Disturbance Observer-Based Motion Control and Visual-Inertial-Actuator Odometry for UAVs

by

Amir Moeini

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Motion control of multirotor Unmanned Aerial Vehicles (UAV) is an area of research which continues to generate significant interest in the community. Being able to accurately follow a broad class of trajectories clearly improves the mission capabilities of the vehicle. Model uncertainty and external disturbances are important factors reducing motion control performance. Improving the robustness of motion control can clearly broaden UAV capability. In this thesis we propose a number of trajectory tracking motion controls which are based on a backstepping design method. By incorporating disturbance observers for external force and torque the proposed methods provide exponentially stable tracking error dynamics for the constant disturbance case. For time-varying disturbances tracking error is proven to be ultimately bounded. The stability analysis accounts for the full nonlinear vehicle model which includes translational and rotational dynamics. This avoids having to make common simplifying assumptions typical of designs with inner outer loop structure (e.g., linear approximation of the rotational dynamics) during the closed-loop stability analysis. The proposed design provides a model-based partial compensation of rotor drag. Software-in-the-loop (SITL) simulation and experimental flight testing results are presented. These results show the effectiveness of the proposed method using the commonly used open-source PX4/Pixhawk development framework. The results demonstrate the methods' practical usefulness including their robustness and tracking error performance.

The developed motion control algorithms require an accurate knowledge of the system's state, in some applications a description of the environment and an estimate of external forces (e.g., aerial manipulation and load transport). Having an algorithm that only depends on onboard sensors can increase autonomy and reliability. There has been an increasing amount of work developing state estimation algorithms using vision and inertial measurements. However, incorporation of the dynamics modelling and actuation data, which are already available on UAVs and can provide

more information about the vehicle's motion, are normally ignored or just an approximate model with unrealistic assumption on force modelling is used. In this thesis, we include an accurate dynamical modelling of a multirotor by considering the effect of rotor drag and also a disturbance observer developed with the assumption of constant force disturbance into an existing open source state estimation approach. The effect of rotor drag is proved to be significant in control and state estimation and its consideration can improve both the estimation and control tasks. In addition, the proposed disturbance observer which is reformulated as a residual term can assist the estimator to differentiate between the constant or slowly time-varying component of the external force and the accelerometer bias providing a more accurate force estimate which is a need in many UAVs applications. Furthermore, this structure increases the odometry accuracy. We evaluate the performance of our proposed method by integrating it with an open source Visual-Inertial Odometry (VIO) system and testing it on benchmark datasets. The results show a significant improvement in the estimation accuracy.

Preface

- Section 3.2 has been published as [A. Moeini, A.F. Lynch, and Q. Zhao, "Disturbance observer-based nonlinear control of a quadrotor UAV," *Advanced Control with Applications*, vol. 2, no. 1, pp. 1–20, 2019.]. I was responsible for theoretical derivation, simulation and manuscript composition. Dr. Lynch and Dr. Zhao provided a supervisory role and helped with concept formation and writing the manuscript.

- Section 3.3 has been published as [A. Moeini, A.F. Lynch, and Q. Zhao, "A backstepping disturbance observer control for multirotor UAVs: theory and experiment," *International Journal of Control*, 2021]. I was responsible for theoretical derivation, experimental implementation and manuscript composition. Dr. Lynch and Dr. Zhao provided a supervisory role and helped with concept formation and writing the manuscript.

- An initial version of Section 3.4 was published as [A. Moeini, M. A. Rafique, Z. Xue, A. F. Lynch, and Q. Zhao, "Disturbance observer-based integral backstepping control for UAVs," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 382–388]. I was responsible for theoretical derivation, simulation and manuscript composition. M.A. Rafique and Z. Xue were responsible for adapting the jMAVSim Simulator included in the PX4 framework. Dr. Lynch and Dr. Zhao provided a supervisory role and helped with concept formation and writing the manuscript.

- The current results in Section 3.4 has been submitted as [A. Moeini, A. Lynch, and Q. Zhao, "Exponentially stable motion control for multirotor UAVs with rotor drag and disturbance compensation," *Journal of Intelligent and Robotic Systems*, vol. 103, no. 1, 2021]. I was responsible for theoretical derivation, simulation and manuscript composition. Dr. Lynch and Dr. Zhao provided a supervisory role and helped with concept formation and writing the manuscript.

- Chapter 4 is under preparation as [A. Moeini, A. Lynch, and Q. Zhao, "State estimation of multirotor UAVs subject to rotor drag and external disturbance using visual-inertial-actuator data"]. I was responsible for theoretical derivation, software implementation, simulation, performance analysis and manuscript composition. Dr. Lynch and Dr. Zhao provided a supervisory role and helped with concept formation and writing the manuscript.

To my lovely wife

Acknowledgements

I would like to express my deep appreciation to my supervisors Professor Alan Lynch and Professor Qing Zhao for their continued support and encouragement during my entire PhD program. I am incredibly grateful for their patience, guidance, and mentorship. Also, I would like to thank Professor Mahdi Tavakoli for his help and constructive comments on my research. In addition, I am extremely grateful to the rest of my dissertation committee members Professor Christopher Nielsen and Professor Stevan Dubljevic for their insightful criticisms and suggestions.

I would kindly appreciate my dear friends Hamid Niazi, Mohammad Khalkhali, and Muhammad Awais Rafique for all their help and encouragement during my PhD studies. Many thanks to my parents for their kindness, patience, and endless support. At last, my deepest gratitude goes to my wife for her consistent emotional support and encouragement throughout my research.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ANCL	Applied Nonlinear Control Lab
ATE	Absolute Trajectory Error
BS	Backstepping
DIY	Do-It-Yourself
DOB-BS	Disturbance Observer-Based Backstepping
DOB-IBS	Disturbance Observer-Based Integral Backstepping
DoF	Degree of Freedom
EKF	Extended Kalman Filter
ESC	Electronic Speed Controller
GWN	Gaussian White Noise
IBS	Integral Backstepping
IMU	Inertial Measurement Unit
KLT	Kanade-Lucas-Tomasi
LQ	Linear Quadratic
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MAP	Maximum-a-Priori
MCS	Motion Capture System
NSS	Nonlinear Sliding Surface
PD	Proportional-Derivative
PDF	Probability Density Function
PID	Proportional-Inegral-Derivative
PnP	Perspective-n-Point
PTAM	Parallel Tracking and Mapping
PWM	Pulse Width Modulation
QGC	QGroundControl

RMS	Root Mean Square
RMSE	Root Mean Square Error
SITL	Software-In-The-Loop
SLAM	Simultaneous Localization and Mapping
UAM	Urban Air Mobility
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
uORB	Micro-Object Request Broker
VIAO	Visual-Inertial-Actuator Odometry
VIO	Visual-Inertial Odometry
VISLAM	Visual Inertial Simultaneous Localization and Mapping
VO	Visual Odometry
VSLAM	Visual Simultaneous Localization and Mapping

Chapter 1

Introduction

Multirotors are a special type of Unmanned Aerial Vehicles (UAV) which have gained popularity in applications due to their unique advantages: simple and robust mechanical structure, maneuverability, available open-source autopilots, and low-cost [2]. Example applications for these UAVs include environment monitoring, terrain mapping, and emergency response [3]. The importance of multirotors is confirmed by the numerous research projects which they have generated. Well known examples are Skydio [1], Urban Air Mobility (UAM) [2], DARPA Fast Lightweight Autonomy [4], and the MBZIRC competition [3]. This thesis is divided into two parts. The first part focuses on improving the robustness of UAV motion control to external wrench with the assumption that the vehicle state is an available measurement. The second part investigates visual-inertial state estimation which provides a vehicle state and an estimate of the external forces. Hence, the results are particularly relevant to improving the system's autonomy and handling applications with external forces.

1.1 Motion Control

Motion control of multirotor Unmanned Aerial Vehicles (UAV) is an area of research which continues to generate significant interest in the community [5]. We remark that this thesis focusses on traditional quadrotor UAVs given their abovementioned benefits. However, in general the results derived are applicable to other rotary-wing UAVs in other configurations, e.g., traditional helicopters with a main and tail rotor. Motion control normally involves tracking specified trajectories for the vehicle's 3D position and yaw. Path following is another related control objective. Being able to accurately and robustly track a broad class of trajectories clearly improves the mission capabilities of the vehicle. Interest in motion control comes from a range of challenges due to: the system's underactuated nonlinear dynamics [6], bounded inputs [7], model uncertainty [8], measurement noise [9], and external disturbances [1]. Although the topic of motion control for these vehicles

[1] https://www.skydio.com/
[2] https://www.easa.europa.eu/domains/urban-air-mobility-uam
[3] https://www.mbzirc.com/

has received significant attention to-date, improving robustness to external disturbance forces and torques remains an important goal which can broaden UAV capability. For example, emerging load transport applications lead to a range of external force and torques which must be compensated [10]. Other examples of disturbances include wind gusts [11] or contact wrench from the environment due to a manipulator attached to the UAV [12, 13]. These disturbances have significant effect on motion control performance [14] and should be compensated. In addition, a force which is usually not explicitly accounted for is rotor drag which has attracted attention recently [15, 16]. Especially for high air speed applications, rotor drag is an important force whose compensation improves motion control.

One approach to account for disturbances uses a disturbance observer (DO) whose estimate is used in the state feedback control to cancel its effect on the closed-loop stability. Usually estimation is done by comparing measurements of the state with the value expected based on a nominal system model. The accuracy of the disturbance estimate depends on an accurate nominal system model used in the observer design. In addition to the robustness to the external disturbances, successful motion control must be robust to uncertainty, implementable on onboard autopilots with specific software and hardware and provide provable bounds for performance. This thesis proposes a disturbance observer-based state feedback to improve the robust performance of motion control.

1.2 Visual-Inertial Odometry (VIO)

The state feedback motion control discussed in Section 1.1 requires an estimate of the vehicle state (i.e., position, linear velocity, attitude, and angular velocity) and in general some applications require a map of the vehicle's environment (e.g., to perform obstacle avoidance or path planning). Which vehicle states can or need to be estimated and whether a map is required depends on the available sensors, the motion control task, and the level of autonomy needed. The literature contains work involving many of these possible combinations. For example, optical flow and an IMU measurement might be sufficient (i.e., no GPS is required) for landing on a moving platform [17]. Performing state estimation using onboard sensors clearly improves the UAV's autonomy, e.g., it enables GPS-denied applications and avoids dependence on a Motion Capture System (MCS) [2, 18]. However, these state estimation problems are particularly challenging for UAVs given their high speed, maneuverability, limited computational power, and operation in challenging environments.

Two closely related particular state estimation problems which have been extensively researched are *Visual Odometry* (VO) and *Visual Simultaneous Localization and Mapping (SLAM)* (VSLAM), which are algorithms that process only images from a computer vision system in order to estimate the UAV state and a map the environment. See [19, 20] for an overview and tutorial. A recent survey of the developments on VSLAM is in [21]. Although other sensors such as Lidar can augment or replace vision, a camera has the advantages of being lightweight, passive, low-power, inexpensive, and suitable for a range of environments. Vision yields vast information about the environment which makes it suitable for mapping [22]. Even in applications where it is not required, estimating

a map alongside the vehicle state improves vehicle state estimation accuracy. Vision data can be captured by a range of sensors, e.g., monocular, RGBD, and stereo cameras. Multiple cameras can also be used. RGBD cameras measure the depth by projecting a pattern on the environment and processing its distortion. Stereo cameras have the advantage of depth estimation which can be useful in state estimation. However, RGBD and stereo cameras suffer from a number of limitations including limited range and increased cost. Hence, this thesis adopts a monocular camera as it is inexpensive lightweight and commonly available sensor.

One problem with monocular vision-only estimation algorithms is determining metric scale and the direction of gravitational force. This limits their use in some applications [23]. Another limitation is their relatively slow estimation frequency which makes them unsuitable for applications such as UAV motion control [24]. These limitations motivate fusing the relatively slow visual data with high frequency inertial measurement units (IMU) data, i.e., to perform so-called Visual-inertial Odometry (VIO) or SLAM (VISLAM). A performance comparison of some recent methods is in [25].

Combining monocular vision with an IMU allows the scale to be estimated. Furthermore, the IMU renders the direction of gravitational force observable which is essential for UAV motion control. In addition, employing an IMU can increase the system's reliability by bridging the gap between a VIO failure and reinitialization. Failures can arise due to motion blur, sudden change of illumination, and texture-less environments [23]. Developing an accurate and robust VIO that is enable to operate in real environments and which can provide state estimates for being used in UAV motion control is a major challenge. This thesis provides an extension to an existing VIO system [26, 27] by integrating the estimation of external forces including rotor drag.

1.3 Motion Control: Literature Review

Traditional Control: Many *traditional* linear and nonlinear motion control laws have been proposed, where *traditional* refers to a little or no analysis for the design's robustness. Typically linear control refers to a design based on an approximate linear model. Examples of such linear designs include Proportional-Integral-Derivative (PID) control in [6] or Linear Quadratic (LQ) optimal control in [28, 29]. Since linear designs lead to a local convergence result, nonlinear control methods, which directly account for all or part of the nonlinearity in the system's dynamics, have been proposed to achieve improved performance supported by a mathematically rigorous proof. Examples of existing nonlinear methods include backstepping in [30, 31], feedback linearization in [32, 33], sliding mode in [34–36], and differential flatness in [15, 37, 38]. We review the relevant subset of these papers below. As discussed below, many nonlinear methods take an inner-outer loop design approach which designs a separate control for the outer translational and inner rotational subsystems. This approach has the advantage of a simple design and is often used in practice, e.g., in the PX4 autopilot firmware [39]. Normally most methods which have this structure ignore the effect of coupling between inner and outer loops on closed-loop stability [40, 41]. On the other

hand, centralized designs account for the entire (i.e., coupled rotational and translational) dynamics in the stability analysis, e.g., [38, 42, 43]. These methods, like the centralized approach proposed in this thesis, result in an improved stability proof with fewer simplifying assumptions.

Original work on nonlinear control uses dynamic state feedback linearization (DSFBL) for a traditional helicopter UAV [32]. With typical modelling assumptions, nominal helicopter dynamics are the same as a quadrotor, and hence the result in [32] can be applied to quadrotors with small modifications in how the inputs appear. The DSFBL extends the state by two dimensions to include thrust input and its time derivative. The approach does not provide a robustness analysis to general model uncertainty, e.g., additive force and torque disturbance entering the translational and rotational dynamics.

Since a quadrotor admits a DSFBL, it is necessarily differentially flat [44] and work in [37, 45] applies this framework for open-loop motion planning or trajectory generation. In this same work, and similar to [46], an inner-outer state feedback control is proposed to avoid the complexities of DSFBL control law expressions. The proposed inner rotational loop uses rotation matrices to avoid the drawbacks associated with parameterizations of SO(3), e.g., the singularities of the commonly-used Euler angles [47]. Related earlier work on an inner-outer loop flatness-based control for a traditional helicopter is in [48].

Backstepping is a popular recursive method for controlling a class of nonlinear systems [49]. This approach has a number of benefits including simpler controller expressions and a natural integration into an adaptive and robust design framework. For example, the non-robust cancellation of model nonlinearity can be avoided. Various forms of backstepping controllers have been proposed for motion control of UAVs using a centralized structure [30, 50, 51] and using an inner-outer loop structure in [52]. As backstepping is performed iteratively subsystem-at-a-time, it leads to a simpler control law relative to the DSFBL in [32]. In particular, backstepping can be implemented using reduced amount of state augmentation [30, 50, 51] as only thrust is taken as the controller state. Alternately, backstepping can be performed using an inner-outer loop approach which eliminates the need for state augmentation, e.g., [52]. Backstepping has been shown to be robust relative to other control methods in experiment [53]. However, these are experimental results and no rigorous theory for the robustness is provided.

Most of the above-mentioned methods suffer from a lack of rigorous robustness analysis to external disturbances, parametric model uncertainty, or unmodeled dynamics. Some exceptions which address robustness include adaptive techniques [1, 54], integral state augmentation [53, 55], sliding surfaces [56, 57], and disturbance observer based control [9, 41, 58, 59]. We focus on robust designs below.

Adaptive Control: A number of researchers have applied adaptive control to improve robustness. Work in [1] proposes an adaptive backstepping control which is proven asymptotically stable for constant force disturbance which is treated as an uncertain parameter. The method augments the state with thrust and its time derivative, and uses two parameter update laws to estimate the

force disturbance. Torque disturbance is not considered and overparameterization [60, Ch. 3] of the disturbance leads to two 3-dimensional parameter update laws. A saturation function-based control from [61] is used to limit thrust while ensuring asymptotic stability. A projection operator is used to avoid wind-up of the parameter estimates. The combination of the projection operator, overparameterization, and saturation functions contributes to the complexity of the design. The method is validated experimentally with the controller implemented on a ground computer using Matlab. The work by the same authors in [62] also takes a backstepping approach which relies on overparameterization. A saturated control and timing law are used to ensure bounded thrust in a path following problem. Both [1, 62] do not control yaw motion or consider the time-varying disturbance case.

Work in [54] designs an adaptive nonlinear control which compensates for disturbances. This work is based on the non-adaptive design [37, 46]. Similar to that work, [54] is developed on the Special Euclidean Group SE(3), i.e., it uses rotation matrices to represent attitude. Separate attitude and position control modes are proposed. In both modes, adaptive laws compensate the effect of torque and force disturbances which are assumed linearly parameterized by an unknown constant parameter. The position mode provides position tracking by relating the direction of desired thrust to the position error. This direction defines a column of the desired rotation matrix. The controller is capable of aggressive maneuvers not commonly considered in the literature. Almost global attractiveness is guaranteed in the case of constant disturbances provided certain gain conditions are satisfied which depend on the model parameters such as mass and inertia. The authors do not consider the effect of mode switching on stability of the hybrid system. The method is validated in simulation and experiment but the effect of external disturbances is not investigated. A local inner-outer loop stability result is provided.

Robust Control: Work in [38] proposes a flatness-based approach which is related to [32], but explicitly accounts for disturbances. As in [32], the state is augmented with a nominal thrust variable and its time derivative. The disturbance is assumed to enter the snap dynamics and its norm is assumed bounded by a linear function of the norm of translational tracking error. This is a restrictive assumption (e.g., nonzero constant disturbances do not satisfy the assumption) which allows for exponential stability to be proven. The control law assumes the magnitudes of roll and pitch are bounded by $\pi/2$ to avoid singularities in the control law which appear in DSFBL using Euler angles.

Sliding mode control is commonly used to improve robustness in a range of applications. A Nonlinear Sliding Surface (NSS) method with inner-outer loop structure is presented in [56]. The NSS provides a variable damping ratio for the closed-loop. This leads to a fast response with low overshoot. Experimental results are presented which include an external disturbance from a small fan. Motion is constrained as the UAV is attached to a stand for safety. Work in [57] combines backstepping and sliding mode techniques. The dynamics is divided into four subsystems: lateral position, altitude, yaw, and actuator dynamics. Similar to [41], to avoid complex expressions for the

control law, low-pass filtered derivatives are employed to estimate derivatives of the virtual control. Asymptotic stability of the closed-loop is derived without accounting for subsystem interaction. A sign function used in the sliding surface equation is approximated by a continuous function to address chattering. However, the effect of this approximation is not investigated in the stability proof. The method is validated on a stand which limits motion to vertical and yaw Degrees of Freedom DoF.

As mentioned above, another way to improve robustness is to add integral action to the feedback. As is well-known for PID control, integral action improves robustness to disturbances and reduces steady-state tracking error [63]. Work in [64] proposes an integral-augmented backstepping control. The iterative backstepping design begins with the integral of position error. Asymptotic rejection of constant disturbances is achieved. However, the disturbance is introduced in the error dynamics in a non-physical way. Further, the norm of the disturbance is assumed bounded by a linear function of the error. This assumption is impractical as it is not valid for constant disturbance. An inner-outer loop control is proposed in [65] where the inner loop reference is generated by an integral augmented backstepping outer loop. However, disturbances or model uncertainty are not considered in the design. Simulations demonstrate that integral-augmentation improves closed-loop robustness. An inner-outer loop control is proposed in [66] where integral-augmented backstepping is combined with sliding mode control. The sliding mode controller is used to achieve asymptotic stability for constant disturbances. Coupling between the inner and outer loops is neglected.

Work [8] proposes an inner-outer loop controller which is adaptive to external force and torque disturbances and the position of centre of mass. The outer loop is divided into vertical and horizontal subsystems. The inner loop controls the rotational dynamics. A Lyapunov stability analysis is provided for the outer loop and asymptotic convergence is proven. The effect of tracking error in the inner loop, which uses gravity compensated PD control, is not considered.

Disturbance Observers: A number of papers adopt a disturbance observer-based approach to achieve robust control [9, 41, 58, 59, 67]. A disturbance observer is an estimator which is designed to estimate the external and internal disturbances using a model of the system and measurements [68]. A disturbance observer provides an estimate of a disturbance which can be used in the controller for disturbance rejection or attenuation. This approach is adopted in this thesis in Chapter 3.

Work in [58] considers a disturbance observer using results from [69]. The observer is combined with a backstepping control and applied to a traditional helicopter. The disturbance observer is designed independently of the feedback control, and the disturbance estimate error is not compensated in the backstepping control. Further, low-pass filters are used to estimate angular acceleration in order to reduce controller complexity. An ultimate boundedness result is obtained due to the estimation error in angular acceleration.

Work in [67] considers a disturbance observer-based method to solve an attitude stabilization problem. Time-varying disturbance torques are considered and a high-gain extended state observer is designed for angular velocity and disturbance. Ultimate boundedness of the estimation error

is proven. A sliding mode feedback control is proposed to prove ultimate boundedness of the closed-loop. Work in [59] presents a disturbance observer-based control for the position control of a hexarotor. This work assumes that linear acceleration is measured, the dynamics are linearized about hover, and a force disturbance enters the translational dynamics. The inverse of the model's transfer function is used to estimate the external force disturbance. This estimate is used in an inner-outer loop control structure. A model inversion controller is used to specify desired roll, pitch, and thrust from the desired acceleration. Stability analysis is performed using the Small Gain Theorem. Simulation and experimental results are presented including a comparison with backstepping without a disturbance observer. Work [41] takes an inner-outer loop structure and combines an extended state observer, backstepping, and dynamic surface control to account for parametric uncertainty and disturbances. The extended state observer estimates disturbances and unmeasured linear and angular velocity states. Dynamic surface control [70] eliminates the complex expressions normally found in backstepping laws. It should be noted that the method [41] assumes simplified rotational dynamics to derive their method. The closed-loop is proven ultimately bounded in the presence of time-varying disturbances. The method is validated in simulation.

The work [9] uses a disturbance observer-based backstepping control with inner-outer loop structure. A simplified linear dynamics is considered for the rotational dynamics and the translational system is approximated about hover. A range of uncertainty is considered: external disturbance, modelling error, and input delay. The outer loop control is based on a simplified dynamics and ultimate boundedness is proven for the outer loop. Simulation and experimental results are presented using a fan as a source of disturbance.

Work in [71] presents an observer-based anti-disturbance control scheme for a UAV subject to wind-gust and suspended payload. A disturbance observer-based control and an extended state observer-based control are employed for the outer loop. A second extended state observer-based control is applied to the rotational dynamics. Experimental results are presented. Work in [72] develops a disturbance observer-based backstepping control using dynamic surfaces to reduce complexity. A minimum performance is ensured by the employing the prescribed performance method.

Rotor Drag Modelling for Control: Rotor drag is a force arising from blade flapping or induced drag [73]. Research has shown that directly compensating for this force improves motion control performance [15, 16]. Rotor drag compensation is particularly important for high-speed trajectories where these aerodynamics effects are significant. The above-mentioned methods either ignore rotor drag or lump these forces into a disturbance. Existing work on rotor drag compensation includes [16] which presents first-order aerodynamic modelling. An inner-outer loop control was proposed with a saturated PD outer loop and quaternion-based inner loop. Integral action is added to the outer loop for improved robustness. Asymptotic stability of the closed-loop is proven. External force and torque disturbances are not considered and performance is validated with Matlab simulation. Paper [15] shows quadrotor position and heading are flat outputs for a dynamics which includes a first order rotor drag model. Flatness is used to calculate a feed-forward control used

in an inner-outer loop controller which compensates rotor drag. An optimization-based approach estimates the rotor drag coefficients. The improved performance of the method is shown experimentally. External disturbances are neglected and wind velocity is assumed zero. The outer-loop is run on the ground using a laptop which sends thrust and angular velocity reference to the onboard flight controller. Work in [73] develops a 3-level inner-outer loop control with outer-level translational DoF control, mid-level rotational DoF control, and inner-level thrust control. Modeling of rotor drag is investigated in detail and used to estimate the varying aerodynamic conditions around the rotor. This estimated model and measured electrical power are used for low-level thrust control. Similar to [16] the rotor drag is decomposed into two components: one in the direction of thrust, and the other independent of attitude which can be compensated in the high-level control. Perfect attitude tracking is assumed in the position/velocity controller, i.e., subsystem coupling is ignored.

In [55] the effect of rotor drag and blade flapping are included in the dynamic model and controller design. Integral augmentation is combined with nonlinear control in an inner-outer loop structure. The outer loop controller is designed to translational DoF. To achieve robustness against model uncertainty and external disturbances, saturated integrators with fast desaturation are employed. The approach is experimentally validated for various reference trajectories and in the presence of wind forces. However, only asymptotic (as opposed to exponential) stability is proven for the outer loop design and inner-outer loop coupling is ignored. This thesis integrates rotor drag modelling into the proposed motion control in order to compensate a component of this force.

1.4 Visual-inertial Odometry: Literature Review

In visual odometry (VO), data is normally processed in two approaches: feature-based and direct methods. In feature-based approaches, the features, which are distinct pieces of information in an image, are extracted and tracked by a feature extraction and tracking system. Therefore, the map is represented by map features (e.g., 3D points) in the environment and the camera pose is estimated by tracking the detected features in the reconstructed map and minimizing the reprojection error. In direct approaches the whole image is processed without feature detection and tracking. The motion estimation is normally based on minimizing the photometric error between two consecutive frames. This approach is normally used to cope with environments with few features. Since there is no need for feature extraction, the direct approaches normally have less computational cost. However, it is known that the extra computational cost makes feature-based methods more robust to changes in illumination [74]. As robustness is an important factor for UAV applications, Chapter 4 and the literature survey below focuses on feature-based methods using a single monocular camera.

VIO involves the fusion of inertial and visual measurements. This can be done in a loosely-coupled [75, 76] or tightly-coupled [77, 78] framework. Loosely-coupled approaches are usually based on fusing the estimates from a VO system with IMU data using a Bayesian filter. Although this approach reduces computational complexity, it ignores coupling between the two subsystems and this reduces estimation accuracy. On the other hand, tightly-coupled algorithms fuse visual

and IMU data by respecting their inherent coupling. The approach incorporates metric scale information from the IMU data into the depth estimation. Hence, scale estimation is not required. A number of researchers have developed VIO for UAV applications [24, 26, 76, 79, 80].

Another property for categorizing VIO is whether it is filter or optimization-based. In filter-based methods, the pose and map are represented by probability density functions (PDF) that are updated using a recursive Bayes filter. This approach requires motion and measurement models which depend on the particular vehicle and sensors [81]. A popular Bayes filter used for VIO is the Extended Kalman Filter (EKF) which relies on a Gaussian White Noise (GWN) model. For example, in [82] the six DoF camera motion and 3D position of the feature points are represented as a state vector. A covariance matrix is propagated to account for uncertainty. New map states are added as new feature points are detected. IMU measurements are used in a motion model to predict camera motion and pose uncertainty. A pinhole camera model in the measurement update step. EKF-based VIO suffer from computational cost that grows in proportion to the number of map points. As well, estimates can become inconsistent due to error in linearization and the GWN assumption [83, 84]. To reduce computational complexity, in [77] a *structureless* approach is proposed where landmark positions are marginalized out of the state vector and only a select history of poses are estimated. However, this method has the disadvantage of not incorporating the correlations between pose and map features.

In optimization-based approaches, pose and map estimation is normally performed by minimizing a cost function constructed from the reprojection error of the mapped points onto the image plane and the integrated IMU measurements between camera frames. Uncertainty in the measurement models assumes GWN and covariance is normally used in the optimization weights. Optimization-based approaches can be divided into full smoothers or batch optimization algorithms, which perform the optimization over the complete history of poses and map points, and fixed-lag smoothers or sliding window optimization algorithms, which consider a sliding window of the latest poses.

Smoothing algorithms are known to be more accurate as they relinearize past measurement models multiple times until a global minimum is reached. Normally in filtering methods linearization happens only once. However, it has been shown that an extension of EKF called iterated EKF is equivalent to the Gauss–Newton algorithm commonly used in optimization-based methods [85]. Relative to full-smoothing algorithms, fixed-lag smoothers are less accurate as they marginalize the states outside the estimation window. Due to this marginalization, fixed-lag smoothers share some of the issues of filtering approaches (inconsistency, and drift due to the buildup of linearization errors). Although full-smoothing algorithms provide higher accuracy as they optimize over the complete history and keep all the correlation between the states optimized, their real-time implementation is a challenge as the number of optimization states increases as the trajectory grows. Therefore, fixed-lag smoothing which compromises between accuracy and computational demand for real-time performance, is normally employed for real-time implementation. Full and fixed-lag smoothing is combined in [86] where full-smoothing runs in a parallel in a low frequency thread to

update the state of the fixed-lag smoothing. This work increases computational efficiency by adding a new state for the pose only at certain frames called keyframes. A keyframe is selected when there exists a large difference between the current frame and one of the previously chosen keyframes. This frame selection reduces computation as it discards frames with less information. According to [87], increasing the number of map points leads to a significant increase in the accuracy of the estimation, while increasing the number of keyframes results improves algorithm robustness.

Work in [77] develops a Multi-state Constraint Kalman Filter (MSCKF) VIO method which is based on a tightly-coupled modified VSLAM system. The state consists of the current camera pose, linear velocity, sensor bias and a history of past camera poses. A measurement model is derived by expressing the geometric constraint arising from viewing a feature from multiple poses. Measurements of each feature are used to define a constraint between all camera poses where the feature is observed. This approach results in computational complexity which is linear in the number of features and therefore real-time operation is possible. The filter propagation is performed by discretization of the motion model which includes the IMU measurement noise models. Least square optimization is employed for estimating the 3D position of the features. Then, the 3D estimates are used to derive a measurement residual based on the poses in which the features were observed. The residual is linearized so that it can be used in the EKF measurement update. The method is validated by experiment using a camera/IMU setup mounted on a car. The final position error is 0.31% of the travelled distance.

In [79] a VISLAM method is proposed using a monocular camera and IMU. The method is based on a modified version of vision only Parallel Tracking and Mapping (PTAM) [86]. Some parameters of PTAM are tuned so that the number of keyframes and map points are reduced. This reduces the amount of computation required. The output of the modified PTAM system is fused with IMU data through an EKF in a loosely-coupled approach to estimate the unknown scale. The mapping capability of the approach enables path planning and obstacle avoidance for a UAV and motion drift cancellation when the UAV is returned to a previously-visited location. The system is implemented in real-time and demonstrated in UAV flight tests.

Paper [80] proposes a tightly-coupled EKF algorithm for data fusion of a camera and IMU onboard a UAV by making use of the platform specific dynamics. The advantages of tightly-coupled approach over the commonly used loosely-coupled ones are robustness improvement of the feature tracking and also computational cost reduction by having a smaller search region for feature extraction. The inverse depth parameterization is used for undelayed feature initialization, reducing the complexity of the measurement model. Results of a comparison with a loosely coupled method show a significant improvement in the estimation accuracy.

The problem of aggressive maneuvering for a quadrotor UAV using only a onboard camera and IMU is investigated in [24]. The objective is to control the UAV trajectory through a small window opening using only onboard sensors. The navigation task consists of state estimation, trajectory planning and trajectory estimation. The state estimation has two modules. The first one is an EKF-based VIO system. The prediction step of the EKF is based on IMU integration and measurement

update is based on the projection of 3D feature points onto the image plane using an inverse depth parametrization. The relative position of the UAV to the window and the structure of the window are estimated. However, the high computational cost of mapping in the VIO leads to low frequency pose estimates which cannot be used for fast and aggressive maneuvers. To compensate for this, an Unscented Kalman Filter (UKF) is employed in a separate module for high rate (500 Hz) estimation of pose. The UKF allows for a larger range of roll and pitch. The UKF uses the VIO output as an initial value and begins to estimate pose by integrating IMU measurements until a new VIO measurement is provided. The trajectory planning generates a feasible path for achieving the aggressive maneuver and a backstepping controller with an inner-outer loop structure performs output tracking.

Paper [26] presents VINS-Mono which is a tightly-coupled VI-SLAM algorithm. A sliding window of preintegrated IMU measurements and feature observations are fused using nonlinear optimization. The camera-to-IMU extrinsic calibration and IMU bias estimation are performed in real-time. Due to the relatively high frequency of the IMU, its measurements are used for pre-estimation of pose between two consecutive frames using a low-cost preintegration method. A loosely-coupled approach is adopted for the initialization phase. The core of the system uses tightly-coupled VIO which is based on bundle adjustment over a sliding window of IMU state and feature observations. A relocalization module for loop detection is proposed to compensate the accumulated drift. The relocalization results are then used in a global optimization module based on a pose graph to ensure consistency of the map and previous poses. Due to the high accuracy of roll and pitch estimation, drift only occurs in 4 DoF and therefore global optimization is performed for these variables. Experimental results are provided for public datasets and closed-loop UAV motion control.

Reference [76] investigates the autonomous navigation problem of multiple UAVs in an unknown and GPS-denied environment. Two navigation modules are developed: local and global. The local navigation module runs in real-time and onboard each UAV. This module contains a SLAM algorithm that is fused with IMU measurements using an EKF, estimating the local pose and the map. It also includes a controller that performs output tracking of the planned trajectories provided by the global navigation module. The global navigation module that runs on a ground station uses data from all UAVs to construct a global map and estimates the pose of each UAV within this map. The path intersections and loop closures are used to increase accuracy of the map and pose estimation. A pose graph-based optimizer for pre-alignment and a bundle adjustment is used in the global navigation module. The updated global pose and the desired trajectories are sent to the UAVs for their local navigation modules. Experimental results for three UAVs are presented.

Most of the existing approaches for VIO do not consider the dynamics of the multirotor. When it is considered, the effect of rotor drag is neglected in the UAV dynamics. VIO which includes some form of force estimation is referred to as Visual-Inertial-Actuator Odometry (VIAO). Recently, rotor drag has been shown to be important in state estimation and control [73]. The proper use of this modelling in a tightly-coupled estimation algorithm can improve its accuracy. An-

other important component to UAV dynamics is the external force disturbance. Estimating this disturbance is important for improving motion control. This can be clearly seen in the many references on force estimation given in Section 1.3. Further, force estimation is especially important in applications where the vehicle is required to sense interaction forces, such as contact inspection, grasping, manipulation, and load transport. It is natural that a tightly-coupled joint estimation of the disturbance with other system states will improve estimation accuracy. Such an approach is proposed in this thesis. Joint estimation requires additional measurements (i.e., actuation or force sensor data) and a model of the system dynamics. To our knowledge the only work to-date that follows this approach is [27]. However, in this work the effect of rotor drag is not considered in the UAV dynamics. Also, the work assumes the external force is zero mean Gaussian noise. In practice, most disturbances encountered are not zero mean (e.g., wind-gust, ground effect, or forces due to load transport). In the case where the external force is not zero mean, the offset is estimated in the accelerometer bias which is not ideal for applications which need an accurate external force estimation.

