 1/p_z$$

$$\boldsymbol{\beta}_3 = \ {}^B\mathbf{v} \quad , \quad \boldsymbol{\beta}_4 = \mathbf{b}_g$$

$$\boldsymbol{\beta}_5 = R({}^B\mathbf{q}_G)^G g\bar{\mathbf{e}}_\mathbf{3} \quad , \quad \boldsymbol{\beta}_6 = \bar{b}_{az}\bar{\mathbf{e}}_3$$

$$\boldsymbol{\beta}_7 = \mathbf{h}_{v2}|_{i=1} = R({}^B\mathbf{q}_G)R({}^B\acute{\mathbf{q}}_G)^T\mathbf{e}_1$$

$$\boldsymbol{\beta}_8 = \mathbf{h}_{v2}|_{i=2} = R({}^B\mathbf{q}_G)R({}^B\acute{\mathbf{q}}_G)^T\mathbf{e}_2$$

$$\boldsymbol{\beta}_9 = \mathbf{h}_a = -\bar{\mathbf{D}}_L{}^B\mathbf{v} + \bar{\mathbf{b}}_a$$

The resulting realization of the system model in terms of the basis functions is,

$$\begin{bmatrix} \dot{\boldsymbol{\beta}}_1 \\ \dot{\beta}_2 \\ \dot{\boldsymbol{\beta}}_3 \\ \dot{\boldsymbol{\beta}}_4 \\ \dot{\boldsymbol{\beta}}_5 \\ \dot{\boldsymbol{\beta}}_6 \\ \dot{\boldsymbol{\beta}}_7 \\ \dot{\boldsymbol{\beta}}_8 \\ \dot{\boldsymbol{\beta}}_9 \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{\bar{\boldsymbol{\beta}}}_1 \left(-\left[\bar{\boldsymbol{\beta}}_1\right]_\times \boldsymbol{\beta}_4 - \boldsymbol{\beta}_2\boldsymbol{\beta}_3 \right) \\ \boldsymbol{\beta}_2 \bar{\mathbf{e}}_3^T \left(\left[\bar{\boldsymbol{\beta}}_1\right]_\times \boldsymbol{\beta}_4 - \boldsymbol{\beta}_2\boldsymbol{\beta}_3 \right) \\ -\left[\boldsymbol{\beta}_3\right]_\times \boldsymbol{\beta}_4 + \boldsymbol{\beta}_5 - \boldsymbol{\beta}_6 \\ \mathbf{0}_{3\times 1} \\ -\left[\boldsymbol{\beta}_5\right]_\times \boldsymbol{\beta}_4 \\ \mathbf{0}_{3\times 1} \\ -\left[\boldsymbol{\beta}_7\right]_\times \boldsymbol{\beta}_4 \\ -\left[\boldsymbol{\beta}_8\right]_\times \boldsymbol{\beta}_4 \\ -\bar{\mathbf{D}}_L \left(\left[\boldsymbol{\beta}_4\right]_\times \boldsymbol{\beta}_3 + \bar{\mathbf{D}}_L \boldsymbol{\beta}_3 + \boldsymbol{\beta}_5 + \boldsymbol{\beta}_6 \right) \end{bmatrix}}_{\mathbf{g}_0} + \underbrace{\begin{bmatrix} \bar{\bar{\boldsymbol{\beta}}}_1 \left[\boldsymbol{\beta}_1\right]_\times \\ \boldsymbol{\beta}_2 \bar{\mathbf{e}}_3^T \left[\boldsymbol{\beta}_1\right]_\times \\ \left[\boldsymbol{\beta}_3\right]_\times \\ \mathbf{0}_3 \\ \left[\boldsymbol{\beta}_5\right]_\times \\ \mathbf{0}_3 \\ \left[\boldsymbol{\beta}_7\right]_\times \\ \left[\boldsymbol{\beta}_8\right]_\times \\ -\bar{\mathbf{D}}_L \left[\boldsymbol{\beta}_3\right]_\times \end{bmatrix}}_{\mathbf{g}_1} \boldsymbol{\omega}_m \tag{3.21}$$

$$\mathbf{h} = \begin{bmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_7 & \boldsymbol{\beta}_8 & \boldsymbol{\beta}_9 \end{bmatrix}^T.$$

In order to prove the observability of the DF-VINS system in (3.20), the observability matrix $\boldsymbol{\Xi}'$ is constructed as follows,

$$\boldsymbol{\Xi}' = \begin{bmatrix} \frac{\partial \mathcal{L}^0 \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_{13} \mathbf{g}_{12}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^1_{\mathbf{g}_0} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_{13} \mathbf{g}_{13}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_0 \mathbf{g}_{12}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^2_{\mathbf{g}_0 \mathbf{g}_0} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_0 \mathbf{g}_{13}} \mathbf{h}}{\partial \beta} & \frac{\partial \mathcal{L}^3_{\mathbf{g}_0 \mathbf{g}_0 \mathbf{g}_0} \mathbf{h}}{\partial \beta} \end{bmatrix}^T$$

The matrix $\boldsymbol{\Xi}'$ is found to be of full column rank and the DF-VINS system has the same unobservable directions and nullspace as the C-VINS that is given in (3.19). The inertial measurement model $\mathbf{h}_a$ does not introduce any additional observable directions for the DF-VINS filter model.

3.2.2 Observability-constrained VINS

The observability-constrained VINS can be used to improve the consistency of the filter by preserving the observability properties of the filter to match with the nullspace of the true nonlinear system. This is achieved by satisfying the following two conditions at each time step,

$$\mathbf{N}_{k+1|k} = \mathbf{\Phi}_{k+1|k}\mathbf{N}_{k|k-1} \tag{3.22}$$

$$\mathbf{H}_k\mathbf{N}_k = \mathbf{0} \tag{3.23}$$

where $\mathbf{\Phi}_{k+1|k}$ is the state transition matrix, $\mathbf{H}_k$ is the measurement Jacobian matrix, and $\mathbf{N}_k$ is the nullspace. Conditions (3.22) and (3.23) can only be fulfilled by appropriately modifying $\mathbf{\Phi}_{k+1|k}$ and $\mathbf{H}_k$. The observability-constrained VINS design proposed in [38] will be followed for the two VINS filters in this work.

3.2.2.1 Modification of the state transition matrix

The constraints are enforced after each new IMU reading during the propagation step to maintain the unobservable directions by preserving the nullspace of the filters given in (3.19). The nullspace $\mathbf{N}$ must satisfy (3.22) by modifying the state transition matrix. The state transition matrix $\mathbf{\Phi}$ is constructed as follows,

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{\Phi}_{12} & \Delta T\mathbf{I}_3 & \mathbf{\Phi}_{14} & \mathbf{\Phi}_{15} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Phi}_{22} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{\Phi}_{25} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{\Phi}_{32} & \mathbf{I}_3 & \mathbf{\Phi}_{34} & \mathbf{\Phi}_{35} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{3.24}$$

Since the two filters have the same construction of the state transition matrix $\mathbf{\Phi}$ and the nullspace $\mathbf{N}$, both will be having the same constraints for observability consistency. From (3.22) and (3.24), the propagated state transition matrix blocks $\mathbf{\Phi}_{12}$, $\mathbf{\Phi}_{22}$, and $\mathbf{\Phi}_{32}$ are constrained as follows,

$$\mathbf{\Phi}_{12} R_{k|k-1}(^{B}\mathbf{q}_{G})^{G} g\bar{\mathbf{e}}_{\mathbf{3}} = [^{G}\mathbf{p}_{B_{k|k-1}}]_{\times}\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} - [^{G}\mathbf{p}_{B_{k+1|k}}]_{\times}\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} \tag{3.25}$$

$$\mathbf{\Phi}_{22} = R(^{B_{k+1|k}}\mathbf{q}_{B_{k|k-1}}) \tag{3.26}$$

$$\mathbf{\Phi}_{32} R(^{B}\mathbf{q}_{G})\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} = \mathbf{0}_{3\times 1} \tag{3.27}$$

where (3.25) and (3.27) can be solved by Karush-Kuhn-Tucker conditions ((63) and (64) in [38]).

3.2.2.2 Modification of the measurement Jacobian

The measurement Jacobian must be modified during each update step to satisfy (3.23). The inertial measurement Jacobian ($\mathbf{H}_i$) given in (3.9) satisfies the condition (3.23), since $\mathbf{H}_i\mathbf{N} = \mathbf{0}$ for all update steps, thus it does not get modified. However, the visual measurement Jacobian ($\mathbf{H}_v$) given in (3.6) does not satisfy the condition, thus some element of $\mathbf{H}_v$ will be modified as follows,

$$\begin{bmatrix} \mathbf{H}_{v1} & \mathbf{H}_{v2} & \mathbf{0}_{1\times 9} & \mathbf{H}_{v6} & \mathbf{H}_{v7} \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & -\left[^{G}\mathbf{p}_{B}\right]_{\times}\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} \\ \mathbf{0}_3 & R(^{B}\mathbf{q}_{G})\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} \\ \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 & -\left[^{G}\acute{\mathbf{p}}_{B}\right]_{\times}\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} \\ \mathbf{0}_3 & R(^{B}\acute{\mathbf{q}}_{G})\ ^{G} g\bar{\mathbf{e}}_{\mathbf{3}} \end{bmatrix} = \mathbf{0} \tag{3.28}$$

where $\mathbf{0}_{i\times j}$ is a $i \times j$ zero matrix. The first block column of (3.28) requires that

$\mathbf{H}_{v6} = -\mathbf{H}_{v1}$. The second block is written as,

$$\begin{bmatrix} \mathbf{H}_{v1} & \mathbf{H}_{v2} & \mathbf{H}_{v7} \end{bmatrix} \begin{bmatrix} \left(\left[{}^{G}\acute{\mathbf{p}}_{B} \right]_{\times} - \left[{}^{G}\mathbf{p}_{B} \right]_{\times} \right) \; {}^{G}g\bar{\mathbf{e}}_{\mathbf{3}} \\ R({}^{B}\mathbf{q}_{G}) \; {}^{G}g\bar{\mathbf{e}}_{\mathbf{3}} \\ R({}^{B}\acute{\mathbf{q}}_{G}) \; {}^{G}g\bar{\mathbf{e}}_{\mathbf{3}} \end{bmatrix} = \mathbf{0} \tag{3.29}$$

where (3.29) can be solved by Karush-Kuhn-Tucker conditions.

3.3 Summary

This chapter presented the design of two error-state Kalman filters, with/without drag force incorporated in the system model, for the visual-inertial navigation system of quadrotor MAVs, where the epipolar geometry constraint has been used as the visual measurement model. The design included the observability analysis of the proposed VINS system to determine the null space of the filter that helped in deducing the observability-based consistency rules to maintain the consistency of the filter during the prediction and update steps. The two filters will be used in Chapter 4 for the design of a multiple model state estimation for the MAV.

Chapter 4

Multiple Model State Estimation

In this chapter, we design an interacting multiple model VINS (IMM-VINS) estimator, where the two VINS models proposed in Chapter 3 will be incorporated in the IMM bank to handle different flight conditions. Additionally, several numerical and experimental studies will be conducted to validate the proposed algorithm.

4.1 Design of Interacting Multiple Model VINS

In this work, an IMM algorithm is used to combine estimated state $\hat{\mathbf{x}}^1$ from C-VINS (filter 1) and state $\hat{\mathbf{x}}^2$ from DF-VINS (filter 2) to find a combined estimate $\hat{\mathbf{x}}$ corresponding to the two models. The structure of the IMM algorithm is shown in Fig. 4.1, where $\hat{\mathbf{x}}^{01}$ and $\hat{\mathbf{x}}^{02}$ are the mixed states, $\hat{\mathbf{P}}^{01}$ and $\hat{\mathbf{P}}^{02}$ are the mixed covariances which are calculated following a state interaction step. The mixed estimates $\hat{\mathbf{x}}^{01}$ and $\hat{\mathbf{x}}^{02}$ are used in filtering algorithms C-VINS and DF-VINS, respectively. The state interaction step makes use of the model probabilities $\hat{\boldsymbol{\mu}}$ and probability for switching from one model to another $\tilde{\boldsymbol{\mu}}^{i|j}$ which is calculated using likelihoods of the two filter models Λ^1 and Λ^2 at each iteration. The state estimates of the two filters are combined (weighted) using the updated model probabilities $\hat{\boldsymbol{\mu}}$ to find the combined estimated state vector $\hat{\mathbf{x}}$ and covariance $\hat{P}$, where $\sum_{i=1}^{2} \hat{\boldsymbol{\mu}}^i = 1$. If the probability model of one filter is $\hat{\boldsymbol{\mu}}^i = 1$, it means the IMM will fully switch to that filter.

Algorithm 4.1. Error state IMM algorithm

1: Initialize estimated state vector $\hat{\mathbf{x}}^i$, covariance matrix $\mathbf{P}^i$, and initial probabilities $\hat{\boldsymbol{\mu}}^i$ for each filter.

2: Compute the mixed estimated states $\hat{\mathbf{x}}^{0j}$ and covariance $\mathbf{P}^{0j}$ as,

$$\hat{\mathbf{x}}^{0j} = \sum_{i=1}^{N} \hat{\mathbf{x}}^i \tilde{\boldsymbol{\mu}}^{i|j}$$

$$\mathbf{P}^{0j} = \sum_{i=1}^{N} \tilde{\boldsymbol{\mu}}^{i|j} \left[\mathbf{P}^i + (\hat{\mathbf{x}}^i \ominus \hat{\mathbf{x}}^{0j})(\hat{\mathbf{x}}^i \ominus \hat{\mathbf{x}}^{0j})^T\right]$$

$$\tilde{\boldsymbol{\mu}}^{i|j} = \frac{1}{\bar{\psi}^j} \rho^{ij} \hat{\boldsymbol{\mu}}^i \quad , \quad \bar{\psi}^j = \sum_{i=1}^{N} \rho^{ij} \hat{\boldsymbol{\mu}}^i$$

3: Propagate and update estimated states and covariance for each filter model.

4: Compute the likelihood Λ^j and estimated probability for each filter using the innovations $\tilde{\mathbf{y}}^j$ and the innovations covariance matrix $\mathbf{S}^j$,

$$\Lambda^j = \frac{1}{\sqrt{|2\pi \mathbf{S}^j|}} e^{\left[-0.5(\tilde{\mathbf{y}}^j)^T (\mathbf{S}^j)^{-1} (\tilde{\mathbf{y}}^j)\right]}$$

$$\tilde{\mathbf{y}}^j = \mathbf{y} \ominus \hat{\mathbf{y}}^j \quad , \quad \mathbf{S}^j = \mathbf{H}^j \mathbf{P}^{oj} (\mathbf{H}^j)^T + \mathbf{R}$$

$$\hat{\boldsymbol{\mu}}^i = \frac{1}{c} \Lambda^i \bar{\boldsymbol{\psi}}^i \quad , \quad c = \sum_{i=1}^{N} \Lambda^i \bar{\boldsymbol{\psi}}^i$$

5: Combine both estimated states and covariances based on estimated probabilities,

$$\hat{\mathbf{x}} = arg \min_{\hat{x} \in M} \sum_{i=1}^{N} \vartheta(\hat{\mathbf{x}}^i, \hat{\boldsymbol{\mu}}^i)$$

$$\mathbf{P} = \sum_{i=1}^{N} \hat{\boldsymbol{\mu}}^i \left[\mathbf{P}^i + (\hat{\mathbf{x}}^i \ominus \hat{\mathbf{x}})(\hat{\mathbf{x}}^i \ominus \hat{\mathbf{x}})^T\right]$$

The IMM algorithm is summarized in Algorithm 4.1, where N is the number of filter models, $\bar{\boldsymbol{\psi}}^i$ is a normalization vector used to normalize the model probability,

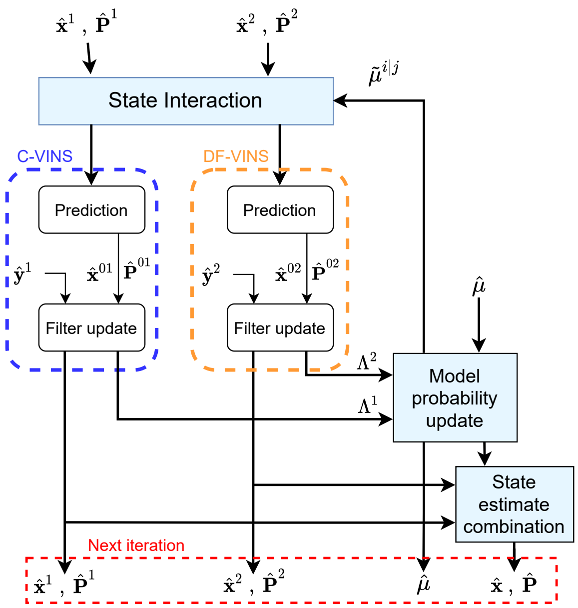


Fig. 4.1: Structure of interacting multiple model algorithm for the proposed two filters, Df-VINS and C-VINS.

and ρ^{ij} is the ij element of the Markov transition probability matrix and represents the switching from model i to model j. The matrix elements are selected parameters that administer the probability of switching from one filter in the IMM filtering bank to another or remaining in the current filter; they are assigned such that $\sum_j^N \rho^{ij} = 1$. The elements of the transition probability matrix and the initial probabilities are selected based on the knowledge of the filters in the IMM filtering bank and their relative priorities with some additional tuning to improve the filter performance as will be demonstrated in section 4.2, i.e., if the two filters in the IMM filtering bank have similar probabilities of occurrence, we set $\rho^{11} = \rho^{22}$ and $\rho^{11}, \rho^{22} >> \rho^{12}, \rho^{21}$, and start with equal initial probability values, $\hat{\boldsymbol{\mu}}^1 = \hat{\boldsymbol{\mu}}^2 = 0.5$.

Compared with the generic IMM algorithm [72, 74], the one proposed in this work has several key modifications. Error-state and measurement residual definitions are used in the state interaction, model probability update, and state combination steps. This makes the algorithm applicable for the error-state Kalman filter VINS formulations presented in this work. Furthermore, a generalized state averaging is performed in the state estimation combination step, where ϑ is the averaging function. ϑ corresponds to the usual vector averaging in case of position, velocity, and biases states (since they are vector spaces, i.e., they $\in \mathbb{R}^3$), while optimal quaternion averaging proposed in [108] is used for averaging the quaternion states (since it $\in$ group S^3), as given in (4.1). The averaging function effectively minimizes the weighted sum of the squared lengths of the error quaternions $\tilde{\mathbf{q}}$.

$$\hat{\mathbf{q}} = \pm \frac{\left[\left(\hat{\mu}^1 - \hat{\mu}^2 + z\right)\hat{\mathbf{q}}_1 + 2\hat{\mu}^2\left(\hat{\mathbf{q}}_1^T\hat{\mathbf{q}}_2\right)\hat{\mathbf{q}}_2\right]}{\|\left(\hat{\mu}^1 - \hat{\mu}^2 + z\right)\hat{\mathbf{q}}_1 + 2\hat{\mu}^2\left(\hat{\mathbf{q}}_1^T\hat{\mathbf{q}}_2\right)\hat{\mathbf{q}}_2\|} \tag{4.1}$$

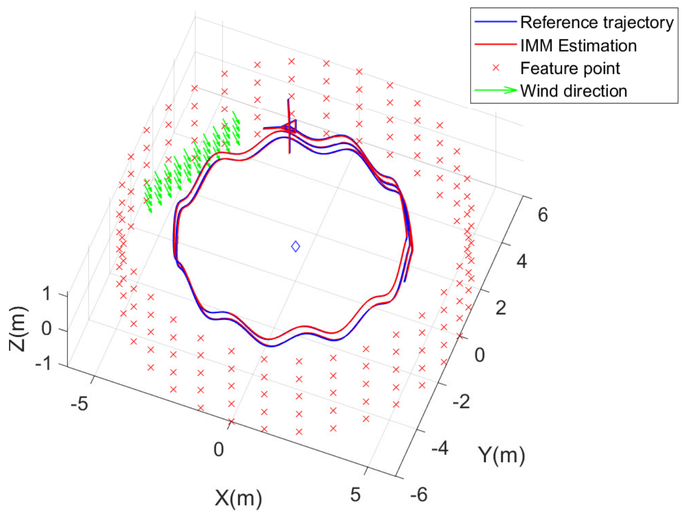
$$\text{where, } z \triangleq \sqrt{\left(\hat{\mu}^1 - \hat{\mu}^2\right)^2 + 4\hat{\mu}^1\hat{\mu}^2\left(\hat{\mathbf{q}}_1^T\hat{\mathbf{q}}_2\right)^2}$$

4.2 Results

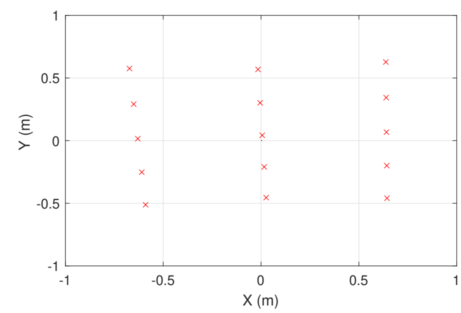
This section presents the numerical and experimental validations of the proposed IMM-VINS. The navigation performance of the conventional and drag force VINS filters are compared along with the performance improvement achieved by implementing an IMM filter using the two filtering models. In the discussion that follows, C-VINS is the conventional VINS with observability-constraints enforced in the filter, DF-VINS is the drag force VINS with observability-constraints enforced in the filter, and IMM is the IMM-VINS filter with observability-constraints enforced in the VINS filters used in the IMM filtering bank. A filter with asterisk superscript (e.g., C-VINS*, DF-VINS*, and IMM*) denotes the filters implemented without the observability consistent design. Those filters without observability-constraints are used to demonstrate the effect of not using the observability-constraints derived in section 3.2 to enforce the unobservable subspace. The experimental validation is presented using the EuRoC dataset (V1_01 easy and V1_02 medium) [104].

4.2.1 Numerical validation

A MATLAB simulator is implemented in order to evaluate the performance of the proposed multi-model estimator (IMM-VINS) and compare its performance with the stand-alone estimators (C-VINS and DF-VINS). The simulated arena includes 495 feature points uniformly distributed on a cylinder with a radius of $6m$ and a height of $2m$, as shown in Fig. 4.2a. The MAV was simulated using the kinematic model given in (3.3) with acceleration ${}^{B}\mathbf{a}$ and the angular speed ${}^{B}\boldsymbol{\omega}$ of the platform selected as inputs. The inputs were designed such that the MAV follows a circular trajectory of a radius of 4 meters completing two laps with additional excitation along the z-axis to result in a wavelike trajectory, as shown in Fig. 4.2a. The input acceleration and angular speeds adhered to differential flatness constraints related to the drag force model, which were realized using the procedure given in [20]. This implicitly enforces the dynamic constraints related to model (3.7) during the simulation as long as there is no external wind disturbance acting on the system. The averages of the magnitudes of the linear and angular velocities used for the simulation are 2.12 m/s and 4.01 rad/s, respectively. In order to verify the switching capability of the proposed IMM estimator in the presence of external disturbance, a 1.76m/s wind has been added in the second lap for a short time, as shown in Fig. 4.2a. The wind was incorporated in the MAV model similar to work in [65]. The IMU and camera measurements of the MAV were simulated at rates of 100 Hz and 10 Hz, respectively. A sample of the camera view is given in Fig. 4.2b. The noise covariance $\mathbf{Q}$ for C-VINS is set as $\mathbf{Diag}(1.1e-3^2\,\mathbf{I}_3;\ 1.3e-2^2\,\mathbf{I}_3;\ 1.8e-2^2\,\mathbf{I}_3;\ 1.7e-4^2\,\mathbf{I}_3)$ and for DF-VINS is set as $\mathbf{Diag}(1.1e-3^2\,\mathbf{I}_3;\ 1.8e-4^2\,\mathbf{I}_3;\ 1.8e-2^2\,\mathbf{I}_3;\ 1.7e-4^2\,\mathbf{I}_3)$ (the International System of Units). The standard deviation of camera measurement is set as 1 pixel.



(a)



(b)

Fig. 4.2: (a) 3D view of the simulator arena used for the experiment and the trajectory followed by the MAV. The simulation arena includes 495 feature points uniformly distributed on a cylinder with a radius of 6m and a height of 2m. (b) Camera view at one iteration including the features.

Since the DF-VINS and C-VINS filters in the IMM filtering bank have the same priorities during the flight, each one is better than the other at certain regions and under certain condition, the IMM algorithm was implemented with a transition probability matrix and initial model probability vector selected as follows,

$$\boldsymbol{\rho} = \begin{bmatrix} 0.96 & 0.04 \\ 0.04 & 0.96 \end{bmatrix} \;,\; \hat{\boldsymbol{\mu}}^i = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{4.2}$$

Fig. 4.2a illustrates the actual and IMM-VINS trajectories of the MAV. Fig. 4.3 and Table 4.1 (Trajectory 3) illustrate the position and orientation estimation accuracy of the stand-alone filters and the IMM-VINS. As seen in Fig. 4.3 and Table 4.1, DF-VINS filter exhibits improved performance than the C-VINS when there is no external wind disturbance acting on the system; on the other hand, the C-VINS filter is not significantly affected in the presence of the wind disturbance as it relies on a

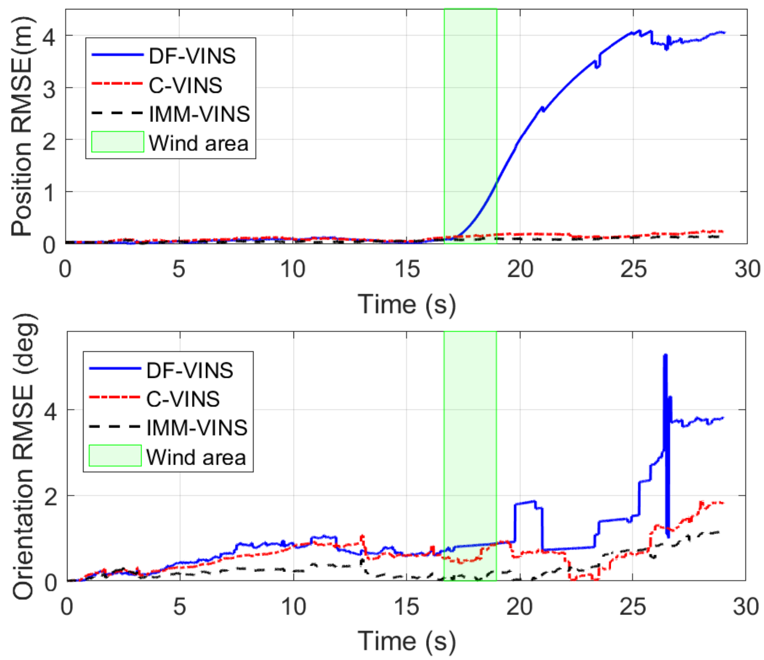


Fig. 4.3: RMSE of the position and orientation of the IMM and the two stand-alone estimators in the presence of wind. The IMM-VINS filter outperforms the stand-alone DF-VINS and C-VINS filters by switching between (or combining) the two filters in its bank.

more robust kinematic system model. The IMM estimator dynamically approximates the model probability as shown in Fig. 4.4 and generates a combined estimate of the state which is more accurate than the stand-alone filters. Moreover, it can also handle the high drift of the DF-VINS during the external disturbance period by switching to (relying on) the correct model based on model probability calculations. Fig. 4.5 shows that the position and orientation errors of the IMM and C-VINS filters agree with the 3σ bound created from the corresponding diagonal elements of the state covariance matrix. However, the covariance corresponding to the position of the DF-VINS exhibits inconsistent estimation in the presence of wind disturbance, i.e., the actual errors are well beyond the 3σ bounds estimated by the filter.

The performance of the C-VINS, DF-VINS, and IMM-VINS filters for three different trajectories are given in Table 4.1. Trajectory 1 does not include any wind dis-

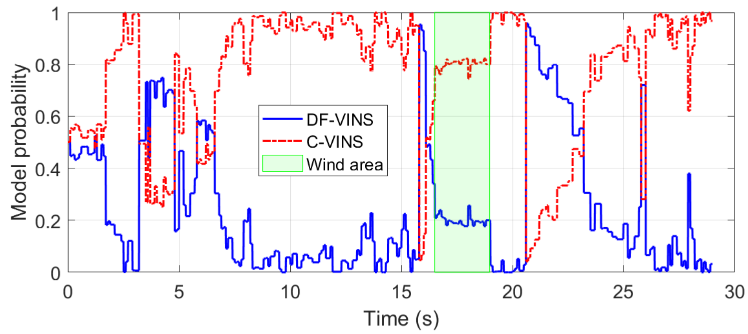


Fig. 4.4: Model probabilities of the IMM filters. The figure shows how the IMM filter switched between the two filters in its bank during the simulation. As the probability of one filter in the IMM bank increases, it implies that the IMM filter relies more on that filter with high probability than the other.