1.5 Overview of Thesis

This thesis is divided into two main parts: robust motion control design for multirotor UAVs subject to external disturbances and multirotor state and map estimation using visual, inertial and actuator data. Before presenting the main contents we present the multirotor dynamical modelling equations with and without rotor drag consideration in Chapter 2. In the same chapter we present our experimental quadrotor platform and discuss the hardware, software and controller implementation. Also the SITL simulation framework is presented.

In Chapter 3 we present different motion control methods for an output tracking problem where the outputs are position and yaw. It is assumed that the multirotor is subject to external force and torque disturbances. Backstepping is employed as the main control design method and estimated wrench from a disturbance observers are used in the backstepping to achieve disturbance rejection. We present the controller-observer design in a coupled and decoupled structure. Exponential convergence is proven in all the designs with the assumption of constant disturbances. In the coupled design, we add an extra term to the observer dynamics in order to improve the theoretical results. However, the onboard implementation is found to be sensitive for this structure. As well, the complexity of the control law expressions make the method difficult to implement. Hence, we only present Matlab simulation results. To address the complexity of this design, we employ a decoupled observer-controller structure. This approach is implemented and flight tested on the PX4 autopilot and simulated in Software in the Loop (SITL). Integral augmentation and rotor drag modelling is included in the decoupled design. In Chapter 4, we investigate the problem of multirotor state, external force and map estimation using visual, inertial and actuator data. We start the chapter by presenting the different coordinate frames and the fundamentals of VIO. Then we present the problem formulation into a tightly-coupled optimization problem that consists of visual, inertial

and dynamic residuals. Afterward we present the derivation of these residuals and the propagation formulas. The effect of rotor drag is incorporated into the vehicle's dynamics. A nonlinear force disturbance observer is reformulated as a residual and incorporated into the dynamic residual. The proposed method is implemented on top of VIMO [27] and tested on benchmark datasets. We show that the proposed method has improved the estimation accuracy. Finally, in Chapter 5 we present our conclusions and future work.

1.6 Contributions

The contribution of this thesis are summarized in this section.

1.6.1 Disturbance Observer-based Control

- Coupled disturbance observer-based backstepping control design for multirotor UAV trajectory tracking [88]. A backstepping controller is designed coupled to two disturbance observers for external force and torque by considering a general form for the observer dynamics. The concept of tuning function is employed to avoid overparameterization. The design leads to a rigorous exponential stability result for the case of constant disturbances. Ultimate boundedness is achieved for time-varying disturbances.

- Decoupled disturbance observer-based backstepping control design for multirotor UAV trajectory tracking [89]. Due to the particular decoupled structure, the disturbance observer error dynamics is LTI, globally exponentially stable, and depends only on disturbance estimate error. The disturbance estimate error feeds the tracking error dynamics in a cascade interconnection. This cascade structure makes the stability analysis, onboard implementation, and controller tuning straightforward. Using cascade stability analysis, *exponential* stability is proven in the presence of constant disturbances.

- Decoupled disturbance observer-based integral backstepping control design with rotor drag compensation [90]. Integral augmentation and the rotor drag compensation is added to the work [89] to improve the robust performance of the motion control. As in [89] the observer is decoupled from the controller. Following [16], rotor drag is compensated by decomposing it into two components: one is independent of UAV attitude, and the other is in the direction of thrust and can be compensated by appropriate choice for desired thrust during backstepping. *Exponential* stability is proven for the case of constant disturbances. We demonstrate the improved tracking performance of the disturbance observer-based integral backstepping control in the presence of wind force disturbance. Also, the benefits of including rotor drag compensation are shown by simulating the proposed method with and without its compensation.

- The full dynamics of the system in the control design [88–90]. This avoids having to make common simplifying assumptions typical of designs with inner outer loop structure (e.g., linear

approximation of the rotational dynamics) during the closed-loop stability analysis of the entire rotational and translational system dynamics [91].

- SITL Validation on PX4 autopilot [89, 90]. The PX4 firmware is a widely-used and open source project intended for research and commercial use. It runs on a number of hardware platforms including the Pixhawk family. SITL simulation is an important step in the development of flight controllers since it allows for safe and efficient debugging and tuning and it captures the many real-world constraints of onboard implementation (e.g., multi-rate measurements, control, and state estimation, bounded signals, and constrained hardware resources)..

- Experimental testing [89]. The proposed method is also implemented experimentally using the Applied Nonlinear Control Lab (ANCL) quadrotor platform in the presence of a wind disturbance. The proposed method is compared with a traditional backstepping controller and the built-in PX4 motion controller. The results demonstrate improved performance for the decoupled disturbance observer-based backstepping control. It is important to note that our implementation does not require simplifying approximations (e.g., small angle assumptions, inner-outer loop structure) that are common in work with experimental validation [8, 9, 41].

- The PX4 code is available online at [92, 93] and should be useful to researchers in nonlinear multirotor control. By basing our work on the open source and commonly used PX4 platform, our results are accessible to the research community.

1.6.2 Visual-Inertial-Actuator Odometry (VIAO)

- VIA state estimation for multirotor UAV with rotor drag consideration. The rotor drag is incorporated into a tightly-coupled state estimation and also a more realistic assumption of slowly time-varying force disturbance is used. These modifications and assumptions have been included by revisiting the vehicles dynamics to include the rotor drag effect and also by deriving a residual term from a disturbance observer proposed for a UAV [89] designed for constant external forces and incorporating the residual into the optimization. We implement our proposed method on top of VIMO [27] which itself is based on VINS-Mono [26], an open source VIO system implemented on Robot Operating System (ROS). We have shown that the proposed method can result in an increased accuracy.

Chapter 2

Quadrotor Modelling and Experimental Platform

This chapter presents the mathematical models and defines the notation that will be used throughout this thesis. Also the experimental platform that is used for performing the flight tests is introduced. In Section 2.1 we introduce the dynamic equations of a quadrotor with and without consideration of rotor drag force. In Section 2.2 the ANCL platform is introduced and details of hardware and software specifications are discussed.

2.1 Modeling

We consider a traditional quadrotor UAV as shown in Figure 2.1. We choose a commonly used nonlinear rigid body model which ignores motor/propeller dynamics, and rotor gyroscopic effects. Further details on modelling where such effects are considered can be found in [53, 73]. We require two reference frames: a fixed inertial navigation frame $\mathcal{N}$ with orthonormal basis $\{n_1, n_2, n_3\}$ and a body frame $\mathcal{B}$ whose origin is at the vehicle's center of mass (CoM) and with orthonormal basis $\{b_1, b_2, b_3\}$. We define b_1 to point in the forward direction of vehicle, b_2 pointing right, and b_3 pointing down. The set of vectors $\{n_1, n_2, n_3\}$ are considered to be orientated north, east, and down, respectively. The configuration of the quadrotor belongs to the special Euclidean group SE(3), and includes the position $p \in \mathbb{R}^3$ of the origin of $\mathcal{B}$ relative to $\mathcal{N}$, and the orientation $R \in \mathrm{SO}(3)$ of $\mathcal{B}$ with respect to $\mathcal{N}$. The distance of each propeller to the origin of $\mathcal{B}$ is ℓ (arm length) and Θ_i is the angle of ith arm with respect to b_1. We assume each propeller generates thrust in the $-b_3$ direction with magnitude of $u_i = k_u \Omega_i^2$ for the ith propeller where Ω_i is the rotational speed of ith propeller and k_u is the thrust coefficient. We define the total thrust due to all propellers by the scalar input $u \coloneqq \sum_{i=1}^{4} u_i > 0$, i.e., the thrust vector is $-ub_3$. The generated torque for each propeller is around the b_3 and is $\tau^i = k_\tau \Omega_i^2$ for $i = 1, 2$, which are rotating clockwise and $\tau^i = -k_\tau \Omega_i^2$ for $i = 2, 3$, which are rotating counterclockwise. The total input torque acting on the vehicle is denoted by $\tau \coloneqq [\tau_1, \tau_2, \tau_3]^\top \in \mathbb{R}^3$ which is expressed in $\mathcal{B}$ and can be calculated

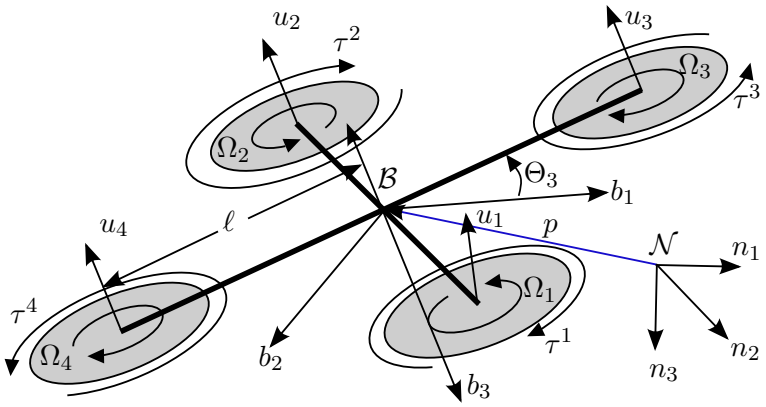


Figure 2.1: Diagram of a quadrotor showing navigation frame $\mathcal{N}$ and body frame $\mathcal{B}$.

by projecting the generated thrusts and torques of each propeller into the direction of b_1 and b_2. For a quadrotor in "cross" configuration where $\Theta_1 = \Theta$, $\Theta_2 = \pi + \Theta$, $\Theta_3 = -\Theta$, $\Theta_4 = \pi - \Theta$, the relation between the u and τ and the rotational speed of each propeller can be written as

$$\begin{bmatrix} u \\ \tau \end{bmatrix} = \begin{bmatrix} k_u & k_u & k_u & k_u \\ -k_u\ell\sin\Theta & k_u\ell\sin\Theta & k_u\ell\sin\Theta & -k_u\ell\sin\Theta \\ k_u\ell\cos\Theta & -k_u\ell\cos\Theta & k_u\ell\cos\Theta & -k_u\ell\cos\Theta \\ k_\tau & k_\tau & -k_\tau & -k_\tau \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{2.1}$$

Since the physical input to the UAV are PWM signals to the Electronic Speed Controller (ESC) which are denoted $W_i, i = 1, 2, 3, 4$, we can write the equation (2.1) as bellow

$$\begin{bmatrix} u \\ \tau \end{bmatrix} = \begin{bmatrix} K_u & K_u & K_u & K_u \\ -K_u\ell\sin\Theta & K_u\ell\sin\Theta & K_u\ell\sin\Theta & -K_u\ell\sin\Theta \\ K_u\ell\cos\Theta & -K_u\ell\cos\Theta & K_u\ell\cos\Theta & -K_u\ell\cos\Theta \\ K_\tau & K_\tau & -K_\tau & -K_\tau \end{bmatrix} \begin{bmatrix} \tilde{W}_1^2 \\ \tilde{W}_2^2 \\ \tilde{W}_3^2 \\ \tilde{W}_4^2 \end{bmatrix} \tag{2.2}$$

where $\tilde{W}_i$ is a normalized PWM signal defined by $\tilde{W}_i = (W_i - W_{\min})/(W_{\max} - W_{\min})$ where $W_{\min}$ and $W_{\max}$ are the minimum and maximum values of W_i, respectively. Therefore, $\tilde{W}_i \in [0, 1]$. Also K_u and K_τ are accordingly normalized thrust and torque coefficients. Thrust model (2.2) clearly shows that u and τ are bounded. To ease presentation of the control design we take torque τ and thrust u as system inputs.

2.1.1 Traditional Modelling

The traditional and widely used dynamics equations that neglect the aerodynamic effects such as rotor drag are given by

$$\dot{p} = v \tag{2.3a}$$

$$m\dot{v} = mgn_3 - uRn_3 + d_f \tag{2.3b}$$

$$\dot{R} = RS(\omega) \tag{2.3c}$$

$$J\dot{\omega} = -\omega \times J\omega + \tau + d_\tau \tag{2.3d}$$

where $v \in \mathbb{R}^3$ is linear velocity expressed in $\mathcal{N}$, $\omega \in \mathbb{R}^3$ is angular velocity expressed in $\mathcal{B}$, m is mass, J is inertia, g is the gravity constant, and $n_3 = [0, 0, 1]^\top$. The disturbance force $d_f \in \mathbb{R}^3$ and disturbance torque $d_\tau \in \mathbb{R}^3$ are unknown and used to model external forces such as wind gusts or other model uncertainty. The skew operator $S(\cdot) : \mathbb{R}^3 \to so(3)$ is defined as

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \quad \text{where } x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2.4}$$

2.1.2 Rotor Drag Modelling

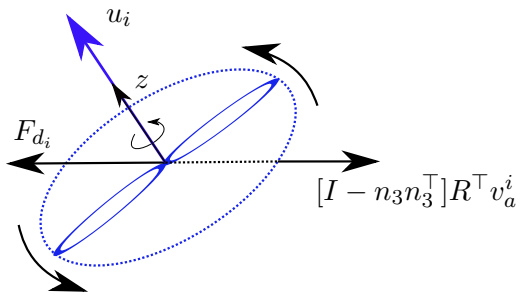


Figure 2.2: Drag force on a single rotor.

With the assumption of rigid blades, (which is reasonable assumption for small multirotors) one aerodynamic effect that affects the multirotor dynamics is the induced drag. The induced drag is generated when the rotor is moving in a specific direction and the advancing blade is experiencing a relatively higher air velocity than the retreating blade as shown in Figure 2.2. Therefore the thrust generated by each blade is different in magnitude and as a result a nonzero net force is created which lies in the horizontal plane of the rotor. This force is called H-Force in aerodynamic literature and can be expressed by [94] (for the ith rotor)

$$F_{d_i} = -\sqrt{u_i}\, d_1 [I - n_3 n_3^\top] v_a^i \tag{2.5}$$

where d_1 is a positive constant, v_a^i is the translational air velocity that the ith rotor is experiencing expressed in $\mathcal{B}$ frame which can be written as $v_a^i = R^\top v_a + S(\omega)\ell_{r_i}$, where $v_a = v - w$, is the air

velocity at the center of the body frame with $w \in \mathbb{R}^3$ as constant wind speed expressed in $\mathcal{N}$ and ℓ_{r_i} is the position of the ith rotor in the body frame. The summation of the above drag force in body frame can be estimated by the following equation considering the thrust for each rotor is equal to $u_i = \frac{mg}{4}$ [16] (near hover condition)

$$F_d = \sum_{i=1}^{4} F_{d_i} = -DR^\top v_a \tag{2.6}$$

with D as the drag coefficient matrix

$$D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{11} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.7}$$

and $d_{11} = 2d_1\sqrt{mg}$. Now, the dynamics of the quadrotor with consideration of the effect of rotor drag can be written by

$$\dot{p} = v \tag{2.8a}$$
$$m\dot{v} = mgn_3 - uRn_3 - RDR^\top v_a + d_f \tag{2.8b}$$
$$\dot{R} = RS(\omega) \tag{2.8c}$$
$$J\dot{\omega} = -\omega \times J\omega + \tau + d_\tau \tag{2.8d}$$

As we mentioned before, the drag force $-RDR^\top v_a$ in (2.8b) is normally neglected in nonlinear controller design which accounts for the entire dynamics (i.e., the coupled rotational and translational dynamics). The compensation of drag force in the proposed method assumes v_a is measured and drag coefficient D is known. Since drag force $-RDR^\top v_a$ has a dependence on R, the backstepping procedure requires its decomposition into two components: one is independent of vehicle attitude and the other is in the direction of thrust:

$$-RDR^\top v_a = -d_{11}v_a + d_{11}(v_a^\top)Rn_3Rn_3 \tag{2.9}$$

This decomposition (shown in Fig. 2.3) is also used in [16] where it is applied to an inner-outer loop design. Although the term $-d_{11}v_a$ can be readily compensated using the proposed backstepping design, the term $d_{11}v_a^\top Rn_3Rn_3$, which is in the direction of thrust, must be counteracted by an appropriate definition of desired thrust which will be explained below.

When deriving the control for heading or yaw below we rely on the ZYX Euler paramterization $\eta = [\phi, \theta, \psi]^\top$, where ϕ, θ and ψ are roll, pitch, and yaw, respectively. The rotation matrix R is

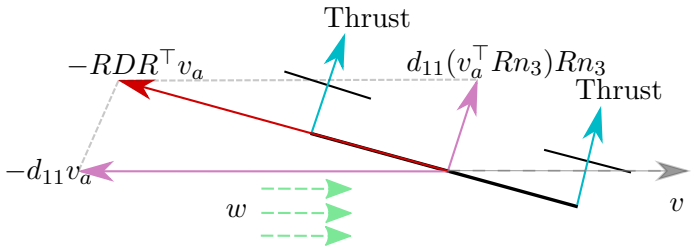


Figure 2.3: Drag force decomposition.

expressed in terms of η as

$$R(\eta) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

where $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$. The rotational kinematics (2.3c) expressed in η are $\dot{\eta} = W(\eta)\omega$ with

$$W(\eta) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}$$

and where $t_\theta = \tan\theta$.

Unit quaternion is a parameterization that can be used for representation of rotations without singularities. However, each rotation does not have a unique unit quaternion since there is double coverage. A unit quaternion is normally represented by $q = [q_0, q_v^\top]^\top$, where $q_0 \in \mathbb{R}$ is the scalar part and $q_v = [q_1, q_2, q_3]^\top \in \mathbb{R}^3$ is the vector part. The unit quaternion set $\mathbb{Q}_u$ is defined by

$$\mathbb{Q}_u = \{q = [q_0, q_v^\top]^\top | q_0 \in \mathbb{R}, q_v \in \mathbb{R}^3, \|q\| = 1\} \tag{2.10}$$

The rotation matrix can be expressed using the unit quaternion

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.11}$$

The quaternion multiplication denoted as $\otimes$ between two unit quaternions q and $\hat{q}$ is defined by

$$q \otimes \hat{q} = \begin{bmatrix} q_0 \hat{q}_0 - q_v^\top \hat{q}_v \\ q_0 \hat{q}_v + \hat{q}_0 q_v + q_v \times \hat{q}_v \end{bmatrix} \tag{2.12}$$

where "$\times$" is the cross product. With the above definitions the rotation kinematics with quaternion is given by [95]

$$\dot{q} = \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} \tag{2.13}$$

which also can be written as

$$\dot{q} = \frac{1}{2}\Omega(\omega)q \tag{2.14}$$

where the operator $\Omega(\cdot)$ is defined on $\mathbb{R}^3$ by

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega^\top \\ \omega & -S(\omega) \end{bmatrix} \tag{2.15}$$

and the operator $S(\cdot)$ was defined in (2.4).

2.2 Experimental Platform

Figure 2.4: The ANCL-Q3 quadrotor vehicle.

Experimental flight test is a necessary step in validating the performance of the control laws. This testing demonstrates if the design can tolerate unmodelled effects such as measurement noise and input saturation under typical operating conditions. In this section we introduce the hardware and software components of ANCL quadrotor platform in details.

2.2.1 Hardware Platform

In this section we present the hardware components of ANCL quadrotor platform that consists of a quadrotor vehicle ANCL-Q3, a Vicon MCS, a QGroundControl ground station (QGC) and a Computer Vision System. The block diagram of the hardware platform is shown in Figure 2.5.

The Quadrotor Vehicle ANCL-Q3

The ANCL-Q3 quadrotor is shown in Figure 2.4 which consists of a 3D Robotics quadrotor do-it-yourself (DIY) frame in a cross configuration equipped with a Pixhawk 1 flight controller (shown in Figure 2.6) which has a 180 MHz ARM CPU, two 3D accelerometers, two 3D gyroscopes, a 3D magnetometer, and a pressure sensor [96]. ANCL-Q3 includes a 2.4 GHz LairdTech transceiver connected to the Pixhawk to receive data from the MCS which is equipped with a matching onboard LairdTech radio transmitter. The Pixhawk is also connected to a Spektrum satellite receiver which is paired to a Spektrum DX8 transmitter. The DX8 enables manual control and allows the operator

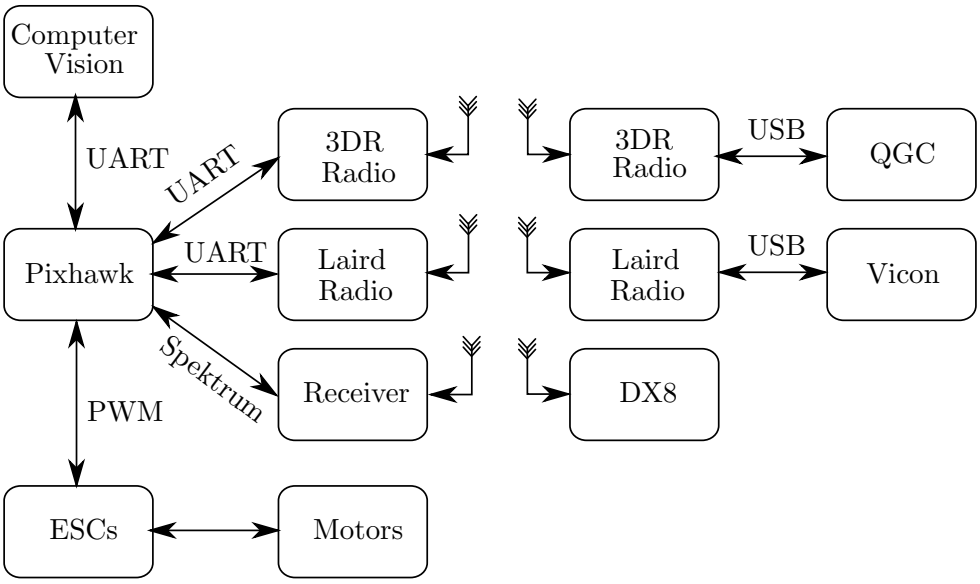


Figure 2.5: Block diagram of the ANCL-Q3.0 quadrotor platform.

Table 2.1: ANCL-Q3 model parameters.

J_1	J_2	J_3	m	ℓ	Θ	K_u	K_τ
$0.03\,\mathrm{kg\,m^2}$	$0.03\,\mathrm{kg\,m^2}$	$0.05\,\mathrm{kg\,m^2}$	$1.6\,\mathrm{kg}$	$0.165\,\mathrm{m}$	$0.58\,\mathrm{rad}$	$7.75\,\mathrm{N}$	$0.8\,\mathrm{N\,m}$

to switch between different control modes. A RN-XV WiFly Module is used to connect the PX4 to the local network via WiFi. ANCL-Q3 is powered by a 12 V, 3 cell, 5000 mA h lithium polymer battery (LiPo). This provides a flight time of about 10 minutes. The Pixhawk outputs a PWM signal to Afro 30 A ESCs which are connected to Turnigy 1100 KV Brushless Outrunner Motors. The APC propellers are 11". The Pixhawk logs data onto its SD card. The mass of the quadrotor is 1.6 kg including the battery. The quadrotor is designed such that it hovers at approximately 50% of its maximum thrust. This ensures enough thrust is available when aggressive maneuvers are required. Table 2.1 lists model parameters in (2.3) and (2.2).

The Motion Capture System (MCS)

The MCS uses a network of eight Vicon Bonita 3 (B3) cameras, a Windows PC, and a 2.4 GHz LairdTech transceiver paired to the one on-board ANCL-Q3. The B3 is a 0.3 MP near infrared (NIR) camera with maximum frame rate of 240 frames per second (fps) and a resolution of 640×480. The quadrotor can be detected by using four 38.1 mm reflective markers mounted on its frame. The position and yaw of the UAV are estimated by the MCS which introduces a constant 10 ms delay [97]. Linear velocity is estimated using a low pass filtered finite difference of position. The data rate of LairdTech transceiver is about 100 Hz and includes a constant 15 ms delay [97].

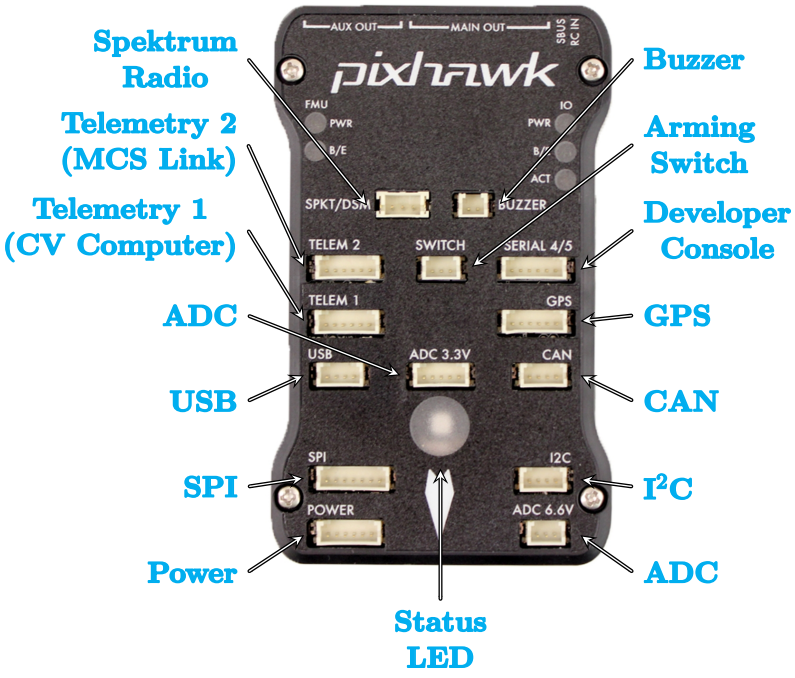


Figure 2.6: Image of Pixhawk 1 autopilot.

The Ground Station

The quadrotor communicates to a ground control station running QGC [98] that is an open-source ground control software which can be used for flashing the Pixhawk, parameter tuning, data visualization, status monitoring, and mission planning. QGC and PX4 are connected through WiFi. The MAVLink protocol [99] is used for communication between from the MCS to ANCL-Q3 and between ANCL-Q3 and QGC.

The Computer Vision System

The computer vision system consisting of an embedded computer system and a camera is a separate system from the PX4, but is still located on the quadrotor and communicating to PX4 through a serial connection using the MAVLink protocol. The embedded computer system that we use is Nvidia Jetson TX2 has a 64 bit ARM central processing unit (CPU), up to 8 GB memory and a graphics processing unit (GPU) with 256 Nvidia CUDA cores. The Jetson TX2 is mounted on an Auvidea carrier board that provides serial connection to for transferring vision data to PX4. Our main camera is the FLIR (previously Point Grey) Chameleon 3 USB 3 camera. It is attached to the quadrotor facing down about 5 cm below the quadrotor's CoM. This is a global shutter camera that can achieve frame rates of up to 149 Hz at a resolution of 1280×1024.

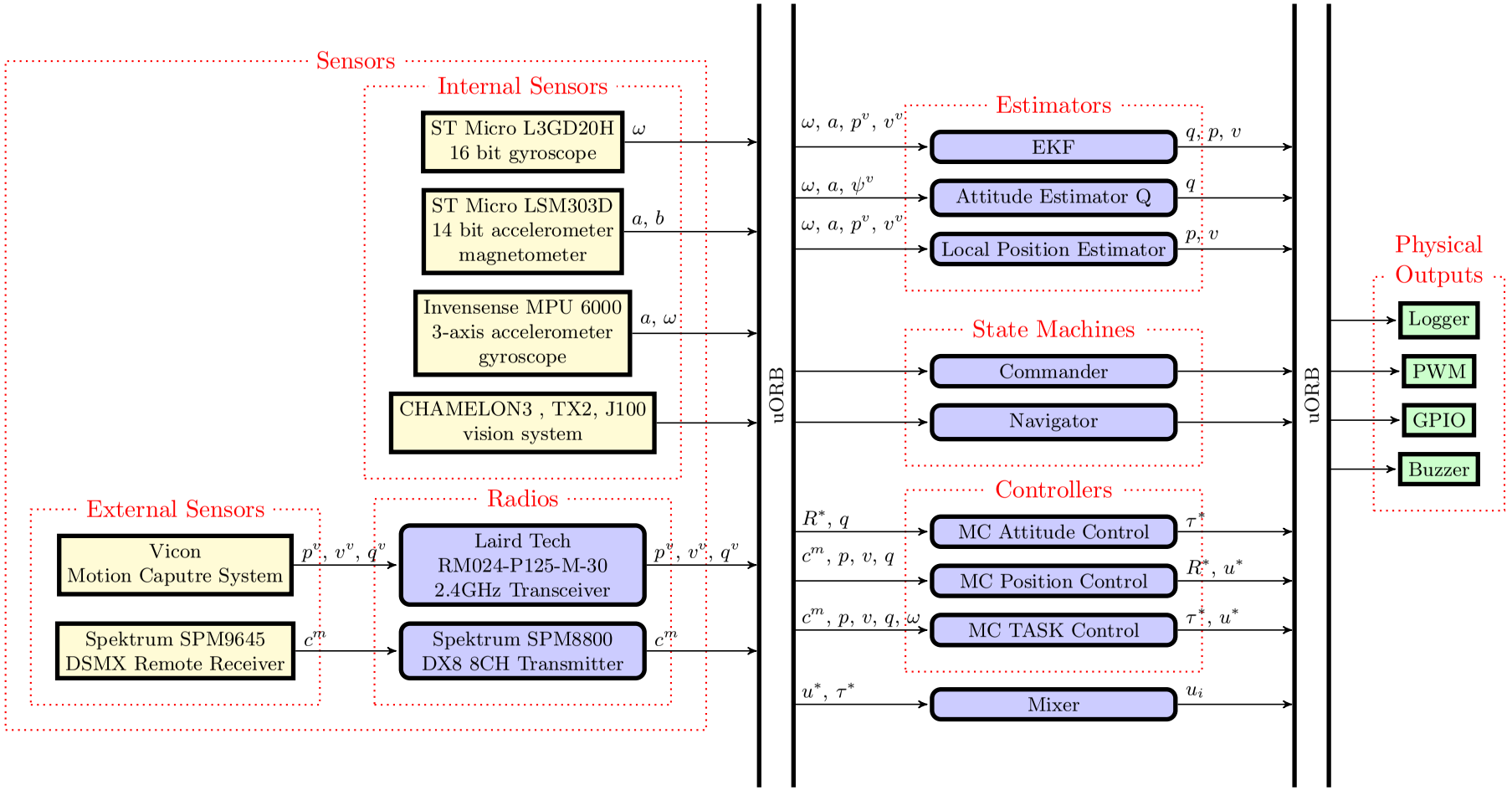


Figure 2.7: Data flow diagram of the ANCL quadrotor. uORB is the object broker for the inter-process communication based on the publish-subscribe design pattern. $(\cdot)^v$: vicon estimates, $(\cdot)^m$: control mode, $(\cdot)^*$: desired value.

2.2.2 Software Platform

We implement our proposed control on the PX4 autopilot firmware which is an active and mature open-source project intended for research and commercial applications [39]. PX4 runs on a number of hardware platforms and various robot types. It has a modular structure where individual autopilot functionality (e.g., motion control, state estimation) is separated into self-contained modules which each runs as a task on NuttX RTOS platforms (e.g., Pixhawk) or as a thread within the main PX4 process on POSIX platforms (e.g., SITL). This modular structure has a number of advantages. For example, adding a new control method to the system is relatively straightforward as only a small component of the code with a well-defined interface needs to be modified. The so-called PX4 middleware supports communication between the modules, sensor device drivers, and communication outside PX4. Module communication is implemented with the micro-object request broker (uORB) which provides a publish/subscribe bus. Using uORB, publishers send messages (e.g., a UAV control input) onto a bus instead of sending the messages directly to specific subscriber modules. Also, subscribers receive messages as soon as there are updates. PX4 provides driver modules for the hardware components (e.g., IMU, GPS and PWM outputs) and libraries for programming (e.g., matrix computation). The diagram of data flow in PX4 is shown in Figure 2.7.

To implement the DOB-BS control we created a new module using the v1.5.5 release of PX4. To simplify its implementation the module uses a so-called "Block" structure in the Controller Library contained in `px4/src/lib/controllib`. This library simplifies the process of subscription and publication. The library also provides functions for numerical integration which are used in the disturbance observer (3.3) and integration for thrust in (3.20). The controller module also uses the PX4 Matrix Library (`https://github.com/PX4/Matrix`) to improve the readability of the control law expressions. Control gains are implemented using PX4's parameter system. This simplifies controller tuning by allowing users to adjust gains from the PXH command line or from QGC. The attitude estimator `attitude_estimator_q` uses quaternions to keep track of the attitude and mixes the roll and pitch estimates from the accelerometer, the heading from the the MCS, and the gyroscope measurements. Position and linear velocity are obtained from `local_position_estimator` which uses the MCS and IMU data. The state machines modules `commander` and `navigator` manages the quadrotorbetween different control modes: Standby, Armed, Manual, ANCL Manual, ANCL Auto 1 (setpoint), ANCL Auto 2 (trajectory tracking). Further details on the modules used are in [97].

2.2.3 SITL Simulation Framework

Performing a simulation that recreates the actual flight conditions and matches on-board implementation has several advantages when developing motion control algorithms. SITL simulation is a method that allows you to run the actual autopilot code and investigate its behaviour without any hardware. In this method, the code is running on a PC and is interfaced with a simulator

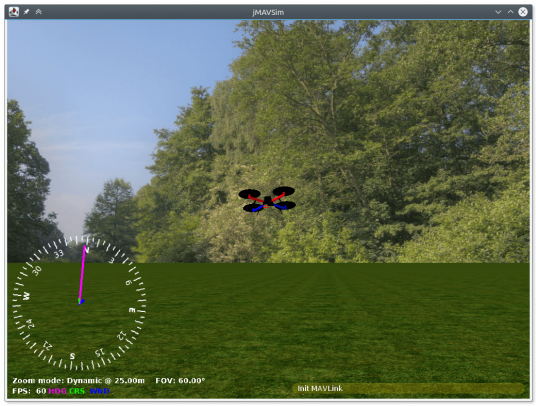


Figure 2.8: Visualizing the simulated motion control in jMAVSim.

that models the vehicle dynamics and the environment conditions (e.g. wind-gust). It recreates the significant constraints of autopilot hardware and software. For example, it ensures that the proposed design can be implemented in a typical autopilot framework where unmodelled effects (e.g., controller saturation, multiple sample rates, computational delays, etc.) influence performance or even prevent the controller from running (e.g., due to limited on-board processing power or memory). Therefore, SITL simulation can be an important step to the actual flight testing, by accelerating the debugging process and controller tuning.

We adopt the open-source PX4 autopilot SITL framework given the lab's investment in this platform to-date [39, 97, 100, 101] and since it is a mature standard in the research community. We choose the lightweight jMAVSim simulator which simplifies implementation while allowing multirotor model and environment customization. SITL and jMAVSim are built into the PX4 project. In general, SITL is not commonly used for validating control laws in the literature. However, it is clearly a safer and faster way of developing controllers which can be flown. A contribution of this paper is to provide simulation source code which can be validated or extended by the community [92, 93]. Our work is based on v1.5.5 of the PX4 firmware. The jMAVSim simulator receives PWM inputs from the autopilot code and outputs GPS position, IMU and barometer measurements related to the simulated UAV motion.

jMAVSim is developed in Java and employs `java3d` library for visualization of a 3D virtual flight environment. Communication between PX4 and jMAVSim is done by the MAVLink protocol using localhost User Datagram Protocol (UDP). A PXH shell, similar to NSH shell in the Pixhawk/NuttX platform, facilitates code debugging and controlling the modules. It can be used to set the parameters, start and stop the modules and listen to the PX4 messages. Figure 2.8 shows a screenshot of the 3D visualization of the simulation.