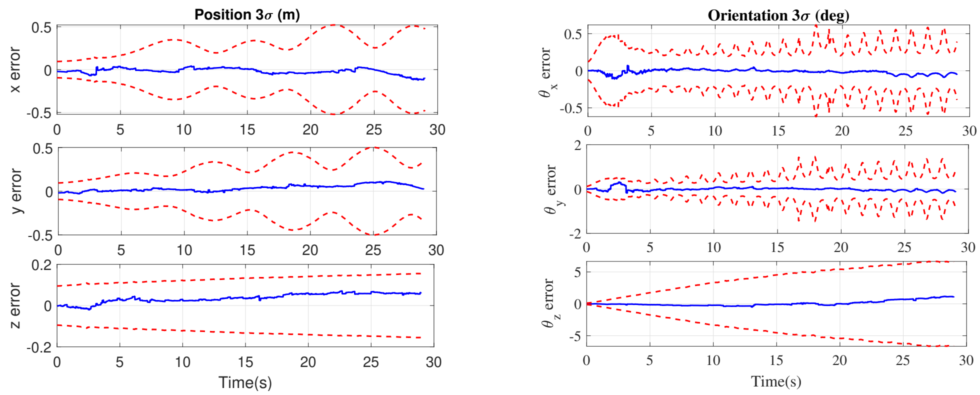
turbance, denoted by vector $\mathbf{V}_w$, while trajectory 2 and 3 include wind disturbances of $\mathbf{V}_w = \begin{bmatrix} 0.83 & -0.83 & 0 \end{bmatrix}^T$ m/s and $\mathbf{V}_w = \begin{bmatrix} 1.76 & -1.76 & 0 \end{bmatrix}^T$ m/s, respectively.

The performance of the three filters without using observability-constrained versions of the filters is also evaluated and presented in Table 4.2. The observability-constrained VINS filters have better performance because they enforce the unobservable directions of the system to prevent inconsistent information gain. As shown in Table 4.3, the IMM filter with observability consistency has more than 30.3% performance improvement over other filters (C-VINS and DF-VINS, and IMM*) for the given numerical simulations.

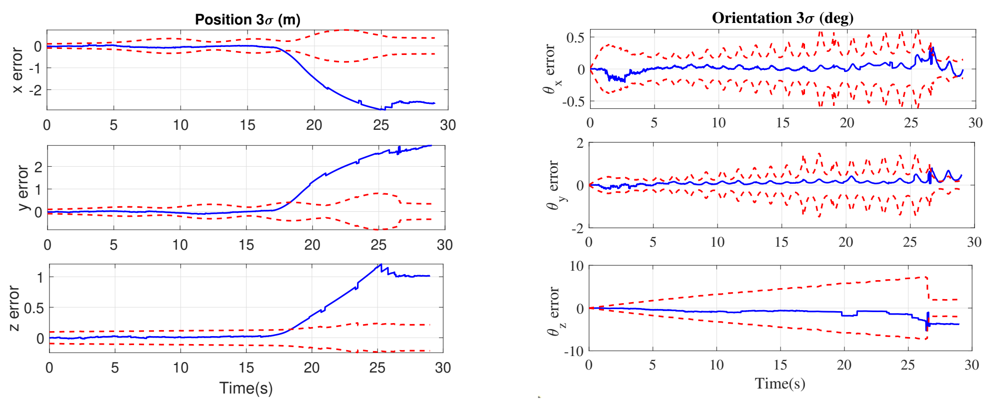
The averaged normalized estimation error squared (NEES) of the three filters (DF-VINS, C-VINS, and IMM) with and without observability-constraints is studied in Fig. 4.6a and Fig. 4.6b. The averaged NEES ($\bar{\eta}_k$) is computed over N_m independent Monte Carlo runs as follows,

$$\bar{\eta}_k = \frac{1}{N_m} \sum_{j=1}^{N_m} \tilde{\mathbf{x}}_{jk}^T P_{\tilde{x}_{jk}}^{-1} \tilde{\mathbf{x}}_{jk} \tag{4.3}$$

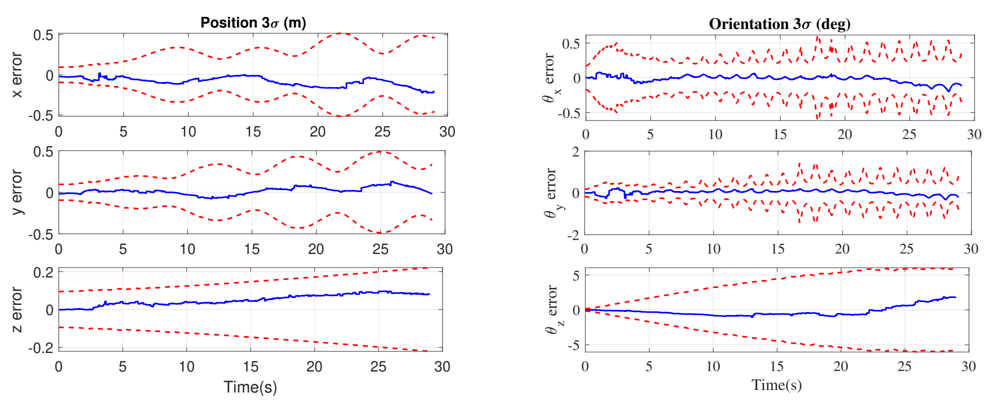
where $\tilde{\mathbf{x}}_{jk}$ is the estimation error at time index k of j^{th} run and $P_{\tilde{x}_{jk}}$ is the corresponding covariance matrix [22]. The NEES performance is another measure of estimation



(a) IMM

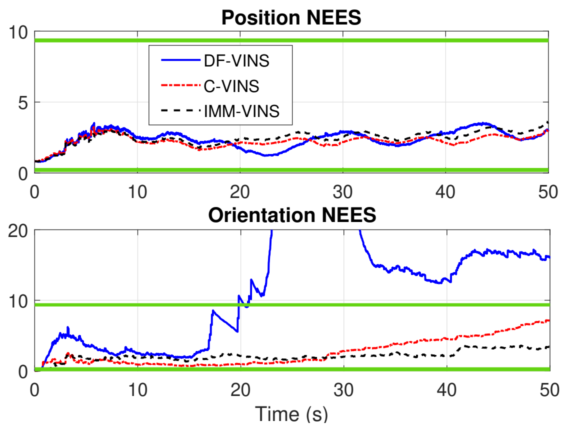


(b) DF-VINS

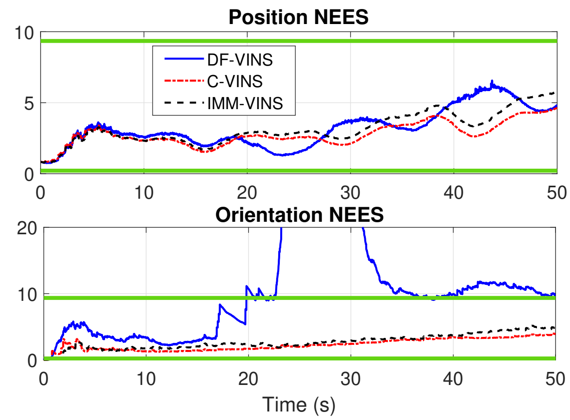


(c) C-VINS

Fig. 4.5: The 3σ bounds (in red color) for the errors for the position and orientation states of the three filters, where θ_x, θ_y, and θ_z are the angles of the 3D rotation vector (rotation axis times the angle). The plotted values are 3-times the square roots of the corresponding diagonal elements of the state covariance matrix of each filter.



(a) With observability-constrained



(b) Without observability-constrained

Fig. 4.6: Average NEES result of 50 Monte Carlo simulations of the IMM and standalone filters with and without observability-constraints enforced in the filters. (a) The three filters exhibit consistent estimation except for the orientation of the Df-VINS that deteriorates during wind disturbance as expected. (b) The position NEES of the three filters, without observability constraints enforced, gradually diverges from the ideal NEES value. The horizontal solid green lines are the two-sided 95% confidence region for a 3-DOF stochastic process.

consistency of a filter. The ideal NEES value is equal to the dimension of the error $\tilde{\mathbf{x}}_{jk}$. NEES performance closer to ideal values is a reasonable indication that the predicted covariance by the estimator is in agreement with the actual state errors of the filter. Fig. 4.6a illustrates the position and orientation NEES plots where the case of observability-constraints are enforced in the filters. The NEES values are closer to the ideal values, as seen in Fig. 3 in [38] and perform better than MSCKF, as seen in Fig. 2 in [86]. Only the orientation NEES of the DF-VINS exhibits inconsistent estimation during the application of wind then shows recovery after the wind is absent. Fig. 4.6b illustrates the position and orientation NEES plots for the filters without observability-constraints enforced. The figure shows that the three positions NEES are gradually diverging from the ideal value of 3.

Table 4.1: Estimation accuracy (RMSE) in the simulation results with observability-constraints enforced in the filters. Pos. is the RMSE in the 3D position vector in (m) and Orien. is the RMSE in the orientation vector in (deg). Trajectory 1 does not include any wind disturbance, while trajectory 2 and 3 include wind disturbances with different wind disturbance intensity.

	C-VINS		DF-VINS		IMM	
	Pos.	Orien.	Pos.	Orien.	Pos.	Orien.
Trajectory 1	0.114	0.66	0.095	0.64	**0.058**	**0.20**
Trajectory 2	0.114	0.65	0.409	0.73	**0.069**	**0.35**
Trajectory 3	0.112	0.62	1.215	1.07	**0.066**	**0.31**

Table 4.2: Estimation accuracy (RMSE) in the simulation results without observability-constraints enforced in the filters. Pos. is the RMSE in the 3D position vector in (m) and Orien. is the RMSE in the orientation vector in (deg). Trajectory 1 does not include any wind disturbance, while trajectory 2 and 3 include wind disturbances with different wind disturbance intensity.

	C-VINS*		DF-VINS*		IMM*	
	Pos.	Orien.	Pos.	Orien.	Pos.	Orien.
Trajectory 1	0.151	1.74	0.129	1.60	**0.111**	**1.48**
Trajectory 2	0.154	1.79	0.394	1.46	**0.099**	**1.32**
Trajectory 3	0.157	1.84	1.262	**1.26**	**0.105**	1.42

4.2.2 Experimental validation

Experimental validation of the proposed IMM-VINS is performed using the EuRoC dataset. In this dataset, a micro hex-rotor helicopter was used to collect navigation data related to different environments and different speeds. The helicopter was equipped with a Visual-Inertial sensor unit that includes a MEMS IMU with an update rate of 200 Hz and two monocular cameras with an update rate of 20 Hz. Only *cam0* measurement is used in this work with feature tracker front-end data imported to MATLAB from the VINS mono ROS package [109]. Two datasets of the Vicon room (shown in Fig. 4.7) will be used for this initial experimental validation of the IMM filter. *V1_01 easy* has a trajectory length of 58.6 m and duration of 144 s.

Table 4.3: Performance comparison of the improvement in RMSE (%) in the simulation results. The IMM outperforms the two stand-alone filters and the IMM filter without observability-constraints enforced in its VINS filters. It provides more than 30.3% performance improvement than the other 3 filters.

	DF-VINS/C-VINS		IMM/C-VINS		IMM/DF-VINS		IMM/IMM*	
	Pos.	Orien.	Pos.	Orien.	Pos.	Orien.	Pos.	Orien.
Trajectory 1	16.7	3.0	49.1	69.7	38.9	68.8	47.8	86.5
Trajectory 2	-258.8	-12.3	39.5	46.2	83.1	52.1	30.3	73.5
Trajectory 3	-984.8	-72.6	41.0	50	94.6	71.0	93.7	78.2

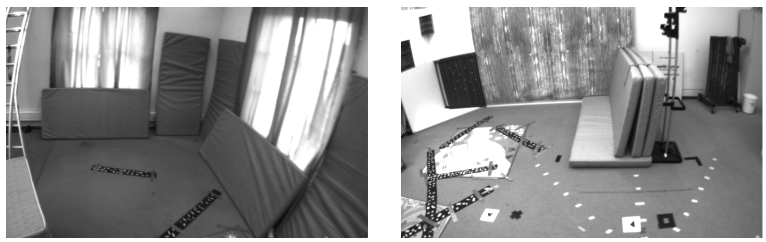


Fig. 4.7: Vicon room of the EuRoC dataset, the size of the room is $8m \times 8.4m \times 4m$.

The average linear and angular velocity are 0.41 m/s and 0.28 rad/s, respectively. *V1_02 medium* has a trajectory length of 75.9 m and duration of 83.5 s. The average linear and angular velocity are 0.91 m/s and 0.56 rad/s, respectively. A nonlinear least-squared optimization was used to estimate the drag parameters $\bar{\mathbf{D}}_L$ using the ground truth data and the IMU measurements of the V1_02 dataset. The optimal values were found as, $k_{1_x} = 0.2$, $k_{1_y} = 0.2$, and $k_{1_z} = 0.0$.

Since the IMU frame $\{I\}$ of the EuRoC datasets is not coinciding with the center of gravity (COG) located at the body frame $\{B\}$ of the MAV, all applied forces including thrust force and drag force will be transformed to the IMU frame $\{I\}$ in order for the drag-force model to work properly. The Vicon frame is almost aligned with the COG of the helicopter except for the small shift in the z-axis; therefore, the Vicon frame will be considered as the helicopter body frame $\{B\}$. The accelerometer bias of the DF-VINS will be updated to include the thrust force. The new bias state will be expressed in $\{B\}$; therefore, the velocity state vector and the inertial measurement model will be updated as given in Appendix A. The accelerometer bias of the C-VINS filter will be expressed in $\{B\}$ as well in order for the IMM to have the corresponding states between both filters expressed in the same frames. Since the extrinsic parameters between the IMU frame $\{I\}$ and Vicon frame are known, the bias will be simply transformed by the rotation matrix $R(^I\mathbf{q}_B)$ that rotates a vector from $\{B\}$ to $\{I\}$.

As a consequence of these changes, the design of the IMM for the EuRoC dataset is challenging because the accelerometer biases of the two filters are different; however, both are expressed in the same frame. The bias state vector of the DF-VINS includes the thrust force; thus, the covariance elements corresponding to this state will be different as well, and this is not the case in the C-VINS filter. Therefore, the state and covariance interaction step in Algorithm 4.1 must be updated to accommodate the state mismatch, which could be solved by handling both filters as unequal dimension states-filters. Both filters have the same states of position, orientation, velocity, and gyroscope bias, but each filter has an extra state, which is the z-axis accelerometer bias. Several methods have been proposed in [34, 110] to address the unequal state dimension problem in IMM. We tested all approaches proposed in both references and selected the unbiased approach shown in [110] as it resulted in the best performance. The unbiased approach has been applied to our IMM estimator with details of the implementation given in Appendix B.

Another challenge arises in the experimental validation using the EuRoC dataset because the helicopter was stationary at the beginning of the trajectory. During stationary conditions, the DF-VINS will diverge, and the state covariance matrix will increase due to the high uncertainties. While the helicopter is stationary, the visual update is off until the helicopter starts to move and detects enough disparity for the visual update as given in (3.11). Therefore, the transition probability matrix and the initial model probability vector cannot be selected as given in (4.2) because the DF-VINS diverges and has poor performance compared to the C-VINS. Therefore, the initial model probability corresponding to the C-VINS should be very high in the beginning in order for the IMM estimator to have better performance, after that, the transition probability matrix and the model probability vector should be updated in a proper manner once the MAV starts to move so that the IMM switches to the filter with better performance. In order to achieve that, the transition probability matrix should be adaptively updated using the posterior probability to improve the performance of the IMM. The IMM algorithm has been updated as proposed in [35] to update the transition probability matrix elements at each new visual update, n, as follows,

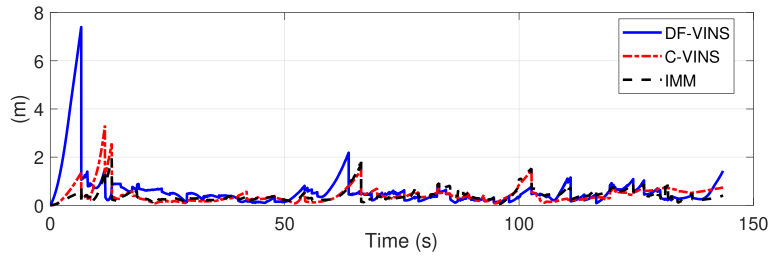
$$\rho^{ij}(n) = \frac{\frac{\hat{\boldsymbol{\mu}}^j(n)}{\hat{\boldsymbol{\mu}}^j(n-1)}\rho^{ij}(n-1)}{\sum_{j=1}^{r}\frac{\hat{\boldsymbol{\mu}}^j(n)}{\hat{\boldsymbol{\mu}}^j(n-1)}\rho^{ij}(n-1)} \tag{4.4}$$

where r is the number of filters used in the IMM. The initial transition probability matrix and models probabilities for the V1_02 have been selected as follows,

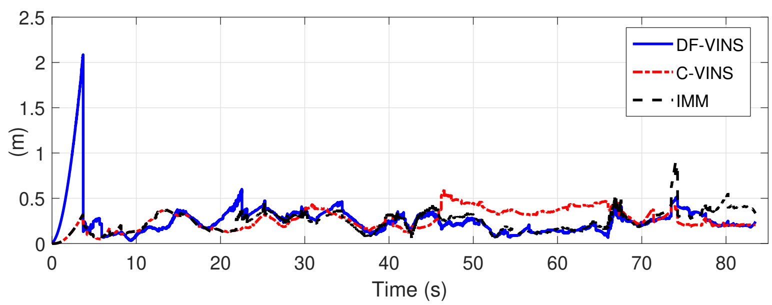
$$\boldsymbol{\rho} = \begin{bmatrix} 0.9999997 & 0.0000003 \\ 0.0001 & 0.9999 \end{bmatrix}, \quad \hat{\boldsymbol{\mu}}^i = \begin{bmatrix} 0.99999997001 \\ 0.00000002999 \end{bmatrix} \tag{4.5}$$

The performance of the DF-VINS, C-VINS, and IMM-VINS filters is illustrated in Fig. 4.8 and Table 4.4. The IMM has improved performance with minimum position RMSE. Fig. 4.9 illustrates the model probability of both filters and shows how the IMM switched to C-VINS filter when the helicopter is stationary at the beginning of both datasets. At slow speeds, the C-VINS model is used by the IMM as seen in the V1_01 dataset, while it switched to the DF-VINS in most of the trajectory due to the validity of the drag force model during fast and aggressive maneuvering [20]. The video of this experiment can be found on https://youtu.be/lsfU2wrHGVg.

It should be noted that the EuRoC dataset was used to show the capability of the IMM algorithm to adaptively switch to the DF-VINS during fast and aggressive maneuvering. However, in order to further validate the performance of IMM for wind disturbance scenarios more representative experimental data should be used as the EuroC dataset has minimal wind disturbance.



(a)



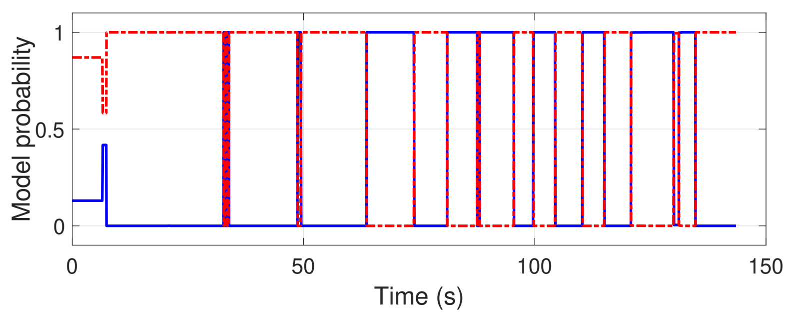
(b)

Fig. 4.8: RMSE of the position of the IMM and the two stand-alone estimators for the experimental validation. (a) is the EuRoC V1_01 easy dataset and (b) is the EuRoC V1_02 medium dataset. The EuRoC V1_01 is a slow trajectory with a flight duration of 144 s and the EuRoC V1_02 is a relatively fast trajectory with a flight duration of 83.5 s.

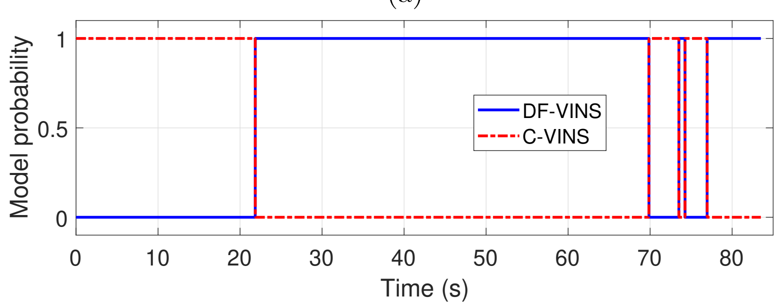
From Fig. 4.8 it can be seen that the IMM filter may have a slightly different performance than the stand-alone filters at some regions due to the state interaction (mixing) given in Algorithm 6.1, where the two filters in the IMM bank are mixed prior to the state update at the beginning of each iteration and then are combined to provide the IMM estimated states after the state update based on the model probability of each filter. Therefore, the two filters in the IMM bank will be having a

Table 4.4: Estimation accuracy (RMSE) of the experimental validation. Pos. is the RMSE in the 3D position vector in (m) and Orien. is the RMSE in the orientation vector in (deg). The results show that the IMM outperforms the two stand-alone filters and the IMM filter without observability-constraints enforced in its VINS filters.

	C-VINS		DF-VINS		IMM*		IMM	
	Pos.	Orien.	Pos.	Orien.	Pos.	Orien.	Pos.	Orien.
V1-01	0.44	2.35	0.59	2.73	0.62	2.42	**0.40**	2.36
V1-02	0.27	0.35	0.26	0.35	0.25	0.35	**0.24**	0.35

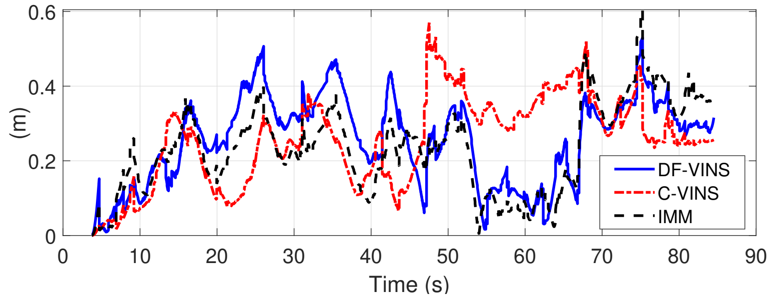


(a)

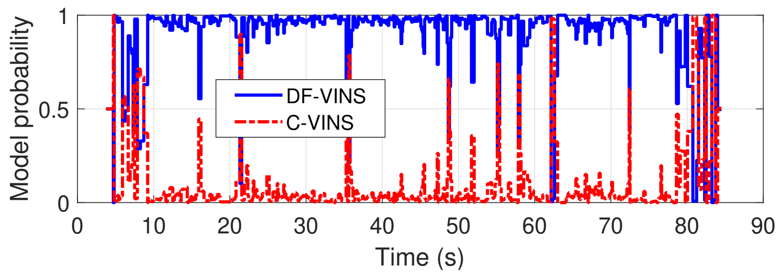


(b)

Fig. 4.9: Model probabilities of the IMM for the experimental validation. (a) is the EuRoC V1_01 easy dataset and (b) is the EuRoC V1_02 medium dataset. The figure shows how the IMM filter relied more on the DF-VINS in V1_02 that includes a faster trajectory and higher maneuver index except for the stationary region at the beginning of the trajectory.



(a)



(b)

Fig. 4.10: Experimental validation of EuRoC V1_02 while starting the experiment after the stationary region of the trajectory. (a) is the RMSE of the position, the RMSE of the DF-VINS is 25 m, C-VINS is 26 m and IMM is 23 m. (b) is the model probabilities of the IMM and it shows mixing between the two filters more comparable to the simulation than the bang-bang behavior of the adaptive IMM.

slightly different performance, but with similar behavior, than the stand-alone filters, DF-VINS and C-VINS, which consequently changes the performance of the IMM.

The IMM algorithm can be used without the adaptation update if the experiment is started after the stationary region of the trajectory. In this case, the transition probability matrix and the initial model probability vector can be selected as given in (4.2). The performance of the three filters starting after the stationary part of this case is given in Fig. 4.10. It is clearly shown that the IMM switched to (relied more on) the filter with lower RMSE to maintain high performance.

4.3 Summary

This chapter presented the design and validation of an IMM estimator for visual-inertial navigation of MAVs. The design strategically selects a conventional VINS model based on a kinematic model, and a drag force VINS model based on rotor drag dynamic constraints for the models of the IMM. An epipolar constraint-based visual measurement model, proposed in Chapter 3, is used for both filters, which addresses the computational complexity concerns of the parallel filters running on IMM. Numerical and experimental results validate the improved performance of the IMM-VINS over the stand-alone versions and highlight the ability of the filter to adaptively transit between the different models to achieve improved performance and navigation consistency.

Chapter 5

Nonlinear Model Predictive Control

In this chapter, we present the dynamic model of the quadrotor MAV (with/without drag force incorporated in the MAV model) and its corresponding discrete-time model which is required for the design of the MPC. Then, an NMPC scheme without stabilizing costs or constraints is presented in order to asymptotically stabilize the system.

5.1 Essential Notation

Throughout the chapter, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{N}$ denote the sets of rational, real, and natural numbers, respectively, and $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ are the sets of non-negative rational, real, and integer numbers, respectively. A function $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class $\mathcal{K}$ if it is continuous and strictly increasing with $\xi(0) = 0$. In addition, if $\xi \in \mathcal{K}$ is unbounded it is of class $\mathcal{K}_\infty$. A function $\zeta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class $\mathcal{KL}$ if it is continuous and $\zeta(\cdot, k) \in \mathcal{K}_\infty$ $\forall k \in \mathbb{R}_{\geq 0}$ and $\zeta(h, \cdot)$ is strictly monotonically decaying to zero for each $h > 0$. The classes $\mathcal{K}_\infty$ and $\mathcal{KL}$ will be used during the chapter for the stability analysis. The class $\mathcal{KL}$-function is illustrated in Fig. 5.1

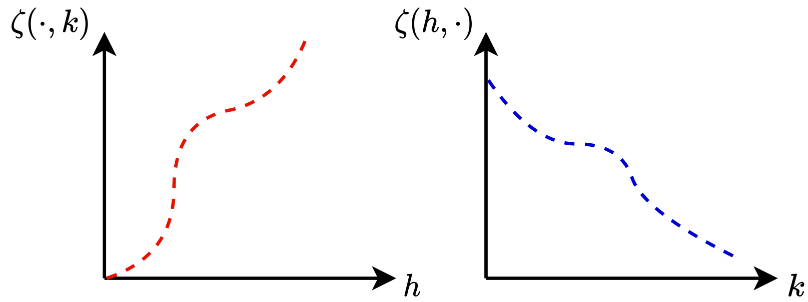


Fig. 5.1: An illustration of the class $\mathcal{KL}$-function ($\zeta(h, k)$).

5.2 System Description

5.2.1 System coordinates

The location of the MAV can be defined by its position and orientation, as shown in Fig. 5.2, where the position can be defined relative to the body frame $\{B\}$ attached to its center of gravity, and the global inertial frame $\{G\}$. The orientation of the MAV is defined by the three Euler angles, namely, roll, pitch, and yaw angles, symbolized as ϕ, θ, and ψ, respectively.

5.2.2 Control problem definition

Consider the following discrete-time nonlinear system,

$$\mathbf{x}^{+} = f_{\delta}(\mathbf{x}, \mathbf{u}), \tag{5.1}$$

where $f_{\delta} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a discrete-time analytic mapping, δ is the sampling time, $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^n$ is the state vector, $\mathbf{x}^{+}$ is the next evolution of $\mathbf{x}$, $\mathbf{u} \in \mathbf{U} \subseteq \mathbb{R}^m$ is the control input vector, n is the number of states, and m is the number of control

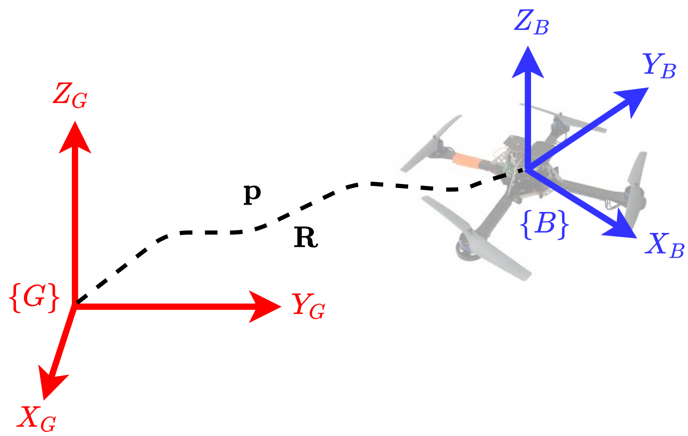


Fig. 5.2: Coordinate frames of the quadrotor MAV system. $\{G\}$ is the global frame and $\{B\}$ is the body frame located at the center of gravity of the MAV.

inputs. The objective is to drive the system from an initial state $\mathbf{x}_0$ to a reference state $\mathbf{x}_r$ using the least amount of control effort and minimum tracking error. For this purpose, the NMPC scheme will be used to stabilize the MAV with the following proposed running cost $\ell(\mathbf{x}, \mathbf{u}) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ that uses the nonlinear model (5.1) to compute a feedback control input.

$$\ell(\mathbf{x}, \mathbf{u}) = \|\mathbf{x} - \mathbf{x}_r\|^2_{\mathbf{Q}} + \|\mathbf{u} - \tilde{\mathbf{u}}\|^2_{\mathbf{R}_u} \tag{5.2}$$

The first term penalizes the tracking error while the second term is the penalty imposed on the control inputs. Where, $\mathbf{Q}$ and $\mathbf{R}_u$ are the weight matrices of the tracking error and control actions, respectively, and $\tilde{\mathbf{u}}$ is the desired input value and will be defined later in the coming sections. The admissibility of an input function is defined as follows.

Definition 5.1. *For a given feasible set of states* $\mathbf{X}$ *and an admissible set of control values* $\mathbf{U}$*, and for initial states* $\mathbf{x}_0 \in \mathbf{X}$ *and* $N \in \mathbb{N}$*, a sequence of control values* $u = (u(0), u(1), \cdots, u(N-1)) \in \mathbf{U}^N$ *is said to be admissible if the state trajectory*

$$x_u(\cdot;\mathbf{x}_0) = (x_u(0;\mathbf{x}_0), x_u(1;\mathbf{x}_0), \cdots, x_u(N;\mathbf{x}_0)),$$

which is iteratively generated by the system model and proceeded from $x_u(0;\mathbf{x}_0) = \mathbf{x}_0$*, satisfies* $x_u(k;\mathbf{x}_0) \in \mathbf{X}\ \forall k \in \{0, 1, \cdots, N\}$*. We denote this admissible control sequence for* $\mathbf{x}_0$ *up to time* N *by* $u \in \mathcal{U}^N(\mathbf{x}_0)$*.*

In this work, we adopt the inner-outer loop control structure [111], shown in Fig. 5.3, where the inner-loop controller (low-level attitude controller) is approximated by a first-order system, as proposed in [52]. The attitude controller tracks the commanded roll and pitch angles; ϕ_{cmd} and θ_{cmd}, and the reference yaw angle ψ_r, and generates the desired torques accordingly. The proposed first-order model sufficiently represents the behavior of the inner-loop controller, which is necessary for the design of the outer-loop controller (NMPC). The parameters of the proposed model can be identified through classic system identification techniques [52].

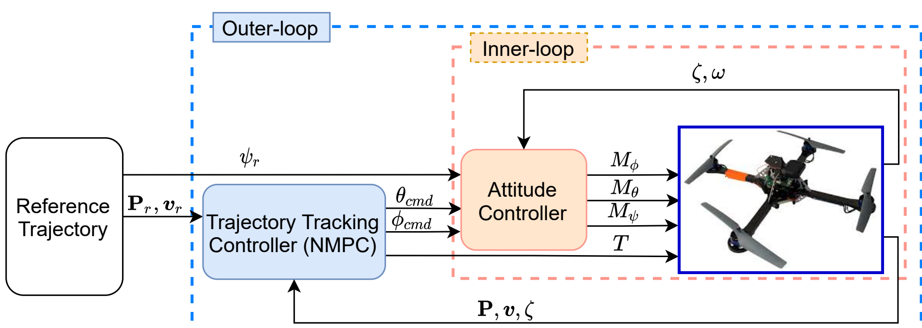


Fig. 5.3: Inner-outer loop control scheme for quadrotor MAV system. $\boldsymbol{\zeta}$ is the Euler angles vector, $\boldsymbol{\omega}$ is the angular velocity vector, $\boldsymbol{v}$ is the MAV velocity vector, M_ϕ, M_θ, and M_ψ are the applied torques for the three Euler angles, and the subscript r denotes the reference value.

5.3 Quadrotor MAV Dynamics

5.3.1 Conventional model without drag force

The nonlinear model of the quadrotor is defined as follows:

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) &= \mathbf{R}\bar{\mathbf{e}}_3 T - \mathrm{g}\bar{\mathbf{e}}_3 \\ \dot{\phi}(t) &= \frac{K_\phi}{\tau_\phi}\phi_{cmd}(t) - \frac{1}{\tau_\phi}\phi(t) \\ \dot{\theta}(t) &= \frac{K_\theta}{\tau_\theta}\theta_{cmd}(t) - \frac{1}{\tau_\theta}\theta(t) \end{aligned} \tag{5.3}$$

where, $\mathbf{p}$ is the position of the MAV $\{B\}$ relative to $\{G\}$ expressed in $\{G\}$; $\mathbf{R} \coloneqq \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi)$ is the rotation matrix from frame $\{B\}$ to $\{G\}$; $\boldsymbol{v}$ is the MAV velocity vector of $\{B\}$ relative to $\{G\}$; g is the gravitational acceleration; T is the mass normalized thrust; τ_ϕ and τ_θ are the time constants of the inner-loop linear first-order model for the roll and pitch angles, respectively; K_ϕ and K_θ are the gains of the same model; and $\bar{\mathbf{e}}_3$ is the standard basis vector $\bar{\mathbf{e}}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

For simplicity, let $\bar{K}_\phi = \frac{K_\phi}{\tau_\phi}$, $\bar{\tau}_\phi = \frac{1}{\tau_\phi}$, $\bar{K}_\theta = \frac{K_\theta}{\tau_\theta}$, and $\bar{\tau}_\theta = \frac{1}{\tau_\theta}$. Therefore, system (5.3) can be written in state-space representation, $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$, as;

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\boldsymbol{x}}_{13} \\ \dot{\boldsymbol{x}}_{46} \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{46} \\ -\mathrm{g}\bar{\mathbf{e}}_3 \\ -\bar{\tau}_\phi x_7 \\ -\bar{\tau}_\theta x_8 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{R}\bar{\mathbf{e}}_3 u_1 \\ \bar{K}_\phi u_2 \\ \bar{K}_\theta u_3 \end{bmatrix} \tag{5.4}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the continuous-time analytic mapping, $\boldsymbol{x}_{13} \coloneqq \mathbf{p}$, $\boldsymbol{x}_{46} \coloneqq \boldsymbol{v}$, $x_7 \coloneqq \phi$, $x_8 \coloneqq \theta$, $u_1 \coloneqq T$, $u_2 \coloneqq \phi_{cmd}$, and $u_3 \coloneqq \theta_{cmd}$. Assuming piecewise constant control inputs on each sampling interval, model (5.4) can be discretized by the sampling time δ (in seconds) using forward Euler method as follows;

$$\mathbf{x}^+ = f_\delta(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \boldsymbol{x}_{13} \\ \boldsymbol{x}_{46} \\ x_7 \\ x_8 \end{bmatrix} + \delta \cdot \begin{bmatrix} \boldsymbol{x}_{46} \\ -\mathrm{g}\bar{\mathbf{e}}_3 \\ -\bar{\tau}_\phi x_7 \\ -\bar{\tau}_\theta x_8 \end{bmatrix} + \delta \cdot \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{R}\bar{\mathbf{e}}_3 u_1 \\ \bar{K_\phi} u_2 \\ \bar{K_\theta} u_3 \end{bmatrix} \tag{5.5}$$

Finally, the discrete-time system model (5.5) will be used in the prediction mechanism of the NMPC.

5.3.2 Dynamic model with drag force

The nonlinear model of the quadrotor MAV is defined as follows;

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \mathbf{R}^B\boldsymbol{v}(t) \\ {}^B\dot{\boldsymbol{v}}(t) &= T\bar{\mathbf{e}}_3 - \mathbf{R}^T\bar{\mathbf{e}}_3\mathrm{g} - \mathbf{D}^B\boldsymbol{v}(t) \\ \dot{\phi}(t) &= \frac{K_\phi}{\tau_\phi}\phi_{cmd}(t) - \frac{1}{\tau_\phi}\phi(t) \\ \dot{\theta}(t) &= \frac{K_\theta}{\tau_\theta}\theta_{cmd}(t) - \frac{1}{\tau_\theta}\theta(t) \end{aligned} \tag{5.6}$$

where ${}^B\boldsymbol{v}$ is the MAV velocity vector of $\{B\}$ relative to $\{G\}$ expressed in $\{B\}$. $\mathbf{D} = \mathrm{diag}(d_x, d_y, d_z)$ is the drag coefficient matrix with normalized drag coefficients, $d_x, d_y,$ and d_z, in the three axes. The aerodynamic rotor drag force ($\mathbf{D}^B\boldsymbol{v}$) [20] of the MAV is one of the main aerodynamics effects that highly influences the tracking errors of the feedback controllers. As a result, incorporating this force in the system model improves the controller performance at aggressive and agile maneuvers with high speeds and accelerations [20].

Similarly, system (5.6) can be written in state-space representation as;

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\boldsymbol{x}}_{13} \\ \dot{\boldsymbol{x}}_{46} \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\boldsymbol{x}_{46} \\ -\mathbf{R}^T \bar{\mathbf{e}}_3 \mathrm{g} - \mathbf{D}\boldsymbol{x}_{46} \\ -\bar{\tau}_\phi x_7 \\ -\bar{\tau}_\theta x_8 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ u_1 \bar{\mathbf{e}}_3 \\ \bar{K_\phi} u_2 \\ \bar{K_\theta} u_3 \end{bmatrix} \tag{5.7}$$

consequently, can be dicretized by the sampling time δ as follows;

$$\mathbf{x}^+ = f_\delta(\mathbf{x}, \mathbf{u}) = \mathbf{x} + \delta \cdot \begin{bmatrix} \mathbf{R}\boldsymbol{x}_{46} \\ -\mathbf{R}^T \mathrm{g} \bar{\mathbf{e}}_3 - \mathbf{D}\boldsymbol{x}_{46} \\ -\bar{\tau}_\phi x_7 \\ -\bar{\tau}_\theta x_8 \end{bmatrix} + \delta \cdot \begin{bmatrix} \mathbf{0}_{3\times 1} \\ u_1 \bar{\mathbf{e}}_3 \\ \bar{K_\phi} u_2 \\ \bar{K_\theta} u_3 \end{bmatrix} \tag{5.8}$$

Similarly, the discrete-time system model (5.8) will be used as well in the prediction mechanism of the NMPC in case of designing the NMPC while incorporating the drag force in the MAV model.

5.3.2.1 States and inputs constraints

For both models, the state vector could be constrained to the working arena as $\mathbf{X} \coloneqq \{\mathbf{x} \in \mathbb{R}^n \mid x_{imin} \leq x_i(k) \leq x_{imax}\}$ for all $i = 1, \cdots, 8$ and $k \in \mathbb{N}_0$. The control input constraints are defined as

$$\mathbf{U} \coloneqq \left\{ \mathbf{u} \in \mathbb{R}^m \mid \begin{pmatrix} u_{1\min} \\ -u_{2\max} \\ -u_{3\max} \end{pmatrix} \leq \mathbf{u}(k) \leq \begin{pmatrix} u_{1\max} \\ u_{2\max} \\ u_{3\max} \end{pmatrix} \right\} \tag{5.9}$$

$\forall k \in \mathbb{N}_0$, where $0 < u_{1min} < u_{1max}$, and $\mathbf{u} = [u_1, u_2, u_3]^T$.

5.4 Nonlinear Model Predictive Control

The NMPC algorithm solves iteratively an optimal control problem (OCP) at each time instant with sampling time $\delta > 0$ and prediction horizon $N \in \mathbb{N}_{\geq 2}$. The solution of the OCP includes the minimization of a cost function $J_N : \mathbf{X} \times \mathbf{U}^N \to \mathbb{R}_{\geq 0}$ that sums up the running costs $\ell(\mathbf{x}, \mathbf{u})$ along the predicted trajectories. The cost function is defined as

$$J_N(\mathbf{x}_k, u) \coloneqq \sum_{r=k}^{k+N-1} \ell(x_u(r, \mathbf{x}_k), \mathbf{u}(r)) \tag{5.10}$$

with an optimal value function $V_N : \mathbf{X} \to \mathbb{R}_{\geq 0}$ defined as

$$V_N(\mathbf{x}_k) \coloneqq \inf_{u \in \mathcal{U}^N(\mathbf{x}_k)} J_N(\mathbf{x}_k, u) \tag{5.11}$$

for $N \in \mathbb{N} \cup \{\infty\}$. Solving the OCP while attaining the infimum induces an optimal control function $u^* = \mathbf{u}^*(\cdot, \mathbf{x}_k)$ with $V_N(\mathbf{x}_k) = J_N(\mathbf{x}_k, u^*)$. Algorithm 5.1 summarizes the proposed NMPC scheme without stabilizing costs or constraints. The NMPC algorithm aims at computing a feedback law $\mu_N : \mathbf{X} \to \mathbf{U}$ (defined as $\mu_N(k, \mathbf{x}) = \mathbf{u}^*(0, \mathbf{x}_k)$) such that for each $\mathbf{x}_k \in \mathbf{X}$, the resulting closed-loop trajectory $x_u(.; \mathbf{x}_k)$ generated by (5.1), where $x_u(0; \mathbf{x}_0) = \mathbf{x}_0$, satisfies the constraints $x_u(k; \mathbf{x}_0) \in \mathbf{X}$ and $\mu_N(x_u(k; \mathbf{x}_0)) \in \mathbf{U}$ for all $k \in \mathbb{N}_0$ and is asymptotically stable. Neither the asymptotic stability of system (5.1) nor the recursive feasibility can be guaranteed under the proposed NMPC algorithm since no stabilizing costs or constraints are incorporated in the proposed OCP and $u \equiv 0_{\mathbb{R}^m} \notin \mathcal{U}^N(\mathbf{x}_k)$, i.e., $u_{1_{min}} > 0$ as in (5.9), therefore, $\mathcal{U}^N(\mathbf{x}_k) \neq \emptyset$ is not guaranteed. In the following section, we will show that the asymptotic stability and recursive stability can be ensured by computing a stabilizing prediction horizon N through deriving a growth sequence that satisfies the cost controllability condition [89]. A closed-loop system is asymptotic stable (e.g., at the origin) if there exists a function $\beta \in \mathcal{KL}$ such that the closed-loop trajectory $x_{\mu_N}(k, \mathbf{x}_0)$ has the following condition, i.e.,

$$\|x_{\mu_N}(k, \mathbf{x}_0)\| \leq \beta(\|\mathbf{x}_0\|, k) \quad \forall k \in \mathbb{N}_0 \tag{5.12}$$

for all $\mathbf{x}_0 \in \mathbf{X}$.

Algorithm 5.1: NMPC Scheme

Input: Prediction horizon N, sampling time δ

Output: NMPC feedback law μ_N

Initialization: Set the time index $k = 0$ and $\mathbf{x}_k \coloneqq \mathbf{x}_0$

1 **Loop**

2 Compute a minimizing control sequence $u^* = (\mathbf{u}^*(0), \mathbf{u}^*(1), \cdots, \mathbf{u}^*(N-1)) \in \mathcal{U}^N(\mathbf{x}_k)$ by solving the OCP that satisfies $J_N(\mathbf{x}_k, u^*) = V_N(\mathbf{x}_k)$.

3 Implement the control input $\mathbf{u}(k) \coloneqq \mu_N(k, \mathbf{x}) \coloneqq \mathbf{u}^*(0, \mathbf{x}_k)$ at the MAV plant.

4 Measure the current state $x_{\mu_N}(k; \mathbf{x}_0) \coloneqq x_{u^*}(k; \mathbf{x}_k)$ and set $\mathbf{x}_k = x_{\mu_N}(k; \mathbf{x}_0)$ and $k = k + 1$.

5.4.1 Stability and recursive feasibility of MPC without stabilizing costs or constraints

This section summarizes the findings from [49, 60, 64], then theses findings will be utilized in section 5.5 to prove the asymptotic stability and recursive feasibility of the discrete-time model of the quadrotor MAV defined in (5.1). We first formulate the following assumptions, which are crucial for proving the asymptotic stability of the proposed NMPC algorithm without stabilizing terminal costs or constraints.

Assumption 5.1. *Assume that there exist*

A5.1.1. A monotonically increasing and bounded sequence $(\gamma_i)_{i\in\mathbb{N}}$ *such that*

$$V_i(\mathbf{x}_0) \leq \gamma_i \left\|\mathbf{x}_0 - \mathbf{x}_r\right\|_{\mathbf{Q}}^2 \quad \forall i \in \mathbb{N} \tag{5.13}$$

holds for each $\mathbf{x}_0 \in \mathbf{X}$, *where* $\|\mathbf{x}\|_{\mathbf{Q}}^2$ *represents the quadratic form* $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ *and* $\mathbf{Q} \in \mathbb{R}^{n\times n}$ *is a diagonal matrix composed of the penalty parameters of the state error such that* $\mathbf{Q} \succ 0$.

A5.1.2. Two functions $\bar{\alpha}, \alpha \in \mathcal{K}_\infty$ *satisfying*

$$\bar{\alpha}(\|\mathbf{x} - \mathbf{x}_r\|) \leq \ell^*(\mathbf{x}) \leq \alpha(\|\mathbf{x} - \mathbf{x}_r\|) \quad \forall \mathbf{x} \in \mathbf{X} \tag{5.14}$$

where $\ell^*(\mathbf{x}) \coloneqq \inf_{u\in\mathcal{U}^1(\mathbf{x})} \ell(\mathbf{x}, u)$.

5.4.1.1 Stability

The performance bound of NMPC closed-loop can be defined as follows.

Definition 5.2. *Let the closed-loop control sequence* $\mu_N(k, \mathbf{x})$ *be admissible and given, and* $J_\infty^{cl}(\mathbf{x}, \mu_N) \coloneqq \sum_{k=0}^{\infty} \ell(x_{\mu_N}(k, \mathbf{x}), \mu_N(k, \mathbf{x}))$ *be the closed-loop costs on the infinite horizon. Then, the performance bound of the proposed NMPC scheme can be defined as follows*

$$J_\infty^{cl}(\mathbf{x}, \mu_N) \leq \alpha_N^{-1} V_\infty(\mathbf{x}) \tag{5.15}$$

where α_N *is the performance index, i.e., degree of suboptimality.*

The asymptotic stability can be ensured using Assumption 5.1 as shown in the following theorem.

Theorem 5.1. *Consider an NMPC problem satisfying Assumption 5.1 and let the performance index α_N be governed by*

$$\alpha_N := 1 - \frac{(\gamma_N - 1)\prod_{k=2}^{N}(\gamma_k - 1)}{\prod_{k=2}^{N}\gamma_k - \prod_{k=2}^{N}(\gamma_k - 1)}. \tag{5.16}$$

Then, if $\alpha_N > 0$, the relaxed Lyapunov inequality

$$V_N(f_\delta(\mathbf{x}, \mu_N(x))) \leq V_N(\mathbf{x}) - \alpha_N \ell(\mathbf{x}, \mu_N(x)) \tag{5.17}$$

holds for all $\mathbf{x} \in \mathbf{X}$ and the NMPC closed loop with prediction horizon N is asymptotically stable. Additionally, Inequality (5.15) holds.

For the proof of Theorem 5.1, see [58] and [49, Chapter 6]. The upper and lower bounds in A5.1.2 can be simply guaranteed as given in [49], however, ensuring A5.1.1 is not straightforward since the derivation of the growth bound $\gamma_i, i \in \mathbb{N}$ is generally sophisticated as shown in 5.5.

5.4.1.2 Recursive Feasibility

In order to guarantee that the proposed NMPC scheme is well defined, it is crucial that at each time step k there exists an admissible control sequence for the closed-loop state $\mathbf{x}_k \coloneqq x_{\mu_N}(k, \mathbf{x}_0)$ for all $k \in \mathbb{N}$, i.e., $\mathcal{U}^N(\mathbf{x}_k) \neq \emptyset$. However, it is difficult to ensure that when there are no terminal constraints or costs. Therefore, we first show the necessary conditions for recursive feasibility in the following definition then show how to ensure it in the following theorem.

Definition 5.3. *Consider an NMPC scheme with optimization horizon N and the feasible set $\mathbf{X}_N$ defined as*

$$\mathbf{X}_N \coloneqq \{\mathbf{x}_k \in \mathbf{X} \mid \mathcal{U}^N(\mathbf{x}_k) \neq \emptyset\} \tag{5.18}$$

and a set $B \subseteq \mathbf{X}_N$. The scheme is called recursively feasible on B, if for each $\mathbf{x}_k \in B$ and each optimal control sequence $u^* \in \mathcal{U}^N(\mathbf{x}_k)$ of (5.11) the condition $x_{u^*}(1, \mathbf{x}_k) \in B$ holds.

The concept of recursive feasibility requires that for each optimal admissible trajectory starting in B remains in B for at least one step. The recursive feasibility can be ensured by means of stability as shown in the following theorem.

Theorem 5.2. *Let there be an NMPC scheme with a state constraint set $\mathbf{X}$ and let Assumption 5.1 hold on $\mathbf{X}_N$ as given by (5.18). Then for each $c > 0$ there exists $N_c > 0$ such that the NMPC algorithm is recursively feasible on the set $\{\mathbf{x}_k \in \mathbf{X}_N \mid V_{N_c} \leq c\}$.*

For the proof of the theorem, we refer to [64, section 5.2]. Therefore, if we can derive a bounding sequence γ_i that ensures A5.1.1 and find the stabilizing prediction horizon N from (5.16) that satisfies $\alpha_N \in (0, 1]$, we can guarantee the asymptotic stability and recursive feasibility of Algorithm 5.1 on the set $\{\mathbf{x}_k \in \mathbf{X}_N \mid V_{N_c} \leq c\}$, as will be shown in the following section.

5.5 Stability Analysis of the Quadrotor MAV and the Growth Function

In this section, we will derive the growth bound γ_i in Assumption 5.1 for the MAV dynamic model with drag force (the derivation of the growth bound for the conventional model is similarly given in Appendix C) and find the shortest possible (minimal) prediction horizon N that stabilizes the NMPC scheme proposed in Algorithm 5.1.

5.5.1 Derivation of the growth bound

Without loss of generality, the desired state is chosen as the hovering point $\mathbf{x}_r = \begin{bmatrix} x_{r,1} & x_{r,2} & x_{r,3} & \mathbf{0}_{1\times5} \end{bmatrix}^T$, where $\mathbf{0}_{1\times5}$ is a 1×5 zero matrix. For this purpose, the running cost (5.2) is tailored as follows,

$$\ell(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{8} q_i \left(x_i - x_{r,i}\right)^2 + r_1(u_1 - \tilde{u}_1)^2 + r_2(u_2 - \tilde{u}_2)^2 + r_3(u_3 - \tilde{u}_3)^2. \tag{5.19}$$

which is the key contribution in this chapter to derive the growth bound in (5.13) and satisfy the asymptotic stability conditions presented in Theorem 5.1 in 5.4.1, where

$$\tilde{u}_1 = \mathrm{g}\cos(x_7)\cos(x_8) + d_z x_6, \quad \tilde{u}_2 = \frac{\bar{\tau}_\phi}{\bar{K}_\phi} x_7, \quad \tilde{u}_3 = \frac{\bar{\tau}_\theta}{\bar{K}_\theta} x_8.$$

Using (5.6) and (5.19), it can be shown that $(u_1 - \tilde{u}_1)$ penalizes the acceleration along Z_B, $(u_2 - \tilde{u}_2)$ penalizes the roll rate, $(u_3 - \tilde{u}_3)$ penalizes the pitch rate, $q_i \in \mathbb{R}_{\geq 0}$, $i = 1, 2, \cdots, 8$ are the i^{th} diagonal elements of the matrix $\mathbf{Q}$, r_1, r_2, and $r_3 \in \mathbb{R}_{\geq 0}$ are the i^{th} diagonal elements of the matrix $\mathbf{R}_u$.

The derivation of the growth bound γ_i given in Assumption 5.1 is, in general difficult, and one of the ways to obtain it is by constructing a summable sequence $c_j \subseteq \mathbb{R}_{\geq 0},\ j \in \mathbb{N}_0$, where $\sum_{j=0}^{\infty} c_j < \infty$, such that $\gamma_i = \sum_{j=0}^{i-1} c_j,\ i \in \mathbb{N}_{\geq 2}$. Additionally, for an admissible control actions $u_{\mathbf{x}_0} = \mathbf{u}_{\mathbf{x}_0}(j),\ j \in \mathbb{N}_0$, the sequence c_j satisfies the following inequality

$$\ell(x_{u_{\mathbf{x}_0}}(j; \mathbf{x}_0), \mathbf{u}_{\mathbf{x}_0}(j)) \leq c_j \left\|\mathbf{x}_0 - \mathbf{x}_r\right\|_{\mathbf{Q}}^2 \tag{5.20}$$

for all $j \in \mathbb{N}_0$ and $\mathbf{x}_0 \in \mathbf{X}$, see [50, section 3] for more details. The computation of the sequence c_j and thus the growth bound γ_i is governed by the following proposition.

Proposition 5.1. *Suppose that the penalty parameters of the control inputs in* (5.19) *are satisfying*

$$r_1 \leq \sigma q_6, \quad r_2 \leq \sigma \bar{K}_\phi^2 q_7, \quad r_3 \leq \sigma \bar{K}_\theta^2 q_8 \tag{5.21}$$

with weighting ratio $\sigma \in \mathbb{N}$. Therefore, condition (5.13) holds with $\gamma_i = \sum_{j=0}^{i-1} c_j,\ i \in \mathbb{N}_{\geq 2}$, where the sequence c_j is governed by

$$c_j := \left[\left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho} \right)^2 + \frac{\sigma}{\delta^2 \lambda^2} \left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2 \right] \tag{5.22}$$

for $j \in \{0, 1, \cdots, \lambda - 1\}$, where $\lambda \in \mathbb{N}_{\geq 2}$ is the number of steps required to perform any given maneuver, the exponent $\rho \in \mathbb{Q}_{\geq 0}$ is adjusted for different trajectory shapes,.