The UAV's model parameters can be hardcoded in the `Simulator` class. To match the real experiment presented in Section 3.3.4, we chose the model parameters as specified in Table 2.1 with a cross (or "x") vehicle configuration. The jMAVSim simulator accepts normalized torque and thrust inputs from PX4 and therefore includes thrust saturation. To model the actuator dynamics,

the normalized inputs are low-pass filtered to generate individual rotor thrust (see class `Rotor`). The time constant of the low-pass filter is set to 5 ms. To recreate the actual flight conditions, jMAVSim adds adjustable time delay to GPS measurements and zero mean Gaussian noise to the gyro, accelerometer and magnetometer measurements. The default numbers for these parameters were chosen. The thrust constant is scaled such that normalized thrust is 0.51 in hover.

$$u_n = 0.51 + \frac{u - mg}{60} \tag{2.16}$$

This value was obtained from the stock PX4 controller `mc_pos_control` when the quadrotor is in hover. The torque commands τ_1, τ_2 and τ_3 are also normalized by factors $n_{\tau_1} = 0.33$, $n_{\tau_2} = 0.33$ and $n_{\tau_3} = 1$, respectively.

Environmental disturbances can be easily generated in jMAVSim. We use this feature to create the force disturbance d_f. The model that generates the force disturbance is in `AbstractMultiCopter` class and the force disturbance it outputs is proportional to the airspeed i.e.,

$$d_f = c_f \cdot (w - v) \tag{2.17}$$

where c_f is a drag coefficient whose default value 0.03 Ns/m is used and $w \in \mathbb{R}^3$ is the wind velocity vector expressed in $\mathcal{N}$. The wind velocity is a random process whose variance and mean can be controlled by the user and is obtained from

$$\frac{dw}{dt} = -\frac{w}{\tau} + \left(\frac{W}{\tau} + \nu\right) \tag{2.18}$$

where we take $W = [0, -20, 0]^T$ m/s as the constant which determines the mean steady state value of w, $\tau = 2$ s and $\nu = [\nu_1, \nu_2, \nu_3]^T \in \mathbb{R}^3$ where its components are Gaussian white noise processes with zero mean and variance $(\tau\sigma_{wk})^2$, i.e., $\mathcal{N}(0, (\tau\sigma_{wk})^2)$, where $\sigma_{w1} = 6$ m/s, $\sigma_{w2} = 8$ m/s, $\sigma_{w3} = 0$. The parameters σ_{wk} and W can be set using the GUI or can be hard-coded in the `Simulator` class. To investigate the effectiveness of the proposed method to reject torque disturbances, a simple model for the torque disturbance is added to the jMAVSim simulation model using

$$d_\tau = -c_\tau \cdot \omega \tag{2.19}$$

where $c_\tau = 0.3$ Nms/rad is the drag coefficient. As well, rotor drag force was added to model using a drag coefficient $d_{11} = 0.3\,\mathrm{s}^{-1}$ which is a reasonable value for the quadrotor considered [15].

Chapter 3

Disturbance Observer-Based Backstepping Controller

This chapter focuses on solving an output tracking problem for the dynamics equation (2.3) (or (2.8) for modelling with rotor drag) where the output includes p and ψ. Given smooth bounded desired trajectories for position and yaw, denoted by $p_d \in \mathbb{R}^3$ and $\psi_d \in \mathbb{R}$, respectively, we derive a dynamic state feedback control for inputs u and τ to ensure exponential convergence of the tracking errors $p - p_d$ and $\psi - \psi_d$ in the presence of constant bounded disturbances d_f and d_τ. The full state measurement is assumed available. The proposed design adopts a backstepping approach inspired by [1] and the references within. As discussed before, the approach in [1] relies on high-dimensional parameter update laws to estimate d_f, whereas our proposed methods uses two 3-dimensional disturbance observers to reject disturbances d_f and d_τ. Since p and ψ can be independently controlled, the design is presented in separate sections: position and yaw control.

The proposed structure for the force and torque disturbance observers that are adopted throughout this research are presented in Section 3.1. In Section 3.2 we take a coupled structure for the controller-observer design with the concept of tuning functions which results in strong stability proof: exponentially stable error dynamics with provable bounds and ultimate boundedness in the case of time-varying disturbances. Matlab simulation results are presented as the control expressions are too complicated for onboard implementation.

To achieve a simpler and implementable control law we adopt a decoupled structure for controller-observer design in Section 3.3 where error dynamics of force and torque disturbance observers is decoupled from the rest of the closed-loop. This important property simplifies the exponential stability proof of the closed-loop. Similar to Section 3.2 still the backstepping approach is employed for the controller design, however, the stability proof is done by adopting the theory of cascaded system for the Linear Time Variant (LTV) systems as we show the error coordinates resulted from the backstepping approach are in the form of LTV system with force and torque disturbance estimation errors as the inputs. Results for SITL simulation and experimental testing are presented for the evaluation of the proposed controller. Section 3.4 extends the decoupled structure in Section

3.3 by adding the integral augmentation and rotor drag modelling in the controller design. The rotor drag is decomposed into two components: one independent of the vehicle attitude which can be readily compensated by an appropriate choice of virtual control in the backstepping procedure and the one which is in the direction of thrust and can be compensated by an appropriate choice for desired thrust during backstepping. SITL simulation results are presented for this controller showing an improved performance as a result of integral augmentation and the rotor drag compensation. The results also show the effectiveness of the controller-observer in the case of time-varying disturbance.

3.1 Disturbance Observers

In this section we present the structure that is used throughout this thesis for the force and torque disturbance observers. The idea for this structure is taken from [102] which originally was proposed for fixed-based manipulators. As we show throughout the rest of this chapter this structure allows us to achieve exponential stability which is known to be essential for controller robustness.

3.1.1 Force Disturbance Observer

To understand the basic structure of the disturbance observer used in the output tracking control, we consider the simplified problem of estimating d_f assuming the system state is measured. In addition, if we assume linear acceleration $\dot{v}$ is measured, a simple disturbance observer for the translational subsystem with traditional modelling (2.3b) is

$$\dot{\hat{d}}_f = -k_{d_f}\hat{d}_f + k_{d_f}(m\dot{v} - mgn_3 + uRn_3) \tag{3.1}$$

where $\hat{d}_f$ denotes estimate, and $k_{d_f} \in \mathbb{R}$ is a positive observer gain. Defining the estimate error $\tilde{d}_f = d_f - \hat{d}_f$, observer (3.1) has error dynamics

$$\dot{\tilde{d}}_f = \dot{d}_f - k_{d_f}\tilde{d}_f \tag{3.2}$$

which is exponentially stable if $\dot{d}_f = 0$ and ultimately bounded if $\dot{d}_f$ is bounded. Since linear acceleration is normally noisy, we take a new structure for the disturbance observer with $z_{d_f} \in \mathbb{R}^3$ as the new state coordinate and observer is introduced as

$$\hat{d}_f = z_{d_f} + k_{d_f}mv \tag{3.3a}$$

$$\dot{z}_{d_f} = -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3). \tag{3.3b}$$

This idea was used for fixed-based manipulators in [102]. Taking the derivative of $\hat{d}_f$ gives

$$\dot{\hat{d}}_f = \dot{z}_{d_f} + k_{d_f}m\dot{v} = -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3) + k_{d_f}(mgn_3 - uRn_3 + d_f) = k_{d_f}\tilde{d}_f$$

which yields error dynamics (3.2), that under the assumption of constant force disturbance $\dot{d}_f = 0$, yields

$$\dot{\tilde{d}}_f = -k_{d_f}\tilde{d}_f \tag{3.4}$$

which is exponentially stable. However, if we consider the modelling with rotor drag (2.8) effect the observer will be

$$\hat{d}_f = z_{d_f} + k_{d_f} m v \tag{3.5a}$$

$$\dot{z}_{d_f} = -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3 - RDR^\top v_a). \tag{3.5b}$$

which again results in exponentially stable error dynamics

$$\begin{aligned}\dot{\tilde{d}}_f &= k_{d_f}\hat{d}_f + k_{d_f}(mgn_3 - uRn_3 - RDR^\top v_a) - k_{d_f}(mgn_3 - uRn_3 - RDR^\top v_a + d_f) \\ &= -k_{d_f}\tilde{d}_f\end{aligned} \tag{3.6}$$

Depending on the modelling assumption, we will use one of the force disturbance observers (3.3) or (3.5) in the controller design.

We remark that u in the equation (3.3b) is theoretically the actual thrust measured by the feedback from the ESCs, not the commanded thrust from the controller, as these two are normally different due to the motor dynamics. However, in this thesis we assume these two are the same as we neglect the motor dynamics.

3.1.2 Torque Disturbance Observer

Also, we consider the following structure for the torque disturbance observer

$$\hat{d}_\tau = z_{d_\tau} + k_{d_\tau} J\omega \tag{3.7a}$$

$$\dot{z}_{d_\tau} = -k_{d_\tau}\hat{d}_\tau - k_{d_\tau}(-\omega \times J\omega + \tau) \tag{3.7b}$$

which results in the estimation error dynamics $\dot{\tilde{d}}_\tau = -k_{d_\tau}\tilde{d}_\tau$ which is exponentially stable.

In the next sections, depending on the structure of the controller-observer design (if coupled), extra terms can be added to the right-hand side of (3.3b) and (3.7b). However, for the decoupled structure the observers structure are kept the same and no extra terms will be added.

3.2 Coupled Controller-Observer Design

In this section, we drive a coupled structure for the controller-observer design using the backstepping method and the observer structure proposed in Section 3.1. The concept of tuning functions is employed in conjunction with coupling the observer to the controller by adding extra terms to achieve rigorous theoretical results. The extra terms which are a function of backstepping tracking error coordinates are added to the right-hand-side of equations (3.3b) and (3.7b). The block dia-

gram of this structure is shown in Fig. 3.1. First the position controller is presented in section 3.2.1 with the assumption of constant force and torque disturbances. However, the stability results are evaluated for the case of time-varying disturbances resulting in ultimate boundedness of the tracking errors. The yaw control is presented in Section 3.2.2. Matlab simulation results are presented in Section 3.2.3.

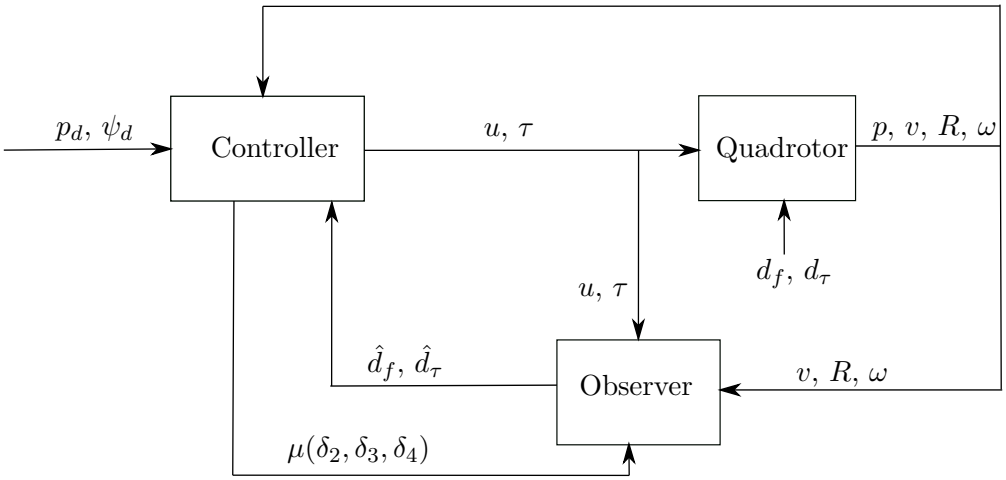


Figure 3.1: Block diagram of the Disturbance Observer-based Control with coupled structure.

3.2.1 Position Tracking Control

In this subsection we derive a tracking control for position. We begin by defining a position tracking error $\delta_1 = p - p_d$ and the first Lyapunov function candidate $V_1 = \frac{1}{2}\|\delta_1\|^2$. Taking the time derivative of V_1 and using (2.3a) gives

$$\dot{V}_1 = \delta_1^\top \dot{\delta}_1 = \delta_1^\top (v - v_d) \tag{3.8}$$

where $v_d = \dot{p}_d$. Taking v as a virtual control to (3.8) we choose $\alpha_1 = v_d - k_1\delta_1$ as its desired value, where $k_1 > 0$ is a scalar controller gain. Defining $\delta_2 = mv - m\alpha_1$ and substituting this expression into (3.8) we have

$$\dot{V}_1 = -k_1\|\delta_1\|^2 + \frac{1}{m}\delta_1^\top \delta_2.$$

Next, we consider the second Lyapunov function candidate $V_2 = V_1 + \frac{1}{2}\|\delta_2\|^2 + \frac{1}{2}\|\tilde{d}_f\|^2$ and take its time derivative

$$\dot{V}_2 = -k_1\|\delta_1\|^2 + \frac{1}{m}\delta_2^\top \delta_1 + \delta_2^\top \dot{\delta}_2 + \tilde{d}_f^\top \dot{\tilde{d}}_f. \tag{3.9}$$

Substituting $d_f = \hat{d}_f + \tilde{d}_f$ and $\dot{\alpha}_1$ into the expression for $\dot{\delta}_2$ gives

$$\dot{\delta}_2 = m\dot{v} - m\dot{\alpha}_1 = mgn_3 - uRn_3 + \hat{d}_d + \tilde{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d. \tag{3.10}$$

where we have used (2.3b). Hence, substituting (3.10) into (3.9) gives

$$\dot{V}_2 = -k_1\|\delta_1\|^2 + \frac{1}{m}\delta_2^\top \delta_1 + \delta_2^\top (mgn_3 - uRn_3 + \hat{d}_f + \tilde{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d) + \tilde{d}_f^\top \dot{\tilde{d}}_f. \tag{3.11}$$

We denote α_2 as the desired value for the second virtual control uRn_3 and take

$$\alpha_2 = mgn_3 + \hat{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d + \frac{1}{m}\delta_1 + k_2\delta_2 \tag{3.12}$$

where $k_2 > 0$ is a controller gain. We remark that the virtual input uRn_3 is the thrust vector expressed in $\mathcal{N}$. Introducing error coordinate $\delta_3 = \alpha_2 - uRn_3$ and substituting (3.12) into (3.11) gives

$$\dot{V}_2 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top \delta_3 + \tilde{d}_f^\top (\delta_2 + \dot{\tilde{d}}_f). \tag{3.13}$$

At this stage it would be possible to compensate the effect of $\tilde{d}_f$ by assigning the estimation law for $\hat{d}_f$. However, instead we tolerate the presence of $\tilde{d}_f$ in (3.13) and introduce the first tuning function $\gamma_1 = \delta_2$. Consider the Lyapunov function candidate

$$V_3 = V_2 + \frac{1}{2}\|\delta_3\|^2. \tag{3.14}$$

Differentiating (3.14) and substituting (3.13) we have

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top \delta_3 + \tilde{d}_f^\top (\gamma_1 + \dot{\tilde{d}}_f) + \delta_3^\top \dot{\delta}_3 \tag{3.15}$$

where

$$\begin{aligned}
\dot{\delta}_3 &= \dot{\alpha}_2 - \dot{u}Rn_3 - uRS(\omega)n_3 \\
&= \dot{\hat{d}}_f - m\ddot{v}_d + k_1(mgn_3 - uRn_3 + d_f) - mk_1\dot{v}_d + \frac{1}{m}(v - v_d) \\
&\quad + k_2(mgn_3 - uRn_3 + d_f - m\dot{v}_d + mk_1 v - mk_1 v_d) - \dot{u}Rn_3 - uRS(\omega)n_3.
\end{aligned} \tag{3.16}$$

As in Section 3.1, we propose a new disturbance observer

$$\hat{d}_f = z_{d_f} + k_{d_f} mv \tag{3.17a}$$

$$\dot{z}_{d_f} = -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3) + \mu(\delta_2, \delta_3, \delta_4) \tag{3.17b}$$

where z_{d_f} is the observer state, and $k_{d_f} > 0$ is an observer gain. The error coordinate δ_4 and function μ will be determined below. Observer (3.17) leads to

$$\dot{\hat{d}}_f = k_{d_f}\tilde{d}_f + \mu. \tag{3.18}$$

It is important to remark that observer (3.17) was chosen to introduce a damping term $-k_{d_f}\tilde{d}_f$ in the expression for $\dot{\tilde{d}}_f$. This ensures the negative definite term $-k_{d_f}\|\tilde{d}_f\|^2$ appears in $\dot{V}_3$ and is eventually necessary to obtain an exponentially stable closed-loop. Without this damping term, only an asymptotically stable result could be proven using the Lassale-Yoshizawa Theorem.

Substituting (3.18) and $d_f = \hat{d}_f + \tilde{d}_f$ in (3.16) we obtain

$$\dot{\delta}_3 = \mu + \beta + (k_{d_f} + k_1 + k_2)\tilde{d}_f - \dot{u}Rn_3 - uRS(\omega)n_3 \tag{3.19}$$

where

$$\begin{aligned}\beta = &-m\ddot{v}_d + k_1 mgn_3 - k_1 uRn_3 + k_1\hat{d}_f - mk_1\dot{v}_d + \frac{1}{m}v - \frac{1}{m}v_d + k_2 mgn_3 - k_2 uRn_3 \\ &+ k_2\hat{d}_f - k_2 m\dot{v}_d + k_2 mk_1 v - k_2 mk_1 v_d.\end{aligned}$$

Hence, substituting (3.19) and $\dot{\tilde{d}}_f = -k_{d_f}\tilde{d}_f - \mu$ into (3.15) gives

$$\begin{aligned}\dot{V}_3 &= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top\delta_3 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_1 - \mu) \\ &\quad + \delta_3^\top(\mu + \beta + (k_{d_f} + k_1 + k_2)\tilde{d}_f - \dot{u}Rn_3 - uRS(\omega)n_3) \\ &= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top\delta_3 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_2 - \mu) \\ &\quad + \delta_3^\top(\mu + \beta - \dot{u}Rn_3 - uRS(\omega)n_3)\end{aligned}$$

where $\gamma_2 = \gamma_1 + (k_{d_f} + k_1 + k_2)\delta_3$ is the second tuning function. We can eliminate the effect of $\tilde{d}_f$ by choosing $\mu = \gamma_2$. However, we retain γ_2 as the second tuning function and tolerate the presence of $\tilde{d}_f$ in $\dot{V}_3$. It should be noted that the terms $\dot{u}Rn_3$ and $uRS(\omega)n_3$ can be written as

$$uRS(\omega)n_3 = R\begin{bmatrix} u\omega_2 \\ -u\omega_1 \\ 0 \end{bmatrix}, \qquad \dot{u}Rn_3 = R\begin{bmatrix} 0 \\ 0 \\ \dot{u} \end{bmatrix}.$$

Since u is the system input we can assign the value of $\dot{u}$ at this stage and continue the backstepping design with $uRS(\omega)n_3$ as a virtual control. We pick

$$\dot{u} = n_3^\top R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2) \tag{3.20}$$

where $k_3 > 0$ is a controller gain. Letting $\delta_4 = \alpha_3 - uRS(\omega)n_3$ where

$$\alpha_3 = R[I - n_3 n_3^\top]R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2)$$

gives

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_2 - \mu) - k_3\|\delta_3\|^2 + \delta_3^\top(\mu - \gamma_2) + \delta_3^\top\delta_4.$$

Considering $V_4 = V_3 + \frac{1}{2}\|\delta_4\|^2$ as a new Lyapunov function candidate, we have

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_2 - \mu) - k_3\|\delta_3\|^2 + \delta_3^\top(\mu - \gamma_2) + \delta_3^\top\delta_4 + \delta_4^\top\dot{\delta}_4$$

where

$$\dot{\delta}_4 = \dot{\alpha}_3 - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 - uRS(\dot{\omega})n_3 \tag{3.21}$$

and

$$\begin{aligned}\dot{\alpha}_3 = {} & RS(\omega)[I - n_3 n_3^\top]R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2) + R[I - n_3 n_3^\top]S(\omega)^\top R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2) \\ & + R[I - n_3 n_3^\top]R^\top(\dot{\beta} + \dot{\delta}_2 + k_3\dot{\delta}_3 + \dot{\gamma}_2).\end{aligned}$$

Calculating expressions for $\dot{\beta}$, $\dot{\delta}_2$, $\dot{\delta}_3$ and $\dot{\gamma}_2$ and substituting $d_f = \hat{d}_f + \tilde{d}_f$ we obtain

$$\begin{aligned}\dot{\beta} &= \dot{\beta}' + (k_1 k_{d_f} + \frac{1}{m^2} + k_2 k_{d_f} + m k_1 k_2)\tilde{d}_f \\ \dot{\delta}_2 &= \dot{\delta}_2' + \tilde{d}_f \\ \dot{\delta}_3 &= \dot{\delta}_3' + (k_{d_f} + k_1 + k_2)\tilde{d}_f \\ \dot{\gamma}_2 &= \dot{\gamma}_2' + (1 + (k_{d_f} + k_1 + k_2)^2)\tilde{d}_f\end{aligned}$$

where $\dot{\beta}'$, $\dot{\delta}_2'$, $\dot{\delta}_3'$ and $\dot{\gamma}_2'$ are the known parts of $\dot{\beta}$, $\dot{\delta}_2$, $\dot{\delta}_3$ and $\dot{\gamma}_2$, respectively, and given by

$$\begin{aligned}\dot{\beta}' = {} & -m\dddot{v}_d - k_1\dot{u}Rn_3 - k_1 uRS(\omega)n_3 + k_1\mu - mk_1\ddot{v}_d + \frac{1}{m^2}(mgn_3 - uRn_3 + \hat{d}_f) - \frac{1}{m}\dot{v}_d \\ & - k_2\dot{u}Rn_3 - k_2 uRS(\omega)n_3 + k_2\mu - k_2 m\ddot{v}_d + k_2 k_1(mgn_3 - uRn_3 + \hat{d}_f) - k_1 k_2 m\dot{v}_d \\ \dot{\delta}_2' = {} & mgn_3 - uRn_3 + \hat{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d \\ \dot{\delta}_3' = {} & \mu + \beta - \dot{u}Rn_3 - uRS(\omega)n_3 \\ \dot{\gamma}_2' = {} & \dot{\delta}_2' + (k_{d_f} + k_1 + k_2)\dot{\delta}_3'.\end{aligned}$$

If we define the known part of $\dot{\alpha}_3$ as

$$\begin{aligned}\dot{\alpha}_3' = {} & RS(\omega)[I - n_3 n_3^\top]R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2) + R[I - n_3 n_3^\top]S(\omega)^\top R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2) \\ & + R[I - n_3 n_3^\top]R^\top(\dot{\beta}' + \dot{\delta}_2' + k_3\dot{\delta}_3' + \dot{\gamma}_2')\end{aligned}$$

then

$$\dot{\alpha}_3 = \dot{\alpha}_3' + kR[I - n_3 n_3^\top]R^\top\tilde{d}_f$$

where

$$k = (k_1 k_{d_f} + \frac{1}{m^2} + k_2 k_{d_f} + m k_1 k_2) + 1 + k_3(k_{d_f} + k_1 + k_2) + 1 + (k_{d_f} + k_1 + k_2)^2.$$

We define desired angular acceleration $\dot{\omega}_d$ satisfying

$$\dot{u}RS(\omega)n_3 + uRS(\omega)^2 n_3 - uRS(n_3)\dot{\omega}_d = \dot{\alpha}_3' + \delta_3 + k_4\delta_4 + \nu \tag{3.22}$$

where $k_4 > 0$ is a controller gain, and ν is a correction term to be determined and added to eliminate the effect of transient disturbance estimation error. From (3.21) and substituting for $\dot{\delta}_4$ gives

$$\begin{aligned}
\dot{V}_4 &= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_2 - \mu) - k_3\|\delta_3\|^2 + \delta_3^\top(\mu - \gamma_2) \\
&\quad + \delta_3^\top\delta_4 + \delta_4^\top(\dot{\alpha}_3 - \dot{\alpha}_3' - \delta_3 - k_4\delta_4 - \nu) \\
&= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_2 - \mu) - k_3\|\delta_3\|^2 + \delta_3^\top(\mu - \gamma_2) \\
&\quad + k\delta_4^\top R[I - n_3 n_3^\top]R^\top\tilde{d}_f - k_4\|\delta_4\|^2 - \delta_4^\top\nu \\
&= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 + \tilde{d}_f^\top(\gamma_3 - \mu) - k_3\|\delta_3\|^2 + \delta_3^\top(\mu - \gamma_2) \\
&\quad - k_4\|\delta_4\|^2 - \delta_4^\top\nu
\end{aligned}$$

where $\gamma_3 = \gamma_2 + kR[I - n_3 n_3^\top]R^\top\delta_4$ is the third tuning function. If we consider

$$\mu = \gamma_3$$

then $\mu - \gamma_2 = kR[I - n_3 n_3^\top]R^\top\delta_4$ and

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \delta_4^\top kR[I - n_3 n_3^\top]R^\top\delta_3 - \delta_4^\top\nu.$$

Taking

$$\nu = kR[I - n_3 n_3^\top]R^\top\delta_3$$

we get

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 < 0.$$

Denoting $\dot{\omega}_d = [\dot{\omega}_{d1}, \dot{\omega}_{d2}, \dot{\omega}_{d3}]^\top$ and solving (3.22) for $\dot{\omega}_{d1}$ and $\dot{\omega}_{d2}$ gives

$$\dot{\omega}_{d1} = -\frac{n_2^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4 + \nu) \tag{3.23a}$$

$$\dot{\omega}_{d2} = \frac{n_1^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4 + \nu). \tag{3.23b}$$

Expressions (3.23a) and (3.23b) have a singularity at $u = 0$. Conditions for avoiding it are given below in Remark 1.

In order to obtain the expressions for the torque input τ which achieves position tracking, we assume one uncontrolled rotational degree of freedom and take $\dot{\omega}_{d3} = 0$. In the case of no torque

disturbance in the rotational dynamics, torque can be derived from the algebraic relation between τ and $\dot{\omega}_d$

$$\tau = J\dot{\omega}_d + \omega \times J\omega$$

which ensures $\dot{\omega} = \dot{\omega}_d$ and, as a result, negative definiteness of $\dot{V}_4$. However, in the case disturbance torque appears, we cannot use $\tau = J\dot{\omega}_d + \omega \times J\omega - d_\tau$ since it depends on d_τ, which is unknown. Therefore, we introduce an observer for d_τ and use its estimate $\hat{d}_\tau$ in the control law

$$\tau = J\dot{\omega}_d + \omega \times J\omega - \hat{d}_\tau. \tag{3.24}$$

From (2.3d), this implies

$$\dot{\omega} = \dot{\omega}_d + J^{-1}\tilde{d}_\tau \tag{3.25}$$

where $\tilde{d}_\tau$ is the torque disturbance estimation error. Now, we can write

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \delta_4^\top uRS(n_3)J^{-1}\tilde{d}_\tau.$$

Considering the new Lyapunov function candidate

$$V_5 = V_4 + \frac{1}{2}\|\tilde{d}_\tau\|^2 \tag{3.26}$$

and its time derivative

$$\begin{aligned}\dot{V}_5 &= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \delta_4^\top uRS(n_3)J^{-1}\tilde{d}_\tau + \tilde{d}_\tau^\top\dot{\tilde{d}}_\tau \\ &= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \tilde{d}_\tau^\top(uJ^{-1}S(n_3)^\top R^\top\delta_4 + \dot{\tilde{d}}_\tau).\end{aligned}$$

Since we have assumed a constant disturbance then $\dot{\tilde{d}}_\tau = -\dot{\hat{d}}_\tau$, and the observer

$$\hat{d}_\tau = z_{d_\tau} + k_{d_\tau}J\omega \tag{3.27a}$$

$$\dot{z}_{d_\tau} = -k_{d_\tau}\hat{d}_\tau - k_{d_\tau}(-\omega \times J\omega + \tau) + uJ^{-1}S(n_3)^\top R^\top\delta_4 \tag{3.27b}$$

provides

$$\dot{V}_5 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_{d_\tau}\|\tilde{d}_\tau\|^2. \tag{3.28}$$

The above derivation of the position tracking control is summarized with the following Theorem.

Theorem 3.1. *Given system (2.3) with constant disturbances d_f, d_τ, bounded smooth reference trajectory p_d, and under the assumption that u and τ do not saturate and $u > 0$, the equilibrium $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \tilde{d}_f^\top, \tilde{d}_\tau^\top] = 0$ of the closed-loop dynamics is exponentially stable with dynamic state*

feedback control

$$
\begin{aligned}
\dot{u} &= n_3^\top R^\top(\beta + \delta_2 + k_3\delta_3 + \gamma_2) \\
\tau &= J\dot{\omega}_d + \omega \times J\omega - \hat{d}_\tau \\
\dot{\omega}_{d1} &= -\frac{n_2^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4 + \nu) \\
\dot{\omega}_{d2} &= \frac{n_1^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4 + \nu) \\
\hat{d}_f &= z_{d_f} + k_{d_f} mv \\
\dot{z}_{d_f} &= -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3) + \mu(\delta_2, \delta_3, \delta_4) \\
\hat{d}_\tau &= z_{d_\tau} + k_{d_\tau} J\omega \\
\dot{z}_{d_\tau} &= -k_{d_\tau}\hat{d}_\tau - k_{d_\tau}(-\omega \times J\omega + \tau) + uJ^{-1}S(n_3)^\top R^\top \delta_4
\end{aligned}
$$

Proof. From (3.28), the time derivative of the quadratic Lyapunov function (3.26) evaluated along the trajectories of the closed-loop dynamics is globally negative definite assuming $u > 0$. □

Remark 1. *To ensure thrust $u > 0$ to avoid the singular points in (3.23) we derive a condition involving initial conditions of the system, observer, controller state, and bounds on d_f, d_τ and $\dot{v}_d$. From $\delta_3 = \alpha_2 - uRn_3$, we have*

$$uRn_3 = \alpha_2 - \delta_3. \tag{3.29}$$

From the definition of α_2 in (3.12) and $v = \frac{1}{m}(\delta_2 + m\alpha_1)$ we have

$$\alpha_2 = mgn_3 + \hat{d}_f - m\dot{v}_d + (\frac{1}{m} - mk_1^2)\delta_1 + (k_1 + k_2)\delta_2.$$

Hence, we can rewrite (3.29) as

$$uRn_3 = mgn_3 + d_f - \tilde{d}_f - m\dot{v}_d + (\frac{1}{m} - mk_1^2)\delta_1 + (k_1 + k_2)\delta_2 - \delta_3.$$

Therefore, a lower bound for thrust is

$$\|u\| \geq mg - b_{d_f} - m\|\dot{v}_d(t)\|_\infty - \|\tilde{d}_f\| - |\frac{1}{m} - mk_1^2|\|\delta_1\| - (k_1 + k_2)\|\delta_2\| - \|\delta_3\|$$

where b_{d_f} is an upper bound for the force disturbance $\|d_f\| \leq b_{d_f}$, and $\|\cdot\|_\infty$ is the infinity norm. If we consider k_m as $k_m = max(1, |\frac{1}{m} - mk_1^2|, k_1 + k_2)$ then

$$
\begin{aligned}
\|\tilde{d}_f\| + |\frac{1}{m} - mk_1^2|\|\delta_1\| + (k_1 + k_2)\|\delta_2\| + \|\delta_3\| &\leq k_m(\|\tilde{d}_f\| + \|\delta_1\| + \|\delta_2\| + \|\delta_3\|) \\
&\leq 2k_m\sqrt{\|\tilde{d}_f\|^2 + \|\delta_1\|^2 + \|\delta_2\|^2 + \|\delta_3\|^2} \\
&\leq 2k_m\sqrt{2V_5(0)}.
\end{aligned}
$$

Therefore, we can write

$$\|u\| \geq mg - b_{d_f} - m\|\dot{v}_d(t)\|_\infty - 2k_m\sqrt{2V_5(0)}$$

and provided

$$2k_m\sqrt{2V_5(0)} \leq mg - b_{d_f} - m\|\dot{v}_d(t)\|_\infty \tag{3.30}$$

then $u > 0$. However, since d_f and d_τ are not known, $V_5(0)$ cannot be calculated and therefore condition (3.30) cannot be verified directly. However, if we consider $\|d_\tau\| \leq b_{d_\tau}$ and choose $\hat{d}_f(0) = 0$, $\hat{d}_\tau(0) = 0$, we can derive an upper bound for $V_5(0)$:

$$V_5(0) \leq \frac{1}{2}\|\delta_1(0)\|^2 + \frac{1}{2}\|\delta_2(0)\|^2 + \frac{1}{2}\|\delta_3(0)\|^2 + \frac{1}{2}\|\delta_4(0)\|^2 + \frac{1}{2}b_{d_f}^2 + \frac{1}{2}b_{d_\tau}^2 \tag{3.31}$$

therefore, condition (3.30) can be written as

$$2k_m\sqrt{\|\delta_1(0)\|^2 + \|\delta_2(0)\|^2 + \|\delta_3(0)\|^2 + \|\delta_4(0)\|^2 + b_{d_f}^2 + b_{d_\tau}^2} \leq mg - b_{d_f} - m\|\dot{v}_d(t)\|_\infty \tag{3.32}$$

3.2.2 Yaw Tracking Control

In this subsection we extend the control presented in Section 3.2.1 to track position and yaw. Given a smooth bounded reference trajectory for yaw ψ_d, we define the tracking error $\epsilon_1 = \psi - \psi_d$, and consider the Lyapunov function candidate $V_{\psi_1} = \frac{1}{2}\|\epsilon_1\|^2$ whose time derivative is $\dot{V}_{\psi 1} = \epsilon_1(\dot{\psi} - \dot{\psi}_d)$. We take $\dot{\psi}$ as a virtual control and define $\epsilon_2 = \dot{\psi} - \alpha_{\psi 1}$, where

$$\alpha_{\psi 1} = \dot{\psi}_d - k_{\psi 1}\epsilon_1$$

is the desired value for $\dot{\psi}$, and $k_{\psi 1} > 0$ is a controller gain. We obtain

$$\dot{V}_{\psi_1} = -k_{\psi 1}\|\epsilon_1\|^2 + \epsilon_1\epsilon_2.$$

Defining the second Lyapunov function candidate as $V_{\psi 2} = V_{\psi 1} + \frac{1}{2}\|\epsilon_2\|^2$ we have

$$\dot{V}_{\psi_2} = \dot{V}_{\psi_1} + \epsilon_2\dot{\epsilon}_2 = -k_{\psi 1}\|\epsilon_1\|^2 + \epsilon_1\epsilon_2 + \epsilon_2(\ddot{\psi} - \dot{\alpha}_{\psi 1}) \tag{3.33}$$

where $\dot{\alpha}_{\psi 1} = \ddot{\psi}_d - k_{\psi 1}(\dot{\psi} - \dot{\psi}_d)$. Since $\ddot{\psi}$ is algebraically related to $\dot{\omega}$ which can be considered as the system input, we can assign its value directly. Next, we present the relation between $\ddot{\psi}$ and $\dot{\omega}$. Time differentiating $\dot{\eta} = W(\eta)\omega$ gives

$$\ddot{\eta} = \dot{W}(\eta)\omega + W(\eta)\dot{\omega}.$$

Taking the last row of this equation gives

$$\ddot{\psi} = n_3^\top \dot{W}(\eta)\omega + \frac{s_\phi}{c_\theta}\dot{\omega}_2 + \frac{c_\phi}{c_\theta}\dot{\omega}_3$$

since $\dot{\omega}_2$ and $\dot{\omega}_3$ are not measurable, we replace them by their desired value using (3.25), and therefore,

$$\ddot{\psi} = n_3^\top \dot{W}(\eta)\omega + \frac{s_\phi}{c_\theta}\dot{\omega}_{d2} + \frac{c_\phi}{c_\theta}\dot{\omega}_{d3} + \frac{s_\phi}{J_{22}c_\theta}\tilde{d}_{\tau 2} + \frac{c_\phi}{J_{33}c_\theta}\tilde{d}_{\tau 3} \tag{3.34}$$

where $\tilde{d}_\tau = [\tilde{d}_{\tau 1}, \tilde{d}_{\tau 2}, \tilde{d}_{\tau 3}]^\top$, and J_{ij} is the (i,j)th entry of J. In order to simplify (3.34) and without loss of generality, we have taken the practical case of J diagonal. Now if we consider $\dot{\omega}_{d3}$ as

$$\dot{\omega}_{d3} = \frac{c_\theta}{c_\phi}(\dot{\alpha}_{\psi 1} - \epsilon_1 - k_{\psi 2}\epsilon_2 - n_3^\top \dot{W}(\eta)\omega - \frac{s_\phi}{c_\theta}\dot{\omega}_{d2}) \tag{3.35}$$

and substitute (3.35) into (3.34), we obtain

$$\ddot{\psi} = \dot{\alpha}_{\psi 1} - \epsilon_1 - k_{\psi 2}\epsilon_2 + \frac{s_\phi}{J_{22}c_\theta}\tilde{d}_{\tau 2} + \frac{c_\phi}{J_{33}c_\theta}\tilde{d}_{\tau 3} \tag{3.36}$$

We observe (3.35) has singular points at $\phi = \pi/2 + k\pi, k \in \mathbb{Z}$. Further, $\theta = \pi/2 + k\pi, k \in \mathbb{Z}$ are singular points for the Euler angles. We must therefore assume $(\phi, \theta) \in \mathcal{S} = \{(\phi, \theta) : -\pi/2 < \phi < \pi/2, -\pi/2 < \theta < \pi/2\}$ when controlling yaw and position. Considering the Lyapunov function candidate $V_6 = V_5 + V_{\psi_2}$ and substituting (3.36) into (3.33) results

$$\begin{aligned}
\dot{V}_6 &= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_{\psi 1}\|\epsilon_1\|^2 - k_{\psi 2}\|\epsilon_2\|^2 \\
&\quad + \delta_4^\top u R S(n_3) J^{-1}\tilde{d}_\tau + G^\top \tilde{d}_\tau + \tilde{d}_\tau^\top \dot{\tilde{d}}_\tau \\
&= -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_{\psi 1}\|\epsilon_1\|^2 - k_{\psi 2}\|\epsilon_2\|^2 \\
&\quad + \tilde{d}_\tau^\top (u J^{-1} S(n_3)^\top R^\top \delta_4 + G + \dot{\tilde{d}}_\tau)
\end{aligned}$$

where

$$G = [0, \frac{s_\phi}{J_{22}c_\theta}\epsilon_2, \frac{c_\phi}{J_{33}c_\theta}\epsilon_2]^\top.$$

The torque disturbance observer is given by

$$\hat{d}_\tau = z_{d_\tau} + k_{d_\tau} J\omega \tag{3.37a}$$

$$\dot{z}_{d_\tau} = -k_{d_\tau}(-\omega \times J\omega + \tau + \hat{d}_\tau) + u J^{-1} S(n_3)^\top R^\top \delta_4 + G \tag{3.37b}$$

so that

$$\dot{V}_6 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_{d_f}\|\tilde{d}_f\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_{\psi 1}\|\epsilon_1\|^2 - k_{\psi 2}\|\epsilon_2\|^2 - k_{d_\tau}\|\tilde{d}_\tau\|^2. \tag{3.38}$$

Theorem 3.2. *Given system (2.3) with constant disturbances d_f, d_τ, bounded smooth reference trajectories p_d, ψ_d, and under the assumption that u and τ do not saturate and also $u > 0$, the*

equilibrium $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \epsilon_1, \epsilon_2, \tilde{d}_f^\top, \tilde{d}_\tau^\top] = 0$ *of the closed-loop system is exponentially stable with dynamic state feedback control*

$$
\begin{aligned}
\dot{u} &= n_3^\top R^\top (\beta + \delta_2 + k_3\delta_3 + \gamma_2) \\
\tau &= J\dot{\omega}_d + \omega \times J\omega - \hat{d}_\tau \\
\dot{\omega}_{d1} &= -\frac{n_2^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4 + \nu) \\
\dot{\omega}_{d2} &= \frac{n_1^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4 + \nu) \\
\dot{\omega}_{d3} &= \frac{c_\theta}{c_\phi}(\dot{\alpha}_{\psi 1} - \epsilon_1 - k_{\psi 2}\epsilon_2 - n_3^\top \dot{W}(\eta)\omega - \frac{s_\phi}{c_\theta}\dot{\omega}_{d2}) \\
\hat{d}_f &= z_{d_f} + k_{d_f} mv \\
\dot{z}_{d_f} &= -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3) + \mu(\delta_2, \delta_3, \delta_4) \\
\hat{d}_\tau &= z_{d_\tau} + k_{d_\tau} J\omega \\
\dot{z}_{d_\tau} &= -k_{d_\tau}(-\omega \times J\omega + \tau + \hat{d}_\tau) + uJ^{-1}S(n_3)^\top R^\top \delta_4 + G
\end{aligned}
$$

Proof. As shown in (3.38) and provided $(\phi, \theta) \in \mathcal{S}$, the time derivative of the quadratic Lyapunov function V_6 evaluated along the trajectories of the closed-loop dynamics is negative definite. □

Remark 2. *In the case of bounded time-varying disturbances, i.e. when* $\dot{\tilde{d}}_f \neq 0$ *and* $\dot{\tilde{d}}_\tau \neq 0$ *and* $\|\dot{d}_f\| \leq c_{d_f}$ *and* $\|\dot{d}_\tau\| \leq c_{d_\tau}$*, for some constants* c_{d_f}, c_{d_τ}*, we can write for the time derivative of* V_6

$$
\begin{aligned}
\dot{V}_6 \leq & -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_{\psi 1}\|\epsilon_1\|^2 - k_{\psi 2}\|\epsilon_2\|^2 \\
& + \|\tilde{d}_f\|(c_{d_f} - k_{d_f}\|\tilde{d}_f\|) + \|\tilde{d}_\tau\|(c_{d_\tau} - k_{d_\tau}\|\tilde{d}_\tau\|).
\end{aligned}
$$

now, based on [103, Theorem. 4.18], the trajectory error $x = [\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \epsilon_1, \epsilon_2, \tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top$ *is ultimately bounded with* $\|x\| \leq max(\frac{c_{d_f}}{k_{d_f}}, \frac{c_{d_\tau}}{k_{d_\tau}})$*. Furthermore, the bound can be reduced by increasing the observer gains* k_{d_f} *and* k_{d_τ}*.*

3.2.3 Matlab Simulation Results

This section presents simulation results to validate the performance of the proposed method and compare it to that in [1]. The parameters of the simulation are chosen to match conditions of the ANCL (Applied Nonlinear Control Lab) quadrotor platform discussed in [97, 100]. The platform parameters are presented in Table 3.1. The desired trajectory has a figure-8 shape and given by $p_d(t) = [A\sin(\frac{2\pi t}{T}), B\sin(\frac{4\pi t}{T}), -1]$ m, where A is the amplitude of the trajectory in the n_1-direction, B is the amplitude in the n_2-direction, and T is the period of the trajectory. The trajectory for yaw is $\psi_d(t) = \text{atan}(\cos(\frac{4\pi t}{T})/\cos(\frac{2\pi t}{T}))$ rad. Based on the available flight volume we choose $A = 1.5$ m, $B = 1$ m, and $T = 12$ s. The desired trajectory is smooth and bounded as required, and sufficiently aggressive enough to test performance of the nonlinear control. The initial conditions of the quadrotor are $p_0 = [0, 1, 0]^\top$ m, $\eta_0 = [0, 0, -0.46]^\top$ rad which result initial

errors $\delta_1(0) = [0, 1, 1]^\top$ m and $\epsilon_1(0) = -0.92$ rad. The initial conditions for the linear and angular velocities are $v(0) = 0$ and $\omega(0) = 0$, respectively. To simulate the noise which appears in the system state in practice, we have added Gaussian white noise to the angular velocity and the position signals. Further, linear velocity is calculated by low-pass filtering the finite difference of the noisy position signal. This accurately simulates the experimental conditions found in our lab (e.g., [97]) where position is available from a motion capture system and the linear velocity is estimated by low-pass filtered finite difference. Based on actual experimental data we use zero mean Gaussian white noise with variance $n_p = [0.0619, 0.2320, 0.3223]^\top \times 10^{-5}$ m for position and $n_\omega = [0.1373, 0.0344, 0.9105]^\top \times 10^{-4}$ rad/s for angular velocity.

Table 3.1: Model parameters.

m	J_{11}	J_{22}	J_{33}	K_u	K_τ	ℓ
1.6 kg	0.03 kg m^2	0.03 kg m^2	0.05 kg m^2	8 N	0.8 N m	0.25 m

Constant Disturbance

In this section we apply constant disturbances $d_f = [-1, 2, -2]^\top$ N and $d_\tau = [0.3, -0.2, 0.1]^\top$N m. The initial values for z_{d_f} and z_{d_τ} are set to zero, and the controller gains are summarized in Table 3.2.

Table 3.2: Observer and controller gains: constant disturbance.

k_1	k_2	k_3	k_4	$k_{\psi 1}$	$k_{\psi 2}$	k_{d_f}	k_{d_τ}
0.4	0.6	0.8	2	2	2	0.05	1

Figure 3.2 shows the 3-D position trajectory of the quadrotor. Postion and tracking error versus time are shown in Figure 3.3(a). Tracking error converges to a sufficiently small neighborhood of zero after about 10 second. Disturbance estimation error is in Figure 3.3(b). We note that $\tilde{d}_f$ converges to an acceptable neighborhood of the origin. Torque disturbance estimation error $\tilde{d}_\tau$ is affected by the measurement noise added. However, due to the system's inertia, the state trajectories do not react to the high frequency noise in the estimate, and motion control performance is not degraded. The trajectories for thrust, torque, and Euler angles are shown in Figure 3.3(c). The trajectory for thrust is far from the singularity at $u = 0$. We conclude that despite relatively large initial condition error and the measurement noise added, position and yaw tracking performance is acceptable. When we remove the measurement noise, asymptotic convergence of the tracking errors is achieved.

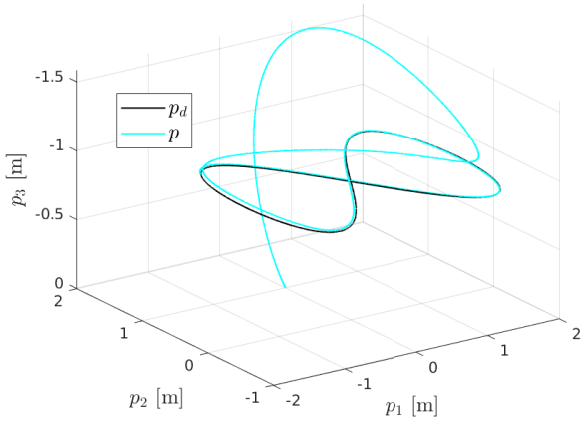
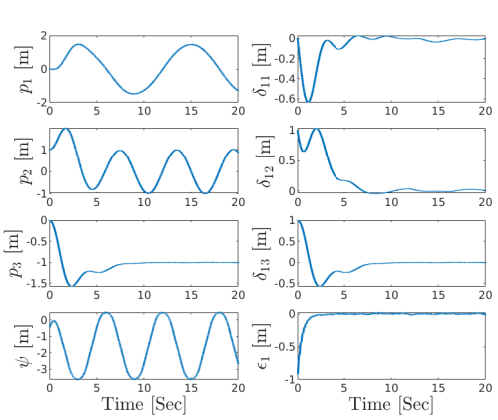
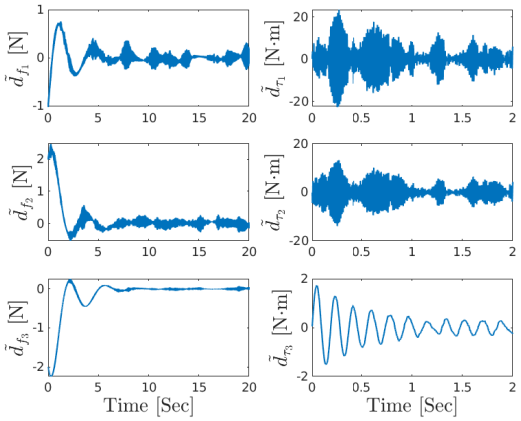


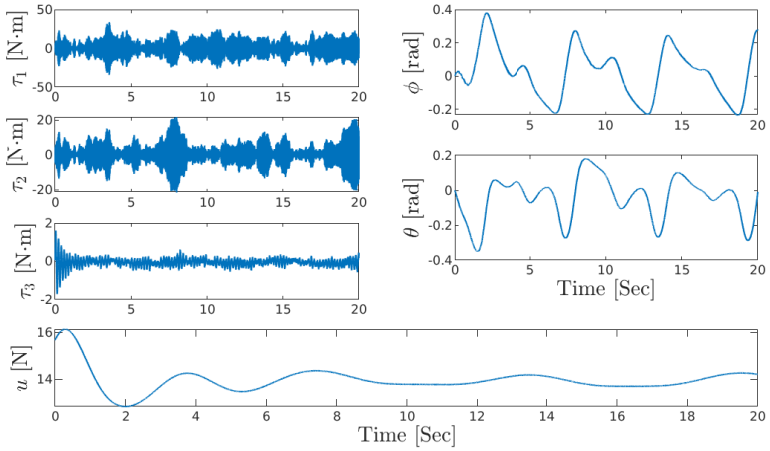
Figure 3.2: 3D Position $p = [p_1, p_2, p_3]^\top$ trajectory: constant disturbance.



(a) Position $p = [p_1, p_2, p_3]^\top$, position error $\delta_1 = [\delta_{11}, \delta_{12}, \delta_{13}]^\top$, yaw error ϵ_1.



(b) Distrubance estimation error.



(c) Inputs torque τ, thrust u, and Euler angles.

Figure 3.3: Simulation results: constant force and torque disturbance.

Time-Varying Disturbance

In this subsection we test the proposed backstepping controller with and without the disturbance observer in the case of time-varying disturbances. The force and torque disturbances are taken as $d_f(t) = [\sin(\pi t),\, \cos(2\pi t),\, -\sin(0.5\pi t)]^\top$ N and $d_\tau(t) = [0.2\sin(\pi t),\, -0.1\cos(2\pi t),\, -0.3\sin(0.5\pi t)]^\top$ N m, respectively. The controller gains are summarized in Table 3.3. Simulation results are shown in Figures 3.4 and 3.5. The tracking error for the backstepping controller without a disturbance observer is in Figure 3.6. For both cases, the initial states are the same as the constant disturbance case. The gains for both controllers are the same, other than k_{d_f} and k_{d_τ} which are not present when no disturbance observer is used. As can be seen from the results, the tracking error converges to a small neighborhood of the origin with the disturbance estimators, while tracking error diverges without an observer. The upper bound of the tracking error can be decreased by increasing the observer gains k_{d_f} and k_{d_τ}. However, large gains lead to large controls which are not practical. We observe that the effect of measurement noise is seen in the torque and its estimate. However, as with the constant disturbance case, the motion control performance is not degraded.

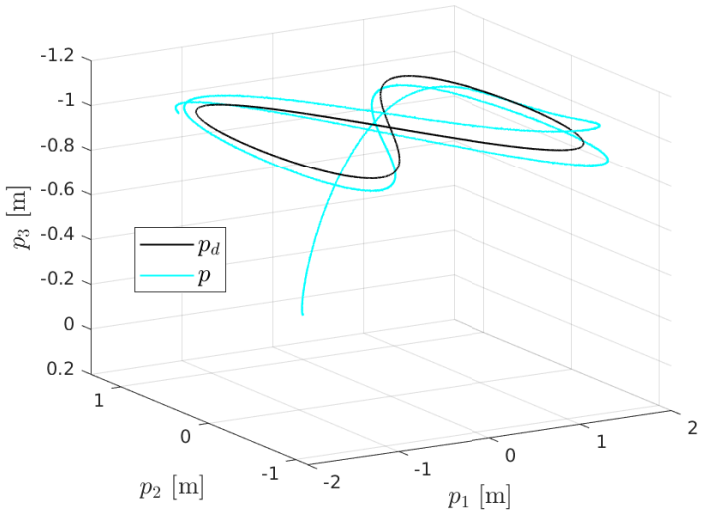
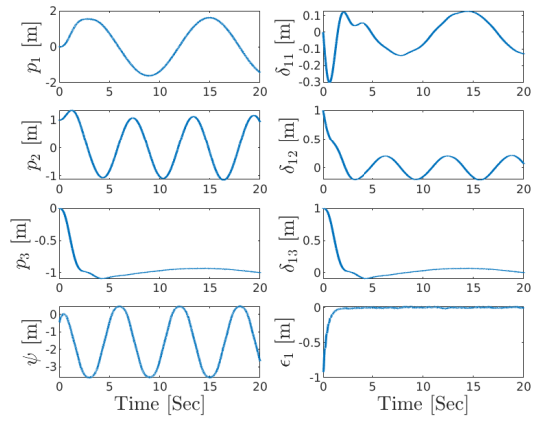


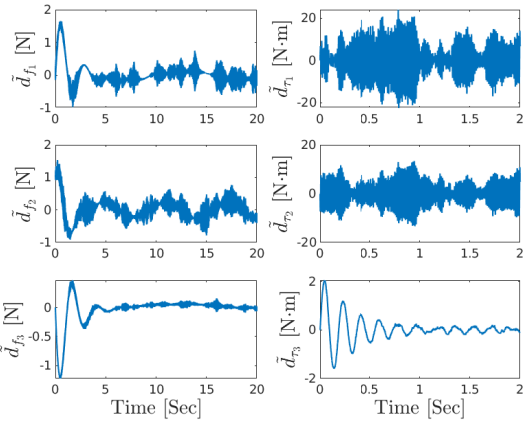
Figure 3.4: 3D Position $p = [p_1, p_2, p_3]^\top$ trajectory: time-varying disturbance.

Table 3.3: Observer and controller gains: time-varying disturbance.

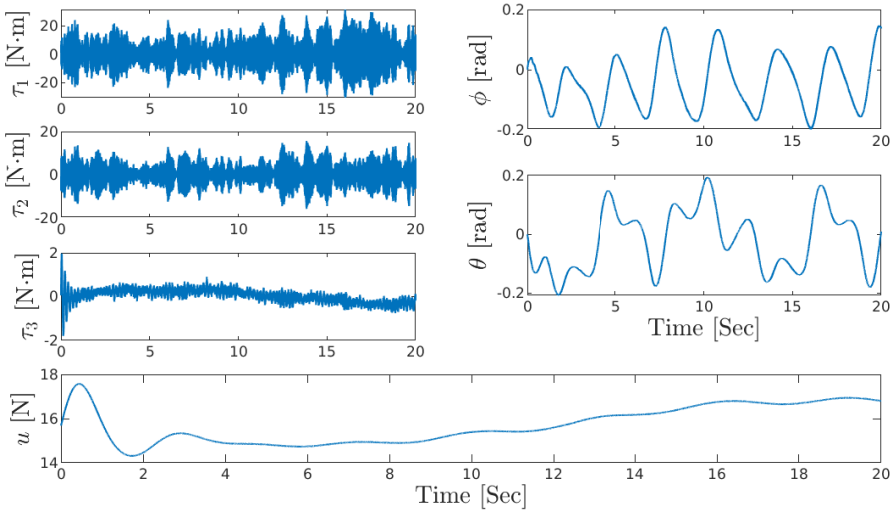
k_1	k_2	k_3	k_4	$k_{\psi 1}$	$k_{\psi 2}$	k_{d_f}	k_{d_τ}
0.8	0.75	1	2.5	2.5	2.5	0.6	3



(a) Position $p = [p_1, p_2, p_3]^\top$, position error $\delta_1 = [\delta_{11}, \delta_{12}, \delta_{13}]^\top$, yaw error ϵ_1.



(b) Disturbance estimation error.



(c) Inputs torque τ, thrust u, and Euler angles.

Figure 3.5: Simulation results: time-varying force and torque disturbance.

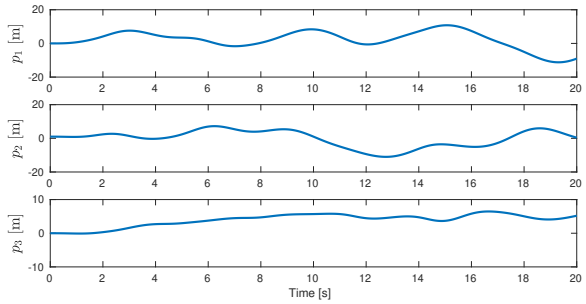


Figure 3.6: Tracking error in the case of time-varying disturbances with no disturbance observer.

Comparison of the Proposed Method

In this subsection we compare the proposed design (denoted D1) with the adaptive backstepping controller in [1] (denoted D2). The reference trajectory for position is the same as the previous section. Since for D2 yaw is uncontrolled and torque disturbances are not considered, we implement D1 without yaw tracking or a torque disturbance observer. Also due to the high complexity of expressions in the control law for the torque actuation case for D2, we simulate the angular velocity actuation case which has reduced complexity. However, D1 is simulated for the torque actuation case. Since the stable gains could not be determined for D2 using the disturbance and initial conditions considered in the previous section, we reduced the magnitudes of these quantities in this section. The initial conditions are $p(0) = [0.5, 0.5, -0.5]^\top$ m, $v(0) = 0$, $R(0) = I$ rad, and $\omega(0) = 0$. The force disturbance is $d_f = [1, 1, 0]^\top$ N, and initial values for z_{d_f} and $\hat{b}$ are zero.

After significant tuning for both methods, the final gains are in Table 3.4 for D1 and $k_3 = 8$ and $k_{b1} = 10$ for D2. As seen in Figures 3.7 and 3.8, D1 yields better transient response in the tracking error and disturbance estimation errors. In particular, $\delta_3, \tilde{d}_{f3}$ show very slow convergence for D2. The improved performance of D1 is attributed to the exponential convergence property of the closed-loop. Further, D2 has a complex implementation so that simulation times are significantly longer than those of the proposed method which implements the torque actuation case. If the torque actuation case were implemented for D2, complexity would be a larger problem. The improved performance of D1 relative to D2 is due to the introduction of the proposed observer to the controller which provides exponential stability of the system and guaranteed rates of convergence for the errors. Figure 3.9 shows the inputs with improved response for D1.

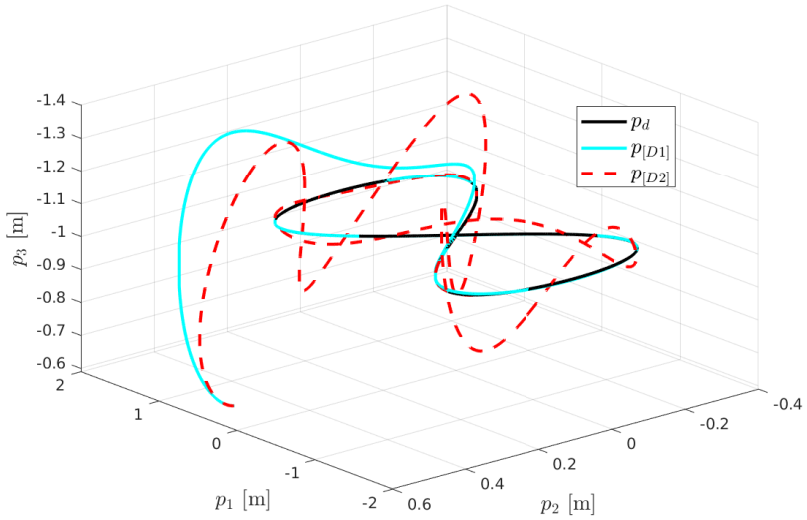


Figure 3.7: 3D Position $p = [p_1, p_2, p_3]^\top$ trajectory: desired, D1 and D2.

Table 3.4: Observer and controller gains of the proposed method

k_1	k_2	k_3	k_4	k_{d_f}
1	0.75	1	2.5	0.05

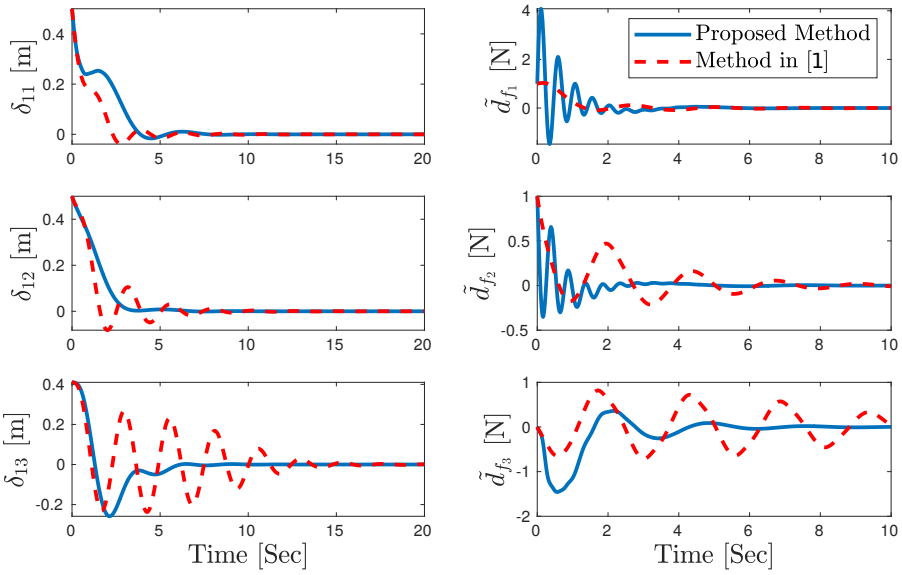


Figure 3.8: Comparison of the proposed method and the adaptive backstepping method in [1].

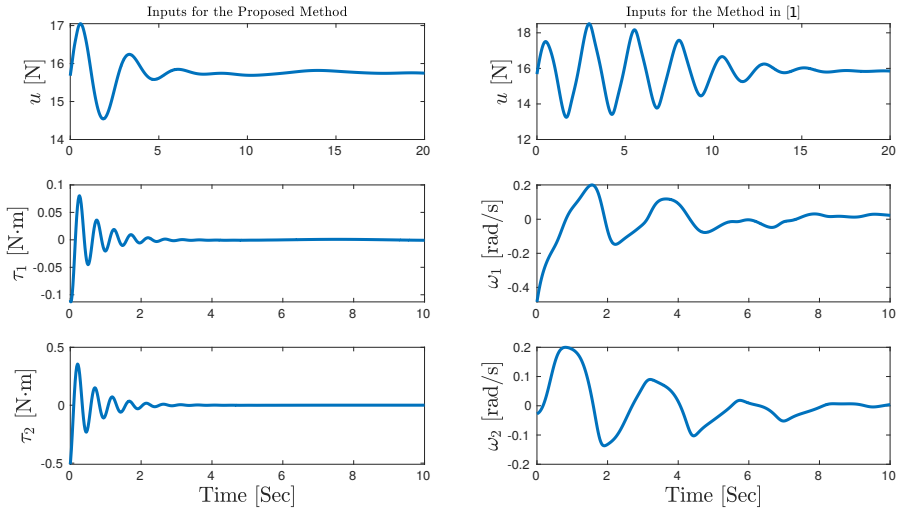


Figure 3.9: Control inputs for the proposed method and the adaptive backstepping method in [1].

3.2.4 Summary

In this section we proposed a nonlinear position and yaw tracking controller which can reject or attenuate the effect of force and torque disturbances. Exponential stability of the closed-loop is proven for constant disturbances, and ultimate boundedness is shown when disturbances are time-varying. The design method uses a backstepping approach. Unlike existing work on backstepping control, our method integrates a low-dimensional disturbance observer using tuning functions. The benefit of the disturbance observers is that exponential stability of the tracking error and disturbance estimate error is obtained when disturbances are assumed constant. Exponential stability also ensures ultimate boundedness of tracking error in the case of time-varying disturbances. The method is validated in simulation and compared to the backstepping method in [1]. The comparison shows the proposed method's reduced complexity and improved transient performance. However, due to the complexity of the control expression, the onboard implementation resulted into failure as the control laws were diverging which is due to the high sensitivity of the controller to the numerical integration error and model uncertainty. This problem can also be predicted from the simulation plots where a large oscillation can be seen in the torques the disturbance estimation error even in Matlab environment where numerical error is significantly smaller than common onboard systems. In the next section, we try to resolve this problem by simplifying the control expressions by adopting a decoupled structure.

3.3 Decoupled Controller-Observer Design

In this section we try to reduce the complexity of the controller-observer design by employing a decoupled structure so that the design is implementable on onboard systems. Again the backstepping approach and the observer proposed in 3.1 are employed. The block diagram of the proposed structure is shown in Fig. 3.10. The position tracking design is presented in section 3.3.1 and the yaw tracking controller is presented in 3.3.2. SITL simulation and experimental results are respectively presented in sections 3.3.3 and 3.3.4.

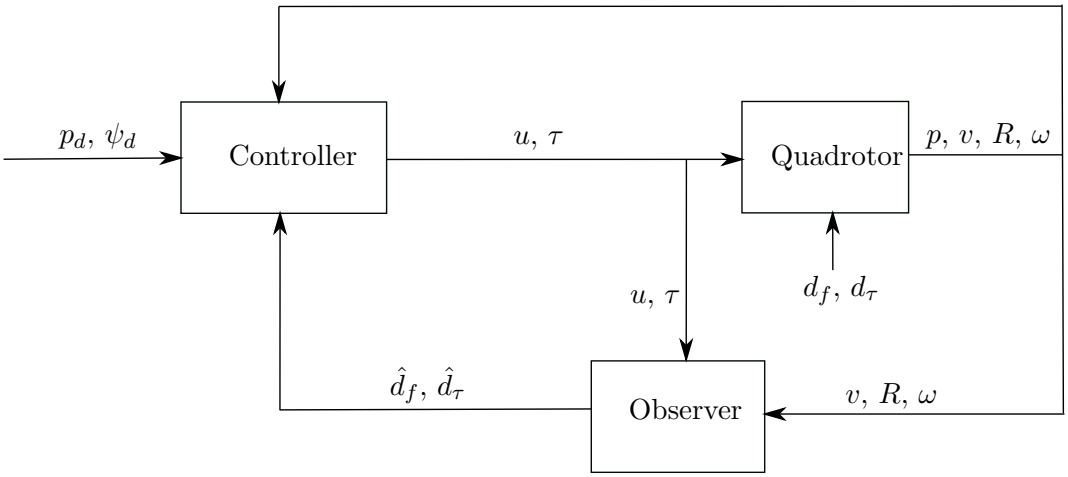


Figure 3.10: Block diagram of the Disturbance Observer-based Control with Decoupled Structure.