Proof. The quadrotor MAV may move in straight lines, curves, or lattice shape [112] based on the initial and final states, prediction horizon, and sampling time. Therefore, the trajectory for each initial state $\mathbf{x}_0 \in \mathbf{X}$ to the reference state $\mathbf{x}_r$ is defined as follows;

$$\begin{aligned} \mathbf{x}[j] &= \left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho} \right) \mathbf{x}_0 + \left(\frac{j^\rho}{\lambda^\rho} \right) \mathbf{x}_r \\ &= \mathbf{x}_0 + \left(\frac{j^\rho}{\lambda^\rho} \right) (\mathbf{x}_r - \mathbf{x}_0) \qquad \forall j \in \{0, 1, .., \lambda - 1\} \end{aligned} \tag{5.23}$$

where $\lambda \in \mathbb{N}_{\geq 2}$ is the number of steps required to perform the maneuver and is chosen large enough such that the constraints (5.9) are satisfied for all $\mathbf{x}_0 \in \mathbf{X}$, and the exponent $\rho \in \mathbb{Q}_{\geq 0}$ depends on the shape of the trajectory. Substitute (5.8) into (5.23) to find the open-loop control inputs ($\mathbf{u}_{\mathbf{x}_0}$) required to perform the maneuver. The first control input u_1 was given in (5.8) as

$$x_6[j+1] = x_6[j] - \delta R_{33}\ \mathrm{g} + \delta u_1[j] - \delta d_z x_6[j]$$

where $R_{33} = \cos(x_7)\cos(x_8)$ is the element in the third row and third column in the rotation matrix $\mathbf{R}$. Thus, control input $u_{1_{\mathbf{x}_0}}$ for all $\mathbf{x}_0 \in \mathbf{X}$ (or u_1 for the sake of simplicity) can be calculated as follows:

$$\begin{aligned}\left(\frac{\lambda^\rho-(j+1)^\rho}{\lambda^\rho}\right)x_{0,6}-(1-\delta d_z)\left(\frac{\lambda^\rho-j^\rho}{\lambda^\rho}\right)x_{0,6}\\ =-\delta R_{33}\mathrm{g}+\delta u_1.\end{aligned} \tag{5.24}$$

Replacing $(j+1)^\rho$ with the binomial expansion, (5.24) reduces to

$$\frac{-\sum_{i=0}^{\rho-1}{}^{\rho}C_i\ j^i+\delta d_z(\lambda^\rho-j^\rho)}{\delta\lambda^\rho}x_{0,6}=u_1-R_{33}\mathrm{g}$$

where ${}^{\rho}C_i=\frac{\rho!}{i!(\rho-i)!}$, that yields,

$$u_1=\frac{-\sum_{i=0}^{\rho-1}{}^{\rho}C_i\ j^i+\delta d_z(\lambda^\rho-j^\rho)}{\delta\lambda^\rho}x_{0,6}+R_{33}\mathrm{g} \tag{5.25}$$

for all $j\in\{0,\cdots,\lambda-1\}$. The second control input u_2 was given in (5.8) as

$$x_7[j+1]=(1-\delta\bar{\tau}_\phi)x_7[j]+\delta\bar{K}_\phi u_2[j].$$

Thus, u_2 is calculated as follows:

$$\left(\frac{\lambda^\rho-(j+1)^\rho}{\lambda^\rho}\right)x_{0,7}-(1-\delta\bar{\tau}_\phi)\left(\frac{\lambda^\rho-j^\rho}{\lambda^\rho}\right)x_{0,7}=\delta\bar{K}_\phi u_2$$

reduces to

$$\frac{-\sum_{i=0}^{\rho-1}{}^{\rho}C_i\ j^i+\delta\bar{\tau}_\phi(\lambda^\rho-j^\rho)}{\lambda^\rho}x_{0,7}=\delta\bar{K}_\phi u_2.$$

Above simplifies as,

$$u_2=\frac{-\sum_{i=0}^{\rho-1}{}^{\rho}C_i\ j^i+\delta\bar{\tau}_\phi(\lambda^\rho-j^\rho)}{\delta\bar{K}_\phi\lambda^\rho}x_{0,7} \tag{5.26}$$

for all $j\in\{0,\cdots,\lambda-1\}$. The calculation of the third input u_3 is similar to u_2 and given by

$$u_3 = \frac{-\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i + \delta\bar{\tau}_\theta(\lambda^\rho - j^\rho)}{\delta\bar{K}_\theta\lambda^\rho} x_{0,8} \tag{5.27}$$

for all $j \in \{0, \cdots, \lambda - 1\}$. Applying (5.23), (5.25), (5.26), and (5.27) into (5.19) yields the running costs (5.28) along the resulting open-loop trajectories.

$$\begin{aligned}\ell(x_{u_{\mathbf{x}_0}}(j;\mathbf{x}_0), \mathbf{u}_{\mathbf{x}_0}(j)) = \left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho}\right)^2 \left(\sum_{i=1}^{8} q_i \left(x_{0,i} - x_{r,i}\right)^2\right) + r_1 \left(\frac{\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i}{\delta\lambda^\rho}\right)^2 x_{0,6}^2 \\ + r_2 \left(\frac{\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i}{\delta\bar{K}_\phi\lambda^\rho}\right)^2 x_{0,7}^2 + r_3 \left(\frac{\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i}{\delta\bar{K}_\theta\lambda^\rho}\right)^2 x_{0,8}^2\end{aligned} \tag{5.28}$$

To this end, the bounding sequence c_j can be found by bounding the running costs (5.28) such that condition (5.20) is satisfied. The second term in (5.28) can be bounded using condition (5.21) and by recalling that $j \leq \lambda$, as follows:

$$r_1 \left(\frac{\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i}{\delta\lambda^\rho}\right)^2 x_{0,6}^2 \leq r_1 \left(\frac{\lambda^{\rho-1}\sum_{i=0}^{\rho-1} {}^{\rho}C_i}{\delta\lambda^\rho}\right)^2 x_{0,6}^2 \leq \sigma q_6 \frac{\left(\sum_{i=0}^{\rho-1} {}^{\rho}C_i\right)^2}{\delta^2\lambda^2} x_{0,6}^2 \tag{5.29}$$

Moreover, the third term in (5.28) can be bounded in the same manner as follows:

$$r_2 \left(\frac{\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i}{\delta\bar{K}_\phi\lambda^\rho}\right)^2 x_{0,7}^2 \leq r_2 \frac{1}{\bar{K}_\phi^2} \frac{\left(\sum_{i=0}^{\rho-1} {}^{\rho}C_i\right)^2}{\delta^2\lambda^2} x_{0,7}^2 \leq \sigma q_7 \frac{\left(\sum_{i=0}^{\rho-1} {}^{\rho}C_i\right)^2}{\delta^2\lambda^2} x_{0,7}^2 \tag{5.30}$$

Similarly, the fourth term in (5.28) is bounded as follows:

$$r_3 \left(\frac{\sum_{i=0}^{\rho-1} {}^{\rho}C_i \; j^i}{\delta\bar{K}_\theta\lambda^\rho}\right)^2 x_{0,8}^2 \leq r_3 \frac{1}{\bar{K}_\theta^2} \frac{\left(\sum_{i=0}^{\rho-1} {}^{\rho}C_i\right)^2}{\delta^2\lambda^2} x_{0,8}^2 \leq \sigma q_8 \frac{\left(\sum_{i=0}^{\rho-1} {}^{\rho}C_i\right)^2}{\delta^2\lambda^2} x_{0,8}^2 \tag{5.31}$$

As a result, the running costs (5.28) can be estimated by

$$
\begin{aligned}
\ell(x_{u_{\mathbf{x}_0}}(j;\mathbf{x}_0),\mathbf{u}_{\mathbf{x}_0}(j)) &\leq \left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho}\right)^2 \|\mathbf{x}_0-\mathbf{x}_f\|_Q^2 + \frac{\sigma}{\delta^2\lambda^2}\left(\sum_{i=0}^{\rho-1} {}^\rho C_i\right)^2 \sum_{i=6}^{8} q_i x_{0,i} \\
&\leq \left[\left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho}\right)^2 + \frac{\sigma}{\delta^2\lambda^2}\left(\sum_{i=0}^{\rho-1} {}^\rho C_i\right)^2\right] \|\mathbf{x}_0-\mathbf{x}_f\|_Q^2
\end{aligned}
\tag{5.32}
$$

for all $j \in \{0, \cdots, \lambda-1\}$. Therefore, the bounding sequence c_j in (5.20) can be attained as in (5.22). □

Finally, the growth bound γ_k, $k \in \mathbb{N}_0$ can be obtained as given in Proposition 5.1 by

$$
\gamma_k := \sum_{j=0}^{k-1} c_j = \sum_{j=0}^{k-1}\left[\left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho}\right)^2 + \frac{\sigma}{\delta^2\lambda^2}\left(\sum_{i=0}^{\rho-1} {}^\rho C_i\right)^2\right]
$$

which can be written as

$$
\gamma_k = \frac{1}{\lambda^{2\rho}}\sum_{j=0}^{k-1}\left[(\lambda^\rho - j^\rho)^2 + \frac{\sigma}{\delta^2}\lambda^{2\rho-2}\left(\sum_{i=0}^{\rho-1} {}^\rho C_i\right)^2\right]. \tag{5.33}
$$

It can be noticed from (5.33) that the growth bound γ_k depends only on the sampling time δ, the exponent value ρ, the number of steps λ, and the weight ratio σ.

5.5.2 Calculation of the minimal stabilizing horizon

In this section, we will calculate the shortest stabilizing prediction horizon using the growth bound γ_k such that $\alpha_N \in (0,1]$ holds. We first expand (5.33) as follows:

$$
\gamma_k = \frac{1}{\lambda^{2\rho}}\sum_{j=0}^{k-1}\left[\lambda^{2\rho} - 2\lambda^\rho j^\rho + j^{2\rho} + \frac{\sigma}{\delta^2}\lambda^{2\rho-2}\left(\sum_{i=0}^{\rho-1} {}^\rho C_i\right)^2\right]
$$

and for the sake of simplicity and for tractable analysis, without loss of generality, assume $\rho = 1$, that yields

$$\gamma_k = \frac{1}{\lambda^2} \sum_{j=0}^{k-1} \left[\lambda^2 - 2\lambda j + j^2 + \frac{\sigma}{\delta^2} \right] = \frac{1}{\lambda^2} \left[\sum_{j=0}^{k-1} j^2 - 2\lambda^2 \sum_{j=0}^{k-1} j + (\lambda^2 + \frac{\sigma}{\delta^2})k \right]. \tag{5.34}$$

Using the sum formulas

$$\sum_{j=0}^{k-1} j^2 = \frac{k(k-1)(2k-1)}{6} \quad \text{and} \quad \sum_{j=0}^{k-1} j = \frac{k(k-1)}{2} \tag{5.35}$$

γ_k can be formulated as

$$\gamma_k = \frac{1}{\lambda^2} \left[k\lambda^2 - (k^2 - k)\lambda + (\frac{k^3}{3} - \frac{k^2}{2} + \frac{k}{6} + \frac{\sigma}{\delta^2} k) \right] \tag{5.36}$$

Thus, the following theorem defines the shortest prediction horizon required to stabilize the proposed NMPC scheme.

Theorem 5.3. *Let the prediction horizon $N = 4$, the sampling time $\delta > 0$, and the weight ratio $\sigma > 0$ be given. If there exist maneuver steps of length λ^* such that for all $\lambda > \lambda^*$ the performance index α_N governed by (5.16) satisfies $\alpha_4 > 0$, thus, the closed-loop asymptotic stability of the quadrotor MAV under NMPC scheme without terminal costs or constraints is ensured for the shortest possible prediction horizon. Additionally, the inequality*

$$V_\infty^{\mu_4} \leq \alpha_4^{-1} V_\infty(\mathbf{x}) \tag{5.37}$$

holds.

Proof. The motion primitives of the quadrotor MAV can be generated in the quadrotor's jerk [113]. Thus, in order to recover the thrust and attitude rates inputs from such a thrice differentiable trajectory, a prediction horizon of 4 is required, i.e., $N \geq 4$. Therefore, the performance estimate α_N given in (5.16) for $N = 4$ is formulated as

$$\alpha_4 = 1 - \frac{(\gamma_2 - 1)(\gamma_3 - 1)(\gamma_4 - 1)^2}{\gamma_2\gamma_3\gamma_4 - (\gamma_2 - 1)(\gamma_3 - 1)(\gamma_4 - 1)}. \tag{5.38}$$

Thus, the conditions for asymptotic stability to be ensured can be determined as

$$0 < \frac{(\gamma_2 - 1)(\gamma_3 - 1)(\gamma_4 - 1)^2}{\gamma_2\gamma_3\gamma_4 - (\gamma_2 - 1)(\gamma_3 - 1)(\gamma_4 - 1)} < 1. \tag{5.39}$$

The left inequality in (5.39) can be ensured with $\gamma_k > 1$. Thus, selecting λ as

$$\lambda > \lambda^* := \frac{(\frac{k^3}{3} - \frac{k^2}{2} + \frac{k}{6} + \frac{\sigma}{\delta^2}k)}{k^2 - k} > 0 \qquad \forall k \geq 2 \tag{5.40}$$

yields $\gamma_K > k > 1$, that ensures the left inequality of (5.39). In addition, substituting the value of λ defined in (5.40) into (5.39) ensures the right inequality. Therefore, $N \geq 4$ ensures the asymptotic stability of the system. If λ is selected as $\lambda = 2\lambda^*$, the performance index becomes

$$\alpha_4 = \frac{360(\frac{\sigma}{\delta^2})^3 + 766(\frac{\sigma}{\delta^2})^2 + 451(\frac{\sigma}{\delta^2}) + 70}{72(\frac{\sigma}{\delta^2})^4 + 444(\frac{\sigma}{\delta^2})^3 + 796(\frac{\sigma}{\delta^2})^2 + 454(\frac{\sigma}{\delta^2}) + 70} \tag{5.41}$$

Equation (5.41) shows that the performance index depends only on the weight parameter σ and sampling time δ. The index α_4 is always positive and converges to 1 as σ/δ^2 is chosen small, as shown in Fig. 5.4. $\square$

Theorem 5.3 shows that the performance of NMPC with infinite prediction horizon can be approached by reducing σ/δ^2 regardless of the shortest prediction horizon, i.e., $N = 4$, used in the NMPC scheme proposed in Algorithm 5.1.

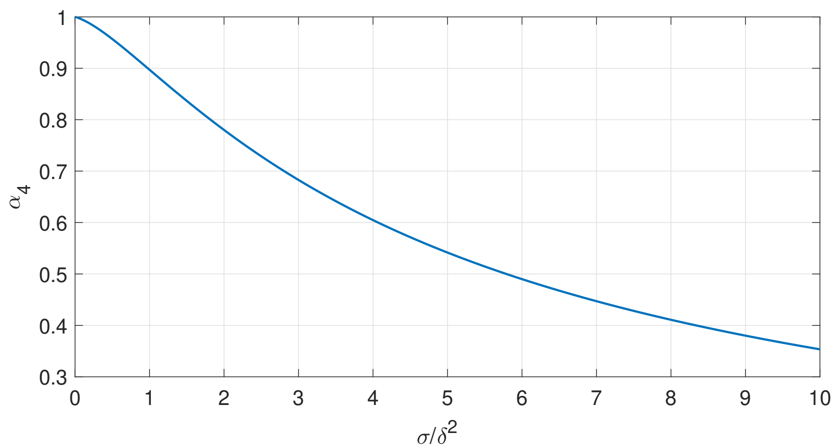


Fig. 5.4: Plot of α_4 vs. σ/δ^2. The performance index α_4, at $N = 4$, converges to 1 at small values of the ratio σ/δ^2

5.6 Results

This section presents several numerical simulations and real-time lab experiments to validate the proposed NMPC scheme for the MAV dynamic model with drag force. In the numerical simulations, the dynamic model (5.6) is used as the system of the quadrotor MAV, while the AscTec Hummingbird Quadrotor [114] is used for all laboratory experiments. In order to obtain the solution of the OCP for both numerical and lab experiments, the dynamic model in (5.8) is implemented for the state prediction process and the running cost in (5.19) is used to construct the objective function (5.10), as shown in Fig. 5.5. Only parameters that need tuning are the q_i $(i = 1, 2, \cdots, 8)$ of the $\mathbf{Q}$ weighting matrix given that r_i $(i = 1, 2, 3)$ parameters are constrained by (5.21), and σ/δ^2 is selected very small for higher performance index, as observed by Fig. 5.4. For all test validations, the parameters in system (5.6) are given in Table 5.1, in which the time constant and gain of the roll and pitch angles were extracted using the open-source code of the system identification of the attitude dynamics of the Hummingbird helicopter, provided by [52][1]. The drag coefficient matrix

[1]available at `https://github.com/ethz-asl/mav_system_identification`, and last accessed on December 2020.

Table 5.1: Quadrotor MAV parameters

Parameter	mass (kg)	τ_ϕ (s)	K_ϕ	τ_θ (s)	K_θ
Value	0.645	0.0914	0.7551	0.0984	0.7226

is given as $\mathbf{D} = \mathrm{diag}(0.01, 0.01, 0)$. The control inputs given in (5.9) are constrained as

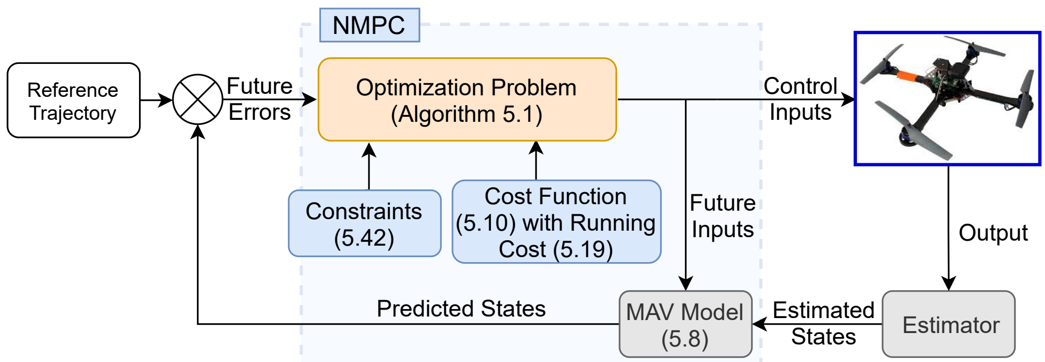


Fig. 5.5: Block diagram of the proposed NMPC and the closed-loop system.

$$\begin{pmatrix} 5\,\mathrm{m/s^2} \\ -10^\circ \\ -10^\circ \end{pmatrix} \leq \begin{pmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{pmatrix} \leq \begin{pmatrix} 15\,\mathrm{m/s^2} \\ 10^\circ \\ 10^\circ \end{pmatrix}. \tag{5.42}$$

The symbolic toolbox CasADi [115] is used to set up the optimal control problem (OCP) proposed in Algorithm 5.1. CasADi was selected due to its high efficiency in implementing nonlinear optimizations and flexibility to be used from C++, Python or MATLAB/Octave. The direct multiple-shooting discretization method [116] is employed to turn the OCP into a nonlinear programming problem (NLP), see [115, section 5.4] for more details. Then, the Interior Point OPTimizer (IPOPT), which is already interfaced with CasADi, is used to solve this NLP. Two main control problems are evaluated, (a) point stabilization (hovering) and (b) trajectory tracking problems. In the point stabilization problem, the algorithm terminates when the following stopping criterion is met

$$\|\mathbf{x}(k) - \mathbf{x}_r\| \le \epsilon,\ \forall k \in \mathbb{N} \tag{5.43}$$

where $\epsilon > 0$ is the error value, otherwise, the process will terminate after reaching the maximum time t_{max}.

5.6.1 Numerical validation

5.6.1.1 Proposed NMPC scheme

The following simulations were conducted on MATLAB. The simulation was run for twelve various initial positions while stabilizing the MAV to the reference point $\mathbf{x}_r = \begin{bmatrix} 0 & 0 & 1 & \mathbf{0}_{1\times5} \end{bmatrix}^T$. The prediction horizon was set to $N = 4$ by means of Theorem 5.3 and the sampling time to $\delta = 0.1$ s. Additionally, the weight ratio σ was selected such that $\sigma/\delta^2 = 0.1$, thus the corresponding performance index is $\alpha_4 = 0.994$. The parameters of the weighting matrix $\mathbf{Q}$ are selected similar to [52] with some fine tuning as $\mathbf{Q} = \mathrm{diag}(100, 100, 120, 80, 80, 100, 10, 10)$. The stopping error value in (5.43) was selected as $\epsilon = 1$ mm and the maximum time $t_{max} = 120$ s. The trajectories of the twelve runs are shown in Fig. 5.6a. The results show that the MAV may go from the initial position to the final position in straight lines, curves, or lattice shape as proposed in the proof of Proposition 5.1. Moreover, the proposed NMPC algorithm with $N = 4$ is able to stabilize the MAV from any random initial state while satisfying the OCP's input constraints, given in (5.42), for all time steps, as shown in Fig. 5.6b, that verifies the fulfillment of the conditions of Theorem 5.2.

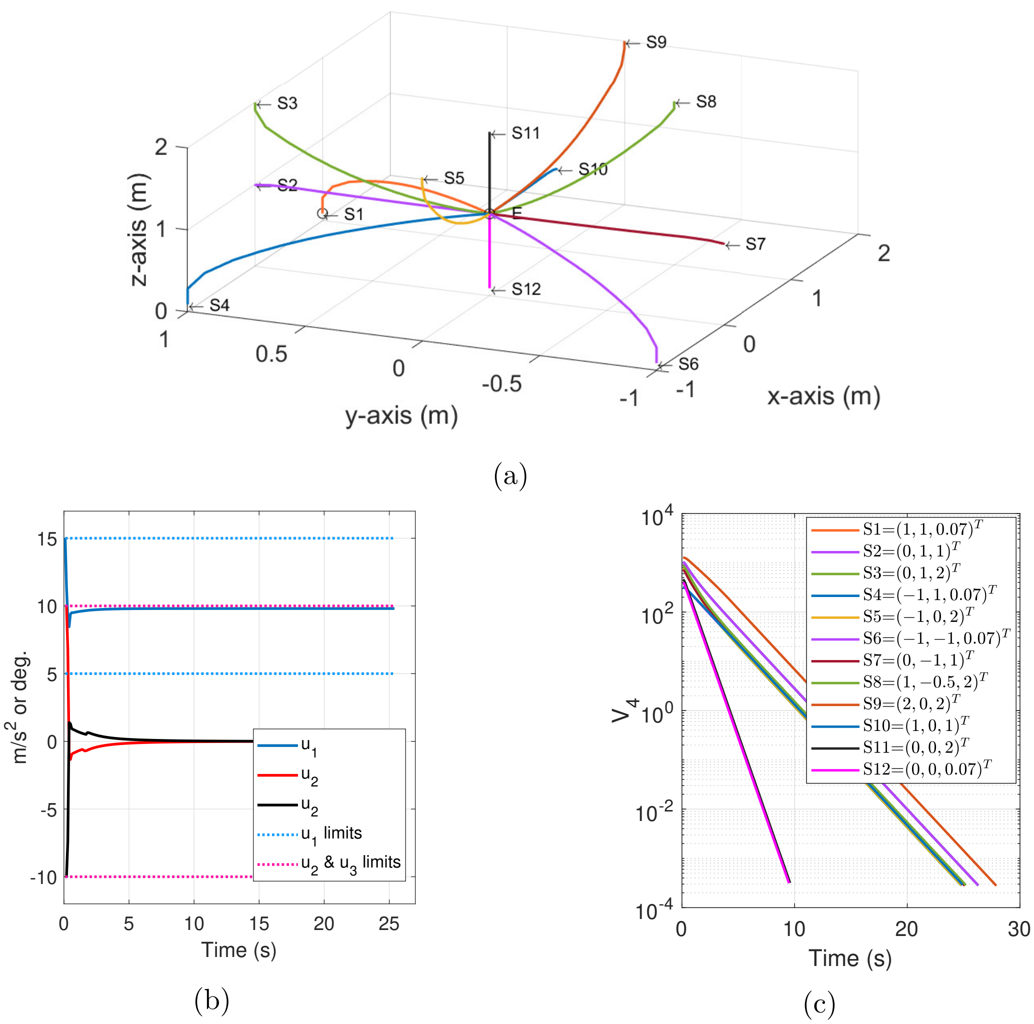
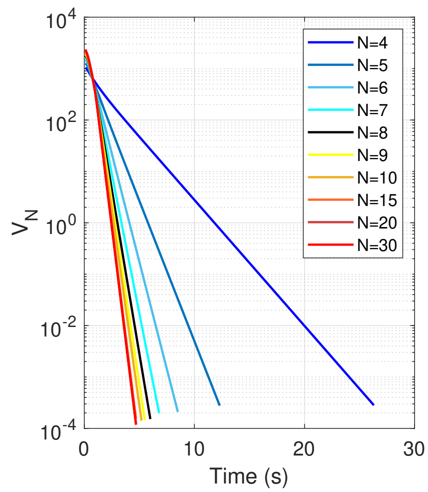
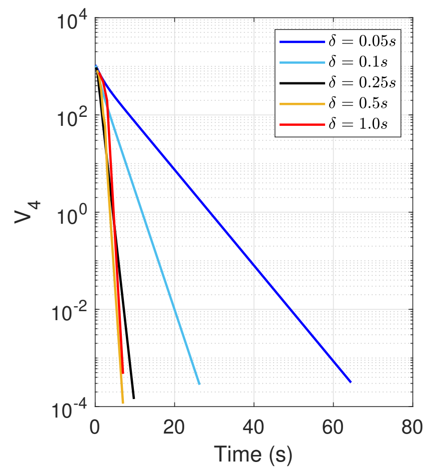


Fig. 5.6: (a) 3D trajectories of twelve simulation runs starting from various initial positions (Si) for i=1,2,..,12 and stabilizing at the final position (E). (b) Feedback control inputs and the input limits given in (5.42) for one simulation run at initial states $\mathbf{x}_0 = (1, 1, 0.07, \mathbf{0}_{1\times 6})^T$. (c) Evolution of the value function V_4 for the twelve various initial positions.

The asymptotic stability of the system can be verified by studying the evolution of the value function V_4 at $N = 4$. Fig. 5.6c shows that V_4 is monotonically decreasing with time for all initial positions, which is leading to the conclusion that the conditions of Theorem 5.1 are successfully met and the system is asymptotically stable for any given initial position. The convergence rate of the MAV to the reference point depends on the prediction horizon N. Increasing the prediction horizon tends to a faster convergence rate and less convergence time, as shown in Fig. 5.7a, i.e., the convergence time is reduced by more than 50% when N increases from $N = 4$ to $N = 5$. Additionally, Fig. 5.7b shows the effect of changing the sampling time δ on the evolution of V_4 and the convergence rate. It is clearly seen that increasing δ decreases the convergence time.



(a)



(b)

Fig. 5.7: (a) Evolution of the value function for various prediction horizon lengths N for one simulation run at initial states $\mathbf{x}_0 = (1, 1, 0.07, \mathbf{0}_{1\times 6})^T$. (b) Evolution of the value function V_4 for various sampling times at the same initial states.

The performance of the proposed controller is also tested for trajectory tracking problems and the results show outstanding tracking performance with the shortest prediction horizon $N = 4$. Fig. 5.8 and Fig. 5.9 show the tracking results of circular and 8-shape trajectories, respectively. The root mean square error is less than 0.017 m for both trajectories.

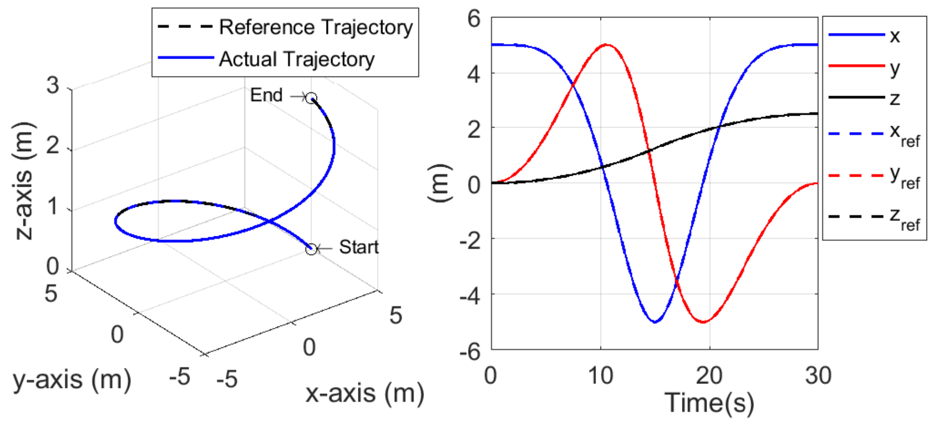


Fig. 5.8: Simulation performance of the circular trajectory tracking of the proposed NMPC scheme at $N = 4$ and $\delta = 0.1$ s. Left: 3D trajectories of the reference and actual trajectories. Right: the reference and actual position in x, y, and z axes.

5.6.1.2 Comparison with traditional NMPC

In this section, the performance of the proposed NMPC scheme without stabilizing costs or constraints is compared against the traditional NMPC scheme which requires addition of the terminal costs in its cost function (see [52][2] for more details). The cost function of the traditional NMPC is given by

$$J_N(\mathbf{x}_k, u) := \sum_{k=1}^{N} \left(\|\mathbf{x} - \mathbf{x}_r\|^2_{\mathbf{Q}} + \|\mathbf{u} - \mathbf{u}_r\|^2_{\mathbf{R}_u} \right) + \|\mathbf{x}_N - \mathbf{x}_r\|^2_{\mathbf{Q}_N} \tag{5.44}$$

in which $\mathbf{Q}_N \in \mathbb{R}^{n \times n}$ is the terminal weight matrix that needs to be computed by solving the following Algebraic Riccati Equation

$$\mathbf{Q}_N \mathbf{A} + \mathbf{A}^T \mathbf{Q}_N - \mathbf{Q}_N \mathbf{B} \mathbf{R}_u^{-1} \mathbf{B}^T \mathbf{Q}_N + \mathbf{Q} = 0 \tag{5.45}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the state and input matrices of the linearized system.

For the point stabilization problem, the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ of the traditional NMPC are selected as in [52] while the $\mathbf{Q}$ matrix of the proposed NMPC is

[2]The work in [52] was considered in our comparative study because it uses an NMPC with terminal costs for the control of the full dynamics of the quadrotor, same as our dynamic model.

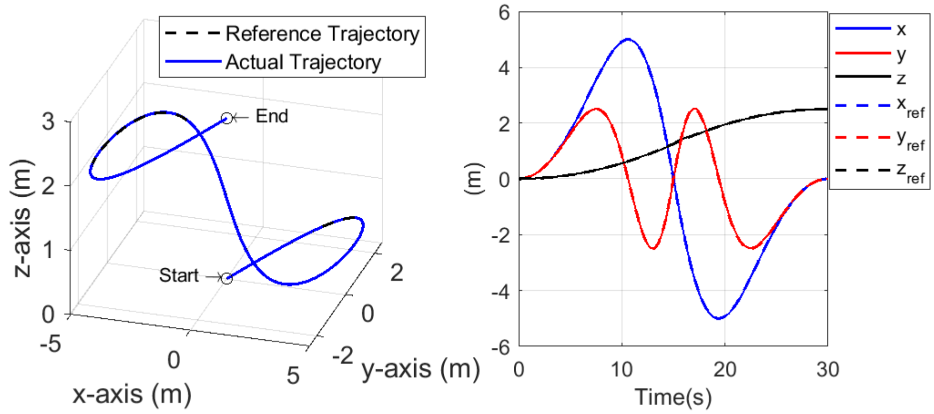
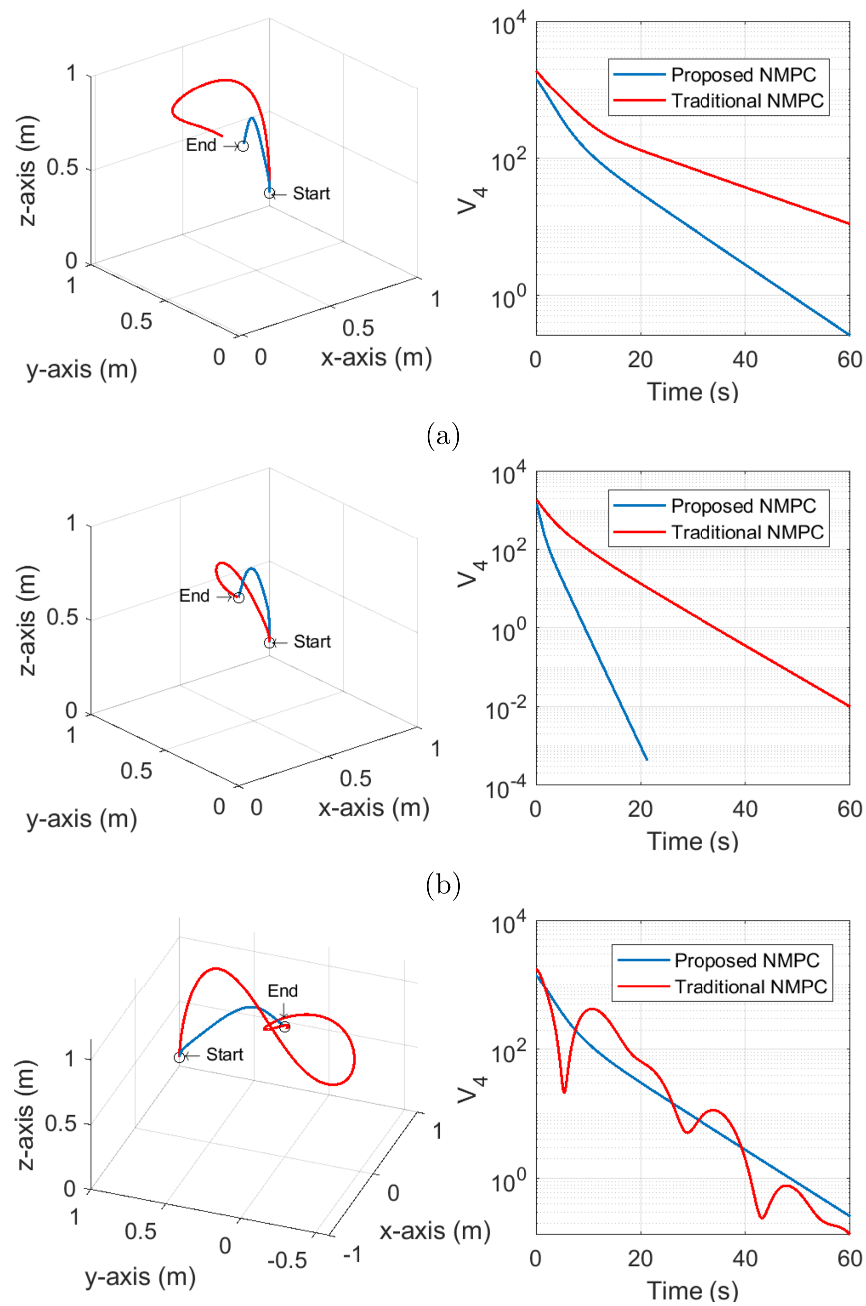


Fig. 5.9: Simulation performance of the 8-shape trajectory tracking of the proposed NMPC scheme at $N = 4$ and $\delta = 0.1$ s. Left: 3D trajectories of the reference and actual trajectories. Right: the reference and actual position in x, y, and z axes.

selected similar to [52] as given in Section 5.6.1. The comparison results are given in Table 5.2. The table shows that the proposed NMPC has asymptotically stabilized the MAV at the reference point while achieving condition (5.43) in finite number of time steps at various prediction horizon N and sampling time δ except for $N = 4$ and $\delta = 0.01$, where the simulation terminated at t_{max} while the MAV was very close to the reference point. However, the traditional NMPC required higher number of iterations to stabilize the MAV (more than the double of the number of iteration at some cases). In addition, at low sampling time, the traditional NMPC could not achieve the stopping criteria in three cases, as shown in Table 5.2, and failed to stabilize the MAV at $\delta = 0.01$ s and $N = 4$. Two samples of the comparison results have been illustrated in Fig. 5.10a and 5.10b. The figures show that the proposed NMPC has faster convergence rate, and smoother and shorter trajectory than the traditional scheme. Moreover, the traditional NMPC controller was tested with the same weighting parameters of the proposed NMPC to accelerate the convergence rate, but the traditional NMPC has failed to asymptotically stabilize the system as well, as shown in Fig. 5.10c, where V_4 is no longer monotonically decreasing.



(a)

(b)

(c)

Fig. 5.10: Simulation point stabilization comparison of the proposed and traditional NMPC schemes at (a) $N = 4$ and $\delta = 0.01$ s. (b) $N = 4$ and $\delta = 0.05$ s. (c) The same as (a) but after tuning the weighting parameters of the traditional NMPC to be similar to the proposed NMPC to accelerate the convergence rate. The left column is the 3D view and the right column is the value function.

Additionally, in order to evaluate the superiority of the proposed NMPC without terminal costs or constraints, the traditional NMPC was tested after disabling the terminal costs in (5.44), i.e., $\mathbf{Q}_N \equiv \mathbf{0}$. The traditional NMPC without terminal costs is tested twice, one with the same weighting parameters as the proposed NMPC and the other as the traditional NMPC with terminal costs. The results are given in Fig. 5.11. In both cases, the systems failed to ensure the asymptotic stability since the decreasing rate of V_4 in Fig. 5.11a is very small and failed to stabilize the system, and the V_4 in Fig. 5.11b is not monotonically decreasing, however, the MAV is converging to the reference point.

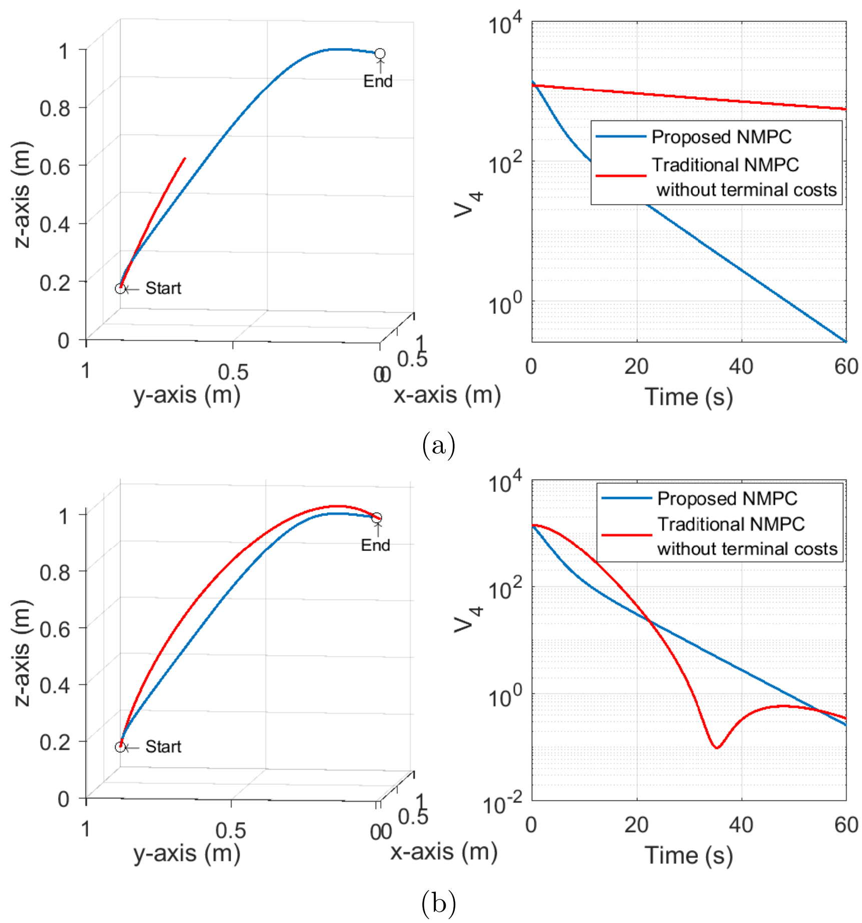
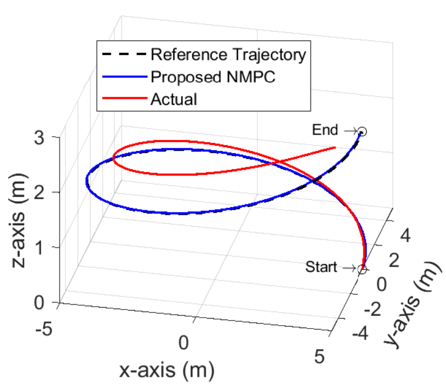
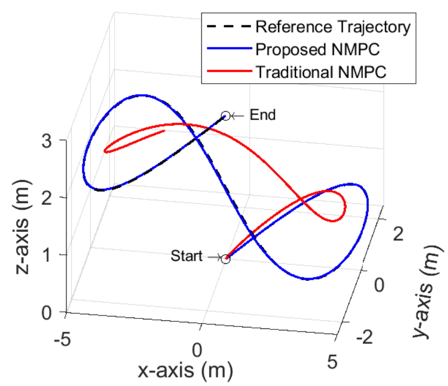


Fig. 5.11: Point stabilization performance of the proposed NMPC and traditional NMPC without terminal costs (terminal costs are disabled) at $N = 4$ and $\delta = 0.01$ s with weighting parameters as (a) the traditional NMPC with terminal costs. (b) the proposed NMPC.

Both controllers were evaluated for the trajectory tracking problems, where the weighting parameters are selected similar to [52]. Table 5.3 and Fig. 5.12 show the trajectory tracking performance of the circular and 8-shape trajectories. It can be observed that the proposed NMPC scheme is robust against changing the controller parameters (N and δ) and outperforms the traditional scheme in terms of the tracking accuracy and computation time of the MPC problem. Furthermore, at sampling times $\delta = 0.1$ s and $\delta = 0.05$ s, the proposed NMPC can achieve outstanding tracking performance with only $N = 4$ similar to the traditional NMPC with $N = 10$ or $N = 20$ while saving more than 40% of the computation time that saves more time and power for the MAV's CPU to accomplish other autonomy tasks. Similarly, at low sampling time, the proposed NMPC provides better tracking performance than the traditional NMPC without the need of tuning the controller parameters, however, the traditional NMPC (with $N = 4$) failed to stabilize the MAV with very high tracking errors.



(a)



(b)

Fig. 5.12: Simulation trajectory tracking comparison of the proposed and traditional NMPC schemes at $N = 4$ and $\delta = 0.01$ s for (a) circular trajectory (b) 8-shape trajectory. The proposed NMPC has superior tracking performance even at small prediction horizons and sampling times, however, he traditional NMPC has failed to track the reference trajectory.

Table 5.2: Simulation comparison results of the proposed NMPC and traditional NMPC for point stabilization problem. RMSE is the root mean square error in meters. # iter. is the number of iterations (time steps) required to stabilize the MAV at the reference point.

N	δ (s)	**Point Stabilization**			
		Proposed		Traditional	
		RMSE	# iter.	RMSE	# iter.
4	0.1	$0.0008 < \sigma$	97	$0.0009 < \sigma$	281
10	0.1	$0.0002 < \sigma$	31	$0.0007 < \sigma$	96
20	0.1	$0.0003 < \sigma$	29	$0.0007 < \sigma$	75
4	0.05	$0.0009 < \sigma$	424	0.004	1199
20	0.05	$0.0003 < \sigma$	59	$0.0007 < \sigma$	192
4	0.01	0.023	5999	0.137 m (Failed)	
10	0.01	$0.0009 < \sigma$	1568	0.033	5999
20	0.01	$0.0006 < \sigma$	670	0.001	5999

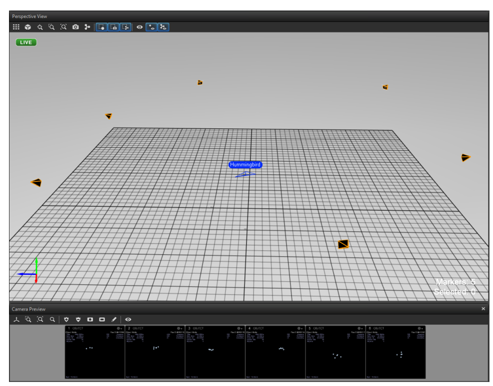
Table 5.3: Simulation comparison results of the proposed NMPC and traditional NMPC for two trajectory tracking problems, 1- circular trajectory, and 2- 8-shape trajectory. RMSE is the root mean square error in meters. T_{MPC} is the average computation time of the optimization process at each iteration in ms.

N	δ (s)	**Circular Trajectory**				**8-shape Trajectory**			
		Proposed		Traditional		Proposed		Traditional	
		RMSE	T_{MPC}	RMSE	T_{MPC}	RMSE	T_{MPC}	RMSE	T_{MPC}
4	0.1	0.005	8.3	0.042	8.7	0.017	8.0	0.057	8.9
10	0.1	0.008	9.9	0.008	10.6	0.013	9.8	0.015	10.9
20	0.1	0.012	12.4	0.013	13.9	0.019	12.3	0.02	13.6
4	0.05	0.008	8.0	0.185	9.2	0.015	7.9	0.197	8.5
20	0.05	0.007	12.0	0.008	13.2	0.011	12.3	0.013	12.8
4	0.01	0.09	9.0	3.045 m (Failed)		0.084	9.3	2.411 m (Failed)	
10	0.01	0.009	12.8	0.428	11.5	0.013	12.3	0.435	13.3
20	0.01	0.001	14.0	0.099	14.5	0.007	13.5	0.108	13.9

5.6.2 Experimental validation

The proposed NMPC scheme has been experimentally evaluated in the lab using the AscTec Hummingbird Quadrotor Helicopter [114], shown in Fig. 5.13a. The experimental platform including the proposed NMPC algorithm has been conducted on the Robot Operating System (ROS). The NMPC node was written in Python since it is compatible with CasADi toolbox. The feedback control actions are sent to the quadrotor via XBee Bluetooth communication as shown in Fig. 5.14. The OptiTrack motion capture system (with six cameras) has been used to provide the quadrotor pose as a feedback to the control system to guarantee accurate measurements since the OptiTrack system can accurately estimate the 6 DOF pose of the quadrotor at high rate up to 120 Hz. For this purpose, five markers have been mounted on the quadrotor to be detected by the OptiTrack system as shown in Fig. 5.13b. A Kalman filter has been used to estimate the velocity of the quadrotor from the measured position since the OptiTrack system does not provide tracked object velocity.

(a)



(b)

Fig. 5.13: (a) Instance of the Asctec Hummingbird quadrotor at hovering position with the proposed NMPC in the lab. Five markers are attached to the quadrotor for visual localization purpose. (b) A screenshot of the OptiTrack Motive with the six cameras around the arena and the quadrotor.

The lab experiments of the point stabilization problem were conducted at sampling

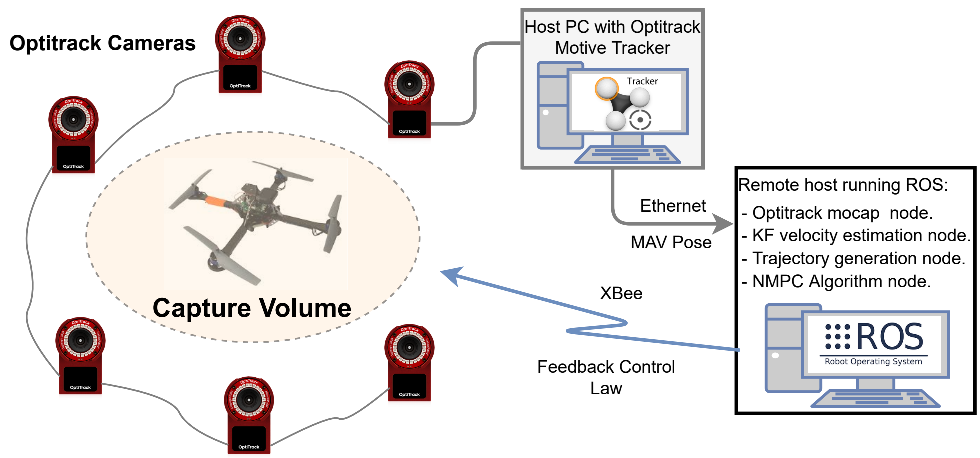
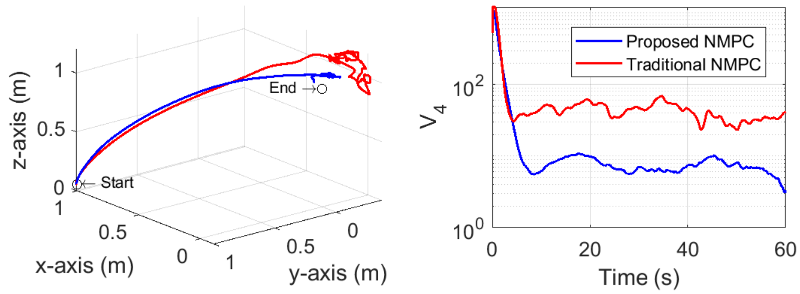
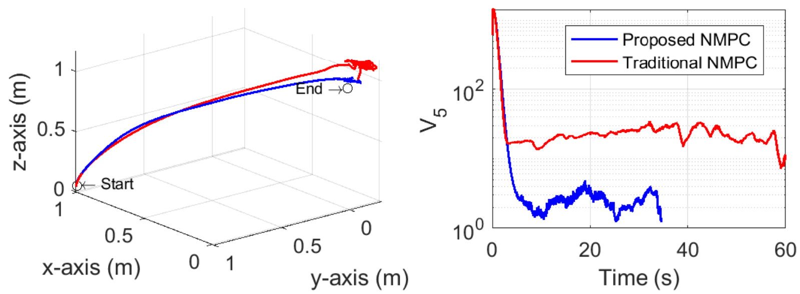


Fig. 5.14: The configuration of the OptiTrack motion capture system and the experimental setup.

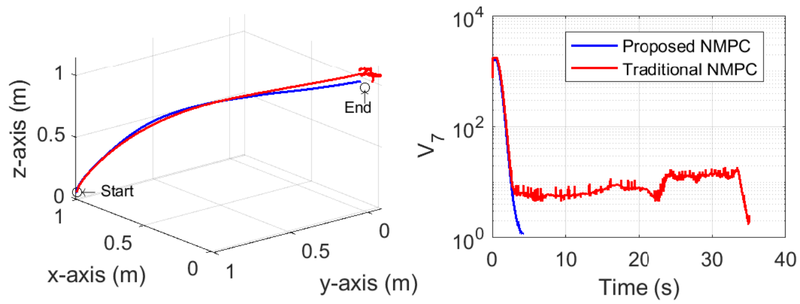
time $\delta = 0.1$ s and various prediction horizons. The weight ratio σ was selected as given in the numerical simulation 5.6.1. The stopping error value in (5.43) was selected as $\epsilon = 50$ mm and the maximum time $t_{max} = 60$ s. The results of four point stabilization experiments of the proposed and traditional schemes are given in Fig. 5.15 and Table 5.4. Both controllers reached the maximum time before achieving the stopping criteria at $N = 4$, as can be seen in Fig. 5.15a, however, the traditional controller provided large RMSE error than the proposed controller and failed to stabilize the MAV close to the reference point. The corresponding value function V_4 of both controllers is monotonically decreasing until some point then keeps almost constant that reflects the reason of the errors. Starting from $N = 5$, the proposed controller was able to stabilize the MAV and achieve the stopping criteria at faster convergence rate than the traditional controller, as shown in Fig. 5.15b, 5.15c, and 5.15d. An instance of the quadrotor under the proposed controller at hovering is shown in Fig. 5.13a.



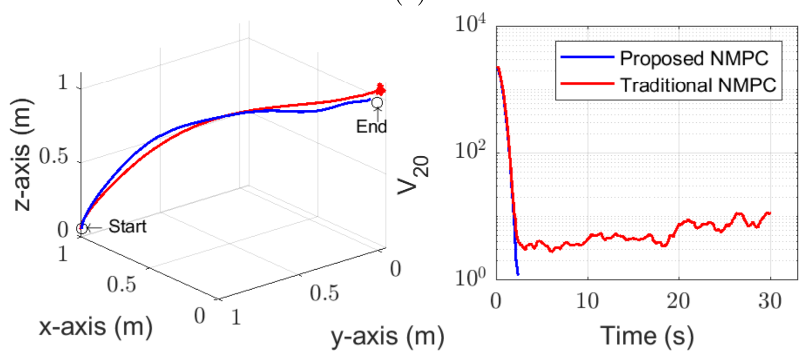
(a)



(b)



(c)



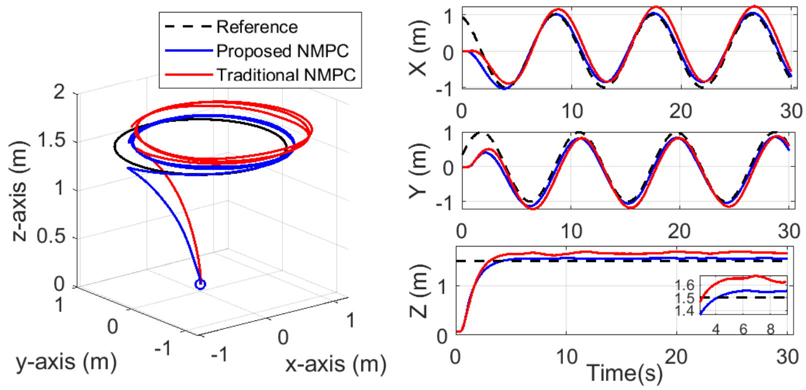
(d)

Fig. 5.15: Experimental point stabilization performance of the proposed and traditional NMPC schemes at $\delta = 0.1$ s and (a) $N = 4$. (b) $N = 5$. (c) $N = 7$. (a) $N = 20$.

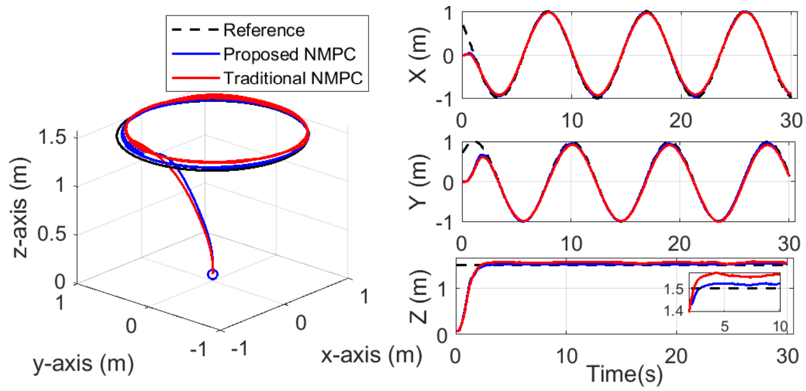
Table 5.4: Experimental comparison results of the point stabilization. RMSE is the root mean square error in meters. # iter. is the number of iterations (time steps) required to stabilize the MAV at the reference point. The sampling time $\delta = 0.1$ s.

N	Proposed		Traditional	
	RMSE	#iter.	RMSE	#iter.
4	0.1	3492	0.317 (Failed)	
5	$0.0498 < \epsilon$	2057	0.155	3485
7	$0.0499 < \epsilon$	236	$0.0499 < \epsilon$	1998
20	$0.0498 < \epsilon$	67	$0.0497 < \epsilon$	802

In order to evaluate the controllers performance for trajectory tracking problem, another ROS node has been created to generate the reference trajectory as shown in Fig. 5.14. The ROS node generates a circular trajectory of radius of 1 m and elevation of 1.5 m. Both controller were tested at two different controller parameters at angular speed of 0.7 rad/s, as shown in Fig. 5.16. The figure shows that the proposed NMPC provides better tracking performance than the traditional controller in both cases. Both controllers are also tested at higher angular velocity of 1.0 rad/s and lower sampling time (we could not go below $\delta = 0.03$ s due to the limitations of the used PC), as shown in Fig. 5.17. The figure shows that the traditional controller failed to track the reference trajectory and the MAV diverged out of the arena, as shown in Fig. 5.17a. The RMSE of the four cases is given in Fig. 5.18, where the outliers come from the fact that the start point of the MAV is the origin, however, the reference trajectory starts from a different point, as seen in Fig. 5.16 and 5.17, therefore, all the experiments have the same maximum outlier. From Fig. 5.16, 5.17 and 5.18, it can be observed that the proposed NMPC outperforms the traditional NMPC regardless of the prediction horizon and sampling time.



(a)

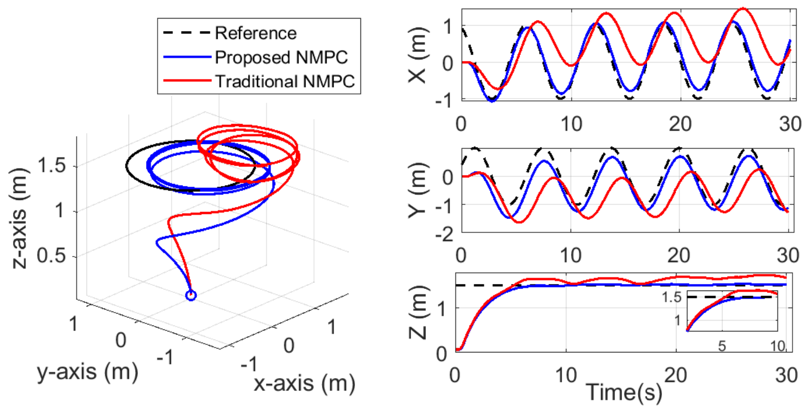


(b)

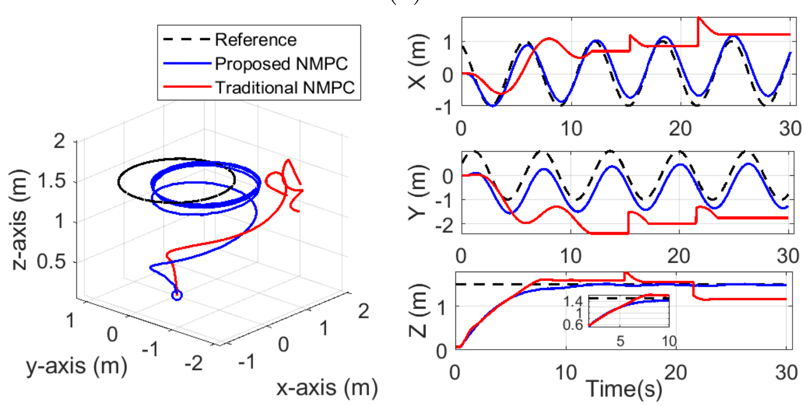
Fig. 5.16: Experimental trajectory tracking performance at angular velocity 0.7 rad/s of the proposed and traditional NMPC schemes at (a) $N = 4$ and $\delta = 0.1$ s. (b) $N = 20$ and $\delta = 0.1$ s.