3.3.1 Position Tracking Control

In this subsection we design a controller for position tracking. We start by defining a tracking error $\delta_1 = p - p_d$ and the first Lyapunov function candidate $V_1 = \frac{1}{2}\|\delta_1\|^2$. Using (2.3a) gives

$$\dot{V}_1 = \delta_1^\top \dot{\delta}_1 = \delta_1^\top (v - v_d) \tag{3.39}$$

where $v_d = \dot{p}_d$. Taking v as a virtual control to (3.8) we choose $\alpha_1 = v_d - k_1\delta_1$ as its desired value, where $k_1 > 0$ is a scalar controller gain. Defining the second error coordinate as $\delta_2 = mv - m\alpha_1$ we get

$$\dot{\delta}_1 = -k_1\delta_1 + \delta_2/m \tag{3.40}$$

Substituting this expression into (3.8) we have $\dot{V}_1 = -k_1\|\delta_1\|^2 + \delta_1^\top \delta_2/m$. Next, considering the second Lyapunov function candidate $V_2 = V_1 + \frac{1}{2}\|\delta_2\|^2$ and taking its time derivative we have

$$\dot{V}_2 = -k_1\|\delta_1\|^2 + \frac{1}{m}\delta_2^\top \delta_1 + \delta_2^\top \dot{\delta}_2. \tag{3.41}$$

Substituting $d_f = \hat{d}_f + \tilde{d}_f$ and $\dot{\alpha}_1$ into the expression for $\dot{\delta}_2$ gives

$$\begin{aligned}\dot{\delta}_2 &= m\dot{v} - m\dot{\alpha}_1 \\ &= mgn_3 - uRn_3 + \hat{d}_f + \tilde{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d.\end{aligned} \tag{3.42}$$

where we have used (2.3b). Hence, substituting (3.42) into (3.41) gives

$$\dot{V}_2 = -k_1\|\delta_1\|^2 + \frac{1}{m}\delta_2^\top \delta_1 + \delta_2^\top (mgn_3 - uRn_3 + \hat{d}_f + \tilde{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d) \tag{3.43}$$

We denote α_2 as the desired value for the second virtual control uRn_3 and take

$$\alpha_2 = mgn_3 + \hat{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d + \frac{1}{m}\delta_1 + k_2\delta_2 \tag{3.44}$$

where $k_2 > 0$ is a controller gain. We remark that the virtual input uRn_3 is the thrust vector expressed in inertial frame $\mathcal{N}$. Introducing the third error coordinate $\delta_3 = \alpha_2 - uRn_3$ and substituting (3.44) into (3.43) gives

$$\dot{V}_2 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top \delta_3 + \tilde{d}_f^\top \delta_2. \tag{3.45}$$

and

$$\dot{\delta}_2 = -\delta_1/m - k_2\delta_2 + \delta_3 + \tilde{d}_f \tag{3.46}$$

Now considering $V_3 = V_2 + \frac{1}{2}\|\delta_3\|^2$ as the third Lyapunov function candidate and taking its time derivative, by substituting (3.45) we get

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top \delta_3 + \tilde{d}_f^\top \delta_2 + \delta_3^\top \dot{\delta}_3 \tag{3.47}$$

where

$$\begin{aligned}\dot{\delta}_3 &= \dot{\alpha}_2 - \dot{u}Rn_3 - uRS(\omega)n_3 \\ &= \dot{\hat{d}}_f - m\ddot{v}_d + k_1(mgn_3 - uRn_3 + d_f) - mk_1\dot{v}_d + \frac{1}{m}(v - v_d) + k_2(mgn_3 - uRn_3 + d_f \\ &\quad - m\dot{v}_d + mk_1 v - mk_1 v_d) - \dot{u}Rn_3 - uRS(\omega)n_3.\end{aligned} \tag{3.48}$$

Substituting (3.1.1) and $d_f = \hat{d}_f + \tilde{d}_f$ in (3.48) we obtain

$$\dot{\delta}_3 = \beta + (k_{d_f} + k_1 + k_2)\tilde{d}_f - \dot{u}Rn_3 - uRS(\omega)n_3 \tag{3.49}$$

where β is a function of known variables

$$\beta = -m\ddot{v}_d + k_1 mgn_3 - k_1 uRn_3 + k_1\hat{d}_f - mk_1\dot{v}_d + \frac{1}{m}v - \frac{1}{m}v_d + k_2 mgn_3 - k_2 uRn_3 + k_2\hat{d}_f$$
$$- k_2 m\dot{v}_d + k_2 mk_1 v - k_2 mk_1 v_d.$$

Hence, substituting (3.49) into (3.47) gives

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top\delta_3 + \delta_3^\top(\beta - \dot{u}Rn_3 - uRS(\omega)n_3) + \tilde{d}_f^\top(\delta_2 + (k_{d_f} + k_1 + k_2)\delta_3)$$

The terms $\dot{u}Rn_3$ and $uRS(\omega)n_3$ can be written as

$$uRS(\omega)n_3 = R\begin{bmatrix} u\omega_2 \\ -u\omega_1 \\ 0 \end{bmatrix}, \qquad \dot{u}Rn_3 = R\begin{bmatrix} 0 \\ 0 \\ \dot{u} \end{bmatrix}.$$

Since u is the system input, we can assign the value of $\dot{u}$ at this stage and continue the backstepping design with $uRS(\omega)n_3$ as a virtual control. So we choose

$$\dot{u} = n_3^\top R^\top(\beta + \delta_2 + k_3\delta_3) \tag{3.50}$$

where $k_3 > 0$ is a controller gain. Letting $\delta_4 = \alpha_3 - uRS(\omega)n_3$ where

$$\alpha_3 = R[I - n_3 n_3^\top]R^\top(\beta + \delta_2 + k_3\delta_3) \tag{3.51}$$

gives us

$$\dot{\delta}_3 = -\delta_2 - k_3\delta_3 + \delta_4 + (k_{d_f} + k_1 + k_2)\tilde{d}_f \tag{3.52}$$

The choice (3.20) and (3.51) cancels the effect of indefinite terms $\delta_2^\top\delta_3$ and $\delta_3^\top\beta$ in $\dot{V}_3$ and adds a damping term $-k_3\|\delta_3\|^2$. We have

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top\delta_4 + \tilde{d}_f^\top(\delta_2 + (k_{d_f} + k_1 + k_2)\delta_3)$$

Considering $V_4 = V_3 + \frac{1}{2}\|\delta_4\|^2$ as a new Lyapunov function candidate, we have

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top\delta_4 + \tilde{d}_f^\top(\delta_2 + (k_{d_f} + k_1 + k_2)\delta_3) + \delta_4^\top\dot{\delta}_4$$

where

$$\dot{\delta}_4 = \dot{\alpha}_3 - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 - uRS(\dot{\omega})n_3 \tag{3.53}$$

and

$$\dot{\alpha}_3 = RS(\omega)[I - n_3 n_3^\top]R^\top(\beta + \delta_2 + k_3\delta_3) + R[I - n_3 n_3^\top]S(\omega)^\top R^\top(\beta + \delta_2 + k_3\delta_3) + R[I - n_3 n_3^\top]R^\top(\dot{\beta} + \dot{\delta}_2 + k_3\dot{\delta}_3).$$

Calculating expressions for $\dot{\beta}$, $\dot{\delta}_2$ and $\dot{\delta}_3$ and substituting $d_f = \hat{d}_f + \tilde{d}_f$ we obtain

$$\begin{aligned}
\dot{\beta} &= \dot{\beta}' + (k_1 k_{d_f} + \frac{1}{m^2} + k_2 k_{d_f} + k_1 k_2)\tilde{d}_f \\
\dot{\delta}_2 &= \dot{\delta}_2' + \tilde{d}_f \\
\dot{\delta}_3 &= \dot{\delta}_3' + (k_{d_f} + k_1 + k_2)\tilde{d}_f
\end{aligned}$$

where $\dot{\beta}'$, $\dot{\delta}_2'$ and $\dot{\delta}_3'$ are the known parts of $\dot{\beta}$, $\dot{\delta}_2$ and $\dot{\delta}_3$, respectively, and given by

$$\begin{aligned}
\dot{\beta}' &= -m\dddot{v}_d - k_1\dot{u}Rn_3 - k_1 uRS(\omega)n_3 - mk_1\ddot{v}_d + \frac{1}{m^2}(mgn_3 - uRn_3 + \hat{d}_f) - \frac{1}{m}\dot{v}_d \\
&\quad - k_2\dot{u}Rn_3 - k_2 uRS(\omega)n_3 - k_2 m\ddot{v}_d + k_2k_1(mgn_3 - uRn_3 + \hat{d}_f) - k_1k_2m\dot{v}_d \\
\dot{\delta}_2' &= mgn_3 - uRn_3 + \hat{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d \\
\dot{\delta}_3' &= \beta - \dot{u}Rn_3 - uRS(\omega)n_3.
\end{aligned}$$

therefore, the known part of $\dot{\alpha}_3$ can be defined by $\dot{\alpha}_3' : \dot{\alpha}_3 = \dot{\alpha}_3' + kR[I - n_3 n_3^\top]R^\top\tilde{d}_f$ where

$$\dot{\alpha}_3' = RS(\omega)[I - n_3 n_3^\top]R^\top(\beta + \delta_2 + k_3\delta_3) + R[I - n_3 n_3^\top]S(\omega)^\top R^\top(\beta + \delta_2 + k_3\delta_3) + R[I - n_3 n_3^\top]R^\top(\dot{\beta}' + \dot{\delta}_2' + k_3\dot{\delta}_3')$$

and

$$k = (k_1 k_{d_f} + \frac{1}{m^2} + k_2 k_{d_f} + k_1 k_2) + 1 + k_3(k_{d_f} + k_1 + k_2).$$

Now we can write the expression for $\dot{V}_4$ as

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_4^\top(\dot{\alpha}_3' + \delta_3 - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 - uRS(\dot{\omega})n_3) + \tilde{d}_f^\top(\delta_2 + (k_{d_f} + k_1 + k_2)\delta_3 + kR[I - n_3 n_3^\top]R^\top\delta_4)$$

At this stage, we define a desired value for the rate of angular velocity $\dot{\omega}_d$, such that

$$\delta_4^\top(\dot{\alpha}_3' + \delta_3 - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 - uRS(\dot{\omega}_d)n_3) = -k_4\delta_4^\top\delta_4$$

where $k_4 > 0$ is a controller gain. This leads to the expression $-k_4\|\delta_4\|^2$ appearing in $\dot{V}_4$. Considering the structure for $S(\dot{\omega}_d)n_3$, we can write the above equation as

$$\delta_4^\top R \begin{bmatrix} \dot{\omega}_{d2} \\ -\dot{\omega}_{d1} \\ 0 \end{bmatrix} u = \delta_4^\top (\dot{\alpha}_3' + \delta_3 - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + k_4\delta_4) \tag{3.54}$$

now using the fact that the third component of $R^\top \delta_4$ is zero, we can conclude that by selecting

$$\dot{\omega}_{d1} = -\frac{n_2^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4) \tag{3.55a}$$

$$\dot{\omega}_{d2} = \frac{n_1^\top R^\top}{u}(\dot{\alpha}_3' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_3 + k_4\delta_4). \tag{3.55b}$$

that (3.54) is satisfied. In order to obtain the expressions for the torque input τ which achieves position tracking, we consider $\dot{\omega}_{d3} = 0$. Now we try to get the expression for torque which is the actual sytem input. From the rotational dynamics, we consider

$$\tau = J\dot{\omega}_d + \omega \times J\omega - \hat{d}_\tau \tag{3.56}$$

where, $\hat{d}_\tau$ is the torque disturbance estimate from (3.7) to cancel the effect of d_τ. From (3.56) and (2.3d) we get

$$\dot{\omega} = \dot{\omega}_d + J^{-1}\tilde{d}_\tau \tag{3.57}$$

Substituting (3.55) and (3.57) in $\dot{\delta}_4$, gives us

$$\dot{\delta}_4 = -\delta_3 - k_4\delta_4 + uRS(n_3)J^{-1}\tilde{d}_\tau + kR[I - n_3 n_3^\top]R^\top \tilde{d}_f \tag{3.58}$$

Now we can write the following expression for $\dot{V}_4$

$$\begin{aligned}\dot{V}_4 = &-k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \delta_4^\top uRS(n_3)J^{-1}\tilde{d}_\tau \\ &+ \tilde{d}_f^\top(\delta_2 + (k_{d_f} + k_1 + k_2)\delta_3 + kR[I - n_3 n_3^\top]R^\top \delta_4)\end{aligned} \tag{3.59}$$

The closed loop stability of the proposed backstepping controller is summarized by the following theorem. Rather than show the negative definiteness of $\dot{V}_4$ in (3.59), we take a more intuitive approach and present the closed-loop stability result for the proposed control by exploiting a cascade of two subsystems: the tracking error dynamics expressed in the δ-coordinates, where $\delta = [\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top]^\top$, and the disturbance estimate error dynamics expressed in the $\tilde{d}$ coordinates, where $\tilde{d} = [\tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top$.

Theorem 3.3. *Given system (2.3) with constant disturbances d_f and d_τ, bounded smooth reference trajectory p_d and assuming $u > 0$, the equilibrium $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top = 0$ of the closed-loop dynamics is exponentially stable with dynamic state feedback control (3.3), (3.7), (3.20), (3.55), and*

(3.56).

Proof. Denoting $\delta = [\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top]^\top$ and $\tilde{d} = [\tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top$ then from (3.40), (3.46), (3.52) and (3.58) the tracking error dynamics is

$$\dot{\delta} = A\delta + B(\delta, t)\tilde{d} \tag{3.60}$$

where

$$A = \begin{bmatrix} -k_1 I_{3\times3} & I_{3\times3}/m & 0_{3\times3} & 0_{3\times3} \\ -I_{3\times3}/m & -k_2 I_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -I_{3\times3} & -k_3 I_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & -k_4 I_{3\times3} \end{bmatrix}$$

$$B(\delta, t) = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} \\ (k_{d_f} + k_1 + k_2) I_{3\times3} & 0_{3\times3} \\ kR[I - n_3 n_3^\top]R^\top & uRS(n_3)J^{-1} \end{bmatrix}$$

We remark that the term $B(\delta, t)\tilde{d}$ that interconnects the δ and $\tilde{d}$ subsystems is time-varying due to the tracking problem considered. We apply the exponential stability result for cascades in [104, Prop. 2.3]. Clearly, the disturbance estimate error dynamics is LTI and exponentially stable. As well, assuming "zero input", i.e., $\tilde{d} = 0$, the dynamics (3.60) are LTI and exponentially stable for any $k_1, k_2, k_3, k_4 > 0$. Hence, it only remains to show

$$\|B(\delta, t)\| \leq C_1 + C_2\|\delta\| \tag{3.61}$$

for some $C_1, C_2 > 0$. Since $\delta_3 = \alpha_2 - uRn_3$, we can deduce $|u| \leq \|\alpha_2\| + \|\delta_3\|$, and from $\alpha_2 = mgn_3 + \hat{d}_f - m\dot{v}_d + mk_1 v - mk_1 v_d + \frac{1}{m}\delta_1 + k_2\delta_2$, and by replacing $\hat{d}_f = d_f - \tilde{d}_f$, $v = \frac{1}{m}(\delta_2 - m\alpha_1)$ and $\alpha_1 = v_d - k_1\delta_1$ we get

$$\alpha_2 = mgn_3 + d_f - \tilde{d}_f - m\dot{v}_d + k_1(\delta_2 - m(v_d - k_1\delta_1)) - mk_1 v_d + \frac{1}{m}\delta_1 + k_2\delta_2$$

therefore, we can obtain an upper bound $\|\alpha_2\| \leq \bar{C}_1\|\delta\| + \bar{C}_2$, $\bar{C}_1, \bar{C}_2 > 0$.

Hence, we conclude all entries of B are linearly bounded in the norm of δ and we obtain (3.61). Therefore, the conditions for [104, Prop. 2.3] are satisfied the equilibrium $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top = 0$ of closed-loop (2.3), (3.3), (3.7), (3.20), (3.55) and (3.56) is exponentially stable. □

3.3.2 Yaw Tracking Control

In this subsection we develop a controller to track trajectories for yaw using τ_3. Given a smooth bounded reference trajectory ψ_d, we define the tracking error $\epsilon_1 = \psi - \psi_d$, and consider the Lyapunov function candidate $V_{\psi_1} = \frac{1}{2}\|\epsilon_1\|^2$ whose time derivative is $\dot{V}_{\psi 1} = \epsilon_1(\dot{\psi} - \dot{\psi}_d)$. We take $\dot{\psi}$ as a virtual control and define $\epsilon_2 = \dot{\psi} - \alpha_{\psi 1}$, where $\alpha_{\psi 1} = \dot{\psi}_d - k_{\psi 1}\epsilon_1$ is the desired value for $\dot{\psi}$, and $k_{\psi 1} > 0$ is a controller gain. We obtain $\dot{V}_{\psi_1} = -k_{\psi 1}\|\epsilon_1\|^2 + \epsilon_1\epsilon_2$. Defining the second Lyapunov

function candidate as $V_{\psi 2} = V_{\psi 1} + \frac{1}{2}\|\epsilon_2\|^2$ we get

$$\dot{V}_{\psi_2} = \dot{V}_{\psi_1} + \epsilon_2\dot{\epsilon}_2 = -k_{\psi 1}\|\epsilon_1\|^2 + \epsilon_1\epsilon_2 + \epsilon_2(\ddot{\psi} - \dot{\alpha}_{\psi 1}) \tag{3.62}$$

where $\dot{\alpha}_{\psi 1} = \ddot{\psi}_d - k_{\psi 1}(\dot{\psi} - \dot{\psi}_d)$. At this stage we can immediately assign the value of $\ddot{\psi}$ since it is directly related to $\dot{\omega}$ which can be considered as the system input. Next, we present the relation between $\ddot{\psi}$ and $\dot{\omega}$. Taking time derivative of $\dot{\eta} = W(\eta)\omega$ gives $\ddot{\eta} = \dot{W}(\eta)\omega + W(\eta)\dot{\omega}$. From the last row of this equation we have

$$\ddot{\psi} = n_3^\top \dot{W}(\eta)\omega + \frac{s_\phi}{c_\theta}\dot{\omega}_2 + \frac{c_\phi}{c_\theta}\dot{\omega}_3 \tag{3.63}$$

Since $\dot{\omega}_2$ and $\dot{\omega}_2$ are not measurable, we replace them by their desired value using equation (3.57)

$$\ddot{\psi} = n_3^\top \dot{W}(\eta)\omega + \frac{s_\phi}{c_\theta}\dot{\omega}_{d2} + \frac{c_\phi}{c_\theta}\dot{\omega}_{d3} + \frac{s_\phi}{J_{22}c_\theta}\tilde{d}_{\tau 2} + \frac{c_\phi}{J_{33}c_\theta}\tilde{d}_{\tau 3} \tag{3.64}$$

where $\tilde{d}_\tau = [\tilde{d}_{\tau 1}, \tilde{d}_{\tau 2}, \tilde{d}_{\tau 3}]^\top$, and J_{ij} is the (i,j)th entry of J. In order to simplify (3.64) and without loss of generality we have considered the practical case of J diagonal. Now, if we consider $\dot{\omega}_{d3}$ as

$$\dot{\omega}_{d3} = \frac{c_\theta}{c_\phi}(\dot{\alpha}_{\psi 1} - \epsilon_1 - k_{\psi 2}\epsilon_2 - n_3^\top \dot{W}(\eta)\omega - \frac{s_\phi}{c_\theta}\dot{\omega}_{d2}) \tag{3.65}$$

and substitute (3.65) into (3.34), we obtain

$$\ddot{\psi} = \dot{\alpha}_{\psi 1} - \epsilon_1 - k_{\psi 2}\epsilon_2 + \frac{s_\phi}{J_{22}c_\theta}\tilde{d}_{\tau 2} + \frac{c_\phi}{J_{33}c_\theta}\tilde{d}_{\tau 3} \tag{3.66}$$

We observe (3.65) has singular points at $\phi = \pi/2 + k\pi, k \in \mathbb{Z}$. Further, from the definition of $W(\eta)$, $\theta = \pi/2 + k\pi, k \in \mathbb{Z}$ are singular points for the Euler angles. We must therefore assume $(\phi, \theta) \in \mathcal{S} = \{(\phi, \theta) : -\pi/2 < \phi < \pi/2, -\pi/2 < \theta < \pi/2\}$ when controlling yaw and position. Considering the Lyapunov function candidate $V_5 = V_4 + V_{\psi_2}$ and substituting (3.36) into (3.33) results

$$\begin{aligned}\dot{V}_5 = &-k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_{\psi 1}\|\epsilon_1\|^2 - k_{\psi 2}\|\epsilon_2\|^2 \\ &+ \tilde{d}_\tau^\top(uJ^{-1}S(n_3)^\top R^\top \delta_4 + G\epsilon_2) + \tilde{d}_f^\top(\delta_2 + (k_{d_f} + k_1 + k_2)\delta_3 + kR[I - n_3 n_3^\top]R^\top \delta_4)\end{aligned}$$

where $G = [0, \frac{s_\phi}{J_{22}c_\theta}, \frac{c_\phi}{J_{33}c_\theta}]^\top$.

Theorem 3.4. *Given system (2.3) with constant disturbances d_f and d_τ and bounded smooth reference trajectories p_d, ψ_d, assuming $u > 0$ and provided $(\phi, \theta) \in \mathcal{S}$, the equilibrium of the closed-loop system $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \epsilon_1, \epsilon_2, \tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top = 0$ is exponentially stable with dynamic state feedback control (3.3), (3.7), (3.20), (3.55), (3.56) and (3.65).*

Proof. The procedure of the proof is similar to the proof of Theorem 3.3 except matrices A and B

are

$$A = \begin{bmatrix} -k_1 I_{3\times3} & I_{3\times3}/m & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ -I_{3\times3}/m & -k_2 I_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & -I_{3\times3} & -k_3 I_{3\times3} & I_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & -k_4 I_{3\times3} & 0_{3\times1} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & -k_{\psi 1} & 1 \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & -1 & -k_{\psi 2} \end{bmatrix}$$

$$B = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} \\ (k_{d_f} + k_1 + k_2) I_{3\times3} & 0_{3\times3} \\ kR(I - n_3 n_3^\top) R^\top & uRS(n_3) J^{-1} \\ 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & G^\top \end{bmatrix}$$

□

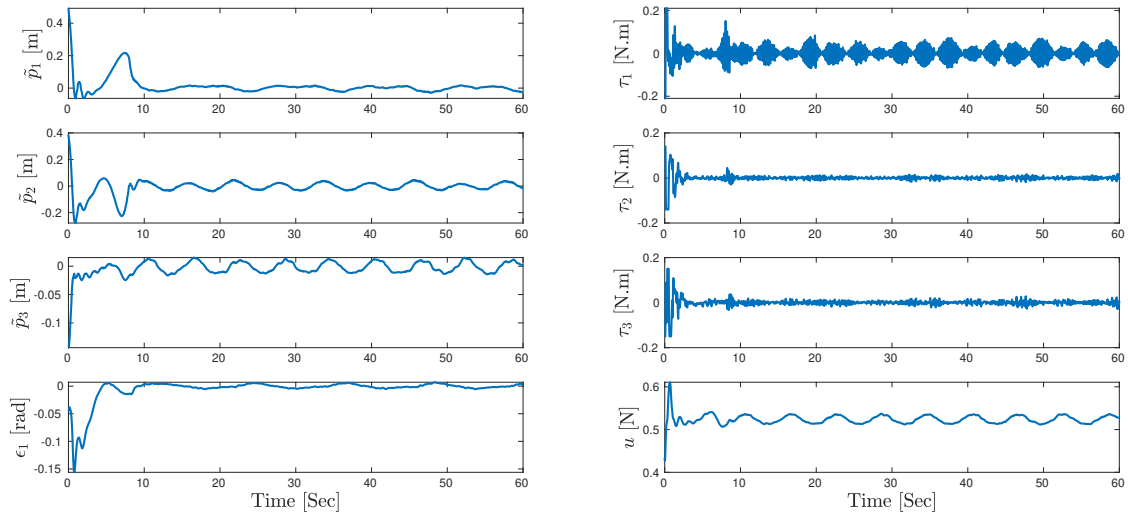
An important contribution of this work is the decoupled structure of the control law. The disturbance observer design is decoupled from the state feedback control and this simplifies the controller expressions and allows for implementation on a commonly used autopilot such as PX4/Pixhawk. Also, assuming $\tilde{d}$ is an input the closed-loop tracking error dynamics is linear, and exponentially stability can be readily proven. This should be compared to the adaptive backstepping control in [1] which achieves asymptotic stability. In fact, most of the mentioned existing work in Section 1.3 only provides asymptotic stability results, sometimes only for an outer loop subsystem [41] or for approximate models [9, 59].

3.3.3 SITL/jMAVSim Simulation

To evaluate the performance of the controller the reference trajectory is taken as a figure-8 given by $p_d(t) = [A\sin(2\pi t/T) + 1, B\sin(4\pi t/T), -0.85]$ m, with $A = 1.5$ and $B = 0.75$. The velocity of the trajectory is increased by decreasing T from 20 at $t = 0$ to 12 in 8 seconds. The setpoint for yaw is zero. In order to investigate the benefit of the disturbance observer, the disturbance observer based backstepping (DOB-BS) control is compared to the backstepping controller (BS) and the PX4 built-in position control (PX4MC). Extensive simulation was performed at different values for controller gains while monitoring the RMS value of the tracking error. This testing led to optimal values $k_1 = k_2 = k_3 = k_4 = 7$. We observed that lower values of gain generally led to larger RMS error while larger values destabilized the UAV. After tuning $k_1 = k_2 = k_3 = k_4$, a similar process was used to find optimal observer gains $k_{d_f} = 0.5$ and $k_{d_\tau} = 0.5$. The vehicle is initialized in hover at $p(0) = [0.5, 0.5, -1]^\top$ m. Hence, a relatively large initial error exists when the controller is switched on, i.e., $\tilde{p}(0) = [0.5, 0.5, -0.15]^\top$ m.

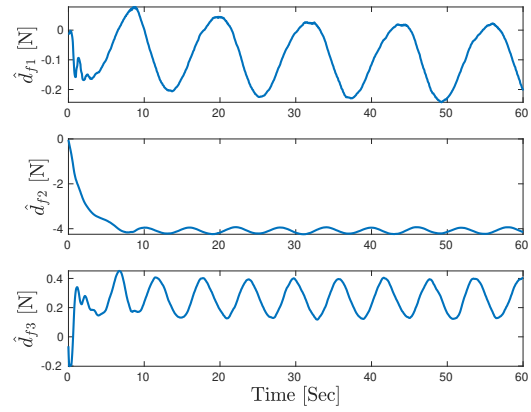
Simulation results for the DOB-BS control are shown in Figure 3.11. As can be observed, the tracking error converges to an acceptable neighborhood of the origin in approximately 10 s and

remains there. The normalized torques and thrust inputs are in Figure 3.11(b). The inputs have reasonable magnitude and remain unsaturated throughout the simulation. As expected, the thrust has an average value close to 0.51 N. The disturbance estimates are shown in Figure 3.11(c) and 3.11(d). We observe $\hat{d}_{f_1}$ has approximately -0.1 N average in steady-state and oscillations at the frequency of the reference trajectory. The average value of $\hat{d}_{f_2}$ in steady-state is about -4 N which is consistent with the applied wind disturbance in the $-n_2$ direction. The steady-state mean of $\hat{d}_{f_3}$ is almost 0.3 N which is due to modeling error. We observe the estimate for the torque disturbance is almost zero. This is consistent with the disturbance the model (2.19) and since the angular velocity of the quadrotor along the figure-8 trajectory is small.

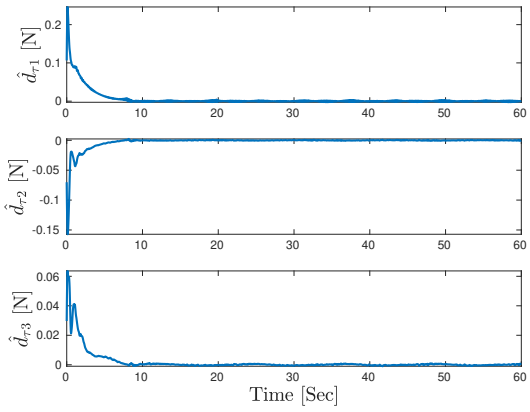


(a) Tracking error.

(b) Normalized controller inputs.



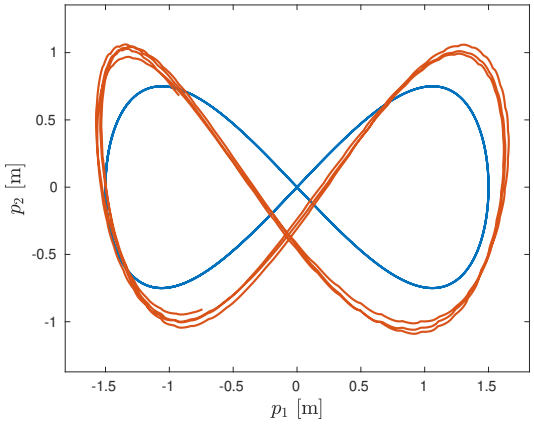
(c) Force disturbance estimates.



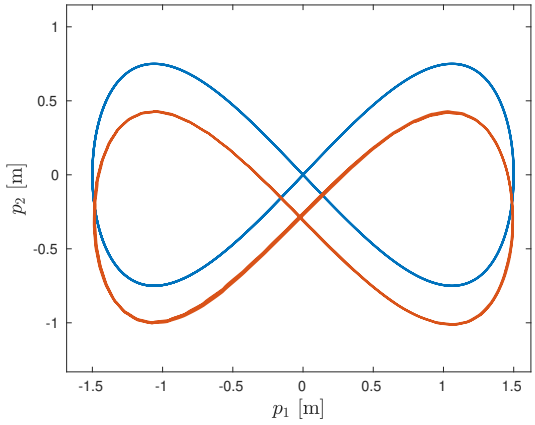
(d) Torque disturbance estimates.

Figure 3.11: Trajectory tracking error, controller inputs, force and torque disturbance estimates for the DOB-BS control in the SITL/jMAVSim simulation.

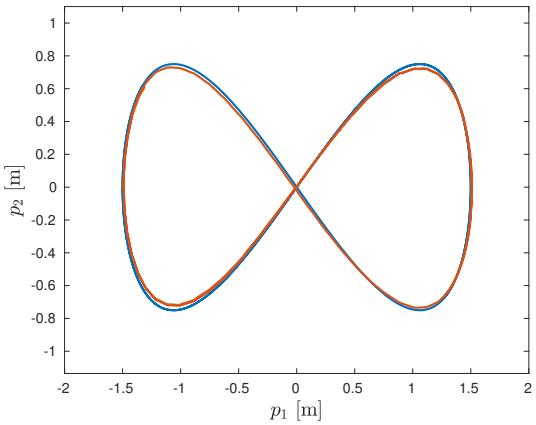
The simulation results for the actual and desired trajectories are shown in 2D for all controllers



(a) PX4MC.



(b) BS.



(c) DOB-BS.

Figure 3.12: Desired and actual trajectories for the control methods simulated using SITL/jMAVSim.

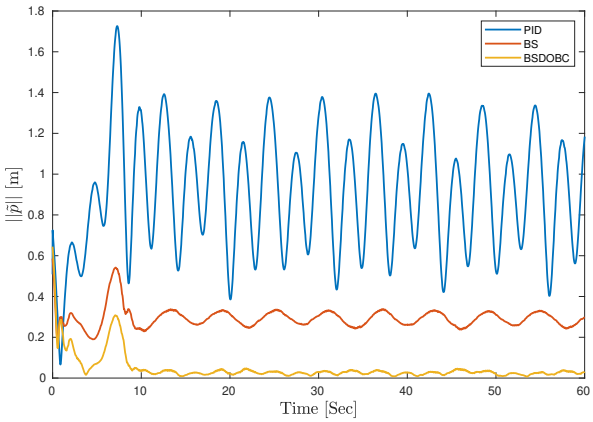


Figure 3.13: Norm of trajectory tracking error of the simulated using SITL/jMAVSim.

in Figure 3.12. Table 3.5 provides the root mean square of the position error (RMSE) in steady-state. The trajectory of the norm of the tracking error is in Figure 3.13. We observe from Table 3.5 the DOB-BS control achieves the lowest steady-state RMSE norm of tracking error. The 2D plots clearly reflect this reduced error and the effectiveness of the proposed disturbance observer on improving the tracking performance. The code which reproduces this simulation is available online for public use [92].

Table 3.5: RMSE of the steady-state position tracking error in the SITL/jMAVSim simulation.

Controller	PX4MC	BS	DOB-BS
RMSE($\|\tilde{p}\|$) m	0.9583	0.2907	0.0268

3.3.4 Physical Experiment

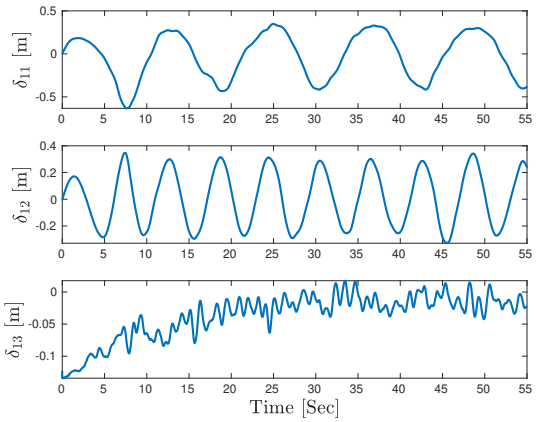
In this section we investigate the performance of the proposed controller implemented on the ANCL quadrotor platform. External disturbances are created with a fan which was placed at a height of about 0.85 m (relative to the fan's centre) and 3 m away from the UAV's initial position. The fan has a blade diameter of about 0.6 m and it produces an airflow of about 1.6 m/s in the direction of the n_1 axis. We remark that the airflow due to the fan will alter the thrust generation of the vehicle, and this change in aerodynamics is difficult to model. As in the simulation, the reference trajectory is a figure-8 given by $p_d(t) = [A\sin(2\pi t/T) + 1, B\sin(4\pi t/T), -0.8]$ m, with $A = 0.7$ and $B = 0.3$. The velocity of the trajectory is increased as a function of time by decreasing T from 20 to 12 in 8 seconds. The setpoint for yaw is zero. As in the simulation, the DOB-BS control is compared with BS and PX4MC controllers. The experiment consists of flying close to the starting point of the reference trajectory manually. Then the controller is switched to setpoint regulation mode (using the stock PX4 control) with setpoint $p_d(0) = [1, 0, -0.8]$ m. Next, one of

Table 3.6: Observer and controller gains used in experiment.

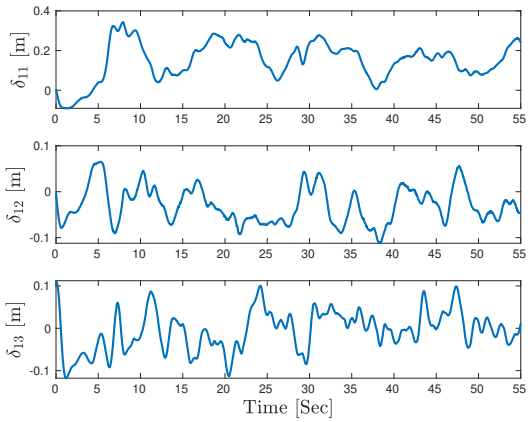
k_1	k_2	k_3	k_4	$k_{\psi 1}$	$k_{\psi 2}$	k_{d_f}
14	8.5	1.5	1.5	3.1	1.3	2.7

the three trajectory tracking controllers is enabled and the fan turned on. The controller gains for the DOB-BS control are in Table 3.6. In order to tune the controller gains, the lateral position subsystem was considered initially. We began with small values of k_1, k_2, k_3, k_4 and the remaining 2 DoF (i.e., yaw and altitude) controlled by the stock PX4 motion controller. The RMS value to the tracking error was observed as k_1, k_2, k_3, k_4 and torque normalization parameters n_{τ_1}, n_{τ_2} and n_{τ_3} were systematically varied. A similar procedure was applied to tune the other 2 DoF which are affected by $k_{\psi 1}, k_{\psi 2}$ and the thrust normalization parameters (2.16).

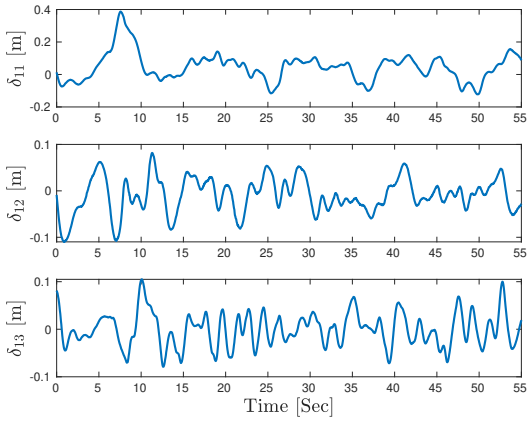
The resulting tracking errors, p_1-p_2 plot, control inputs and disturbance estimates are shown in Figures 3.14, 3.15, 3.16 and 3.17, respectively. As can be observed from Figures 3.14 and 3.15,



(a) PX4MC.

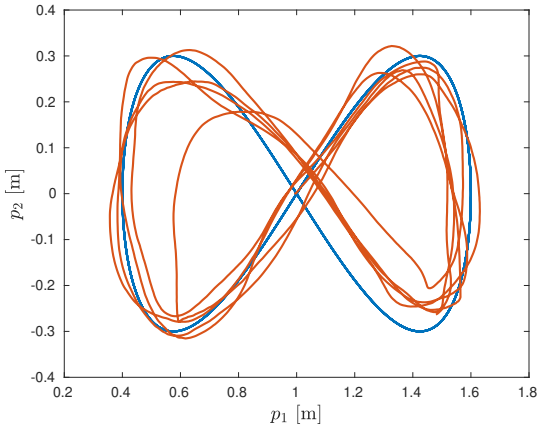


(b) BS.

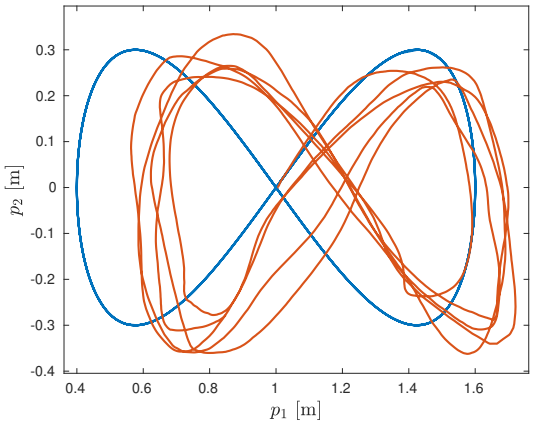


(c) DOB-BS.

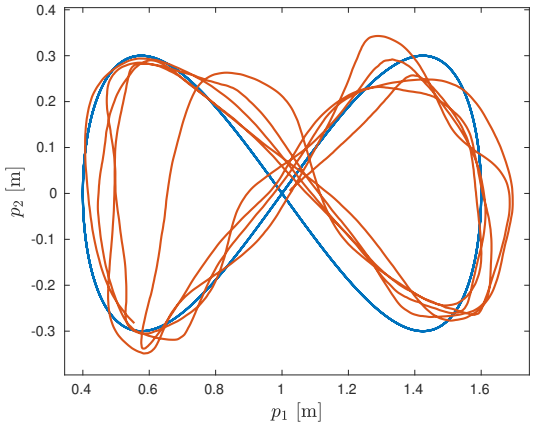
Figure 3.14: Experimental results: tracking error.



(a) PX4MC.



(b) BS.



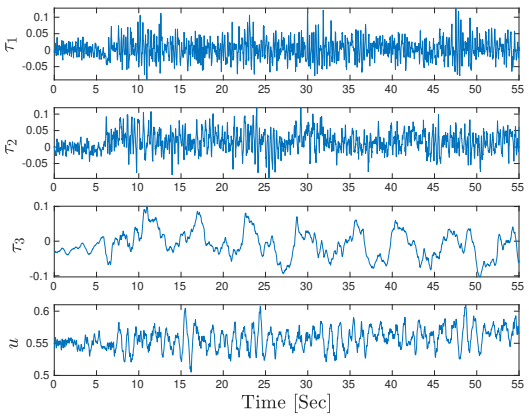
(c) DOB-BS.

Figure 3.15: Experimental results: p_1-p_2 plot.

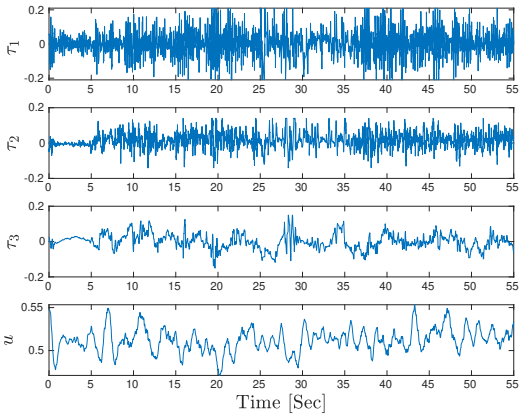
the DOB-BS controller provides better performance in the steady state compared with the BS and PX4MC controllers. We observe an offset about 20 cm in the n_1 direction for the tracking error of the BS controller. This offset is due to the applied force disturbance from the fan and since the BS control has no disturbance compensation. The force disturbance estimate is shown in Figure 3.17. The average steady state value of $\hat{d}_{f_1}$ is about 0.5 N. From Figure 3.16 we observe the torques are noisy which is mainly due to the noise in angular velocity measurements. The mean value and the root mean square error (RMSE) of the tracking error for the three methods are shown in Table 3.7. We observe the DOB-BS controller achieves better performance which is to be expected as it estimates and compensates for the disturbance force.

Figure 3.18 shows a picture of the experimental setup. A video of the experiment is provided at [1].

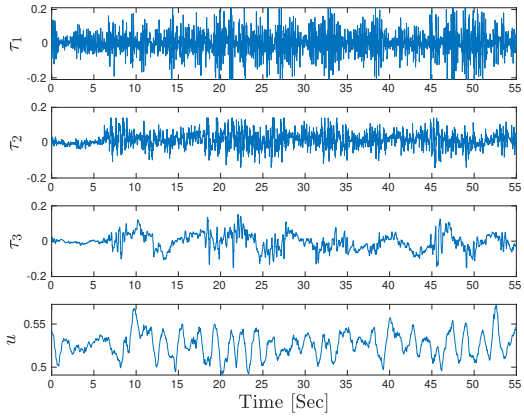
[1] https://youtu.be/-90w_FPmNWE



(a) PX4MC.



(b) BS.



(c) DOB-BS.

Figure 3.16: Experimental results: normalized control inputs.

Table 3.7: Experimental results: mean and RMSE of position tracking error.

Controller	$\tilde{p}_1$(mean)	$\tilde{p}_2$(mean)	$\tilde{p}_3$(mean)	$\tilde{p}_1$(RMSE)	$\tilde{p}_2$(RMSE)	$\tilde{p}_3$(RMSE)	$\|\tilde{p}\|$(RMSE)
PX4MC (m)	-0.0193	0.0037	-0.0337	0.2598	0.2036	0.0467	0.2810
BS (m)	0.1509	-0.0312	-0.0089	0.1796	0.0497	0.0475	0.1501
DOB-BS (m)	0.0433	-0.0101	-0.0010	0.0952	0.0380	0.0346	0.0668

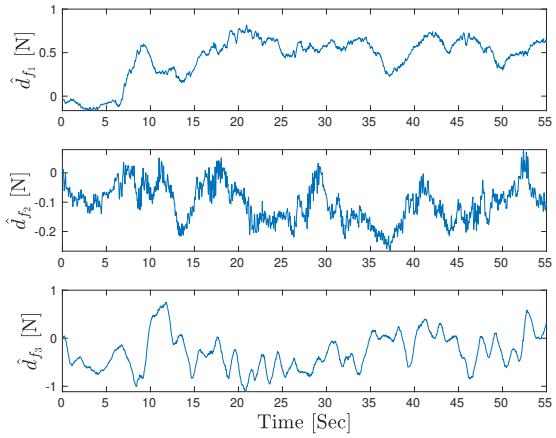


Figure 3.17: Experimental results: disturbance estimates.

Figure 3.18: Experimental setup.

3.3.5 Summary

In this section we proposed a nonlinear position and yaw tracking control which combines backstepping with a disturbance observer to increase its robustness to external force disturbances. The disturbance observer error dynamics is decoupled from the tracking error dynamics and the cascade structure of the closed-loop makes the stability analysis, implementation, and tuning straightforward. Exponential stability is proven for the constant disturbance case. The method is validated in SITL simulation and flight experiment using the commonly used PX4 autopilot firmware. The DOB-BS control is compared with a stock PX4 inner outer loop control and a traditional backstepping method. The DOB-BS control demonstrates improved tracking performance and robustness relative to these controllers.

3.4 Decoupled Controller-Observer Design with Integral Augmentation and Rotor Drag Effect

In this section we extend the work presented in Section 3.3 by adding integral augmentation and rotor drag modelling to the controller design. SITL simulation results are provided showing increased performance when the integral augmentation is added. Also the importance of rotor drag compensation is shown in simulation where a considerable performance deterioration is observed when rotor drag is ignored in the controller-observer design.

3.4.1 Position Tracking Control

We begin backstepping with integral augmentation by defining $\delta_1 = \int_{t_0}^{t}(p(\xi) - p_d(\xi))d\xi$ and the first Lyapunov function candidate $V_1 = \frac{1}{2}\|\delta_1\|^2$. This yields $\dot{V}_1 = \delta_1^\top \dot{\delta}_1 = \delta_1^\top (p - p_d)$. Taking p as the first virtual control and its desired value as $\alpha_1 = -k_1\delta_1 + p_d$, gives

$$\dot{V}_1 = \delta_1^\top(\delta_2 + \alpha_1 - p_d) = -k_1\|\delta_1\|^2 + \delta_1^\top\delta_2$$

where $\delta_2 = p - \alpha_1$. Now, defining the second Lyapunov function $V_2 = V_1 + \frac{1}{2}\|\delta_2\|^2$ gives $\dot{V}_2 = -k_1\|\delta_1\|^2 + \delta_1^\top\delta_2 + \delta_2^\top\dot{\delta}_2$, where $\dot{\delta}_2 = v + k_1(p - p_d) - \dot{p}_d$. Hence,

$$\dot{V}_2 = -k_1\|\delta_1\|^2 + \delta_1^\top\delta_2 + \delta_2^\top(v + k_1(p - p_d) - \dot{p}_d)$$

Denoting α_2 as the desired value for virtual control v and taking

$$\alpha_2 = -\delta_1 - k_1(p - p_d) + \dot{p}_d - k_2\delta_2$$

gives $\dot{V}_2 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top\delta_3/m$ where $\delta_3 = mv - m\alpha_2$. Next, taking $V_3 = V_2 + \frac{1}{2}\|\delta_3\|^2$ we get

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top\delta_3/m + \delta_3^\top\dot{\delta}_3$$

Using (2.9) and defining a new input as $u_s = u - d_{11}v_a^\top Rn_3$, which is the sum of the rotor thrust and drag force component in the direction of rotor thrust vector, we can write $\dot{\delta}_3$ as

$$\dot{\delta}_3 = mgn_3 - u_sRn_3 - d_{11}v_a + d_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) - m\ddot{p}_d + mk_2\dot{\delta}_2$$

where we have used

$$u_sRn_3 = uRn_3 - d_{11}v_a^\top Rn_3Rn_3. \tag{3.67}$$

Now, we consider u_sRn_3 as the next virtual control and take its desired value as

$$\alpha_3 = \delta_2/m + k_3\delta_3 + mgn_3 - d_{11}v_a + \hat{d}_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) - m\ddot{p}_d + mk_2\dot{\delta}_2$$

Note that we have replaced d_f appearing in $\dot{\delta}_3$ from α_3 by its estimate $\hat{d}_f$ from the disturbance observer (3.5). Defining $\delta_4 = \alpha_3 - u_s R n_3$ gives

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top \delta_4 + \tilde{d}_f^\top \delta_3$$

Defining $V_4 = V_3 + \frac{1}{2}\|\delta_4\|^2$ so that

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top \delta_4 + \tilde{d}_f^\top \delta_3 + \delta_4^\top \dot{\delta}_4 \tag{3.68}$$

where $\dot{\delta}_4 = \dot{\alpha}_3 - \dot{u}_s R n_3 - u_s R S(\omega) n_3$ and $\dot{\alpha}_3$ can be expressed as

$$\begin{aligned}
\dot{\alpha}_3 = {} & \dot{\delta}_2/m + k_3(mgn_3 - u_s R n_3 - d_{11} v_a + d_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) - m\ddot{p}_d + mk_2\dot{\delta}_2) \\
& - \frac{d_{11}}{m}(mgn_3 - u_s R n_3 - d_{11} v_a + d_f) + k_{d_f}\tilde{d}_f + m(\dot{p} - \dot{p}_d) \\
& + k_1(mgn_3 - u_s R n_3 - d_{11} v_a + d_f - m\ddot{p}_d) \\
& - m\dddot{p}_d + k_2(mgn_3 - u_s R n_3 - d_{11} v_a + d_f + mk_1(v - \dot{p}_d) - m\dot{v}_d)
\end{aligned} \tag{3.69}$$

We note the expression for $\dot{\alpha}_3$ in (3.69) depends on d_f, the unknown force disturbance. Therefore, $\dot{\alpha}_3$ is estimated by β whose expression is (3.69) with d_f replaced by its estimate. Hence, we obtain

$$\dot{\alpha}_3 = \beta + (k_1 + k_2 + k_3 + k_{d_f} - \frac{d_{11}}{m})\tilde{d}_f$$

where we get the following expression for $\dot{\delta}_4$

$$\dot{\delta}_4 = \beta + (k_1 + k_2 + k_3 + k_{d_f} - \frac{d_{11}}{m})\tilde{d}_f - \dot{u}_s R n_3 - u_s R S(\omega) n_3 \tag{3.70}$$

substituting (3.70) into (3.68) gives

$$\begin{aligned}
\dot{V}_4 = {} & -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top \delta_4 + \tilde{d}_f^\top(\delta_3 + (k_1 + k_2 + k_3 + k_{d_f} - \frac{d_{11}}{m})\delta_4) \\
& + \delta_4^\top(\beta - \dot{u}_s R n_3 - u_s R S(\omega) n_3)
\end{aligned} \tag{3.71}$$

Now, we assign expressions for $\dot{u}_s$ and $u_s R S(\omega) n_3$ to make $\dot{V}_4$ negative definite. At this stage of the backstepping, we cancel β, the indefinite term $\delta_3^\top \delta_4$, and add a damping term $-k_4\|\delta_4\|^2$. Hence, we define the desired value for $\dot{u}_s$ as

$$\dot{u}_{sd} = n_3^\top R^\top(\beta + \delta_3 + k_4\delta_4) \tag{3.72}$$

and also we take $\alpha_4 = R[I - n_3 n_3^\top]R^\top(\beta + \delta_3 + k_4\delta_4)$ as the desired value for the virtual control $u_s R S(\omega) n_3$. From the definition of u_s ($u_s = u - d_{11} v_a^\top R n_3$) and using the assumption of constant

wind velocity $\dot{w} = 0$, the following expression for $\dot{u}$ can make $\dot{u}_s$ equal to its desired value in (3.72):

$$\dot{u} = n_3^\top R^\top(\beta + \delta_3 + k_4\delta_4) + \frac{d_{11}}{m}(mgn_3 - u_s Rn_3 - d_{11}v_a + d_f)^\top Rn_3 + d_{11}v_a^\top RS(\omega)n_3 \tag{3.73}$$

However, since d_f is unknown we replace it with its estimate in (3.73) and this leads to an extra term proportional to disturbance estimation error in (3.73). That is,

$$\dot{u}_s = n_3^\top R^\top(\beta + \delta_3 + k_4\delta_4) - \frac{d_{11}}{m}\tilde{d}_f^\top Rn_3 \tag{3.74}$$

Now by defining a new error coordinates as $\delta_5 = \alpha_4 - u_s RS(\omega)n_3$ and using the equality $\delta_4^\top \tilde{d}_f^\top Rn_3 Rn_3 = \tilde{d}_f^\top (Rn_3)^\top Rn_3\delta_4$ we get

$$\begin{aligned}\dot{V}_4 = &-k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 \\ &+ \tilde{d}_f^\top(\delta_3 + (k_1 + k_2 + k_3 + k_{d_f} - \frac{d_{11}}{m} + \frac{d_{11}}{m}(Rn_3)^\top Rn_3)\delta_4) + \delta_4^\top\delta_5\end{aligned}$$

Now by choosing $V_5 = V_4 + \frac{1}{2}\|\delta_5\|^2$,

$$\begin{aligned}\dot{V}_5 = &-k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 \\ &+ \tilde{d}_f^\top(\delta_3 + (k_1 + k_2 + k_3 + k_{d_f} - \frac{d_{11}}{m} + \frac{d_{11}}{m}(Rn_3)^\top Rn_3)\delta_4) + \delta_4^\top\delta_5 + \delta_5^\top\dot{\delta}_5\end{aligned}$$

where $\dot{\delta}_5 = \dot{\alpha}_4 - \dot{u}_s RS(\omega)n_3 - u_s RS(\omega)^2 n_3 - u_s RS(\dot{\omega})n_3$ and $\dot{\alpha}_4$ is given by

$$\begin{aligned}\dot{\alpha}_4 = &RS(\omega)[I - n_3n_3^\top]R^\top(\beta + \delta_3 + k_4\delta_4) + R[I - n_3n_3^\top]S(\omega)^\top R^\top(\beta + \delta_3 + k_4\delta_4) \\ &+ R[I - n_3n_3^\top]R^\top(\dot{\beta} + \dot{\delta}_3 + k_4\dot{\delta}_4).\end{aligned} \tag{3.75}$$

Similar to Section 3.3 the expression for $\dot{\alpha}_4$ is unknown due to the presence of d_f, and $\tilde{d}_f$ which appears in $\dot{\beta}, \dot{\delta}_3$, and $\dot{\delta}_4$. However, the expressions for these quantities can be estimated by replacing d_f by its estimate. We denote these "estimated" quantities as $\dot{\beta}'$, $\dot{\delta}_3'$ and $\dot{\delta}_4'$. Since these expressions are complicated, they are omitted here. Using the expressions for the estimates $\dot{\beta}'$, $\dot{\delta}_3'$ and $\dot{\delta}_4'$ in (3.75) we can obtain an estimate of $\dot{\alpha}_4$ which is denoted $\dot{\alpha}_4'$. We have

$$\dot{\alpha}_4 = \dot{\alpha}_4' + KR[I - n_3n_3^\top]R^\top\tilde{d}_f$$

where K is a constant matrix depending on controller and observer gains; and the system parameters m and d_{11}. At this stage, we try to define a desired value for the rate of angular velocity $\dot{\omega}_d$ so that we have

$$\delta_5^\top(\dot{\alpha}_4' - \dot{u}_{sd}RS(\omega)n_3 - u_s RS(\omega)^2 n_3 - u_s RS(\dot{\omega}_d)n_3) = -k_5\delta_5^\top\delta_5 - \delta_5^\top\delta_4 \tag{3.76}$$

Note that we used $\dot{u}_{sd}$ since $\dot{u}_s$ is unknown. Now, considering the structure for $S(\dot{\omega}_d)n_3$ and using the fact that the third component of $R^\top \delta_5$ is zero, we can conclude that by selecting

$$\dot{\omega}_{d1} = \frac{-n_2^\top R^\top}{u_s}(\dot{\alpha}_4' - \dot{u}_{sd}RS(\omega)n_3 - u_sRS(\omega)^2 n_3 + \delta_4 + k_5\delta_5) \tag{3.77a}$$

$$\dot{\omega}_{d2} = \frac{n_1^\top R^\top}{u_s}(\dot{\alpha}_4' - \dot{u}_{sd}RS(\omega)n_3 - u_sRS(\omega)^2 n_3 + \delta_4 + k_5\delta_5) \tag{3.77b}$$

then (3.76) will be satisfied. In the above equation $k_5 > 0$ is a controller gain. Replacing (3.76) in $\dot{V}_5$ and using $\dot{\omega} = \dot{\omega}_d + J^{-1}\tilde{d}_\tau$ and $\dot{u}_s = \dot{u}_{sd} - \frac{d_{11}}{m}\tilde{d}_f^\top Rn_3$ gives us

$$\begin{aligned}\dot{V}_5 = &-k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_5\|\delta_5\|^2 \\ &+ \tilde{d}_f^\top(\delta_3 + (k_1 + k_2 + k_3 + k_{d_f} - \frac{d_{11}}{m} + \frac{d_{11}}{m}(Rn_3)^\top Rn_3)\delta_4 + (KR[I - n_3n_3^\top]R^\top \\ &+ \frac{d_{11}}{m}(Rn_3)^\top RS(\omega)n_3)\delta_5) + \delta_5^\top u_sRS(n_3)J^{-1}\tilde{d}_\tau\end{aligned}$$

Again we take $\dot{\omega}_{d3} = 0$ in order to obtain the expression for τ using the equation

$$\tau = J\dot{\omega}_d + \omega \times J\omega - \hat{d}_\tau. \tag{3.78}$$

We remark that since the yaw dynamics is independent from the position dynamics, the controller proposed in Section 3.3.2 can be used for the yaw control. The closed-loop exponential stability of the proposed backstepping controller is summarized with the following theorem.

Theorem 3.5. *Consider the closed-loop (2.3) with dynamic state feedback control (3.3), (3.7), (3.73), (3.77) and (3.78). Assuming constant disturbances d_f and d_τ, bounded smooth reference trajectory p_d, bounded wind velocity w, $u_s \geq \epsilon$, for some $\epsilon > 0$ the equilibrium of the closed-loop system $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \delta_5^\top, \tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top = 0$ is exponentially stable.*

Proof. By defining $\delta = [\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \delta_5^\top]^\top$ and $\tilde{d} = [\tilde{d}_f^\top, \tilde{d}_\tau^\top]^\top$ the tracking error dynamics is

$$\dot{\delta} = A\delta + B(\delta, t)\tilde{d} \tag{3.79}$$

where

$$A = \begin{bmatrix} -k_1I_{3\times3} & I_{3\times3}/m & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ -I_{3\times3}/m & -k_2I_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -I_{3\times3} & -k_3I_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & -k_4I_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & -k_5I_{3\times3} \end{bmatrix}$$

$$B(\delta,t)=\begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \\ I_{3\times3} & 0_{3\times3} \\ (k_1+k_2+k_3+k_{d_f}-\frac{d_{11}}{m}+\frac{d_{11}}{m}(Rn_3)^\top Rn_3)I_{3\times3} & 0_{3\times3} \\ KR[I-n_3n_3^\top]R^\top+\frac{d_{11}}{m}(Rn_3)^\top RS(\omega)n_3 & u_sRS(n_3)J^{-1} \end{bmatrix}$$

We remark that the term $B(\delta,t)\tilde{d}$ that interconnects the δ and $\tilde{d}$ subsystems is time-varying due to the tracking problem considered. We apply the exponential stability result for cascades in [104, Prop. 2.3]. Clearly, the disturbance estimate error dynamics is LTI and exponentially stable. As well, assuming "zero input", i.e., $\tilde{d}=0$, the dynamics (3.79) are LTI and exponentially stable for any $k_1,k_2,k_3,k_4,k_5>0$. Hence, it only remains to show

$$\|B(\delta,t)\|\leq C_1+C_2\|\delta\| \tag{3.80}$$

for some $C_1,C_2>0$. Since $\delta_4=\alpha_3-u_sRn_3$, we can deduce $|u_s|\leq\|\alpha_3\|+\|\delta_4\|$, and from $\alpha_3=\delta_2/m+k_3\delta_3+mgn_3-d_{11}v_a+\hat{d}_f+m\dot{\delta}_1+mk_1(v-\dot{p}_d)-m\ddot{p}_d+mk_2\dot{\delta}_2$, and by replacing $\dot{\delta}_2=v+k_1(p-p_d)-\dot{p}_d$, $v_a=v-w$, $\hat{d}_f=d_f-\tilde{d}_f$, $v=\frac{1}{m}(\delta_3+m\alpha_2)$, $\alpha_2=-\delta_1-k_1(p-p_d)+\dot{p}_d-k_2\delta_2$, $\dot{\delta}_1=p-p_d$ and $p=\delta_2-k_1\delta_1+p_d$, we get

$$\begin{aligned}\alpha_3=\delta_2/m+k_3\delta_3+mgn_3-d_{11}(\frac{1}{m}(\delta_3+m(-\delta_1-k_1(\delta_2-k_1\delta_1)+\dot{p}_d-k_2\delta_2))-w)+d_f-\tilde{d}_f\\ +m(\delta_2-k_1\delta_1)+mk_1(\frac{1}{m}(\delta_3+m(-\delta_1-k_1(\delta_2-k_1\delta_1)+\dot{p}_d-k_2\delta_2))-\dot{p}_d)-m\ddot{p}_d\\ +mk_2(\frac{1}{m}(\delta_3+m(-\delta_1-k_1(\delta_2-k_1\delta_1)+\dot{p}_d-k_2\delta_2))+k_1(\delta_2-k_1\delta_1)-\dot{p}_d)\end{aligned}$$

therefore, we can obtain an upper bound $\|\alpha_3\|\leq\bar{C}_1\|\delta\|+\bar{C}_2$, $\bar{C}_1,\bar{C}_2>0$. Now we can deduce

$$|u_s|\leq\tilde{C}_1\|\delta\|+\tilde{C}_2,\quad \tilde{C}_1,\tilde{C}_2>0$$

Next, we obtain a bound for $\|RS(\omega)n_3\|$. From $\delta_5=\alpha_4-u_sRS(\omega)n_3$, we can deduce $\|RS(\omega)n_3\|\leq\frac{1}{\epsilon}(\|\alpha_4\|+\|\delta_5\|)$, and from $\alpha_4=R[I-n_3n_3^\top]R^\top(\beta+\delta_3+k_4\delta_4)$ and the relation for β we can find a linear bound for $\|RS(\omega)n_3\|$ for some $\tilde{C}_3,\tilde{C}_4>0$

$$\|RS(\omega)n_3\|\leq\tilde{C}_3\|\delta\|+\tilde{C}_4,$$

Hence, we conclude all entries of B are linearly bounded in the norm of δ and we obtain (3.80).

Therefore, the conditions for [104, Prop. 2.3] are satisfied and the equilibrium of the closed-loop system (2.3), (3.3), (3.7), (3.73), (3.77) and (3.78) $[\delta_1^\top,\delta_2^\top,\delta_3^\top,\delta_4^\top,\delta_5^\top,\tilde{d}_f^\top,\tilde{d}_\tau^\top]^\top=0$ is exponentially stable.

□

3.4.2 SITL Simulation Results

The reference trajectory is taken as a circle given by $p_d(t) = [3\sin(\frac{2\pi t}{T}), 3\cos(\frac{2\pi t}{T}), -1.1]^\top$ m. The velocity of the trajectory is increased by decreasing T from 20 at $t = 0$ to 10 in 10 seconds, reaching a maximum speed of 1.8850 m/s which ensures disturbances and drag forces are sufficiently large. We use the yaw control in [88] with a zero yaw setpoint. The controller and observer gains are selected as $k_1 = k_2 = k_3 = k_4 = k_5 = k_{d_f} = k_{d_\tau} = 2$. The vehicle is initialized in hover at $p(0) = [0.0, 0.0, -0.85]^\top$ m. Hence, a relatively large initial error exists when the controller is switched on, i.e., $\tilde{p}(0) = [0.0, -3.0, -0.15]^\top$ m.

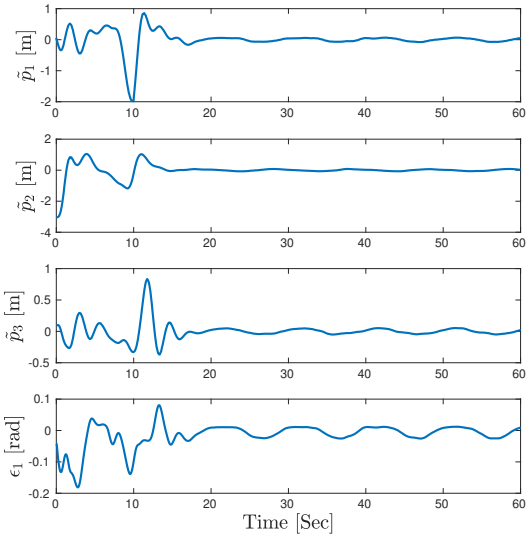


Figure 3.19: Position tracking error for the DOB-IBS controller

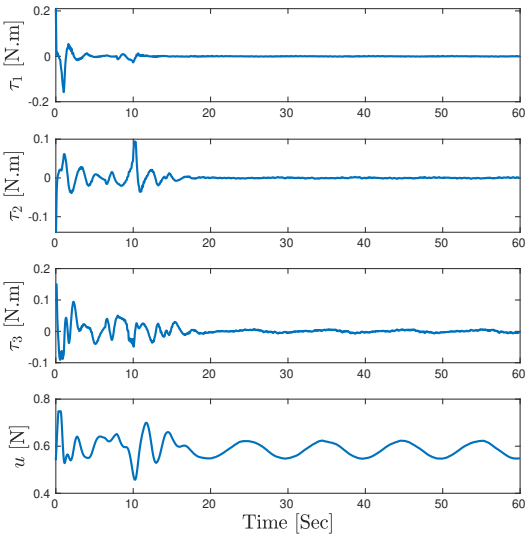


Figure 3.20: Control inputs for the DOB-IBS controller

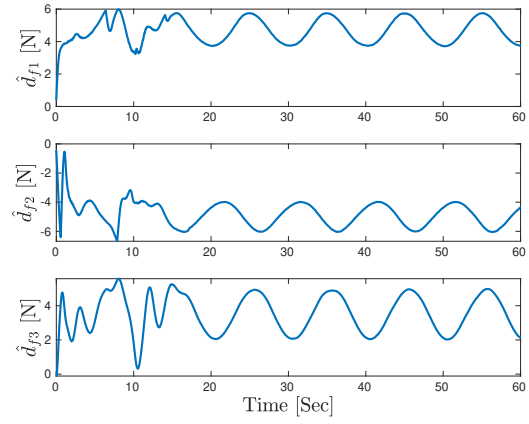


Figure 3.21: Force disturbance estimates for the DOB-IBS controller.

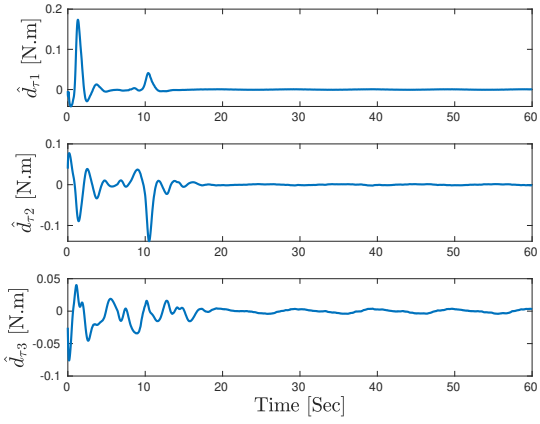


Figure 3.22: Torque disturbance estimates for the DOB-IBS controller.

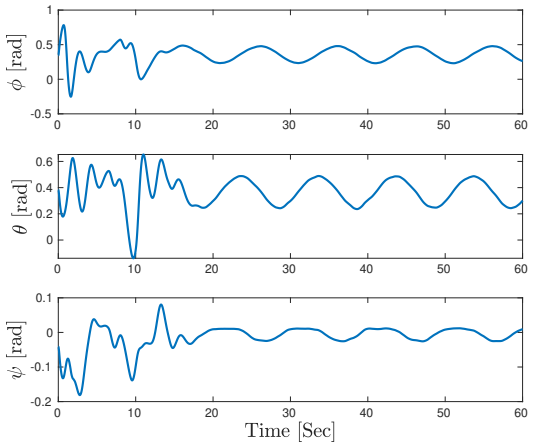


Figure 3.23: Euler angles for the DOB-IBS controller

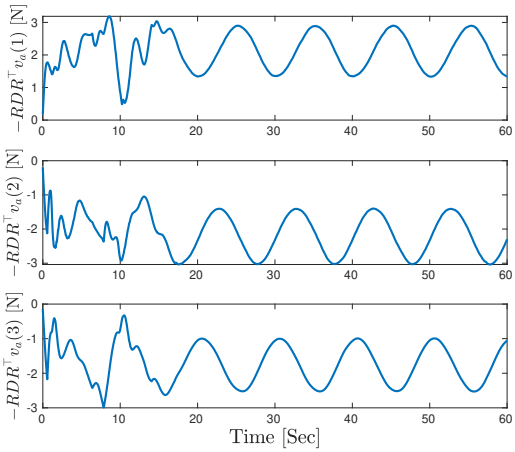


Figure 3.24: Drag force $-RDR^\top v_a$.

Simulation results for the proposed control (referred to as DOB-IBS) are shown in Figures 3.19–3.24. We observe the tracking error in Figure 3.19 converges to an acceptable neighborhood of the origin (i.e., $\|\tilde{p}\| \approx 5\,\text{cm}$ in steady-state) in about 20 s and remains there. The normalized torques and thrust are in Figure 3.20 and have reasonable magnitudes. The force and torque disturbance estimates are shown in Figure 3.21 and Figure 3.22. We observe the average steady-state of $\hat{d}_{f1}$ converges to a value close to 5 N and has oscillations at the frequency of the reference trajectory. The average value is consistent with the applied wind disturbance in the n_1 direction. Similarly, the steady-state average value of $\hat{d}_{f2}$ is -5 N and near its expected value. The steady-state average of $\hat{d}_{f3}$ is about 5 N which is due to error in thrust scaling required to convert force to normalized input in the controller. From Figure 3.22 it can be seen that the torque disturbance estimates are small and quickly converge to zero. This is consistent with equation (2.19) as the angular velocity ω is close to zero for the desired trajectory. Plots for the Euler angles are shown in Figure 3.23. Figure 3.24 shows the rotor drag force $-RDR^\top v_a$ has a significant amplitude.

To show the effect of compensating the drag force $-RDR^\top v_a$ in the controller design, we tested the DOB-IBS controller without drag force compensation. The 2D plots of the desired trajectory and DOB-IBS controller with and without drag compensation are shown in Figure 3.25. The compensation leads to noticeable tracking performance improvement.

In order to investigate the effect of the disturbance observer and integral augmentation on performance, the DOB-IBS control is compared with a number of related control laws: Backstepping (BS), Disturbance Observer-based Backstepping (DOB-BS), Integral Backstepping (IBS). The simulation results for the actual and desired trajectories are shown in 2D for all controllers and in steady-state, $20 \leq t \leq 40$ s, in Figure 3.27. Table 3.8 provides the root mean square (RMS) of the position error in steady-state. The trajectory for the norm of the tracking error is in Figure 3.26. We remark that all controllers exhibit some level of oscillation at the frequency of the reference input. In theory the error for the DOB-IBS should converge to zero. However, jMAVSim includes

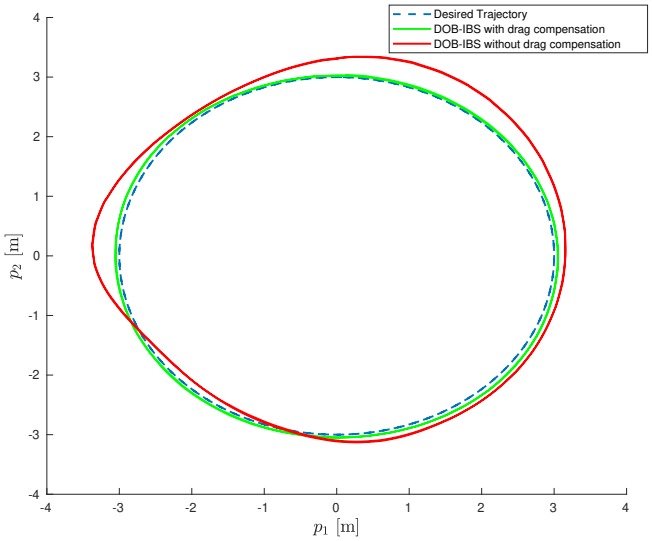


Figure 3.25: 2D plots of the desired Trajectory, DOB-IBS controller with drag compensation and DOB-IBS controller without drag compensation

a number of unmodelled effects which explain the lack of convergence: rotor inertia, sensor noise, and error in thrust scaling. We observe from Table 3.8 that the DOB-IBS control achieves the best performance. The 2D plots clearly reflect this reduced error. The BS control suffers from a lack of robustness to the wind forces and has the largest steady-statetate error. Adding the disturbance observer (DOB-BS) or the integral term (IBS) improves robustness, but not to the level of the DOB-IBS control. Controller gains could be adjusted to improve steady-state response of the other controllers. However, gains cannot be found to match the size of the DOB-IBS error and smaller error leads to reduced stability margin.