5.7 Summary

This work has successfully developed and demonstrated a computationally efficient NMPC scheme for quadrotor MAVs to navigate on fast trajectories. In order to achieve fast trajectories, the NMPC scheme was designed with minimum computational cost with a shortest possible prediction horizon. In all experiments, this controller has shown robust performance against the traditional NMPC controller since the proposed controller can stabilize the system under numerous initial conditions with varied system configurations. The stability for this controller was performed without having to follow any constraint or conditions and as a result, the controller has a larger re-



(a)



(b)

Fig. 5.17: Experimental trajectory tracking performance at angular velocity 1.0 rad/s of the proposed and traditional NMPC schemes at (a) $N = 4$ and $\delta = 0.05$ s. (b) $N = 4$ and $\delta = 0.03$ s.

gion of attraction as opposed to many other stability-based controllers reported in the literature. Due to this reason the controller was able to perform well even under small sampling time intervals. The results showed that the proposed NMPC scheme outperforms the traditional schemes that incorporate terminal conditions in its cost function in terms of tracking accuracy and convergence rate. The proposed scheme (with prediction horizon, $N = 4$ or $N = 5$) can achieve the same task done by the traditional scheme (with $N = 10$ or $N = 20$) while attaining better tracking and stabilization accuracy, and saving more than 40% of the computation time. In addition, it still can provide high tracking performance at low sampling time without the need

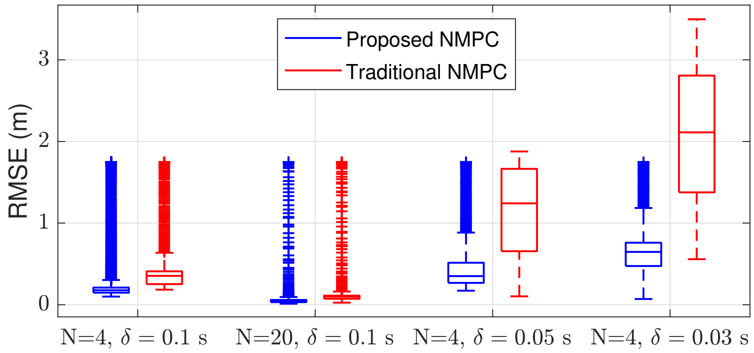


Fig. 5.18: RMSE of the lab experiments at various controller configurations.

of tuning the controller parameters.

Chapter 6

Multiple Model Control

In this chapter, we design a Multiple Nonlinear Model Predictive Controller (Multiple-NMPC), where the two NMPC controllers proposed in Chapter 5 will be incorporated in the NMPC bank to handle different flight conditions. Additionally, numerical case studies will be conducted to validate the proposed algorithm. The Multiple-NMPC validation will be limited to numerical simulations in this thesis.

6.1 Design of Multiple-NMPC

In this work, the Multiple-NMPC algorithm is used to switch between the two NMPC designed in 5.4. The first controller (DF-NMPC) contains the rotor drag force in its model given in (5.6), while the second controller (C-NMPC) uses the conventional model given in (5.3). The DF-NMPC is expected to perform better during fast and aggressive trajectories, while the C-NMPC is expected to be sturdy during external wind disturbances. Therefore, the C-NMPC will be included in the NMPC bank to overcome the deficiency of the DF-NMPC and improve the overall performance of the system. In order for the multiple-NMPC to be computationally efficient while using optimization-based control, one controller should be active at a time. For this

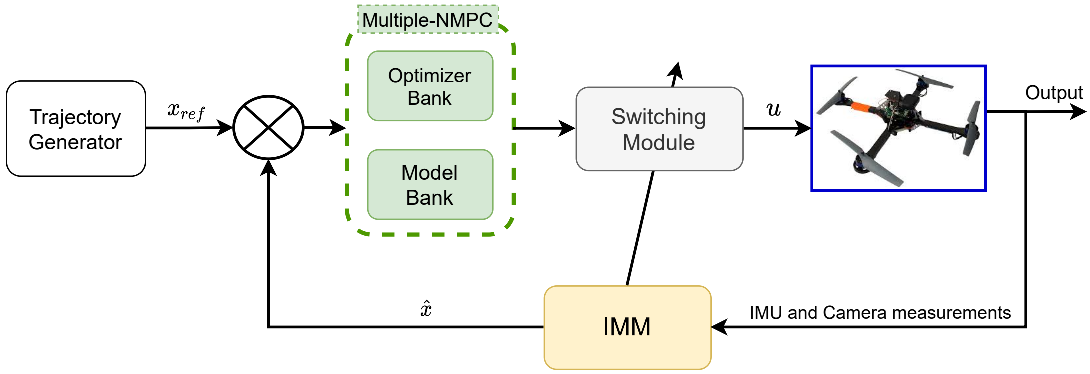


Fig. 6.1: Multiple-NMPC control scheme for quadrotor system.

purpose, the IMM estimator proposed in Algorithm 4.1, Section 4.1, will be used to switch between the controllers/models in the NMPC bank, as shown in Fig. 6.1. The advantage of this scheme, in addition to its lower computationally cost, is that each controller can have an independent model and its controller parameters, such as prediction horizon or weighting parameters.

The IMM algorithm iteratively updates the estimated weighting probability of each filter based on its residual and likelihood; therefore, the IMM can recognize the flight periods with external wind disturbances or aggressive flights. As a result, the estimated weighting probabilities are used to switch between the two controllers, as shown in Algorithm 6.1. The IMM algorithm runs at a higher rate, 100 Hz, while the controller runs at 50 Hz to guarantee the availability of the feedback ahead of each optimization cycle. Since the multiple-NMPC is a switching-based algorithm, the smooth transition is a critical consideration. As a result, Algorithm 6.1 employs the same sampling time for both controllers and uses the state of the previous controller as the initial state of the next controller to avoid discontinuities in the computed control input, as demonstrated in [99].

Algorithm 6.1 requires low number of tuning parameters, i.e., the elements of error weighting matrix $\mathbf{Q}$ in the running cost (5.2). The elements of inputs weighting matrix $\mathbf{R}_u$ in (5.2) are constrained by Proposition 5.1 and the prediction horizon could be selected as $N \geq 4$, as given by Theorem 5.3 and demonstrated in Section 5.6. Additionally, the elements of the Markov transition probability matrix $\boldsymbol{\rho}$ and values of initial probabilities $\hat{\boldsymbol{\mu}}^i$ used in the IMM algorithm are selected based on the probabilities of occurrence of each filter in the IMM bank, as explained in Section 4.1. In our given problem, both filters have similar probabilities of occurrence, therefore, equal initial probabilities are selected.

Algorithm 6.1, at each iteration, computes the estimated state vector $\hat{\mathbf{x}}_k$ from the proposed DF-VINS and C-VINS filters using the estimated model probabilities $\hat{\boldsymbol{\mu}}^i$. Afterwards, the algorithm uses the estimated model probability of each filter to detect the flight condition then trigger the appropriate controller, as given in step 9 in Algorithm 6.1. The NMPC solvers, which include the system model and optimizer, are created ahead of the loop during the initialization of the algorithm. Then the solver is used to solve the OCP and compute the minimizing control sequence $\mathbf{u}^*$ that will be implemented at the MAV plant. The performance of the controller is evaluated by computing the tracking error of the actual trajectory $\mathbf{x}_k$ after applying the optimal control input.

Algorithm 6.1: Multiple-NMPC Scheme

Input: Prediction horizon N, control-loop rate $f_c = 50$ Hz, and estimation-loop rate $f_e = 100$ Hz

Output: MAV closed-loop pose and velocity.

Initialization: Set $k = 0$, actual state $\mathbf{x}_k \coloneqq \mathbf{x}_0$, estimated state $\hat{\mathbf{x}}_k \coloneqq \mathbf{x}_0$, initial estimated probabilities $\hat{\boldsymbol{\mu}}^i = 0.5$ for $i = 1, 2$, and the NMPC optimizer *solver*^{i} for $i = 1, 2$ corresponding to each model (as demonstrated in Section 5.4).

Loop *at* f_e *rate*
    Compute the mixed estimated state vectors (Algorithm 4.1, step 2).
    Collect the IMU measurements at a rate of 100 Hz and Camera measurements at a rate of 20 Hz.
    Propagate and update the estimated state of each filter (Section 3.1.4).
    Compute the estimated probabilities $\hat{\boldsymbol{\mu}}^i$ (Algorithm 4.1, step 4).
    Combine both states to compute $\hat{\mathbf{x}}_k$ using $\hat{\boldsymbol{\mu}}^i$ (Algorithm 4.1, step 5).
    Loop *at* f_c *rate*
        Generate $\mathbf{x}_{ref}(k : k + N)$ and set $\mathbf{x}_0 = \hat{\mathbf{x}}_k$ and $\mathbf{u}_0 = \mathbf{u}^*$.
        if $\hat{\boldsymbol{\mu}}^1 > \hat{\boldsymbol{\mu}}^2$ **then**
            NMPC_solver= *solver*^{1}
        else
            NMPC_solver= *solver*^{2}
        Compute a minimizing control sequence $\mathbf{u}^*$ using *NMPC_solver*.
        Implement the control input $\mathbf{u}(k) \coloneqq \mathbf{u}^*(0, \mathbf{x}_k)$ at the MAV plant.
    Measure the current close-loop state $\mathbf{x}_k \coloneqq x_{\mathbf{u}^*}(k; \mathbf{x}_k)$ and compute the tracking error $\|\mathbf{x}_{ref}(k) - \mathbf{x}_k\|_2$.
    Set $k = k + 1$.

6.2 Comparison with the State-of-the-art

As has been previously reported in the literature, several methods are proposed to address the multiple model predictive control problem. However, the existing methods suffer from certain drawbacks that have been resolved in our proposed work. In this section, we will present the key elements of improvement in our proposed Multiple-NMPC algorithm and compare it with the state-of-the-art multiple model predictive controller, found in the literature, in terms of robustness, computational efficiency, and applicability.

1. **Robustness:** The work in [97–99, 103] proposed the use of a set of linear model/MPC pairs at different operating points. Similarly, the work in [100–102] proposed the use of a single linear MPC and a set of linearized models. The accuracy and robustness of these reported methods depend on the number of models incorporated in the model back, and determining the optimal number of models is considerably difficult. Therefore, for a system with changing dynamics, e.g., MAVs[1], a large number of models are required to yield acceptable control system performance, and a lower number of models may cause system failure or unpredictable behavior [99, 101]. Additionally, it would be difficult to effectively tune the weights and parameters of the single MPC proposed in [100–102] to perform robustly in the possible range of prediction models in the model bank. The mistuning consequently affects the robustness of the proposed method along with the used linearized models and different fight conditions. On the other hand, our proposed method includes a set of nonlinear models/controllers pairs that take advantage of the given nonlinear system dynamics to provide high-performance designs. As shown in Algorithm 6.1, several NMPC solvers can be designed

[1]The dynamic system model significantly changes due to external disturbances.

for the MAV control problem that covers all operating points of the nonlinear system and addresses additional flight conditions.

2. **Computational efficiency:**

   The weighting and switching functions proposed in [97–99] rely on the residual or probability distribution of the outputs of the controllers to decide on the appropriate model/MPC pairs. This method requires all the controllers to be active at each iteration, which is computationally expensive especially when optimization is used for each controller. On the other hand, our proposed algorithm triggers only one controller at a time by means of an order of a magnitude faster filtering-based IMM algorithm, as shown in Algorithm 6.1 (step 9). The Multiple-NMPC makes the decision based on the estimated model probability computed by the IMM algorithm at each iteration that depends mainly on the individual VINS filters in the IMM filtering bank, not the NMPCs in the controller bank.

3. **Applicability:** Most of the reported methods in the literature [97–102] are proposed for process and medical applications. Although they showed improved performance than individual controllers, they cannot be applied to MAVs since the MAV system requires fast control actions. As a result, only computationally efficient algorithms could be implemented onboard due to the limited computation resources and power. Our proposed algorithm includes computationally efficient VINS filters in the IMM filtering bank that are designed using error-state filtering formulation, and use the epipolar constraint as a measurement model. Additionally, our algorithm includes computationally efficient NMPCs that are designed without terminal costs or constraints in the cost function. Also, the performed stability analysis showed that the NMPC is asymptotically

stable with a relatively small prediction horizon, which consequently reduces the computational cost of the optimization problem. Therefore, the proposed Multiple-NMPC algorithm can be applied to MAVs to operate robustly under various and changing operating conditions with improved trajectory tracking performance, as will be demonstrated in the next section.

6.3 Results

This section presents the numerical validations of the overall system, including the multiple-NMPC and IMM algorithms. The performance of the stand-alone DF-NMPC is compared along with the performance improvement achieved by implementing a multiple-NMPC scheme using the two controllers (DF-NMPC and C-NMPC) in its bank.

A MATLAB simulator is implemented in order to evaluate the performance of the proposed system. The same simulated arena proposed in Section 4.2.1, given in Fig. 4.2a, was used. The reference trajectory was generated using the kinematic model given in (3.3), where the acceleration and the angular speed of the platform were designed such that the MAV follows a circular trajectory of a radius of 4 meters completing two laps with additional excitation along the z-axis to result in a wavelike trajectory, as shown in Fig. 6.2. The input acceleration and angular speeds adhered to differential flatness constraints related to the drag force model, which were realized using the procedure given in [20]. This implicitly enforces the dynamic constraints related to model (3.7) during the simulation as long as there is no external wind disturbance acting on the system. In order to verify the superiority of the multiple-NMPC over the stand-alone DF-NMPC in the presence of external disturbance, a normally distributed random wind, with a maximum wind velocity $\|V_m\| = 6.8$m/s, has been added in both laps for a short time.

Since the DF-VINS and C-VINS filters in the IMM filtering bank have the same priorities during the flight, each one is better than the other at certain regions and under certain conditions, the IMM algorithm was implemented with equal initial model probabilities. For a fair comparison, both controllers, multiple-NMPC and DF-NMPC, have the IMM as feedback. The system has been simulated with the same parameters given in Section 4.2.1 and Section 5.6.1. The prediction horizon is selected as $N = 6$ for both controllers.

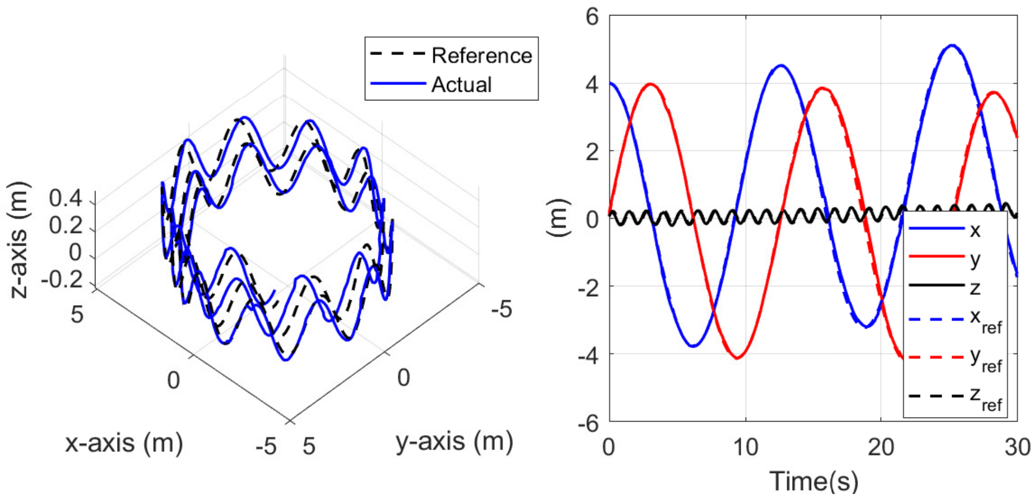
6.3.1 Numerical validation Without disturbance

The performance of both controllers, multiple-NMPC and DF-NMPC, has been tested first in an aggressive flight without wind disturbance. The trajectory tracking performance for both controllers is given in Fig. 6.2. The figure shows the 3D trajectory tracking of the DF-NMPC (Fig. 6.2a) and Multiple-NMPC (Fig. 6.2b). Both controllers were successfully able to track the reference trajectory and showed comparable performance (Fig. 6.3).

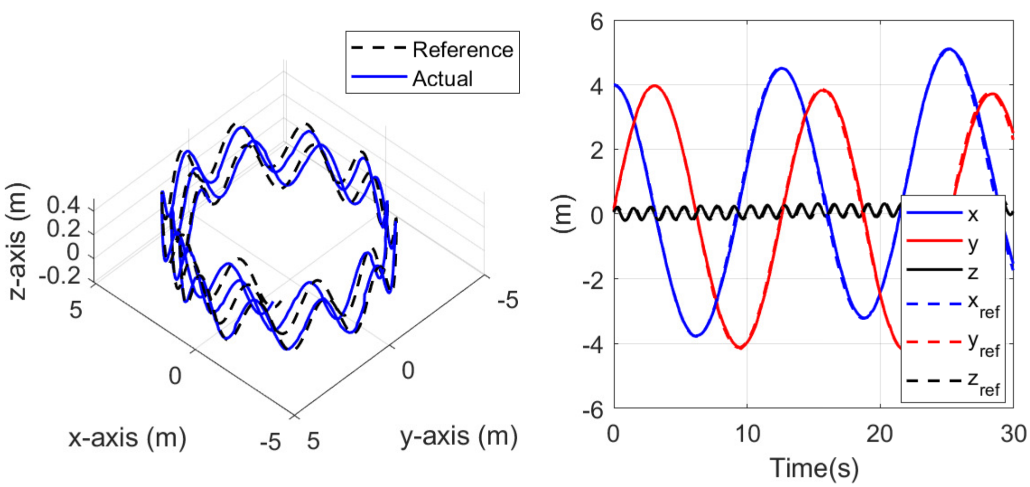
The root mean square error (RMSE) of the position tracking has been computed for each controller and given in Table 6.1. The Multiple-NMPC has only about 3.6% performance improvement over the DF-NMPC since the Multiple-NMPC sticks to the DF-NMPC most of the flight, as given by the values of the model probabilities shown in Fig. 6.4. The DF-Model has a higher estimated model probability during most of the flight. Therefore, based on the switching condition in Algorithm 6.1, the Multiple-NMPC will select the first solver associated to the DF-NMPC, which is supposed to provide better performance than the solver associated to the C-NMPC.

Table 6.1: Tracking accuracy (RMSE) in the simulation results with/without wind disturbance and the corresponding performance improvement.

	RMSE (m)		Percent of improvement
	DF-NMPC	Multiple-NMPC	Multiple-NMPC/DF-NMPC
No wind trajectory	0.140	**0.135**	3.6 %
Wind trajectory	0.284	**0.125**	56 %



(a)



(b)

Fig. 6.2: Simulation performance of (a) DF-NMPC and (b) Multiple-NMPC. Left: 3D trajectories of the reference and actual trajectories. Right: the reference and actual position in x, y, and z axes.

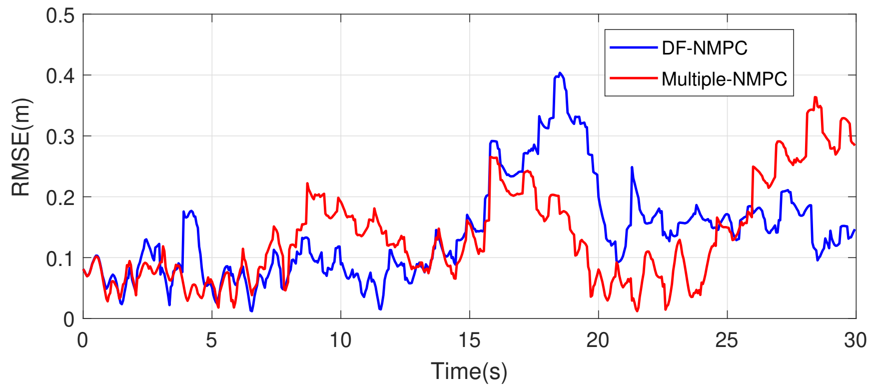


Fig. 6.3: RMSE of the position of the Multiple-NMPC and DF-NMPC.

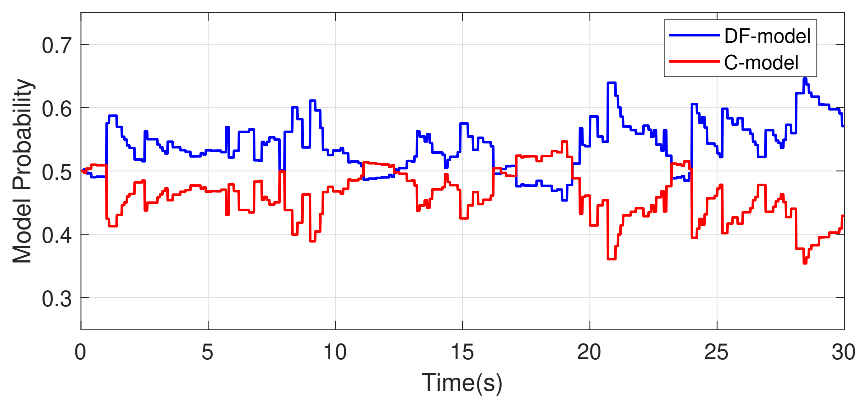
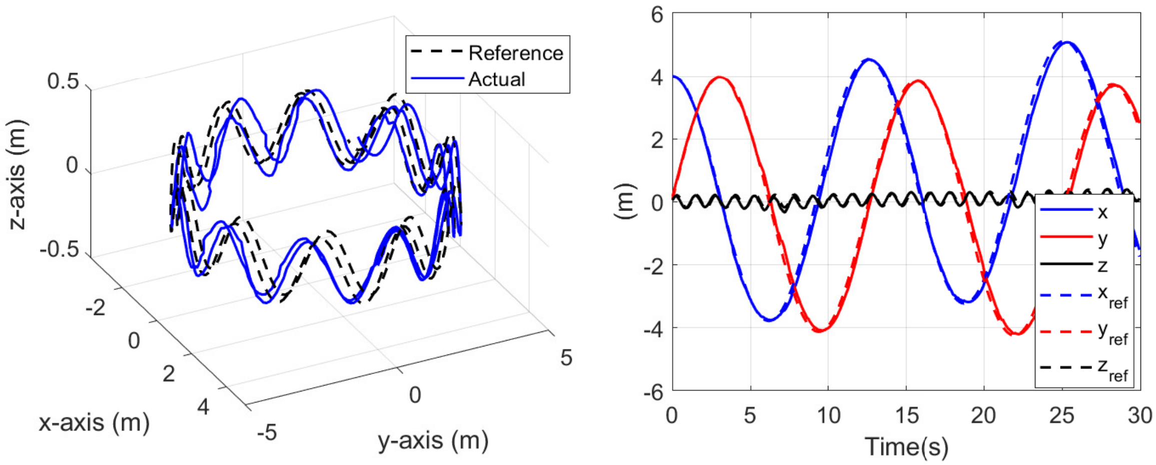


Fig. 6.4: Model probabilities of the IMM filters.

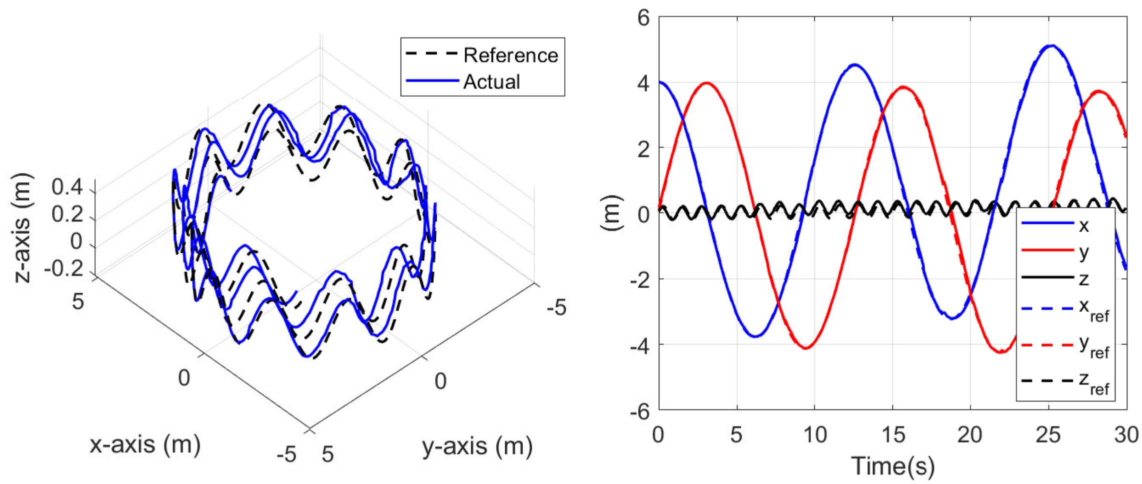
6.3.2 Numerical validation with wind disturbance

On the other hand, the superiority of the proposed Multiple-NMPC can be demonstrated in the presence of wind disturbance, where the DF-NMPC is expected to get affected by the external force exerted by the wind, as shown in Fig. 6.5. The Multiple-NMPC outperforms the DF-NMPC and provides more than 50% performance improvement, as shown in Fig. 6.6 and Table 6.1. This improvement has been achieved due to the ability of the Multiple-NMPC to switch between the two NMPCs in the controller bank based on their probability to represent the actual flight mode, as shown in Fig. 6.7.

Monte Carlo numerical simulation of 20 runs is conducted to validate the performance of the proposed controller at different sensor random noises and random wind disturbances. The results in Fig. 6.8 show that the Multiple-NMPC still outperforms the DF-NMPC in the existence of wind disturbances, where the overall RMSE of the DF-NMPC is 0.30 m while the RMSE of the Multiple-NMPC is 0.22 m.



(a)



(b)

Fig. 6.5: Simulation performance during the presence of wind disturbance of (a) DF-NMPC and (b) Multiple-NMPC. Left: 3D trajectories of the reference and actual trajectories. Right: the reference and actual position in x, y, and z axes.

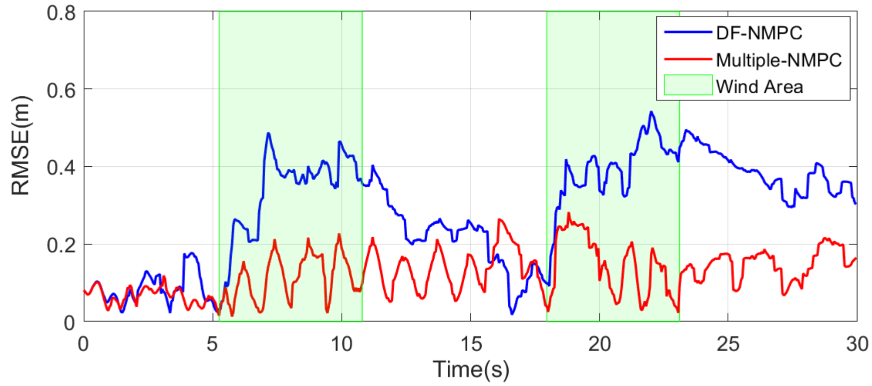


Fig. 6.6: RMSE of the position of the Multiple-NMPC and DF-NMPC during the presense of wind disturbance.