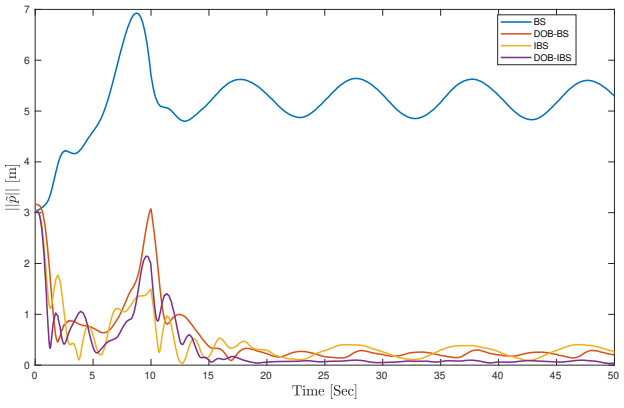


Figure 3.26: Norm of trajectory tracking error

Next, we perform a simulation to investigate the effect of parametric model error and time-varying disturbance. To introduce parameter error, we $m = 1.2$kg (25% parameter error) and

Table 3.8: Root mean square (RMS) of the steady-state position tracking error

Controller	BS	DOB-BS	IBS	DOB-IBS	DOB-IBS (no drag)
RMS ($\|\tilde{p}\|$) [m]	5.2454	0.2201	0.2830	0.0743	0.2959

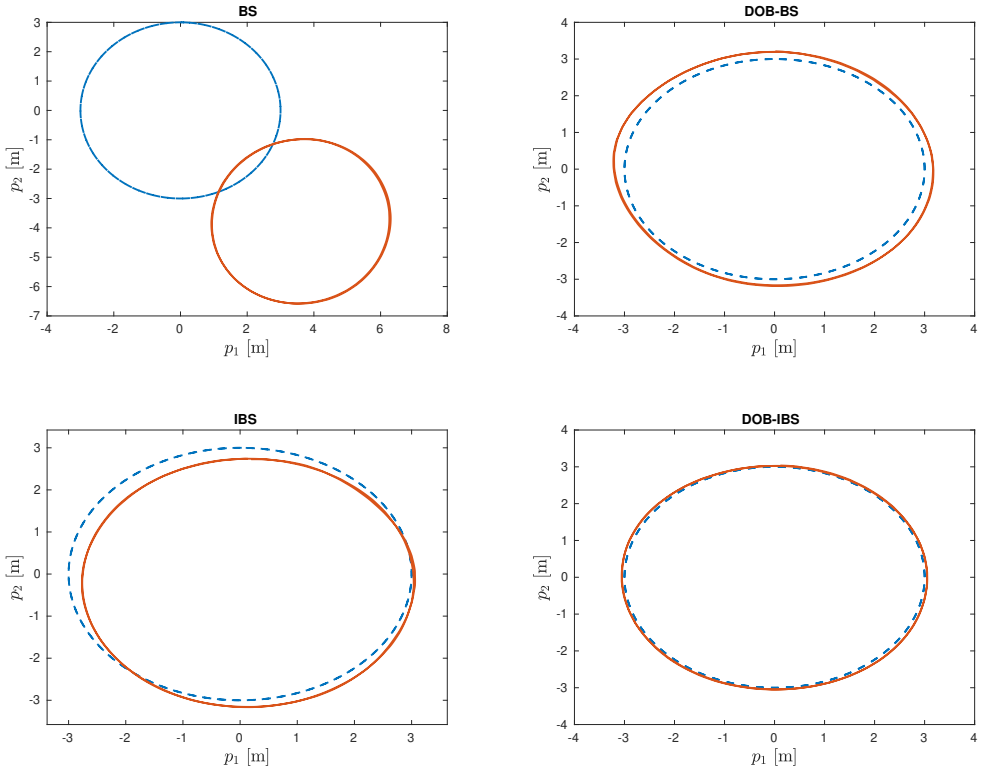


Figure 3.27: Desired and actual trajectories

$J = \text{diag}(0.02, 0.02, 0.03)\text{kg m}^2$ in the control law. We introduce a time-varying wind velocity whose n_1 and n_2 components vary on $[-10, 10]$ m/s. All the other conditions of the simulation are the same as above. The wind velocity and position tracking errors can be seen in Figures 3.28-3.29. The disturbance estimates are given in Figures 3.30-3.31. We observe that even with a significantly time-varying and large wind velocity, the tracking errors are ultimately bounded to a practically small neighbourhood of the origin. The RMS of the steady-state error is 0.2023 m which is acceptable given the simulation conditions. From Figure 3.30 we observe that the force disturbance estimate has a similar shape to the wind velocity as the wind velocity in the n_1 and n_2. This implies the disturbance observer is tracking the low frequency components of the disturbance force. There is a 5 N offset in $\hat{d}_{f_3}$ which corresponds to the 0.4 kg error in vehicle mass. Hence, the disturbance observer is able to compensate for parametric uncertainty.

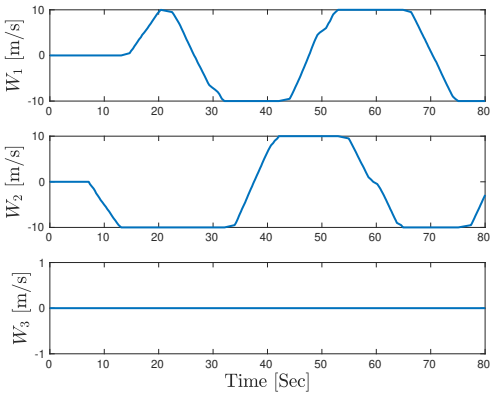


Figure 3.28: Wind velocity.

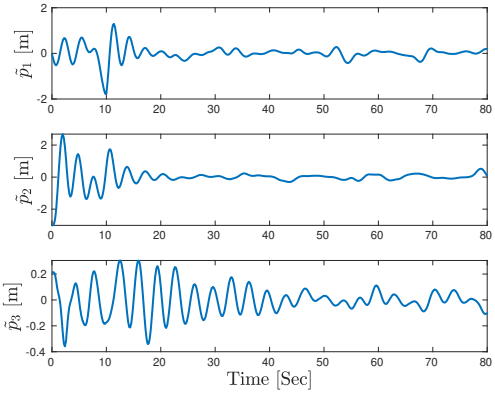


Figure 3.29: Position error for the case of time-varying wind and model uncertainty.

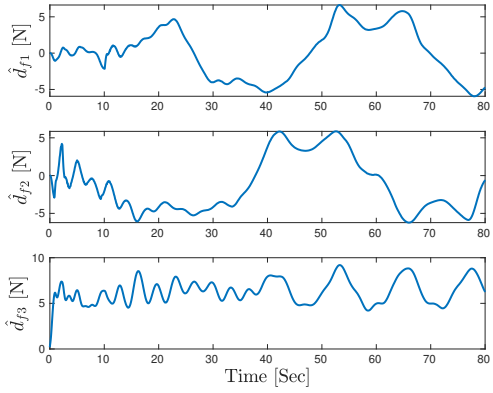


Figure 3.30: Force disturbance estimates for the case of time-varying wind and model uncertainty.

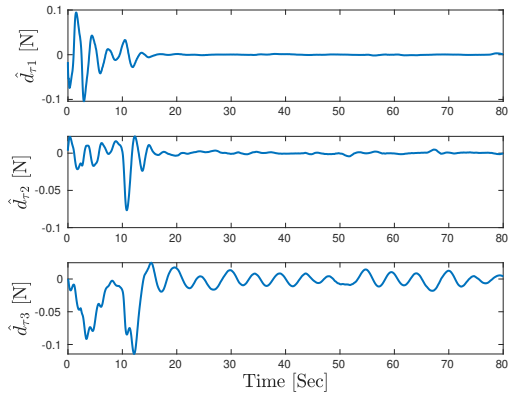


Figure 3.31: Torque disturbance estimates for the case of time-varying wind and model uncertainty.

3.4.3 Summary

In this section a disturbance observer-based integral backstepping controller was proposed for trajectory tracking of multirotor UAVs. We included the rotor drag in the translational dynamics. The drag effect is compensated by decomposing it into two components: the first one is independent of the vehicle attitude. This component can be readily compensated by an appropriate choice of virtual control in the backstepping procedure. The second component of the drag force is in the direction of thrust and can be compensated by an appropriate choice for desired thrust during backstepping. Exponential stability of the closed loop is proven assuming constant wind velocity and constant disturbances. SITL simulations demonstrate improved controller performance of the method, and the importance of rotor drag compensation. The simulations illustrate the approach can compensate for time-varying disturbances and parametric uncertainty. Self-contained code for the PX4 SITL simulation is available online at [93].

Chapter 4

Visual-Inertial-Actuator Odometry

The motion control algorithms presented in the previous chapter are state feedback laws which require knowledge about the system states. A combination of camera and IMU is considered to be an ideal choice of sensors as they are normally onboard UAV and possess complementary characteristics. In some UAV applications, external force estimation is necessary in order to improve the motion control performance or perform specified tasks where UAV is interacting with the environment. Force estimation is possible by using the vehicle's dynamics and the actuator inputs which are either the rotor thrust and torque or the rotor speeds. In addition to providing force estimation, including the vehicle dynamics and input can increase state estimation accuracy.

This chapter presents a sliding window optimization-based method for the problem of multirotor UAV VIAO. Inertial measurements, visual data and actuator information are used in conjunction with the vehicle's dynamic model. The effect of rotor drag is included in the vehicle's modelling. Also, the assumption found in [27] that the external disturbance force is zero mean white noise is replaced by a more realistic Gaussian white noise plus a random walk process. Furthermore, the force disturbance observer proposed in 3.1 which was designed for the constant case is formulated as a residual and added to the optimization.

The organization of this chapter is as follows: in Section 4.1 the vehicle's state is introduced and vision measurements, IMU measurements, noise models and estimator initialization are discussed. Section 4.2 presents for tightly-coupled optimization framework. A cost function is introduced which is a sum of norm-squared of so-called residual terms. The inertial residual is derived in Section 4.3 and propagation formulas are given. Section 4.4 discusses how the vehicle dynamics including rotor drag is integrated into a residual A disturbance observer is presented to estimate slowly time-varying force disturbances. The simulation results for evaluating the proposed method are discussed in Section 4.6. Finally, results are summarized in Section 4.7.

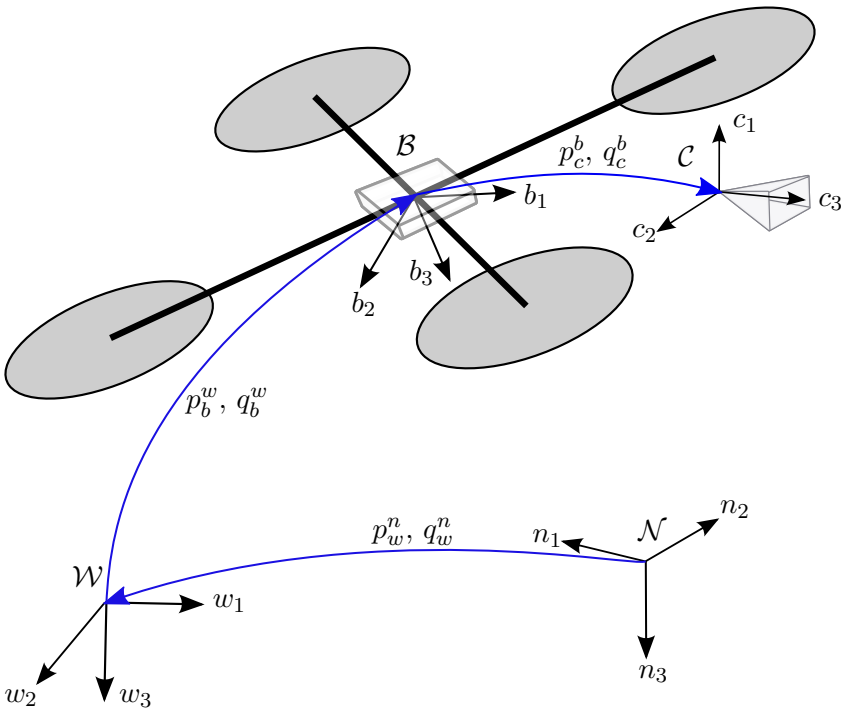


Figure 4.1: Multirotor indicated with body $\mathcal{B}$, camera $\mathcal{C}$, world $\mathcal{W}$ and $\mathcal{N}$ navigation frames.

4.1 Preliminaries

We assume three coordinate frames as depicted in Figure 4.1: a body frame $\mathcal{B}$ which is assumed fixed and aligned with the IMU frame, a camera frame $\mathcal{C}$ which is fixed with respect to the body frame and a world frame $\mathcal{W}$ as the inertial frame. As discussed later, the world frame is established in the initialization step of the VIAO and the location of its origin is set as the first camera frame in the initialization sliding window with its w_3 axis aligned with the direction of the gravity. There is no control over the directions of the w_1 and w_2 axes as they depend on the direction of the first camera frame. For these reasons we distinguish between the world frame established by the VIAO system and the navigation frame used in the previous chapters which is normally coming from the MCS. Also, we define b_k and c_k respectively as the body and camera frames at time t_k, which is the time when the kth keyframe is taken. With these definitions the UAV state at time t_k is defined by

$$x_k = [{p_k}^\top, {v_k}^\top, {q_{b_k}^w}^\top, {b_{a_k}}^\top, {b_{\omega_k}}^\top]^\top \tag{4.1}$$

where $p_k \in \mathbb{R}^3$ is the position of the origin of $\mathcal{B}$ relative to the origin of $\mathcal{W}$ expressed in $\mathcal{W}$, $v_k \in \mathbb{R}^3$ is the linear velocity of $\mathcal{B}$ with respect to $\mathcal{W}$, expressed in $\mathcal{W}$, $q_{b_k}^w$ is the rotation of $\mathcal{B}$ with respect to $\mathcal{W}$ expressed as a quaternion, and $b_{a_k} \in \mathbb{R}^3$, $b_{\omega_k} \in \mathbb{R}^3$ are accelerometer and gyroscope sensor biases, respectively. Although the rotation is represented by quaternion, whenever necessary we use the rotation matrix for vector transformation. Specifically the rotation matrix is used when a vector needs to be transformed between body and world frames. We assume that the rigid transformation

from camera frame to body frame as (p_c^b, R_c^b) and assume it is known through calibration. This can be calculated by several available toolbox [105].

For the feature points we use inverse depth parametrization [106] which is a direct parametrization of the inverse depth of point features relative to the camera locations where they were first observed. This parametrization is known to generate measurement models with high degree of linearity and also suitable for far features which show low parallax during camera motion. The inverse depth parametrization for the sth feature is denoted by a 6-dimensional vector

$$l_s = [p_{cj}^\top, \theta_s, \phi_s, r_s]^\top$$

where p_{cj} is the location of the camera frame at jth frame (where the feature received its first observation), θ_s and ϕ_s are respectively azimuth and elevation angles for the sth feature in the camera frame and r_s is the inverted distance of the sth feature to the camera frame. The relation between the vector l_s and the Euclidean parametrization of the feature point, represented by $[X_s, Y_s, Z_s]^\top$, is

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = p_{cj} + \frac{1}{r_s} \mathrm{m}(\theta_s, \phi_s) \tag{4.2}$$

with

$$\mathrm{m}(\theta_s, \phi_s) = \begin{bmatrix} \cos\phi_s \sin\theta_s \\ -\sin\phi_s \\ \cos\phi_s \cos\theta_s \end{bmatrix} \tag{4.3}$$

4.1.1 Vision Measurements Processing

For every new image, which is denoted by $\bar{c}$, new features are detected and existing features are tracked by a feature tracking algorithm (KLT sparse optical flow in [26]). Normally, the algorithm enforces a minimum number of features for each image and also a minimum pixel distance between two adjacent features. The keyframe selection is performed at this stage. Keyframes are special image frames that are selected when there exists a large difference between the current frame and the previously last keyframe. This keyframe selection leads to efficient use of computational resources as we can discard the frames that provide less information. The average parallax from the previous keyframe (if greater than a certain threshold) and number of tracked features (if less than a certain threshold) are the two criteria for selecting the keyframes. These criteria guarantee that that there exist enough differences between the keyframes and therefore the computational resources are used efficiently by discarding the frames that provide less important information.

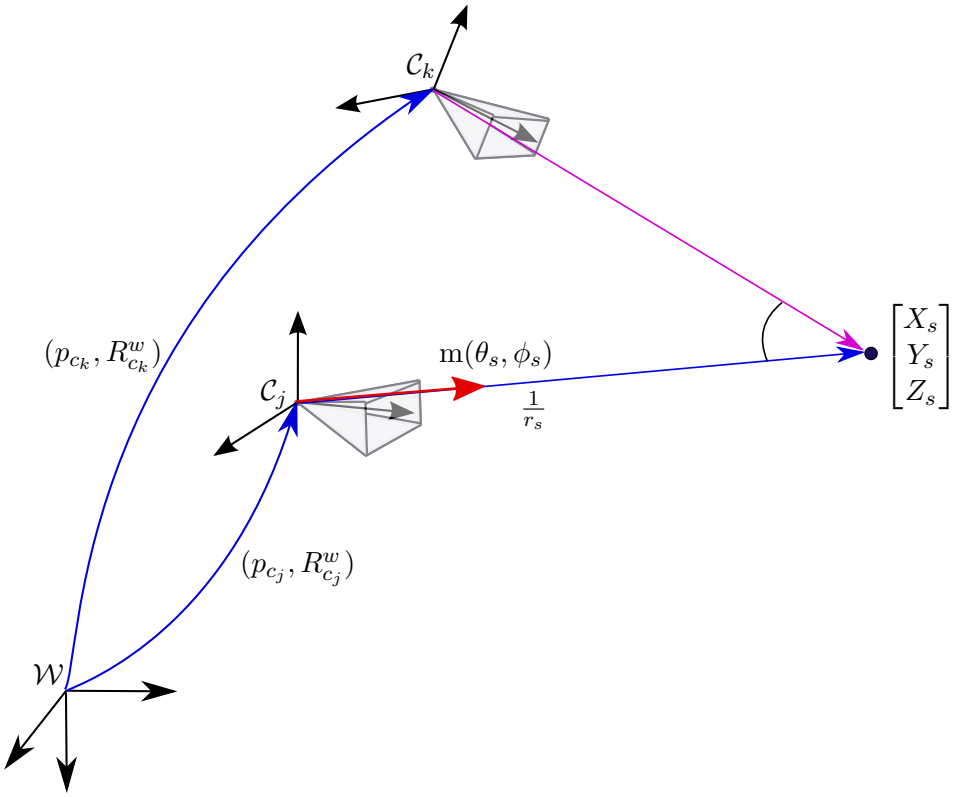


Figure 4.2: Inverse depth parametrization of a feature which received its first observation in jth camera frame.

4.1.2 IMU Measurements

The IMU normally consists of a three-axis gyro and a three-axis accelerometer which respectively measure the angular velocity and the acceleration in the body frame with respect to an inertial frame. The measurements of the gyro and accelerometer are respectively, $\bar{\omega}^b = \omega^b + b_\omega + \eta_\omega$ and $\bar{a}^b = a^b - R^b_w g + b_a + \eta_a$, where ω^b is the angular velocity of $\mathcal{B}$ with respect to $\mathcal{W}$, expressed in $\mathcal{B}$. Also, a^b is the acceleration of the sensor expressed in $\mathcal{B}$ and $g = [0, 0, 9.81]^\top$ m/s^2 is the gravity vector in $\mathcal{W}$. As previously defined b_a and b_ω are unknown slowly time-varying sensor biases whose derivatives are zero mean Gaussian noise, $\dot{b}_a = \eta_{b_a}$ and $\dot{b}_\omega = \eta_{b_\omega}$, with $\eta_{b_a} \sim \mathcal{N}(0, \sigma^2_{b_a} I_3)$ and $\eta_{b_\omega} \sim \mathcal{N}(0, \sigma^2_{b_\omega} I_3)$, where $\mathcal{N}(\bar{\mu}, \sigma^2 I_3)$ denotes a Gaussian distribution with mean as μ and σI_3 as covariance. Also, η_ω and η_a are measurement noise, $\eta_\omega \sim \mathcal{N}(0, \sigma^2_\omega I_3)$, $\eta_a \sim \mathcal{N}(0, \sigma^2_a I_3)$. With these definitions the accelerometer measures the specific force $(a^b - R^b_w g)$ in the ideal case ($b_a = 0$ and $\eta_a = 0$). The kinematic model of the body frame is as follows

$$\dot{p} = v \tag{4.4a}$$

$$\dot{v} = a = R^w_b a^b = R^w_b(\bar{a}^b + R^b_w g - b_a - \eta_a) = R^w_b(\bar{a}^b - b_a - \eta_a) + g \tag{4.4b}$$

$$\dot{q}^w_b = \frac{1}{2} q^w_b \otimes \begin{bmatrix} 0 \\ \bar{\omega}^b - b_\omega - \eta_\omega \end{bmatrix} \tag{4.4c}$$

Since the IMU frequency is much higher than the camera output, in order to reduce the computational cost, the IMU measurements are normally integrated between two consecutive image frames and a single constraint is formed. The formulation for deriving such a constraint is explained in detail in Section 4.3.

4.1.3 Estimator Initialization

Since the tightly coupled visual-inertial optimization is a highly nonlinear problem, a relatively accurate initial value for the state estimate is required for the optimization to converge to the global minimum. This is normally addressed by a loosely-coupled alignment of IMU preintegration with the vision-only up-to-scale pose and feature points estimation. In [26] an up-to-scale translation and relative rotation between the two frames in the sliding window (the frames that have the largest parallax) is derived using the five-point algorithm [107] and all the features seen in these two frames are triangulated. Then a perspective-n-point (PnP) method [108] is applied to find the poses of all other frames in the window. With the assumption of known camera to the body frame transformation, all poses from the camera frame are translated to the body frame. Using the constraints derived from the integration of the IMU measurements between the frames, a loosely-coupled linear least-square problem is constructed to solve for linear velocities, gravity and the scale. The world frame is selected as the first camera frame with the z axis aligned with the direction of the gravity. After setting the world frame, all variables are translated to the world frame and a sufficiently accurate initial estimate is available for the tightly-coupled VIO.

4.2 Tightly Coupled VIA Odometry

We assume an optimization window of n recent keyframes plus the current frame (which might not necessarily be a keyframe). The optimization variables consist of

$$\chi = [{x_0}^\top, {d^b_{f_0}}^\top, {x_1}^\top, {d^b_{f_1}}^\top, ..., {x_{n-1}}^\top, {d^b_{f_{n-1}}}^\top, {x_n}^\top, {l_1}^\top, ..., {l_m}^\top]^\top \tag{4.5}$$

where $d^b_{f_k}$ is the external force applied to the multirotor in the interval $[t_k, t_{k+1})$ for $k = 0, ..., n-1$ expressed in the body frame, m is the number of features seen by all frames existing in the optimization window and l_s is the inverse depth of the sth feature. We assume an unknown slowly time-varying external force disturbance (in $\mathcal{W}$) whose derivative is zero mean Gaussian noise, $\dot{d}_f = \eta_{d_f}$, where $\eta_{d_f} \sim \mathcal{N}(0, \sigma^2_{d_f})$. Considering B as the set of keyframes in the sliding window plus the current frame, L as the set of features that have been seen at least in two frames within the sliding window, $\bar{I}^k_{k+1}$ as the set of IMU measurements between two consecutive keyframes (or the last keyframe and the current frame), $\bar{U}^k_{k+1}$ as the set of actuator measurements between two consecutive keyframes (or the last keyframe and the current frame), $\bar{Z}^{c_k}_s$ as the measurement of the sth feature in the kth frame with c_k as the kth camera frame. The posterior probability of χ with the available visual and inertial measurements in the current sliding window (denoted by $\mathcal{Z}$) and

the prior $p(\chi_0)$ is

$$p(\chi|\mathcal{Z}) \propto p(\chi_0)p(\mathcal{Z}|\chi) = p(\chi_0) \prod_{k\in B} p(\bar{I}^k_{k+1}|\chi) \prod_{k\in B} p(\bar{U}^k_{k+1}|\chi) \prod_{k\in B, s\in L} p(\bar{Z}^{c_k}_s|\chi)$$

The maximum-a-priori (MAP) estimate corresponds to the minimum of the negative log-posterior. Under the assumption of zero-mean Gaussian noise for the measurement models which are presented later by deriving the relation between the states of the two consecutive keyframes and the measurements, the negative log-posterior can be written as the sum of the squared residual errors derived from the mismatch between the measurements and their predicted values

$$\min_{\chi}\{\sum_{k\in B}\|r_I(\bar{I}^k_{k+1}, x_k, x_{k+1})\|^2_{P^I_k} + \sum_{k\in B}\|r_D(\bar{U}^k_{k+1}, x_k, d^b_{f_k}, x_{k+1}, d^b_{f_{k+1}})\|^2_{P^D_k} + \sum_{s\in L,\, k\in B}\|r_V(\bar{Z}^{c_k}_s, \chi)\|^2_{P^V_{sk}} + \|r_P(\chi)\|^2\} \tag{4.6}$$

In order to appropriately represent the uncertainty, the residual errors are weighted by the covariance matrices using the Mahalanobis distance $\|\cdot\|$ which measures the difference between a point and a distribution. In the above equation r_I and r_D are respectively the inertial and dynamic residuals. The P^I_k and P^D_k are the covariance matrices representing the uncertainty in r_I and r_D respectively between the times t_k and t_{k+1}. The r_V is the vision residual calculated from the difference between the expected measurement and its actual measurement and P^V_{sk} is the covariance matrix representing the uncertainty in the vision residual. Furthermore, r_P is the prior information from marginalization in order to keep the number of frames in the sliding window less than n. The block diagram of the visual-inertial-actuator odometry system is shown in Figure 4.3 illustrating the structure of the system and the data flow. In the next sections we derive the explicit relations for the residuals in equation (4.6).

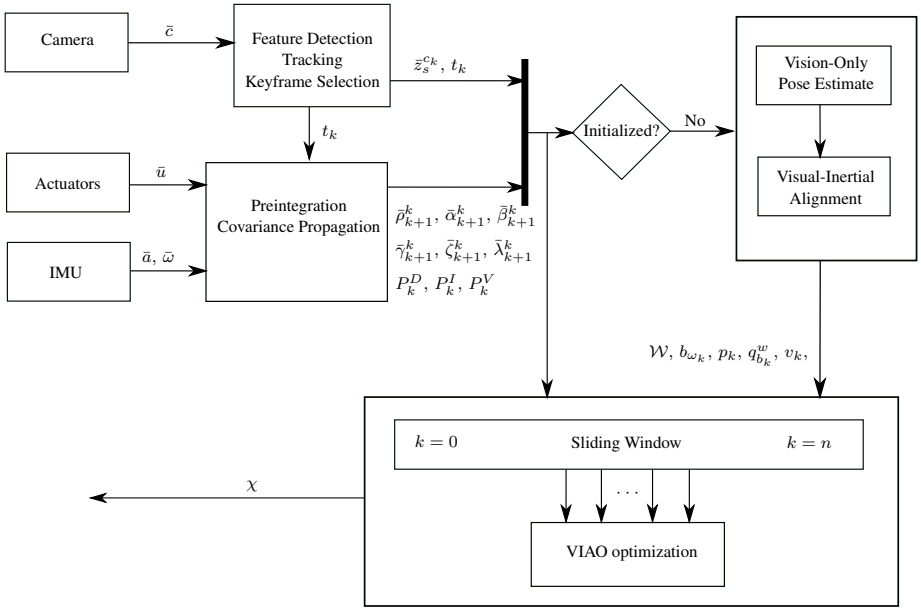


Figure 4.3: Block diagram of the Visual-Inertial-Actuator Odometry.

4.3 The Inertial Factor

We follow the procedure presented in [26] to derive the inertial residual. To derive the constraints between two consecutive keyframes, equations (4.4) are integrated over the interval $[t_k, t_{k+1}]$

$$p_{k+1} = p_k + \int_{t_k}^{t_{k+1}} v_k d\tau + \int\int_{t_k}^{t_{k+1}} R^w_{b_\tau} a^b_\tau d\tau^2 \tag{4.7a}$$

$$v_{k+1} = v_k + \int_{t_k}^{t_{k+1}} R^w_{b_\tau} a^b_\tau d\tau \tag{4.7b}$$

$$q^w_{b_{k+1}} = q^w_{b_k} \otimes \int_{t_k}^{t_{k+1}} \frac{1}{2}\Omega(\omega^b_\tau) q^{b_k}_{b_\tau} d\tau \tag{4.7c}$$

where the operator $\Omega(\cdot)$ was defined in (2.15) and the double integration in (4.7a) is defined as

$$\int\int_{t_k}^{t_{k+1}} R^w_{b_\tau} a^b_\tau d\tau^2 = \int_{t_k}^{t_{k+1}} (\int_{t_k}^{\mu} R^w_{b_\tau} a^b_\tau d\tau) d\mu$$

Considering $\Delta t_k = t_{k+1} - t_k$ and substituting for a^{b_τ} and ω^{b_τ} we obtain

$$p_{k+1} = p_k + v_k \Delta t_k + \frac{1}{2} g \Delta t_k^2 + \int\int_{t_k}^{t_{k+1}} R^w_{b_\tau} (\bar{a}^b_\tau - b_{a_k} - \eta_a) d\tau^2 \tag{4.8a}$$

$$v_{k+1} = v_k + g\Delta t_k + \int_{t_k}^{t_{k+1}} R^w_{b_\tau} (\bar{a}^b_\tau - b_{a_k} - \eta_a) d\tau \tag{4.8b}$$

$$q^w_{b_{k+1}} = q^w_{b_k} \otimes \int_{t_k}^{t_{k+1}} \frac{1}{2}\Omega(\bar{\omega}^b_\tau - b_{\omega_k} - \eta_\omega) q^{b_k}_{b_\tau} d\tau \tag{4.8c}$$

We remark that the integrals in (4.8) have to be recalculated whenever the states at time t_k change during every iteration of the optimization. Performing this recomputation would lead to excessive computation. To reduce the dependence of the integral terms on the optimization variables at time t_k, we change the reference frame from w to b_k. This leads to the following expressions

$$R^{b_k}_w (p_{k+1} - p_k - v_k \Delta t_k - \frac{1}{2} g \Delta t_k^2) = \alpha^k_{k+1} \tag{4.9a}$$

$$R^{b_k}_w (v_{k+1} - v_k - g\Delta t_k) = \beta^k_{k+1} \tag{4.9b}$$

$$q^{b_k}_w \otimes q^w_{b_{k+1}} = \gamma^k_{k+1} \tag{4.9c}$$

with

$$\alpha^k_{k+1} = \int\int_{t_k}^{t_{k+1}} R^{b_k}_{b_\tau} (\bar{a}^b_\tau - b_{a_k} - \eta_a) d\tau^2 \tag{4.10a}$$

$$\beta^k_{k+1} = \int_{t_k}^{t_{k+1}} R^{b_k}_{b_\tau} (\bar{a}^b_\tau - b_{a_k} - \eta_a) d\tau \tag{4.10b}$$

$$\gamma^k_{k+1} = \int_{t_k}^{t_{k+1}} \frac{1}{2}\Omega(\bar{\omega}^b_\tau - b_{\omega_k} - \eta_\omega) \gamma^k_\tau d\tau \tag{4.10c}$$

with $\alpha^k_{k+1} \in \mathbb{R}^3$, $\beta^k_{k+1} \in \mathbb{R}^3$ and $\gamma^k_{k+1} \in \mathbb{Q}_u$. We also define $\gamma^k_\tau = q^{b_k}_{b_\tau}$. It can be seen that the dependence of integrals on optimization variables in (4.10) has diminished from $R^w_{b_k}$, b_{a_k} and b_{ω_k} to b_{a_k} and b_{ω_k} in (4.10). Note that $R^w_{b_\tau} = R^w_{b_k} R^{b_k}_{b_\tau}$ and $R^{b_k}_{b_\tau}$ depends on the IMU measurements between t_k and τ. We observe the right hand side of (4.8) still depends on the sensor biases at time t_k. This problem is tackled by assuming b_{a_k} and b_{ω_k} are known and remain constant between two keyframes and then evaluate the integrals. But when the biases vary slightly during optimization, instead of reintegration, the integrals are updated by their first order approximations with respect to the changes in the biases. This procedure is given in Section 4.3.2. When the change in bias is above a threshold, the integrals are recalculated.

Equation (4.9) can be considered as measurement model. The equation (4.10) has a complex dependence on measurement noise and it is known that, for a MAP estimator a clear definition of the measurement densities, which in this case are represented by covariance matrix, is required. Also α^k_{k+1}, β^k_{k+1} and γ^k_{k+1} cannot be calculated as the η_a and η_ω are white noises. Therefore, we compute their estimates, denoted $\bar{(\cdot)}$, by assuming $\eta_a = \eta_\omega = 0$

$$\bar{\alpha}^k_{k+1} = \int\int_{t_k}^{t_{k+1}} R^{b_k}_{b_\tau}(\bar{\gamma}^k_\tau)(\bar{a}^b_\tau - b_{a_k}) d\tau^2 \tag{4.11a}$$

$$\bar{\beta}^k_{k+1} = \int_{t_k}^{t_{k+1}} R^{b_k}_{b_\tau}(\bar{\gamma}^k_\tau)(\bar{a}^b_\tau - b_{a_k}) d\tau \tag{4.11b}$$

$$\bar{\gamma}^k_{k+1} = \int_{t_k}^{t_{k+1}} \frac{1}{2}\Omega(\bar{\omega}^b_\tau - b_{\omega_k})\bar{\gamma}^k_\tau d\tau \tag{4.11c}$$

Therefore, we can write our measurement model as

$$\begin{bmatrix} \bar{\alpha}^k_{k+1} \\ \bar{\beta}^k_{k+1} \\ \bar{\gamma}^k_{k+1} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R^{b_k}_w(p_{k+1} - p_k - v_k\Delta t_k - \frac{1}{2}g\Delta t_k^2) \\ R^{b_k}_w(v_{k+1} - v_k - g\Delta t_k) \\ q^{b_k}_w \otimes q^w_{b_{k+1}} \\ b_{a_{k+1}} - b_{a_k} \\ b_{\omega_{k+1}} - b_{\omega_k} \end{bmatrix} \tag{4.12}$$

which gives us the following residual

$$r_I(\bar{I}^k_{k+1}, x_k, x_{k+1}) = \begin{bmatrix} R^{b_k}_w(p_{k+1} - p_k - v_k\Delta t_k - \frac{1}{2}g\Delta t_k^2) - \bar{\alpha}^k_{k+1} \\ R^{b_k}_w(v_{k+1} - v_k - g\Delta t_k) - \bar{\beta}^k_{k+1} \\ 2[q^{b_k}_w \otimes q^w_{b_{k+1}} \otimes (\bar{\gamma}^k_{k+1})^{-1}]_v \\ b_{a_{k+1}} - b_{a_k} \\ b_{\omega_{k+1}} - b_{\omega_k} \end{bmatrix} \tag{4.13}$$

in above equation the operator $[\cdot]_v$ extracts the vector part of the quaternion $q = [q_0, q_v^\top]^\top$.

The residual (4.13) is called the inertial factor and will be used later in a siding-window opti-

mization for VIAO. As mentioned before, the covariance matrix is required for the weight in the optimization (4.6). In the next section we present the procedure for calculation of α^k_{k+1}, β^k_{k+1}, γ^k_{k+1} and also the covariance propagation of the inertial factor.