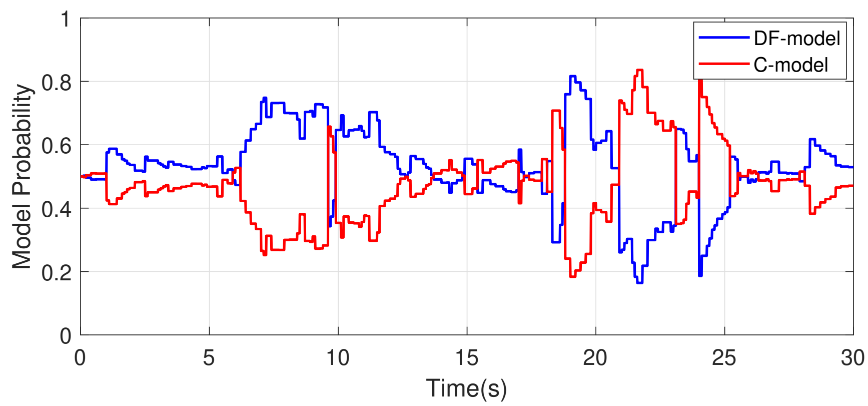


Fig. 6.7: Model probabilities of the IMM filters during the presence of wind disturbance.

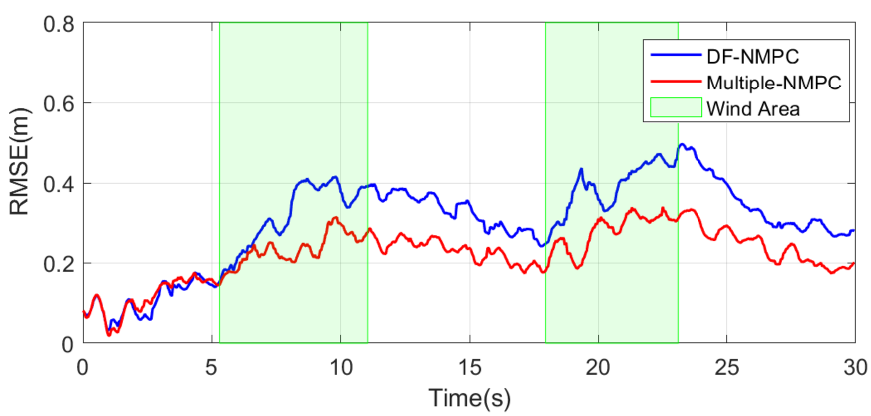


Fig. 6.8: Tracking accuracy (RMSE) in the simulation results for 20 Monte Carlo runs in the existence of wind disturbance.

6.4 Summary

This work has numerically validated the design of a multiple-NMPC scheme where two controllers are included in the controller bank to handle different flight trajectories or conditions. The proposed scheme showed superior performance compared to the stand-alone NMPC that incorporates the rotor drag force in its model. The simulation studies showed tracking performance improvement by more than 50% during external wind disturbances. For future work, the proposed algorithm will be experimentally validated, as part of an industrial project in Winter 2022, with additional NMPCs incorporated in the controller bank to address more flight conditions and modes.

Chapter 7

Summary and Outlook

In this chapter, we summarize the accomplished work towards the research objectives presented in 1.3 and the contributions of the thesis, while a list of scientific articles and future research directions are presented.

The primary focus of this research study was to adopt the multiple model approach for the state estimation and control of quadrotor MAVs. The work included the design of computation-efficient state estimators and feedback controllers to be suitable for the multiple model framework. The conducted research formed four main objectives:

1. Design of a computational-efficient error-state visual inertial navigation system (VINS) with improved accuracy and consistency for MAVs.

2. Design of a novel interacting multiple model for VINS (IMM-VINS) that supports operation during periods with aggressive flights and/or external disturbances (e.g., wind).

3. Design of a novel computationally efficient and stable nonlinear model predictive controller (NMPC) without terminal conditions for the control of MAVs with/without drag force incorporated in the system model.

4. Design of a novel computational-efficient multiple-NMPC scheme that supports various operation and flight conditions.

The research summary related to each objective is presented in the following sections.

7.1 Research Summary Based on Objective I

For the first objective, two error-state visual-inertial navigation systems, with/without drag force, have been designed for the state estimation for MAVs. After an extensive literature review, it was concluded that the computational complexity and performance of VINS filters are affected by the number of poses in the measurement model and the consistency of the filter. Therefore, in this study, we designed the VINS filter using the epipolar constraint as the measurement model since it is computationally efficient. Additionally, nonlinear observability analysis has been conducted to study the observability properties of the designed filter to identify the unobservable subspace corresponding to the nonlinear systems, which are needed to design observability-constrained consistent filters for VINS. The observability analysis has shown that the system has four unobservable directions, as indicated by the derived nullspace matrix. Those unobservable directions correspond to the three degrees of freedom global translation of the current and previous positions pair together, and the global rotation about the gravity axis. Therefore, the observability-based consistency rules for the filters are deduced to preserve the observability properties of the filter to match with the nullspace of the true nonlinear system to prevent the filter from updating along any unobservable direction. The designed filters will be then used for the fulfillment of Objective II.

The proposed VINS filters share several assumptions as the state-of-the-art work, e.g., Gaussian white noise and proper initialization. To address these limitations, strategies such as Gaussian mixture noises [117], robust initialization [118] and failure recovery modules [119] should be implemented. Additionally, the proposed filters assume the availability of sufficient feature points to ensure the existence of matched features. For flights with high altitudes, spare sensors, e.g., multiple cameras or LiDAR, could be added to the filter to resolve the limitation of lack of matched features.

7.2 Research Summary Based on Objective II

Secondly, we considered the design of an interacting multiple model for VINS (IMM-VINS) where two VINS filters, conventional VINS and drag force VINS, are incorporated in the IMM filter bank to support operation during periods with aggressive flights and/or external disturbances (e.g., wind). The numerical and experimental studies validated the capability of the IMM to produce improved VINS performance over stand-alone versions of filters in its bank. IMM achieves this because the stand-alone filters are meant to serve different navigation capabilities (accuracy, stability, robustness, self-calibration) and different flight conditions (aggressive, hover, nominal), while IMM allows a synergistic combination of the capabilities of the filters during the flight to generate improved performance over the stand-alone versions as shown in this work. However, the IMM implementation of this work does not have improved pose estimation accuracy over the generic MSCKF [15] or MSCKF-MONO [120] of the EuRoC datasets in our experiments [121]. This is due to the use of the epipolar measurement model in the design to reduce the computational complexity. A recent study [121] identifies that the filter execution time is more than 90% faster as a result of the epipolar measurement model when compared with MSCKF filters. This means that the IMM design has significant headroom for improvement. While this additional computational headroom can be used for accuracy improvement, what is more interesting in this strategy is the ability to instill backup filters for fault tolerance and opportunistic self-calibration of the VINS.

Parameter selection of the IMM algorithm needs to be carefully considered as some of the individual filters in the filtering bank can perform sub-optimally at initialization. For example, the stationary part at the beginning of the EuRoC dataset flight, where the drag force filter is expected to diverge and the state covariance matrix is expected to increase due to the high uncertainties. In order to resolve this issue, an adaptive IMM algorithm [35] should be in use to update the elements of the transition probability matrix over time to optimize the performance of the IMM, as demonstrated in Section 4.2.2. Additionally, incorporation of robust initialization schemes available for VINS can ensure proper initialization and failure recovery [118, 119, 122, 123].

7.3 Research Summary Based on Objective III

Thirdly, we consider the design of a novel NMPC algorithm for the control and stabilization of a quadrotor MAV. A tailored cost function has been proposed without incorporating terminal costs or constraints that facilitates the stability analysis of the quadrotor and the derivation of a growth bound on the proposed value function to achieve the controllability conditions. This growth bound has been used to calculate the minimal stabilizing prediction horizon of the proposed scheme, which was proved to be four. The design provides a unique analytical methodology that requires minimal tuning parameters compared to other controllers in the literature. The numerical and experimental studies showed that the proposed NMPC scheme outperforms the traditional schemes that incorporate terminal conditions in its cost function in terms of tracking accuracy and convergence rate. The proposed scheme with a small prediction horizon, e.g., $N = 4$ or $N = 5$ can achieve the same task done by the traditional scheme with relatively high prediction horizon, e.g., $N = 10$ or $N = 20$ while attaining better tracking and stabilization accuracy, and saving more than 40% of the computation time. In addition, it still can provide high tracking performance at low sampling time without the need of tuning the controller parameters. This superior performance makes the proposed NMPC adequate for the multiple model control scheme as presented in Objective IV.

7.4 Research Summary Based on Objective IV

Finally, we considered the multiple model control, where a Multiple Nonlinear Model Predictive Controller has been designed for the control of quadrotor Micro Aerial Vehicles (MAVs). In this respect, two NMPC controllers, with/without drag force in the system model, were incorporated in the NMPC bank to handle different flight conditions. One of the main objectives of this work is the design of a computationally efficient algorithm to effectively switch between the controllers. Therefore, the IMM estimator, designed in Objective II, was used to recognize the flight mode/condition then trigger the appropriate controller. The conducted numerical simulations have validated the proposed algorithm and showed more than 50% performance improvement of the Multiple-NMPC over the stand-alone drag force NMPC.

The NMPCs in the controller bank are ensured to be asymptotically stable and the multiple-NMPC algorithm uses the state of the previous controller as the initial state of the next controller to avoid discontinuities in the computed control input. As future research directions, the stability analysis of the switching control system needs to be performed to ensure the overall asymptotic stability of the switching system.

7.5 Discussion

The proposed multiple-NMPC algorithm showed improved performance compared to the stand-alone DF-NMPC. The simulation results have numerically validated the proposed algorithm and manifested its superiority and capability of switching between the NMPCs in the controller bank to handle different flight conditions. Although only two models are used in this study, the same framework can be used to incorporate more models, such as taking off and landing models. The following steps can be followed to add more models to the multiple model scheme.

1. Select an additional MAV dynamic model that addresses a different flight mode or conditions.

2. Design an error-state filter for this model and compute its process and measurement filtering Jacobian matrices (as demonstrated in Section 3.1). (Note: The filter could be designed as a VINS or any other type of filter, based on the type of sensors used in the filter and the kind of measurement model.)

3. Perform the observability analysis of this additional model and find the null space (as demonstrated in Section 3.2.1).

4. Design an observability-constrained VINS using the identified unobservable subspace to improve the consistency of the filter (as demonstrated in Section 3.2.2).

5. Incorporate the designed observability-constrained VINS filter in the IMM filtering bank by selecting a suitable initial probability and suitable matrix elements for the transition probability matrix (as demonstrated in Section 4.1).

6. Design an NMPC for the additional model by selecting a suitable objective function without terminal costs or constraints (as demonstrated in Section 5.4).

7. Perform the stability analysis for the designed NMPC by deriving the growth bound that ensues the asymptotic stability of the closed-loop system and calculate the minimal stabilizing prediction horizon (as demonstrated in Section 5.5).

8. Incorporate the designed NMPC in the multiple-NMPC controller bank by attaching it to its corresponding filter to be triggered when the associated model probability is higher than other models' probabilities (as demonstrated in Section 6.1).

7.6 Summary of Contributions

In summary, following are the key contributions of the thesis objectives in using multiple model approach for state estimation and control for quadrotor MAVs.

1. **Contributions related to Objective I:**

    - A novel design of two error-state VINS filters with epipolar constraints as the measurement model. (Section 3.1)
    - Observability analysis of the drag force VINS filter and the epipolar constraint measurement models. (Section 3.2.1)
    - Development of observability-constrained VINS to maintain the consistency of the filter. (Section 3.2.2)

2. **Contributions related to Objective II:**

    - A novel design of an error-state IMM estimator for VINS applications. (Section 4.1)

- Numerical and experimental validations of the proposed IMM-VINS using Matlab simulator and the EuRoC dataset. (Section 4.2)

3. **Contributions related to Objective III:**

   - A design of a computationally-efficient NMPC scheme with improved stability characteristics for the control of quadrotor MAVs without the use of terminal costs or constraints. (Section 5.4)
   - Stability analysis of the proposed controller to prove its asymptotic stability by deriving a growth bound on the proposed MPC value function. (Section 5.5)
   - Calculation of a minimal stabilizing prediction horizon, which effectively minimizes the computational cost. (Section 5.5.2)
   - Numerical and experimental validations of the proposed controller at various initial conditions, system configurations, and various trajectories, and demonstrating its robustness against the traditional NMPC schemes in the literature. (Section 5.6)

4. **Contributions related to Objective IV:**

   - A novel design of a multiple model control scheme, depending on the IMM filter, that can effectively recognize the flight mode and then trigger the appropriate NMPC from the controller's bank. (Section 6.1)
   - Numerical validation of the proposed Multiple-NMPC system for quadrotor MAVs. (Section 6.3)

7.6.1 List of Publications

This research led to the following scientific articles and publications:

Journal Articles:

- M.A.K. Gomaa, O. De Silva, G.K.I. Mann, R. Gosine “Computationally Efficient Stability-based Nonlinear Model Predictive Control Design for Quadrotor MAVs”, IEEE Transactions on Control Systems Technology, 2021 (Submitted and Preparing 1st revised version).

- M.A.K. Gomaa, O. De Silva, G.K.I. Mann, R. Gosine, "Nonlinear MPC Without Terminal Costs or Constraints for Multi-Rotor Aerial Vehicles," in IEEE Control Systems Letters, vol. 6, pp. 440-445, 2022. (Presented at the 60th IEEE conference on Decision and Control "CDC 2021", Texas, USA)

- M.A.K. Gomaa, O. De Silva, G.K.I. Mann, R. Gosine, "Observability-Constrained VINS for MAVs Using Interacting Multiple Model Algorithm," in IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 3, pp. 1423-1442, June 2021

Conference Publications:

- M.A.K. Gomaa, O. De Silva, G.K.I. Mann, R. Gosine, R. Hengeveld “Computationally Efficient Multiple-NMPC for Quadrotor Micro Aerial Vehicles”, The 30th Annual Newfoundland Electrical and Computer Engineering Conference (NECEC), Nov. 2021, NL, Canada.

- M.A.K. Gomaa, O. De Silva, G.K.I. Mann, R. Gosine “Interacting Multiple Model Navigation System for Quadrotor Micro Aerial Vehicles Subject to Rotor Drag”, 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2020, Las Vegas, USA.

- M.A.K. Gomaa, O. De Silva, G.K.I. Mann, R. Gosine, R. Hengeveld "ROS Based Real-Time Motion Control for Robotic Visual Arts Exhibit Using Decawave Local Positioning System", 2020 American Control Conference (ACC), July 2020, Colorado, USA.

7.7 Future Research Directions

The research work presented in this thesis has a number of possible potential extensions. These future developments aim at improving the performance of the overall multiple model state estimation and control system in terms of the computational efficiency and accuracy.

Onboard implementation of the proposed scheme: Implement the proposed multiple model state estimation and control system onboard and test it on outdoor aggressive trajectories. A new experimental setup, using VOXL m500 drone[1], is being prepared to conduct the lab experiments of the multiple-NMPC. The VOXL m500 drone is equipped with high-resolution sensors and an autonomy computer and flight controller (VOXL Flight Deck), which makes it suitable for fast onboard computations and advanced autonomy development. The overall multiple model scheme will be implemented on ROS while using CasADi toolbox [115] as the NMPC optimization solver, as demonstrated in Chapter 5.

Incorporation of additional filters and model/NMPC pairs in the multiple model bank: In light of the computational efficiency of the proposed multiple model scheme, incorporating additional filters and model/NMPC pairs in the multiple model bank would improve the tracking performance of the MAV and increase its robustness in most indoor and outdoor applications. Multiple model methods for VINS

[1]Available on `https://www.modalai.com/products/voxl-m500`

can be further investigated by including additional filters that consider different noise conditions and filters that can guarantee robustness such as sliding mode methods in the filtering bank. Additionally, filters with models of learned dynamics can be incorporated to handle the unknown dynamics of the MAV. Similarly, the multiple model control can be further investigated by including additional NMPCs that address other flight conditions, such as, ground effect, proximity to walls or objects in the environment, taking off, and landing.

Development of optimization-based VINS: Optimization-based VINS has improved estimation accuracy and solution robustness but demands considerable computational power, where a fixed history of vehicle states and environment features are optimized using nonlinear optimization [66]. A recent study in [17] showed that optimization-based VINS is consistently accurate and robust across the different hardware platforms used in the study. However, this superior performance came at the cost of a high level of resource usage. Therefore, optimization-based techniques can be further investigated to improve its computational efficiency and make it adequate for the implementation in the multiple model schemes. Epipolar measurement models in the optimization framework and novel cost functions to improve optimization performance/consistency can be considered.

Using machine learning techniques as a weighting/switching module: Recently, machine learning techniques have shown promising results in robotics applications such as terrain classification [124] and image classification for path-following [125] using sensors data, e.g., images or acceleration from an IMU. As a result, they were used in MAV applications to recognize some flight conditions such as ground effect [126] or proximity to wall [7]. Therefore, machine learning techniques, such as Neural Network [127], can be used in the multiple model estimation/control algorithm to recognize the flight modes or conditions and then trigger the appropriate filter and model/NMPC pair to improve the overall performance of the MAV. The MAV sensors, including the camera and IMU, and possibly additional sensors, can be utilized for the feature selection of each flight mode.

References

[1] Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I. L., Ruess, F., Suppa, M., and Burschka, D., "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[2] Garimella, G. and Kobilarov, M., "Towards model-predictive control for aerial pick-and-place," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4692–4697.

[3] Mancini, F., Dubbini, M., Gattelli, M., Stecchi, F., Fabbri, S., and Gabbianelli, G., "Using unmanned aerial vehicles (uav) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments," *Remote Sensing*, vol. 5, no. 12, Dec 2013.

[4] Kumar, V. and Michael, N., *Opportunities and Challenges with Autonomous Micro Aerial Vehicles.* Cham: Springer Intern. Publishing, 2017, pp. 41–58.

[5] Hoffmann, G., Waslander, S., and Tomlin, C., "Quadrotor helicopter trajectory tracking control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2008, pp. 1–14.

[6] Kan, X., Thomas, J., Teng, H., Tanner, H. G., Kumar, V., and Karydis, K., "Analysis of ground effect for small-scale uavs in forward flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[7] McKinnon, C. D. and Schoellig, A. P., "Estimating and reacting to forces and torques resulting from common aerodynamic disturbances acting on quadrotors," *Robotics and Autonomous Systems*, vol. 123, p. 103314, 2020.

[8] Chatfield, A. B., *Fundamentals of high accuracy inertial navigation.* American Institute of Aeronautics and Astronautics, Inc., 1997.

[9] Maenaka, K., "Mems inertial sensors and their applications," in *2008 5th International Conference on Networked Sensing Systems*, 2008, pp. 71–73.

[10] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, N. M. K. and Kasi, V., "Reviews on various inertial measurement unit (imu) sensor applications," *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, December 2013.

[11] Abeywardena, D., Huang, S., Barnes, B., Dissanayake, G., and Kodagoda, S., "Fast, on-board, model-aided visual-inertial odometry system for quadrotor micro aerial vehicles," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1530–1537.

[12] Indelman, V., Williams, S., Kaess, M., and Dellaert, F., "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721 – 738, 2013.

[13] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P., "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[14] Indelman, V., Melim, A., and Dellaert, F., "Incremental light bundle adjustment for robotics navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1952–1959.

[15] Mourikis, A. I. and Roumeliotis, S. I., "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3565–3572.

[16] Nguyen, T., Mann, G., Vardy, A., and Gosine, R., "Developing computationally-efficient nonlinear cubature kalman filtering for visual inertial odometry," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, 02 2019.

[17] Delmerico, J. and Scaramuzza, D., "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE International Confer. on Robotics and Automation (ICRA)*, May 2018, pp. 2502–2509.

[18] Zhang, X., Li, X., Wang, K., and Lu, Y., "A survey of modelling and identification of quadrotor robot," *Abstr. and Appl. Analys.*, vol. 2014, p. 1–16, 2014.

[19] Jackson, J., Nielsen, J., McLain, T., and Beard, R., "Improving the robustness of visual-inertial extended kalman filtering," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 4703–4709.

[20] Faessler, M., Franchi, A., and Scaramuzza, D., "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, April 2018.

[21] Bouabdallah, S. and Siegwart, R., "Full control of a quadrotor," in *2007 IEEE/RSJ Inter. Confer. on Intell. Robots and Syst.*, Oct 2007, pp. 153–158.

[22] Bar-shalom, Y., Li, X., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley Sons, Ltd, 2002.

[23] Glavine, P. J., De Silva, O., Mann, G., and Gosine, R., "Gps integrated inertial navigation system using interactive multiple model extended kalman filtering," in *2018 Moratuwa Engineer, Research Conf.*, May 2018, pp. 414–419.

[24] Duan, L., Huang, X., Luo, B., and Li, Q., "Target tracking with interactive multiple model in geodetic coordinate system for naval ships cooperative engagement," in *2008 11th Intern, Conf, on Info... Fusion*, June 2008, pp. 1–8.

[25] Bayat, M., Vanni, F., and Aguiar, A. P., "Online mission planning for cooperative target tracking of marine vehicles," *IFAC Proceedings Volumes*, vol. 42, no. 18, pp. 185 – 189, 2009, 8th IFAC Conference on Manoeuvring and Control of Marine Craft.

[26] Busch, M. T. and Blackman, S. S., "Evaluation of imm filtering for an air defense system application," vol. 2561, 1995.

[27] Miller, M. D., Drummond, O. E., and Perrella Jr., A. J., "Multiple-model filters for boost-to-coast transition of theater ballistic missiles," vol. 3373, 1998.

[28] Burlet, J., Aycard, O., Spalanzani, A., and Laugier, C., "Adaptive interacting multiple models applied on pedestrian tracking in car parks," in *2006 IEEE/RSJ Intern.. Conference on Intelligent Robots and Systems*, 2006, pp. 525–530.

[29] Vasuhi, S. and Vaidehi, V., "Target tracking using interactive multiple model for wireless sensor network," *Information Fusion*, vol. 27, pp. 41 – 53, 2016.

[30] Cho, C. H., Chong, S. Y., and Song, T. L., "Imm filtering for vehicle tracking in cluttered environments with glint noise," in *2017 Intern. Conference on Control, Automation and Information Sciences (ICCAIS)*, Oct 2017, pp. 98–105.

[31] Tissainayagam, P. and Suter, D., "Visual tracking with automatic motion model switching," *Pattern Recognition*, vol. 34, no. 3, pp. 641 – 660, 2001.

[32] Jo, K., Chu, K., Lee, K., and Sunwoo, M., "Integration of multiple vehicle models with an imm filter for vehicle localization," in *2010 IEEE Intelligent Vehicles Symposium*, June 2010, pp. 746–751.

[33] Toledo-Moreo, R., Zamora-izquierdo, M. A., and Gomez-skarmeta, A. F., "Imm-ekf based road vehicle navigation with low cost gps/ins," in *2006 IEEE Intern. Conf. on Multisensor Fusion and Integ. for Intell. Syst.*, Sep. 2006, pp. 433–438.

[34] Granström, K., Willett, P., and Bar-Shalom, Y., "Systematic approach to imm mixing for unequal dimension states," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pp. 2975–2986, Oct 2015.

[35] Li, B., Pang, F., Liang, C., Chen, X., and Liu, Y., "Improved interactive multiple model filter for maneuvering target tracking," in *Proceedings of the 33rd Chinese Control Conference*, July 2014, pp. 7312–7316.

[36] Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., Jose Taylor, C., and Kumar, V., "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Lett.*, vol. PP, 11 2017.

[37] Webb, T. P., Prazenica, R. J., Kurdila, A. J., and Lind, R., "Vision-based state estimation for autonomous micro air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 3, pp. 816–826, 2007.

[38] Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I., "Camera-imu-based localization: Observability analysis and consistency improvement," *The Intern. Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[39] Goodarzi, F., Lee, D., and Lee, T., "Geometric nonlinear pid control of a quadrotor uav on se(3)," in *2013 European Control Conference*, 2013, pp. 3845–3850.

[40] Mellinger, D., Michael, N., and Kumar, V., "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[41] Lee, T., Leok, M., and McClamroch, N. H., "Nonlinear robust tracking control of a quadrotor uav on se(3)," in *2012 American Control Conference (ACC)*, 2012, pp. 4649–4654.

[42] Naldi, R., Furci, M., Sanfelice, R. G., and Marconi, L., "Robust global trajectory tracking for underactuated vtol aerial vehicles using inner-outer loop control paradigms," *IEEE Trans. Automatic Control*, vol. 62, no. 1, pp. 97–112, 2017.

[43] Lei, W., Li, C., and Chen, M. Z. Q., "Robust adaptive tracking control for quadrotors by combining pi and self-tuning regulator," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2663–2671, 2019.

[44] Dydek, Z. T., Annaswamy, A. M., and Lavretsky, E., "Adaptive control of quadrotor uavs: A design trade study with flight evaluations," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1400–1406, 2013.

[45] Tal, E. and Karaman, S., "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Trans.Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2021.

[46] Madani, T. and Benallegue, A., "Backstepping control for a quadrotor helicopter," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3255–3260.

[47] Mayne, D., Rawlings, J., Rao, C., and Scokaert, P., "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[48] Mayne, D. Q., "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.

[49] Grüne, L. and Pannek, J., *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer London, 01 2011.

[50] Worthmann, K., Mehrez, M. W., Zanon, M., Mann, G. K. I., Gosine, R. G., and Diehl, M., "Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1394–1406, 2016.

[51] Gomaa, M. A. K., De Silva, O., Mann, G. K. I., Gosine, R. G., and Hengeveld, R., "Ros based real-time motion control for robotic visual arts exhibit using decawave local positioning system," in *2020 American Control Conference (ACC)*, 2020, pp. 653–658.

[52] Kamel, M., Stastny, T., Alexis, K., and Siegwart, R., *Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System*. Cham: Springer International Publishing, 2017, pp. 3–39.

[53] Lindqvist, B., Mansouri, S. S., Agha-mohammadi, A., and Nikolakopoulos, G., "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robot. and Automat. Lett.*, vol. 5, no. 4, 2020.

[54] Müller, M. A. and Worthmann, K., "Quadratic costs do not always work in mpc," *Automatica*, vol. 82, pp. 269–277, 2017.

[55] Chen, H. and Allgower, F., "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[56] Primbs, J. A. and Nevistić, V., "Feasibility and stability of constrained finite receding horizon control," *Automatica*, vol. 36, no. 7, 2000.

[57] Scokaert, P. O. M., Mayne, D. Q., and Rawlings, J. B., "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.

[58] Grüne, L., "Analysis and design of unconstrained nonlinear mpc schemes for finite and infinite dimensional systems," *SIAM Journal on Cont. and Optimiz.*, vol. 48, pp. 1206–1228, 2009.

[59] Worthmann, K., “Stability analysis of unconstrained receding horizon control schemes,” Ph.D. dissertation, Bayreuth University, 2011.

[60] Worthmann, K., “Estimates of the prediction horizon length in mpc: a numerical case study,” *IFAC Proceedings Volumes*, vol. 45, no. 17, 2012, 4th IFAC Conference on Nonlinear Model Predictive Control.

[61] Keertht, S. S. and Gilbert, E. G., “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations,” *Journal of Optimization Theory and Application*, vol. 57, p. 265–293, 1988.

[62] Worthmann, K., Mehrez, M. W., Mann, G. K., Gosine, R. G., and Pannek, J., “Interaction of open and closed loop control in mpc,” *Automatica*, vol. 82, pp. 243 – 250, 2017.

[63] Boccia, A., Grüne, L., and Worthmann, K., “Stability and feasibility of state constrained mpc without stabilizing terminal constraints,” *Systems Control Letters*, vol. 72, pp. 14 – 21, 2014.

[64] Grüne, L., “Nmpc without terminal constraints,” *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 1 – 13, 2012.

[65] Waslander, S. and Wang, C., “Wind disturbance estimation and rejection for quadrotor position control,” in *AIAA Infotech@Aerospace Conference*, 04 2009.

[66] Shen, S., Michael, N., and Kumar, V., “Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs,” in *2015 IEEE Intern. Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5303–5310.

[67] Michael, N., Mellinger, D., Lindsey, Q., and Kumar, V., “The grasp multiple micro-uav testbed,” *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 56–65, Sep. 2010.

[68] Leishman, R. C., Macdonald, J. C., Beard, R. W., and McLain, T. W., “Quadrotors and accelerometers: State estimation with an improved dynamic model,” *IEEE Control Systems Magazine*, vol. 34, no. 1, pp. 28–41, Feb 2014.

[69] Abeywardena, D., Kodagoda, S., Dissanayake, G., and Munasinghe, R., “Improved state estimation in quadrotor mavs: A novel drift-free velocity estimator,” *IEEE Robotics Automation Magazine*, vol. 20, no. 4, pp. 32–39, Dec 2013.

[70] Fernando, E., Mann, G. K., De Silva, O., and Gosine, R. G., “Design and analysis of a pose estimator for quadrotor MAVs With modified dynamics and range measurements,” in *Proceedings of the ASME 2017 Dynamic Systems and Control Conference*, vol. 3, 10 2017.

[71] Bar-Shalom, Y., Chang, K. C., and Blom, H. A. P., "Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm," *IEEE Trans. Aerospace & Electr. Syst.*, vol. 25, no. 2, pp. 296–300, March 1989.

[72] Genovese, A. F., "Interacting multiple model algorithm for accurate state estimation of maneuvering targets," *Johns Hopkins APL Technical Digest*, vol. 22, no. 4, pp. 614–623, 2001.

[73] Kirubarajan, T. and Bar-Shalom, Y., "Kalman filter versus imm estimator: when do we need the latter?" *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1452–1457, Oct 2003.

[74] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J., "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, Jan 1998.

[75] Punithakumar, K., Kirubarajan, T., and Sinha, A., "Multiple-model probability hypothesis density filter for tracking maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 87–98, January 2008.

[76] Jwo, D., Chung, F., and Yu, K., "Gps/ins integration accuracy enhancement using the interacting multiple model nonlinear filters," *Journal of Applied Research and Technology*, vol. 11, no. 4, pp. 496 – 509, 2013.

[77] Hausman, K., Weiss, S., Brockers, R., Matthies, L., and Sukhatme, G. S., "Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a uav," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4289–4296.

[78] Amoozgar, M. H., Chamseddine, A., and Zhang, Y. M., "Experimental test of an interacting multiple model filtering algorithm for actuator fault detection and diagnosis of an unmanned quadrotor helicopter," in *Intelligent Robotics and Applications*, Su, C.-Y., Rakheja, S., and Liu, H., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 473–482.

[79] Lu, P., Eykeren, L. V., Kampen, E.-J. V., Visser, C. D., and Chu, Q., "Double-model adaptive fault detection and diagnosis applied to real flight data," *Control Engineering Practice*, vol. 36, pp. 39 – 57, 2015.

[80] Hermann, R. and Krener, A., "Nonlinear controllability and observability," *IEEE Trans.. Automatic Control*, vol. 22, no. 5, pp. 728–740, October 1977.

[81] Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I., "Observability-based rules for designing consistent ekf slam estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[82] Trawny, N., Zhou, X. S., Zhou, K., and Roumeliotis, S. I., "Interrobot transformations in 3-d," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 226–243, April 2010.

[83] De Silva, O., Mann, G. K. I., and Gosine, R. G., "Observability Analysis of Relative Localization Filters Subjected to Platform Velocity Constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 5, 03 2017.

[84] Huang, G. P., Trawny, N., Mourikis, A. I., and Roumeliotis, S. I., "Observability-based consistent ekf estimators for multi-robot cooperative localization," *Autonomous Robots*, vol. 30, no. 1, pp. 99–122, Jan 2011.

[85] Hesch, J. A., Kottas, D. G., Bowman, S. L., and Roumeliotis, S. I., "Observability-constrained vision-aided inertial navigation," University of Minnesota, Department of Computer Science and Engineering, MARS Lab, Technical report 001, 2012, available at: http://www.seas.upenn.edu/~seanbow/papers/tr_2012_1.pdf.

[86] Wu, K., Zhang, T., Su, D., Huang, S., and Dissanayake, G., "An invariant-ekf vins algorithm for improving consistency," in *2017 IEEE/RSJ Intern. Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 1578–1585.

[87] Nevistić, V. and Primbs, J. A., "Receding horizon quadratic optimal control: Performance bounds for a finite horizon strategy," in *1997 European Control Conference (ECC)*, 1997, pp. 3584–3589.

[88] Grimm, G., Messina, M. J., Tuna, S. E., and Teel, A. R., "Model predictive control: for want of a local control lyapunov function, all is not lost," *IEEE Trans. Automatic Control*, vol. 50, pp. 546–558, 2005.

[89] Coron, J.-M., Grüne, L., and Worthmann, K., "Model predictive control, cost controllability, and homogeneity," *SIAM Journal on Control and Optimization*, vol. 58, pp. 2979–2996, 2020.

[90] Tuna, S. E., Messina, M. J., and Teel, A. R., "Shorter horizons for model predictive control," in *2006 Amer. Cont. Conf.*, 2006, pp. 863–868.

[91] Mehrez, M. W., Worthmann, K., Cenerini, J. P., Osman, M., Melek, W. W., and Jeon, S., "Model predictive control without terminal constraints or costs for holonomic mobile robots," *Robotics and Autonomous Systems*, vol. 127, p. 103468, 2020.

[92] Kim, H., Shim, D., and Sastry, S., "Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles," in *Proceedings of the 2002 American Control Conf.*, vol. 5, 2002, pp. 3576–3581 vol.5.

[93] Owis, M., El-Bouhy, S., and El-Badawy, A., "Quadrotor trajectory tracking control using non-linear model predictive control with ros implementation," in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, 2019, pp. 243–247.

[94] Andriën, A., Kremers, D., Kooijman, D., and Antunes, D., "Model predictive tracking controller for quadcopters with setpoint convergence guarantees," in *2020 American Control Conference (ACC)*, 2020.

[95] Zhang, K., Shi, Y., and Sheng, H., "Robust nonlinear model predictive control based visual servoing of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 700–708, 2021.

[96] Nguyen, N. T., Prodan, I., and Lefèvre, L., "Stability guarantees for translational thrust-propelled vehicles dynamics through nmpc designs," *IEEE Transactions on Control Systems Technology*, vol. 29, 2021.

[97] Schott, K. D. and Bequette, B. W., *Multiple Model Adaptive Control (MMAC)*. Dordrecht: Springer Netherlands, 1998, pp. 33–57.

[98] Yu, C., Roy, R., Kaufman, H., and Bequette, B., "Multiple-model adaptive predictive control of mean arterial pressure and cardiac output," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 8, pp. 765–778, 1992.

[99] Chi, Q. and Liang, J., "A multiple model predictive control strategy in the pls framework," *Journal of Process Control*, vol. 25, pp. 129–141, 2015.

[100] Aufderheide, B. and Bequette, B., "Extension of dynamic matrix control to multiple models," *Computers Chemical Engineering*, vol. 27, no. 8, pp. 1079–1096, 2003, 2nd Pan American Workshop in Process Systems Engineering.

[101] Rao, R., Aufderheide, B., and Bequette, B., "Experimental studies on multiple-model predictive control for automated regulation of hemodynamic variables," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 3, pp. 277–288, 2003.

[102] Nandola, N. N. and Bhartiya, S., "A multiple model approach for predictive control of nonlinear hybrid systems," *Journal of Process Control*, vol. 18, no. 2, pp. 131–148, 2008.

[103] Alexis, K., Nikolakopoulos, G., and Tzes, A., "Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances," *Control Engineering Practice*, vol. 19, no. 10, pp. 1195–1207, 2011.

[104] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., and Siegwart, R., "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[105] Furgale, P., Rehder, J., and Siegwart, R., "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1280–1286.

[106] Hartley, R. I. and Zisserman, A., *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[107] Shuster, M. D., "Survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, pp. 439–517, Oct. 1993.

[108] Markley, F. L., Cheng, Y., Crassidis, J. L., and Oshman, Y., "Averaging quaternions," *Jour. of Guidance, Cont., and Dyn.*, vol. 30, no. 4, pp. 1193–1197, 2007.

[109] Qin, T., Li, P., and Shen, S., "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[110] Yuan, T., Bar-Shalom, Y., Willett, P., Mozeson, E., Pollak, S., and Hardiman, D., "A multiple imm estimation approach with unbiased mixing for thrusting projectiles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3250–3267, October 2012.

[111] Cao, N. and Lynch, A. F., "Inner–outer loop control for quadrotor uavs with input and state constraints," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1797–1804, 2016.

[112] Andersson, O., Ljungqvist, O., Tiger, M., Axehill, D., and Heintz, F., "Receding-horizon lattice-based motion planning with dynamic obstacle avoidance," in *2018 IEEE Conf. on Decision and Control*, 2018.

[113] Mueller, M. W., Hehn, M., and D'Andrea, R., "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[114] "Asctec hummingbird with autopilot user's manual," Ascending Technologies GmbH, Tech. Rep. [Online]. Available: http://robotics.caltech.edu/~ndutoit/wiki/images/7/70/AscTec_AutoPilot_manual_v1.0_small.pdf

[115] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M., "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[116] Bock, H. and Plitt, K., "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[117] Brunot, M., "A gaussian uniform mixture model for robust kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 2656–2665, 2020.

[118] Qin, T. and Shen, S., "Robust initialization of monocular visual-inertial estimation on aerial robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4225–4232.

[119] Faessler, M., Fontana, F., Forster, C., and Scaramuzza, D., "Automatic reinitialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1722–1729.

[120] Zhu, A. Z., Atanasov, N., and Daniilidis, K., "Event-based visual inertial odometry," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5816–5824.

[121] Thalagala, R. G., "Comparison of state marginalization techniques in visual inertial navigation filters," Master's thesis, Memorial University of Newfoundland, NL, Canada, 2019.

[122] Li, J., Bao, H., and Zhang, G., "Rapid and robust monocular visual-inertial initialization with gravity estimation via vertical edges," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6230–6236.

[123] Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V., *Initialization-Free Monocular Visual-Inertial State Estimation with Application to Autonomous MAVs.* Cham: Springer International Publishing, 2016.

[124] Weiss, C., Frohlich, H., and Zell, A., "Vibration-based terrain classification using support vector machines," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4429–4434.

[125] Born, W. K. and Lowrance, C. J., "Application of convolutional neural network image classification for a path-following robot," in *2018 IEEE MIT Undergraduate Research Technology Conference (URTC)*, 2018, pp. 1–4.

[126] From, H., "Predicting the ground effect in drone landing with online learning," Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2019.

[127] Zhang, G., "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.

Appendix A

Transformation to the Body Frame at the C.G. of the Helicopter

All the forces including thrust force, drag force, and gravitational acceleration force and the accelerometer measurements given in Fig. A.1 will be resolved at $\{G\}$ to compute the velocity of $\{I\}$ w.r.t $\{G\}$ expressed in $\{G\}$, ${}^{G}\mathbf{v}_{GI}$, then transform it back to $\{I\}$ to get ${}^{I}\mathbf{v}_{GI}$ or ${}^{I}\mathbf{v}$ for simplicity. The drag force at $\{B\}$ is ${}^{B}\bar{\mathbf{D}}_{L}{}^{B}\mathbf{v}_{GB}$, where ${}^{B}\bar{\mathbf{D}}_{L}$ is the drag parameters matrix expressed in $\{B\}$ and ${}^{B}\mathbf{v}_{GB}$ is the velocity of $\{B\}$ w.r.t $\{G\}$ expressed in $\{B\}$. Due to this coordinate transformation there will be an extra term in the velocity state dynamics as follows,

$$
{}^{I}\dot{\hat{\mathbf{v}}} = R({}^{I}\hat{\mathbf{q}}_{G})^{G}g\bar{\mathbf{e}}_{\mathbf{3}} + R({}^{I}\hat{\mathbf{q}}_{B})^{B}\hat{\bar{\mathbf{b}}}_{a}\bar{\mathbf{e}}_{3} - (\boldsymbol{\omega}_{m} - \hat{\mathbf{b}}_{g}) \times {}^{I}\hat{\mathbf{v}} - {}^{I}\bar{\mathbf{D}}_{L}{}^{I}\hat{\mathbf{v}} - {}^{I}\bar{\mathbf{D}}_{L}(\boldsymbol{\omega}_{m} - \hat{\mathbf{b}}_{g}) \times {}^{I}\mathbf{p}_{IB}
$$

where ${}^{I}\mathbf{p}_{IB}$ and $R({}^{I}\mathbf{q}_{B})$ are the translation and orientation from $\{B\}$ to $\{I\}$ that are given in the dataset, ${}^{I}\bar{\mathbf{D}}_{L}$ is the drag parameters matrix expressed in $\{I\}$ and is calculated as ${}^{I}\bar{\mathbf{D}}_{L} = R({}^{I}\mathbf{q}_{B})^{B}\bar{\mathbf{D}}_{L}R({}^{I}\mathbf{q}_{B})^{T}$, and ${}^{B}\hat{\bar{\mathbf{b}}}_{a}$ is the new accelerometer bias state that includes the thrust force and expressed in $\{B\}$. The inertial measurement

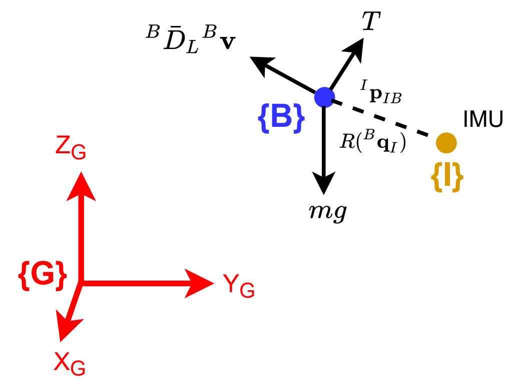


Fig. A.1: Applied forces at the body frame at the C.G. of the helicopter.

model will be updated as well as follows,

$$\mathbf{h}_a = -{}^I\bar{\mathbf{D}}_L{}^I\mathbf{v} + R({}^I\hat{\mathbf{q}}_B){}^B\bar{\mathbf{b}}_a - {}^I\bar{\mathbf{D}}_L(\boldsymbol{\omega}_m - \mathbf{b}_g) \times {}^I\mathbf{p}_{IB} + \mathbf{n}_a$$

For the C-VINS filter, the only change happens in the velocity state equation due to the transformation of the accelerometer bias from $\{I\}$ to $\{B\}$, as follows,

$${}^I\dot{\hat{\mathbf{v}}} = R({}^I\hat{\mathbf{q}}_G){}^Gg\bar{\mathbf{e}}_\mathbf{3} + (\mathbf{a}_m - R({}^I\mathbf{q}_B){}^B\mathbf{b}_a) - (\boldsymbol{\omega}_m - \mathbf{b}_g) \times {}^I\hat{\mathbf{v}}$$

The process noise covariance matrix corresponding to the accelerometer bias in both filters has to be transformed to $\{B\}$ as well, as follows,

$${}^BQ_w = R({}^I\mathbf{q}_B)^T Q_w R({}^I\mathbf{q}_B)$$

Appendix B

Unbiased Approach for Unequal Dimension States IMM

The DF-VINS process model and state covariance matrix can be rewritten as,

$$\hat{\mathbf{x}}^{(1)}_{k|k} = \begin{bmatrix} \hat{\bar{\mathbf{x}}}^{(1)}_{k|k} \\ {}^{B}\hat{\bar{\mathbf{b}}}^{(1)}_{a\ k|k} \end{bmatrix} \quad , \quad \mathbf{P}^{(1)}_{k|k} = \begin{bmatrix} \mathbf{P}^{(\bar{x}\bar{x},1)}_{k|k} & \mathbf{P}^{(\bar{x}\bar{b}_a,1)}_{k|k} \\ \mathbf{P}^{(\bar{b}_a\bar{x},1)}_{k|k} & \mathbf{P}^{(\bar{b}_a\bar{b}_a,1)}_{k|k} \end{bmatrix}$$

where ${}^{B}\bar{\mathbf{b}}_a$ is the accelerometer bias including the thrust force expressed in $\{B\}$ and $\bar{\mathbf{x}}$ is state vector that includes the other states proposed in (3.7). The same for the C-VINS filter, the process model and state covariance matrix can be rewritten as,

$$\hat{\mathbf{x}}^{(2)}_{k|k} = \begin{bmatrix} \hat{\bar{\mathbf{x}}}^{(2)}_{k|k} \\ {}^{B}\hat{\mathbf{b}}^{(2)}_{a\ k|k} \end{bmatrix} \quad , \quad \mathbf{P}^{(2)}_{k|k} = \begin{bmatrix} \mathbf{P}^{(\bar{x}\bar{x},2)}_{k|k} & \mathbf{P}^{(\bar{x}b_a,2)}_{k|k} \\ \mathbf{P}^{(b_a x,2)}_{k|k} & \mathbf{P}^{(b_a b_a,2)}_{k|k} \end{bmatrix}$$

where ${}^{B}\mathbf{b}_a$ is the accelerometer bias expressed in $\{B\}$ and $\bar{\mathbf{x}}$ is state vector that includes the other states proposed in (3.3). Using the unbiased approach for unequal dimension states, the state interaction can be implemented as follow,

$$\hat{\mathbf{x}}^{(01)} = \tilde{\mu}^{1|1}\hat{\mathbf{x}}_{k|k}^{(1)} + \tilde{\mu}^{2|1}\hat{\mathbf{x}}_{k|k}^{(2|1)}$$

$$\hat{\mathbf{x}}^{(02)} = \tilde{\mu}^{1|2}\hat{\mathbf{x}}_{k|k}^{(1|2)} + \tilde{\mu}^{2|2}\hat{\mathbf{x}}_{k|k}^{(2)}$$

where

$$\hat{\mathbf{x}}_{k|k}^{(2|1)} = \begin{bmatrix} \hat{\hat{\mathbf{x}}}_{k|k}^{(2)} \\ {}^{B}\hat{\hat{\mathbf{b}}}_{a}^{(1)}{}_{k|k} \end{bmatrix} \quad , \quad \hat{\mathbf{x}}_{k|k}^{(1|2)} = \begin{bmatrix} \hat{\hat{\mathbf{x}}}_{k|k}^{(1)} \\ {}^{B}\hat{\mathbf{b}}_{a}^{(2)}{}_{k|k} \end{bmatrix}$$

While the covariance interaction is implemented as follow,

$$\begin{aligned} \mathbf{P}^{(01)} = \tilde{\mu}^{1|1} &\left[\mathbf{P}^{(1)} + \left(\hat{\mathbf{x}}^{(1)} - \hat{\mathbf{x}}^{(01)}\right)\left(\hat{\mathbf{x}}^{(1)} - \hat{\mathbf{x}}^{(01)}\right)^{T}\right] \\ +\tilde{\mu}^{2|1} &\left[\mathbf{P}^{(2|1)} + \left(\hat{\mathbf{x}}^{(2|1)} - \hat{\mathbf{x}}^{(01)}\right)\left(\hat{\mathbf{x}}^{(2|1)} - \hat{\mathbf{x}}^{(01)}\right)^{T}\right] \end{aligned}$$

$$\begin{aligned} \mathbf{P}^{(02)} = \tilde{\mu}^{1|2} &\left[\mathbf{P}^{(1|2)} + \left(\hat{\mathbf{x}}^{(1|2)} - \hat{\mathbf{x}}^{(02)}\right)\left(\hat{\mathbf{x}}^{(1|2)} - \hat{\mathbf{x}}^{(02)}\right)^{T}\right] \\ +\tilde{\mu}^{2|2} &\left[\mathbf{P}^{(2)} + \left(\hat{\mathbf{x}}^{(2)} - \hat{\mathbf{x}}^{(02)}\right)\left(\hat{\mathbf{x}}^{(2)} - \hat{\mathbf{x}}^{(02)}\right)^{T}\right] \end{aligned}$$

where

$$\mathbf{P}^{(1|2)} = \begin{bmatrix} \mathbf{P}^{(\bar{x}\bar{x},1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{(b_a b_a,2)} \end{bmatrix} \quad , \quad \mathbf{P}^{(2|1)} = \begin{bmatrix} \mathbf{P}^{(\bar{x}\bar{x},2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{(\bar{b}_a \bar{b}_a,1)} \end{bmatrix}$$

Appendix C

Stability Analysis for Conventional NMPC

The running cost in (5.2) is tailored as follows,

$$\ell(\mathbf{x},\mathbf{u}) = \sum_{i=1}^{8} q_i \left(x_i - x_{r,i}\right)^2 + r_1(u_1 - \tilde{u}_1)^2 + r_2(u_2 - \tilde{u}_2)^2 + r_3(u_3 - \tilde{u}_3)^2, \tag{C.1}$$

which is designed to satisfy the asymptotic stability conditions presented in Theorem 5.1 in Section 5.4.1, where

$$\tilde{u}_1 = \frac{\mathrm{g}}{\mathrm{c}x_7\mathrm{c}x_8},\ \tilde{u}_2 = \frac{\bar{\tau}_\phi}{\bar{K}_\phi}x_7,\ \tilde{u}_3 = \frac{\bar{\tau}_\theta}{\bar{K}_\theta}x_8,$$

$\mathrm{c}x_7 \coloneqq \cos(x_7)$, and $\mathrm{c}x_8 \coloneqq \cos(x_8)$.

The growth bound γ_i given in Assumption 5.1 can be obtained by constructing a summable sequence $c_j \subseteq \mathbb{R}_{\geq 0},\ j \in \mathbb{N}_0$ that satisfies Inequality (5.20), such that $\gamma_i = \sum_{j=0}^{i-1} c_j,\ i \in \mathbb{N}_{\geq 2}$. The computation of the sequence c_j and thus the growth bound γ_i is governed by the following proposition.

Proposition C.1. *Consider the system model* (5.5) *and running costs* (C.1)*. Let the* *penalty parameters in (C.1) are given as*

$$r_1 \leq \frac{\sigma}{16} q_6, \quad r_2 \leq \sigma \bar{K}_\phi^2 q_7, \quad r_3 \leq \sigma \bar{K}_\theta^2 q_8 \tag{C.2}$$

with weighting ratio $\sigma \in \mathbb{Q}$, and let θ and $\phi \in [-60°, 60°]$. Then, condition (5.13) holds with $\gamma_i = \sum_{j=0}^{i-1} c_j$, $i \in \mathbb{N}_{\geq 2}$, where sequence c_j is governed by (5.22). Also, there exists a prediction horizon $N \in \mathbb{N}$ such that condition (5.16) holds and the NMPC closed-loop with N is asymptotically stable.

Proof. The quadrotor trajectories can be chosen as straight lines, curves, or lattice shapes, as given by function (5.23). Substitute (5.5) into (5.23) in order to find the required open-loop control inputs ($\mathbf{u}_{\mathbf{x}_0}$) for the maneuver. The first control input u_1 was given in (5.5) as

$$x_6[j+1] = x_6[j] - \delta \mathrm{g} + \delta R_{33} u_1[j]$$

where, $R_{33} = cx_7 cx_8$ is the element in the third row and third column in the rotation matrix R. Thus, the control input $u_{1_{\mathbf{x}_0}}$ for all $\mathbf{x}_0 \in \mathbf{X}$ (or u_1 for the sake of simplicity) can be calculated as follows:

$$\left(\frac{\lambda^\rho - (j+1)^\rho}{\lambda^\rho}\right) x_{0,6} - \left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho}\right) x_{0,6} = -\delta \mathrm{g} + \delta R_{33} u_1. \tag{C.3}$$

Using the binomial expansion, (C.3) reduces to

$$\left(\frac{-\sum_{i=0}^{\rho-1} {}^\rho C_i \; j^i}{\delta \lambda^\rho}\right) x_{0,6} + \mathrm{g} = R_{33} u_1.$$

Let $\mathcal{C}_\rho = \sum_{i=0}^{\rho-1} {}^\rho C_i \; j^i$ that yields,

$$u_1 = \frac{\mathrm{g} - \frac{\mathcal{C}_\rho}{\delta \lambda^n} x_{0,6}}{cx_7 cx_8}. \tag{C.4}$$

The calculations of the second and third control inputs are similar to (5.26) and (5.27), respectively, as follows

$$u_2 = \frac{-\mathcal{C}_\rho + \delta\bar{\tau}_\phi(\lambda^\rho - j^\rho)}{\delta\bar{K}_\phi\lambda^\rho}x_{0,7}. \tag{C.5}$$

$$u_3 = \frac{-\mathcal{C}_\rho + \delta\bar{\tau}_\theta(\lambda^\rho - j^\rho)}{\delta\bar{K}_\theta\lambda^\rho}x_{0,8}. \tag{C.6}$$

Applying (5.23), (C.4), (C.5), and (C.6) into (C.1) yields the running costs (C.7) along the resulting open-loop trajectories.

$$\begin{aligned}\ell(x_{u_{\mathbf{x}_0}}(j;\mathbf{x}_0), \mathbf{u}_{\mathbf{x}_0}(j)) = \left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho}\right)^2 \sum_{i=1}^{8} q_i\left(x_{0,i} - x_{r,i}\right)^2 + r_1\frac{(\frac{\mathcal{C}_\rho}{\delta\lambda^\rho})^2}{c^2x_7c^2x_8}x_{0,6}^2 \\ +r_2\left(\frac{-\mathcal{C}_\rho}{\delta\bar{K}_\phi\lambda^\rho}\right)^2 x_{0,7}^2 + r_3\left(\frac{-\mathcal{C}_\rho}{\delta\bar{K}_\theta\lambda^\rho}\right)^2 x_{0,8}^2\end{aligned} \tag{C.7}$$

To this end, the bounding sequence c_j can be found by bounding the running costs (C.7) such that condition (5.20) is satisfied. We use the limit on the attitude angles as θ and $\phi \in [-60°, 60°]$ as this bound captures the nominal non-aggressive flight trajectories of the quadrotors. As a result, the second term in (C.7) can be bounded as

$$r_1\frac{(\frac{\mathcal{C}_\rho}{\delta\lambda^\rho})^2}{\cos^2(x_7)\cos^2(x_8)}x_{0,6}^2 \le r_1\frac{16}{\delta^2}\left(\frac{\sum_{i=0}^{\rho-1}{}^\rho C_i\, j^i}{\lambda^\rho}\right)^2 x_{0,6}^2$$

where, $1/\cos^2(x_7)\cos^2(x_8)$ has an upper bound of 16. Using condition (C.2) and recalling that $j \le \lambda$ yield

$$r_1\frac{(\frac{\mathcal{C}_\rho}{\delta\lambda^\rho})^2}{\cos^2(x_7)\cos^2(x_8)}x_{0,6}^2 \le r_1\frac{16}{\delta^2}\left(\frac{\lambda^{\rho-1}\sum_{i=0}^{\rho-1}{}^\rho C_i}{\lambda^\rho}\right)^2 x_{0,6}^2 \le \sigma q_6\frac{\left(\sum_{i=0}^{\rho-1}{}^\rho C_i\right)^2}{\delta^2\lambda^2}\, x_{0,6}^2$$

Moreover, the third and fourth terms in (C.7) can be bounded in the same manner as follows:

$$r_2 \left(\frac{-\mathcal{C}_\rho}{\delta \bar{K}_\phi \lambda^\rho} \right)^2 x_{0,7}^2 \leq r_2 \frac{1}{\delta^2 \bar{K}_\phi^2} \frac{\left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2}{\lambda^2} x_{0,7}^2 \leq \sigma q_7 \frac{\left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2}{\delta^2 \lambda^2} x_{0,7}^2$$

$$r_3 \left(\frac{-\mathcal{C}_\rho}{\delta \bar{K}_\theta \lambda^\rho} \right)^2 x_{0,8}^2 \leq \sigma q_8 \frac{\left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2}{\delta^2 \lambda^2} x_{0,8}^2$$

As a result, the running costs (C.7) can be estimated by

$$\begin{aligned} \ell(x_{u_{\mathbf{x}_0}}(j;\mathbf{x}_0), \mathbf{u}_{\mathbf{x}_0}(j)) &\leq \left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho} \right)^2 \|\mathbf{x}_0 - \mathbf{x}_f\|_Q^2 + \frac{\sigma}{\delta^2 \lambda^2} \left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2 \sum_{i=6}^{8} q_i x_{0,i} \\ &\leq \left[\left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho} \right)^2 + \frac{\sigma}{\delta^2 \lambda^2} \left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2 \right] \|\mathbf{x}_0 - \mathbf{x}_f\|_Q^2 \end{aligned} \tag{C.8}$$

Therefore, the bounding sequence c_j in (5.20) can be attained as in (5.22). Finally, the growth bound $\gamma_k \coloneqq \sum_{j=0}^{k-1} c_j$, $k \in \mathbb{N}_0$ can be obtained as given in Theorem C.1 by

$$\gamma_k = \sum_{j=0}^{k-1} \left[\left(\frac{\lambda^\rho - j^\rho}{\lambda^\rho} \right)^2 + \frac{\sigma}{\delta^2 \lambda^2} \left(\sum_{i=0}^{\rho-1} {}^\rho C_i \right)^2 \right], \tag{C.9}$$

which is the same bounding sequence as in (5.33).

□