4.3.1 Preintegration and Covariance Propagation

The integrals in (4.11) are in continuous time. For implementation the integrals need to be discretized. Different methods can be used for discretization. Here we take the trapezoidal rule for integration. We define i as the discrete time between two consecutive keyframes at t_k and t_{k+1} and δt as the time difference between i and $i+1$ IMU samples. We also take $\alpha^k_{k+1}[0] = 0$, $\beta^k_{k+1}[0] = 0$ and $\gamma^k_{k+1}[0] = [1, 0_{1\times 3}]^\top$ as the initial values for the preintegrations. With these assumptions the discretized relations with trapezoidal rule for integrals (4.10) and accelerometer and gyro biases will be

$$\alpha^k_{k+1}[i+1] = \alpha^k_{k+1}[i] + \beta^k_{k+1}[i]\delta t + \frac{1}{2}\frac{R(\gamma^k_{k+1}[i])(\bar{a}^b[i] + \eta_a[i] - b_{a_k}[i]) + R(\gamma^k_{k+1}[i+1])(\bar{a}^b[i+1] + \eta_a[i+1] - b_{a_k}[i+1])}{2}\delta t^2 \tag{4.14a}$$

$$\beta^k_{k+1}[i+1] = \beta^k_{k+1}[i] + \frac{R(\gamma^k_{k+1}[i])(\bar{a}^b[i] + \eta_a[i] - b_{a_k}[i]) + R(\gamma^k_{k+1}[i+1])(\bar{a}^b[i+1] + \eta_a[i+1] - b_{a_k}[i+1])}{2}\delta t \tag{4.14b}$$

$$\gamma^k_{k+1}[i+1] = \gamma^k_{k+1}[i] \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\frac{\bar{\omega}^b[i] + \eta_\omega[i] + \bar{\omega}^b[i+1] + \eta_\omega[i+1]}{2} - b_{\omega_k}[i])\delta t \end{bmatrix} \tag{4.14c}$$

$$b_{a_k}[i+1] = b_{a_k}[i] + \eta_{b_a}[i]\delta t \tag{4.14d}$$

$$b_{\omega_k}[i+1] = b_{\omega_k}[i] + \eta_{b_\omega}[i]\delta t \tag{4.14e}$$

Given the above equations, the linear approximation of the error term due to the measurement noise is

$$\begin{bmatrix} \delta\alpha^k_{k+1}[i+1] \\ \delta\beta^k_{k+1}[i+1] \\ \delta\theta^k_{k+1}[i+1] \\ \delta b_{a_k}[i+1] \\ \delta b_{\omega_k}[i+1] \end{bmatrix} = \underbrace{\begin{bmatrix} I & \delta t & F_{13} & F_{14} & F_{15} \\ 0 & I & F_{23} & -\frac{R(\gamma^k_{k+1}[i]) + R(\gamma^k_{k+1}[i+1])}{2}\delta t & F_{25} \\ 0 & 0 & I - S(\frac{\bar{\omega}^b[i] + \bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i]) & 0 & -\delta t \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}}_{F[i]} \begin{bmatrix} \delta\alpha^k_{k+1}[i] \\ \delta\beta^k_{k+1}[i] \\ \delta\theta^k_{k+1}[i] \\ \delta b_{a_k}[i] \\ \delta b_{\omega_k}[i] \end{bmatrix}$$
$$+ \underbrace{\begin{bmatrix} \frac{1}{4}R(\gamma^k_{k+1}[i])\delta t^2 & V_{12} & \frac{1}{4}R(\gamma^k_{k+1}[i+1])\delta t^2 & V_{14} & 0 & 0 \\ \frac{1}{2}R(\gamma^k_{k+1}[i])\delta t & V_{22} & \frac{1}{2}R(\gamma^k_{k+1}[i+1])\delta t & V_{24} & 0 & 0 \\ 0 & \frac{1}{2}\delta t & 0 & \frac{1}{2}\delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta t \end{bmatrix}}_{V[i]} \begin{bmatrix} \eta_a[i] \\ \eta_\omega[i] \\ \eta_a[i+1] \\ \eta_\omega[i+1] \\ \eta_{b_a}[i] \\ \eta_{b_\omega}[i] \end{bmatrix} \tag{4.15}$$

where $\delta\theta^k_{k+1}[i] \in \mathbb{R}^3$ is the error term considered as a perturbation about the mean of $\hat{\gamma}^k_{k+1}[i]$ defined by

$$\gamma^k_{k+1}[i] = \hat{\gamma}^k_{k+1}[i] \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta^k_{k+1}[i] \end{bmatrix}$$

and

$$F_{13} = \frac{-R(\gamma^k_{k+1}[i])S(\bar{a}^b[i] - b_{a_k}[i])\delta t^2}{4} + \frac{-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])(I - S(\frac{\bar{\omega}^b[i]+\bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i])\delta t)\delta t^2}{4} \tag{4.16a}$$

$$F_{23} = \frac{-R(\gamma^k_{k+1}[i])S(\bar{a}^b[i] - b_{a_k}[i])\delta t}{2} + \frac{-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])(I - S(\frac{\bar{\omega}^b[i]+\bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i])\delta t)\delta t}{2} \tag{4.16b}$$

$$F_{14} - \frac{1}{4}(R(\gamma^k_{k+1}[i]) + R(\gamma^k_{k+1}[i+1]))\delta t^2 \tag{4.16c}$$

$$F_{15} = \frac{1}{4}(-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])\delta t^2)(-\delta t) \tag{4.16d}$$

$$F_{25} = \frac{1}{2}(-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])\delta t)(-\delta t) \tag{4.16e}$$

$$V_{12} = \frac{1}{4}(-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])\delta t^2)(\frac{1}{2}\delta t) \tag{4.16f}$$

$$V_{14} = \frac{1}{4}(-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])\delta t^2)(\frac{1}{2}\delta t) \tag{4.16g}$$

$$V_{22} = \frac{1}{2}(-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])\delta t)(\frac{1}{2}\delta t) \tag{4.16h}$$

$$V_{24} = \frac{1}{2}(-R(\gamma^k_{k+1}[i+1])S(\bar{a}^b[i+1] - b_{a_k}[i])\delta t)(\frac{1}{2}\delta t) \tag{4.16i}$$

Now, using the linear approximation (4.15) the Jacobian matrix can be propagated recursively by the initial condition $J^I_k[0] = I$ and

$$J^I_k[i+1] = F[i]J^I_k[i] \tag{4.17}$$

Also the covariance of the measurement model noise can also be propagated using the formula

$$P^I_k[i+1] = F[i]P^I_k[i]F[i]^\top + V[i]Q_dV[i]^\top \tag{4.18}$$

with $P^I_k[0] = 0$ and $Q_d \approx \delta t Q_t$ as the discrete-time covariance matrix where Q_t is the continuous time covariance matrix available from the sensor specifications

$$Q_t = \begin{bmatrix} \sigma_a^2 I_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\omega^2 I_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_a^2 I_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\omega^2 I_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{b_a}^2 I_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{b_\omega}^2 I_3 \end{bmatrix}.$$

4.3.2 Accelerometer and Gyro Bias Correction

Normally the bias estimate at time t_k changes during the optimization. This requires updating the integrated terms as the current bias estimate is different. One way to perform this is to reintegrate the equation (4.11) which is computationally demanding. One way to avoid this is to replace the integrals by their first order approximations using the Jacobians with respect to the biases when the changes are less than certain thresholds

$$\bar{\alpha}_{k+1}^{k} = \bar{\alpha}_{k+1}^{k} + [J_k^I]_{b_a}^{\bar{\alpha}} \delta b_{a_k} + [J_k^I]_{b_\omega}^{\bar{\alpha}} \delta b_{\omega_k} \tag{4.19a}$$

$$\bar{\beta}_{k+1}^{k} = \bar{\beta}_{k+1}^{k} + [J_k^I]_{b_a}^{\bar{\beta}} b_{a_k} + [J_k^I]_{b_\omega}^{\bar{\beta}} \delta b_{\omega_k} \tag{4.19b}$$

$$\bar{\gamma}_{k+1}^{k} = \bar{\gamma}_{k+1}^{k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}[J_k^I]_{b_\omega}^{\bar{\gamma}} \delta b_{\omega_k} \end{bmatrix} \tag{4.19c}$$

Otherwise, we can repropagate the integration (4.11) with the new bias estimates derived from the optimization.

4.4 The Dynamic Factor

The multirotor dynamics subject to external force disturbance and rotor drag is

$$\dot{p} = v \tag{4.20a}$$

$$m\dot{v} = -mg + R_b^w T^b - R_b^w D R_b^{w\top} v_a + d_f \tag{4.20b}$$

$$\dot{q}_b^w = \frac{1}{2} q_b^w \otimes \begin{bmatrix} 0 \\ \omega^b \end{bmatrix} \tag{4.20c}$$

where T^b is the thrust vector generated by the propellers expressed in body frame, d_f is the external force disturbance expressed in the world frame, m is mass, $v_a = v - w$ is the air velocity with $w \in \mathbb{R}^3$ as constant wind speed, D is a matrix including the rotor drag coefficients (2.7). As the wind velocity is normally unknown, in the rest of the paper, we assume $w = 0$, which means $v_a = v$. Considering $\bar{u}$ as the collective thrust measured from the ESCs, we propose a measurement model for T^b as

$$\bar{T}^b = \begin{bmatrix} 0 \\ 0 \\ \bar{u} \end{bmatrix} + \eta_T \tag{4.21}$$

where η_T is a GWN as the measurement noise.

In the following we present the residual term r_D by employing the dynamical model and the thrust measurements between two consecutive keyframes at times t_k and t_{k+1}.

4.4.1 Position Residual

If we consider the time interval between two consecutive image frames (or key-frames) as $\Delta t_k = t_{k+1} - t_k$, then we have

$$p_{k+1} = p_k + v_k \Delta t_k - \frac{1}{2} g \Delta t_k^2 + \frac{1}{m} \int \int_{t_k}^{t_{k+1}} R_{b_\tau}^w (T_\tau^b - D R_{b_\tau}^{w\top} v_\tau + d_{f_\tau}^b) d\tau^2 \tag{4.22a}$$

$$v_{k+1} = v_k - g \Delta t_k + \frac{1}{m} \int_{t_k}^{t_{k+1}} R_{b_\tau}^w (T_\tau^b - D R_{b_\tau}^{w\top} v_\tau + d_{f_\tau}^b) d\tau \tag{4.22b}$$

$$q_{k+1}^w = q_k^w \otimes \int_{t_k}^{t_{k+1}} \frac{1}{2} \Omega(\omega_\tau^b) q_\tau^k d\tau \tag{4.22c}$$

where $R_{b_\tau}^w = R_{b_k}^w R_{b_\tau}^{b_k}$. Changing the reference frame from w to b_k (multiplying by $R_w^{b_k}$) we get

$$R_w^{b_k}(p_{k+1} - p_k - v_k \Delta t_k + \frac{1}{2} g \Delta t_k^2) = \frac{1}{m} \int \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} (T_\tau^b - D R_{b_\tau}^{b_k\top} R_{b_k}^{w\top} v_\tau + d_{f_\tau}^b) d\tau^2 \tag{4.23a}$$

$$R_w^{b_k}(v_{k+1} - v_k + g \Delta t_k) = \frac{1}{m} \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} (T_\tau^b - D R_{b_\tau}^{b_k\top} R_{b_k}^{w\top} v_\tau + d_{f_\tau}^b) d\tau \tag{4.23b}$$

To bring v_τ outside of the integral and simplify the integration we consider $R_{b_\tau}^{b_k} \approx I$, $R_{b_\tau}^{b_k} D R_{b_\tau}^{b_k\top} \approx D$ which gives $R_{b_\tau}^{b_k} D R_{b_\tau}^{b_k\top} R_{b_k}^{w\top} v_\tau \approx D R_{b_k}^{w\top} v_\tau$, therefore

$$R_w^{b_k}(p_{k+1} - p_k - v_k \Delta t_k + \frac{1}{2} g \Delta t_k^2) = \frac{1}{m} \int \int_{t_k}^{t_{k+1}} (R_{b_\tau}^{b_k} T_\tau^b - D R_{b_k}^{w\top} v_\tau + R_{b_\tau}^{b_k} d_{f_\tau}^b) d\tau^2 \tag{4.24a}$$

$$R_w^{b_k}(v_{k+1} - v_k + g \Delta t_k) = \frac{1}{m} \int_{t_k}^{t_{k+1}} (R_{b_\tau}^{b_k} T_\tau^b - D R_{b_k}^{w\top} v_\tau + R_{b_\tau}^{b_k} d_{f_\tau}^b) d\tau \tag{4.24b}$$

considering v_τ to be constant in the interval $[t_k, t_{k+1}]$, we can write the above equation as

$$R_w^{b_k}(p_{k+1} - p_k - v_k \Delta t_k + \frac{1}{2} g \Delta t_k^2) + \frac{1}{2m} D R_{b_k}^{w\top} v_k \Delta t_k^2 = \zeta_{k+1}^k \tag{4.25a}$$

$$R_w^{b_k}(v_{k+1} - v_k + g \Delta t_k) + \frac{1}{m} D R_{b_k}^{w\top} v_k \Delta t_k = \lambda_{k+1}^k \tag{4.25b}$$

where

$$\zeta_{k+1}^k = \frac{1}{m} \int \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} (T_\tau^b + d_{f_\tau}^b) d\tau^2 \tag{4.26a}$$

$$\lambda_{k+1}^k = \frac{1}{m} \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} (T_\tau^b + d_{f_\tau}^b) d\tau \tag{4.26b}$$

Due to the presence of measurement noise the actual value of T^b is not exactly known, therefore, the measured thrust can be treated as an estimate and used to calculate $\bar{\zeta}_{k+1}^k$ and $\bar{\lambda}_{k+1}^k$ which are

the estimates for ζ_{k+1}^k and λ_{k+1}^k

$$\bar{\zeta}_{k+1}^k = \frac{1}{m}\int\int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k}(\bar{T}_\tau^b + d_{f_\tau}^b)d\tau^2 \tag{4.27a}$$

$$\bar{\lambda}_{k+1}^k = \frac{1}{m}\int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k}(\bar{T}_\tau^b + d_{f_\tau}^b)d\tau \tag{4.27b}$$

4.4.2 Disturbance Observer

Considering the dynamics of the multirotor (4.20), and with the assumption of random walk process for d_f (i.e., $\dot{d}_f = \eta_{d_f} \approx 0$) (which is an accurate assumption for slowly time-varying disturbances) the following disturbance observer [89] has a globally exponential stable dynamics

$$\hat{d}_f = z_{d_f} + k_{d_f} m v \tag{4.28a}$$

$$\dot{z}_{d_f} = -k_{d_f}\hat{d}_f - k_{d_f}(-mg + R_b^w T^b - R_b^w D {R_b^w}^\top v). \tag{4.28b}$$

where $z_{d_f} \in \mathbb{R}^3$ is the observer state, $\hat{d}_f \in \mathbb{R}^3$ is disturbance estimate expressed in the world frame, and $k_{d_f} > 0$ is an observer gain. Defining $\tilde{d}_f = d_f - \hat{d}_f$, observer (3.3) has the error dynamics

$$\begin{aligned}\dot{\tilde{d}}_f &= -(\dot{z}_{d_f} + k_{d_f} m \dot{v}) \\ &= k_{d_f}\hat{d}_f + k_{d_f}(-mg + R_b^w T^b - R_b^w D {R_b^w}^\top v) - k_{d_f}(-mg + R_b^w T^b - R_b^w D {R_b^w}^\top v + d_f) \\ &= -k_{d_f}\tilde{d}_f \end{aligned} \tag{4.29}$$

which is globally exponentially stable. Now, we write the observer dynamics as a residual and include it in the optimization. From equation (4.28b), the force disturbance estimate at time t_{k+1} can be written as

$$\hat{d}_{f_{k+1}} = z_{d_f}^{k+1} + k_{d_f} m v_{k+1} \tag{4.30}$$

where

$$z_{d_f}^{k+1} = z_{d_f}^k + \int_{t_k}^{t_{k+1}} (-k_{d_f}\hat{d}_{f_\tau} - k_{d_f}(-mg + R_{b_\tau}^w T_\tau^b - R_{b_\tau}^w D {R_{b_\tau}^w}^\top v_\tau))d\tau.$$

Assuming constant $\hat{d}_{f_\tau}$ within the time interval $[t_k, t_{k+1}]$ we can write

$$z_{d_f}^{k+1} = z_{d_f}^k - k_{d_f}\hat{d}_{f_k}\Delta t_k + k_{d_f} mg \Delta t_k - k_{d_f}\int_{t_k}^{t_{k+1}} (R_{b_\tau}^w T_\tau^b - R_{b_\tau}^w D {R_{b_\tau}^w}^\top v_\tau)d\tau$$

and considering $R_{b_\tau}^w = R_{b_k}^w R_{b_\tau}^{b_k}$ yields

$$z_{d_f}^{k+1} = z_{d_f}^{k} - k_{d_f}\hat{d}_{f_k}\Delta t_k + k_{d_f} mg\Delta t_k - k_{d_f} R_{b_k}^{w} \int_{t_k}^{t_{k+1}} (R_{b_\tau}^{b_k} T_\tau^b - R_{b_\tau}^{b_k} D {R_{b_\tau}^{b_k}}^\top {R_{b_k}^{w}}^\top v_\tau) d\tau \tag{4.31}$$

Now considering $R_{b_\tau}^{b_k} \approx I$, $R_{b_\tau}^{b_k} D {R_{b_\tau}^{b_k}}^\top \approx D$ we can write

$$z_{d_f}^{k+1} = z_{d_f}^{k} - k_{d_f}\hat{d}_{f_k}\Delta t_k + k_{d_f} mg\Delta t_k - k_{d_f} R_{b_k}^{w} \int_{t_k}^{t_{k+1}} (R_{b_\tau}^{b_k} T_\tau^b - D {R_{b_k}^{w}}^\top v_\tau) d\tau \tag{4.32}$$

and by the assumption of constant velocity within the time interval $[t_k, t_{k+1}]$

$$z_{d_f}^{k+1} = z_{d_f}^{k} - k_{d_f}\hat{d}_{f_k}\Delta t_k + k_{d_f} mg\Delta t_k + k_{d_f} R_{b_k}^{w} D {R_{b_k}^{w}}^\top v_k \Delta t_k - k_{d_f} R_{b_k}^{w} \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} T_\tau^b d\tau \tag{4.33}$$

which gives us

$$\rho_{k+1}^{k} = -\frac{1}{k_{d_f}} R_w^{b_k} (z_{d_f}^{k+1} - z_{d_f}^{k} + k_{d_f}\hat{d}_{f_k}\Delta t_k - k_{d_f} mg\Delta t_k - k_{d_f} R_{b_k}^{w} D {R_{b_k}^{w}}^\top v_k \Delta t_k) \tag{4.34}$$

where

$$\rho_{k+1}^{k} = \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} T_\tau^b d\tau \tag{4.35}$$

Also, by using the relations

$$z_{d_f}^{k+1} = \hat{d}_{f_{k+1}} - k_{d_f} m v_{k+1} \tag{4.36}$$

and

$$z_{d_f}^{k} = \hat{d}_{f_k} - k_{d_f} m v_k \tag{4.37}$$

we can write

$$\begin{aligned}
\rho_{k+1}^{k} &= -\frac{1}{k_{d_f}} R_w^{b_k} (\hat{d}_{f_{k+1}} - k_{d_f} m v_{k+1} - \hat{d}_{f_k} + k_{d_f} m v_k + k_{d_f}\hat{d}_f^k \Delta t_k - k_{d_f} mg\Delta t_k - k_{d_f} R_{b_k}^{w} D {R_{b_k}^{w}}^\top v_k \Delta t_k) \\
&= \frac{1}{k_{d_f}} R_w^{b_k} (-\hat{d}_{f_{k+1}} + k_{d_f} m v_{k+1} + \hat{d}_{f_k} - k_{d_f} m v_k - k_{d_f}\hat{d}_{f_k}\Delta t_k + k_{d_f} mg\Delta t_k + k_{d_f} R_{b_k}^{w} D {R_{b_k}^{w}}^\top v_k \Delta t_k) \\
&= \frac{1}{k_{d_f}} R_w^{b_k} (\hat{d}_{f_k} - \hat{d}_{f_{k+1}}) + R_w^{b_k} (m v_{k+1} - m v_k - \hat{d}_{f_k}\Delta t_k + mg\Delta t_k + R_{b_k}^{w} D {R_{b_k}^{w}}^\top v_k \Delta t_k) \\
&= \frac{1}{k_{d_f}} (\hat{d}_{f_k}^b - R_{b_{k+1}}^{b_k} \hat{d}_{f_{k+1}}^b) + R_w^{b_k} (m v_{k+1} - m v_k + mg\Delta t_k) - \hat{d}_{f_k}^b \Delta t_k + D {R_{b_k}^{w}}^\top v_k \Delta t_k
\end{aligned} \tag{4.38}$$

The difference between this constraint and the one in equation (9) in VIMO [27] is that the assumption of slowly time-varying force disturbance is included in the residual. Since the actual value of thrust is unknown, we consider its measurements and derive an estimate for ρ_{k+1}^{k} by

$$\bar{\rho}_{k+1}^{k} = \int_{t_k}^{t_{k+1}} R_{b_\tau}^{b_k} \bar{T}_\tau^b d\tau \tag{4.39}$$

4.4.3 Dynamic Residual

The other assumption that can be added to the residual is the assumption of slowly time-varying force disturbance in the world frame which is a realistic assumption for modelling the force disturbances as a result of wind-gust or ground effect. We can express this assumption by the following equation

$$\hat{d}_{f_k} - \hat{d}_{f_{k+1}} = R^w_{b_k}\hat{d}^b_{f_k} - R^w_{b_{k+1}}\hat{d}^b_{f_{k+1}} \approx 0 \tag{4.40}$$

The following dynamic residual is proposed which consists of (4.25a), (4.38) and (4.40)

$$r_D(\bar{U}^k_{k+1}, x_k, \hat{d}^b_{f_k}, x_{k+1}, \hat{d}^b_{f_{k+1}}) = \begin{bmatrix} R^{b_k}_w(p_{k+1} - p_k - v_k\Delta t_k + \frac{1}{2}g\Delta t_k^2) + \frac{1}{2m}DR^{w\top}_{b_k}v_k\Delta t_k^2 - \bar{\zeta}^k_{k+1} \\ \frac{1}{k_{d_f}}(\hat{d}^b_{f_k} - R^{b_k}_{b_{k+1}}\hat{d}^b_{f_{k+1}}) + R^{b_k}_w(mv_{k+1} - mv_k + mg\Delta t_k) - \hat{d}^b_{f_k}\Delta t_k - \bar{\rho}^k_{k+1} \\ R^w_{b_k}\hat{d}^b_{f_k} - R^w_{b_{k+1}}\hat{d}^b_{f_{k+1}} \end{bmatrix} \tag{4.41}$$

As defined before, $\bar{U}^k_{k+1}$ is the set of all control inputs within the time interval $[t_k, t_{k+1}]$. We remark that for the tightly coupled optimization (4.6) we consider $d_f = \hat{d}_f$.

4.4.4 Preintegration and Covariance Propagation

The integrals in (4.10), (4.26) and (4.39) are in continuous time. For implementation the integrals need to be discretized. Similar to Section 4.3.1 here we take the trapezoidal rule for integration. We define i as the discrete time between two consecutive keyframes at t_k and t_{k+1} and δt as the time difference between i and $i+1$ samples. We also take $\zeta^k_{k+1}[0] = 0$, $\lambda^k_{k+1}[0] = 0$, $\rho^k_{k+1}[0] = 0$ and $\gamma^k_{k+1}[0] = [1, 0_{1\times 3}]^\top$ as the initial values for the preintegrations. With these assumptions the discretized relations with trapezoidal rule for integrals (4.26), (4.10c) (4.39) and force disturbance and gyro biases will be

$$\zeta^k_{k+1}[i+1] = \zeta^k_{k+1}[i] + \rho^k_{k+1}[i]\delta t + \frac{1}{2}\frac{R(\gamma^k_{k+1}[i])(\bar{T}^b[i] + \eta_T[i] - \hat{d}^b_{f_k}[i]) + R(\gamma^k_{k+1}[i+1])(\bar{T}^b[i+1] + \eta_T[i+1] - \hat{d}^b_{f_k}[i+1])}{2}\delta t^2 \tag{4.42a}$$

$$\lambda^k_{k+1}[i+1] = \lambda^k_{k+1}[i] + \frac{R(\gamma^k_{k+1}[i])(\bar{T}^b[i] + \eta_T[i] - \hat{d}^b_{f_k}[i]) + R(\gamma^k_{k+1}[i+1])(\bar{T}^b[i+1] + \eta_T[i+1] - \hat{d}^b_{f_k}[i+1])}{2}\delta t \tag{4.42b}$$

$$\rho^k_{k+1}[i+1] = \rho^k_{k+1}[i] + \frac{R(\gamma^k_{k+1}[i])(\bar{T}^b[i] + \eta_T[i]) + R(\gamma^k_{k+1}[i+1])(\bar{T}^b[i+1] + \eta_T[i+1])}{2}\delta t \tag{4.42c}$$

$$\gamma^k_{k+1}[i+1] = \gamma^k_{k+1}[i] \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\frac{\bar{\omega}^b[i] + \eta_\omega[i] + \bar{\omega}^b[i+1] + \eta_\omega[i+1]}{2} - b_{\omega_k}[i])\delta t \end{bmatrix} \tag{4.42d}$$

$$\hat{d}^b_{f_k}[i+1] = \hat{d}^b_{f_k}[i] + \eta_{d_f}[i]\delta t \tag{4.42e}$$

$$b_{\omega_k}[i+1] = b_{\omega_k}[i] + \eta_{b_\omega}[i]\delta t \tag{4.42f}$$

The linear approximation of the error term in the above equation in the presence of measurement noise is

$$\begin{bmatrix} \delta\zeta_{k+1}^k[i+1] \\ \delta\lambda_{k+1}^k[i+1] \\ \delta\rho_{k+1}^k[i+1] \\ \delta\theta_{k+1}^k[i+1] \\ \delta\hat{d}_{f_k}^b[i+1] \\ \delta b_{\omega_k}[i+1] \end{bmatrix} = \underbrace{\begin{bmatrix} I & \delta t & 0 & \bar{F}_{14} & \bar{F}_{15} & \bar{F}_{16} \\ 0 & I & 0 & \bar{F}_{24} & -\frac{R(\gamma_{k+1}^k[i])+R(\gamma_{k+1}^k[i+1])}{2}\delta t & \bar{F}_{26} \\ 0 & 0 & I & \bar{F}_{34} & -\frac{R(\gamma_{k+1}^k[i])+R(\gamma_{k+1}^k[i+1])}{2}\delta t & \bar{F}_{36} \\ 0 & 0 & 0 & I - S(\frac{\bar{\omega}^b[i]+\bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i]) & 0 & -\delta t \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}}_{\bar{F}[i]} \begin{bmatrix} \delta\zeta_{k+1}^k[i] \\ \delta\lambda_{k+1}^k[i] \\ \delta\rho_{k+1}^k[i] \\ \delta\theta_{k+1}^k[i] \\ \delta\hat{d}_{f_k}^b[i] \\ \delta b_{\omega_k}[i] \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} \frac{1}{4}R(\gamma_{k+1}^k[i])\delta t^2 & \bar{V}_{12} & \frac{1}{4}R(\gamma_{k+1}^k[i+1])\delta t^2 & \bar{V}_{14} & 0 & 0 \\ \frac{1}{2}R(\gamma_{k+1}^k[i])\delta t & \bar{V}_{22} & \frac{1}{2}R(\gamma_{k+1}^k[i+1])\delta t & \bar{V}_{24} & 0 & 0 \\ \frac{1}{2}R(\gamma_{k+1}^k[i])\delta t & \bar{V}_{32} & \frac{1}{2}R(\gamma_{k+1}^k[i+1])\delta t & \bar{V}_{34} & 0 & 0 \\ 0 & \frac{1}{2}\delta t & 0 & \frac{1}{2}\delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta t \end{bmatrix}}_{\bar{V}[i]} \begin{bmatrix} \eta_T[i] \\ \eta_\omega[i] \\ \eta_T[i+1] \\ \eta_\omega[i+1] \\ \eta_{d_f}[i] \\ \eta_{b_\omega}[i] \end{bmatrix} \tag{4.43}$$

where

$$\bar{F}_{14} = \frac{-R(\gamma_{k+1}^k[i])S(\bar{T}^b[i] - \hat{d}_{f_k}^b[i])\delta t^2}{4} + \frac{-R(\gamma_{k+1}^k[i+1])S(\bar{T}^b[i+1] - \hat{d}_{f_k}^b[i])(I - S(\frac{\bar{\omega}^b[i]+\bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i])\delta t)\delta t^2}{4} \tag{4.44a}$$

$$\bar{F}_{24} = \frac{-R(\gamma_{k+1}^k[i])S(\bar{T}^b[i] - \hat{d}_{f_k}^b[i])\delta t}{2} + \frac{-R(\gamma_{k+1}^k[i+1])S(\bar{T}^b[i+1] - \hat{d}_{f_k}^b[i])(I - S(\frac{\bar{\omega}^b[i]+\bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i])\delta t)\delta t}{2} \tag{4.44b}$$

$$\bar{F}_{15} - \frac{1}{4}(R(\gamma_{k+1}^k[i+1]) + R(\gamma_{k+1}^k[i]))\delta t^2 \tag{4.44c}$$

$$\bar{F}_{16} = \frac{1}{4}(-R(\gamma_{k+1}^k[i+1])S(\bar{T}^b[i+1] - \hat{d}_{f_k}^b[i])\delta t^2)(-\delta t) \tag{4.44d}$$

$$\bar{F}_{26} = \frac{1}{2}(-R(\gamma_{k+1}^k[i+1])S(\bar{T}^b[i+1] - \hat{d}_{f_k}^b[i])\delta t)(-\delta t) \tag{4.44e}$$

$$\bar{F}_{34} = \frac{-R(\gamma_{k+1}^k[i])S(\bar{T}[i])\delta t}{2} + \frac{-R(\gamma_{k+1}^k[i+1])S(\bar{T}^b[i+1])(I - S(\frac{\bar{\omega}^b[i]+\bar{\omega}^b[i+1]}{2} - b_{\omega_k}[i])\delta t)\delta t}{2} \tag{4.44f}$$

$$\bar{F}_{36} = \frac{1}{2}(-R(\gamma_{k+1}^k[i+1])S(\bar{T}^b[i+1])\delta t)(-\delta t) \tag{4.44g}$$

and

$$\bar{V}_{12} = \frac{1}{4}(-R(\gamma_{k+1}^{k}[i+1])S(\bar{T}^{b}[i+1] - \hat{d}_{f_k}^{b}[i])\delta t^2)(\frac{1}{2}\delta t) \tag{4.45a}$$

$$\bar{V}_{14} = \frac{1}{4}(-R(\gamma_{k+1}^{k}[i+1])S(\bar{T}^{b}[i+1] - \hat{d}_{f_k}^{b}[i])\delta t^2)(\frac{1}{2}\delta t) \tag{4.45b}$$

$$\bar{V}_{22} = \frac{1}{2}(-R(\gamma_{k+1}^{k}[i+1])S(\bar{T}^{b}[i+1] - \hat{d}_{f_k}^{b}[i])\delta t)(\frac{1}{2}\delta t) \tag{4.45c}$$

$$\bar{V}_{24} = \frac{1}{2}(-R(\gamma_{k+1}^{k}[i+1])S(\bar{T}^{b}[i+1] - \hat{d}_{f_k}^{b}[i])\delta t)(\frac{1}{2}\delta t) \tag{4.45d}$$

$$\bar{V}_{32} = \frac{1}{2}(-R(\gamma_{k+1}^{k}[i+1])S(\bar{T}^{b}[i+1])\delta t)(\frac{1}{2}\delta t) \tag{4.45e}$$

$$\bar{V}_{34} = \frac{1}{2}(-R(\gamma_{k+1}^{k}[i+1])S(\bar{T}^{b}[i+1])\delta t)(\frac{1}{2}\delta t) \tag{4.45f}$$

Now the Jacobian matrix can be propagated recursively by the initial condition $J_k^D[0] = I$ and

$$J_k^D[i+1] = \bar{F}[i]J_k^D[i] \tag{4.46}$$

The covariance can also be propagated from the formula

$$\bar{P}_k^D[i+1] = \bar{F}[i]\bar{P}_k^D[i]\bar{F}[i]^\top + \bar{V}[i]\bar{Q}_d\bar{V}[i]^\top \tag{4.47}$$

with $\bar{P}_k^D[0] = 0$ and $\bar{Q}_d$ as the discrete-time covariance matrix and $\bar{Q}_d = \delta t\bar{Q}_t$, where $\bar{Q}_t$ is the continuous time covariance matrix available from the sensors specifications

Therefore, the residual weight used in the VIAO optimization (4.6) is given by

$$P_k^D = \begin{bmatrix} \bar{P}^D_{k(1:3,1:3)} & \bar{P}^D_{k(1:3,7:9)} & \bar{P}^D_{k(1:3,13:15)} \\ \bar{P}^D_{k(7:9,1:3)} & \bar{P}^D_{k(7:9,7:9)} & \bar{P}^D_{k(7:9,13:15)} \\ \bar{P}^D_{k(13:15,1:3)} & \bar{P}^D_{k(13:15,7:9)} & \bar{P}^D_{k(13:15,13:15)} \end{bmatrix} \tag{4.48}$$

4.4.5 Force Disturbance and Gyro Bias Correction

Similar to the previous section, the force disturbance estimate might change during the optimization. Here in order to avoid repropagation of preintegrations, a first order approximation is used when the changes are less than a threshold

$$\bar{\zeta}_{k+1}^{k} = \bar{\zeta}_{k+1}^{k} + [J_k^D]^{\bar{\zeta}}_{\hat{d}_f^b}\delta\hat{d}_{f_k}^{b} + [J_k^D]^{\bar{\zeta}}_{b_\omega}\delta b_{\omega_k} \tag{4.49a}$$

$$\bar{\lambda}_{k+1}^{k} = \bar{\lambda}_{k+1}^{k} + [J_k^D]^{\bar{\lambda}}_{\hat{d}_f^b}\delta\hat{d}_{f_k}^{b} + [J_k^D]^{\bar{\lambda}}_{b_\omega}\delta b_{\omega_k} \tag{4.49b}$$

$$\bar{\gamma}_{k+1}^{k} = \bar{\gamma}_{k+1}^{k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}[J_k^D]^{\bar{\gamma}}_{b_\omega}\delta b_{\omega_k} \end{bmatrix} \tag{4.49c}$$

However, if the changes are greater than the specified value, the integrals are recalculated.

4.5 Vision Residual

We denote the image measurement at kth image frame as c_k. The measurement model for the sth feature (which we assume received its first observation in the jth image) observed in the kth image is defined [26] as

$$Z_s^{c_k} = \frac{V_s^{c_k}}{\|V_s^{c_k}\|} \tag{4.50}$$

with

$$V_s^{c_k} = R_b^c(R_w^{b_k}(R_{b_j}^w(R_c^b\frac{1}{r_s}\pi_c^{-1}\left(\begin{bmatrix}\bar{u}_s^{c_j}\\ \bar{v}_s^{c_j}\end{bmatrix}\right) + p_c^b) + p_j - p_k) - p_c^b) \tag{4.51}$$

where $[\bar{u}_s^{c_j}, \bar{v}_s^{c_j}]^\top$ is the first pixel measurement of the sth feature in the jth image. Also, π_c^{-1} is a projection function that project a pixel location into the unit sphere using camera intrinsic parameters. The actual output of the vision system (which consists of feature detector and tracker) is $\bar{z}_s^{c_k} = [\bar{u}_s^{c_k}, \bar{v}_s^{c_k}]^\top$ that is the observation of the same feature (that was first observed in the jth image) in the kth image. With the above definition, the vision residual can be defined as

$$r_V(\bar{z}_s^{c_k}, \chi) = [h_1\ h_2]^\top \cdot (\bar{Z}_s^{c_k} - Z_s^{c_k}) \tag{4.52}$$

with

$$\bar{Z}_s^{c_k} = \pi_c^{-1}\left(\begin{bmatrix}\bar{u}_s^{c_k}\\ \bar{v}_s^{c_k}\end{bmatrix}\right) \tag{4.53}$$

and $[h_1\ h_2]^\top$ as the vector space spanning the tangent plane of $\bar{Z}_s^{c_k}$. A visual illustration for definition of the vision residual is shown in Figures 4.4.

The covariance of the residual which is needed as a weight in the optimization (4.6) can be directly propagated from projection of the pixel coordinates onto the unit sphere. A Pseudo code of the algorithm can be seen below.

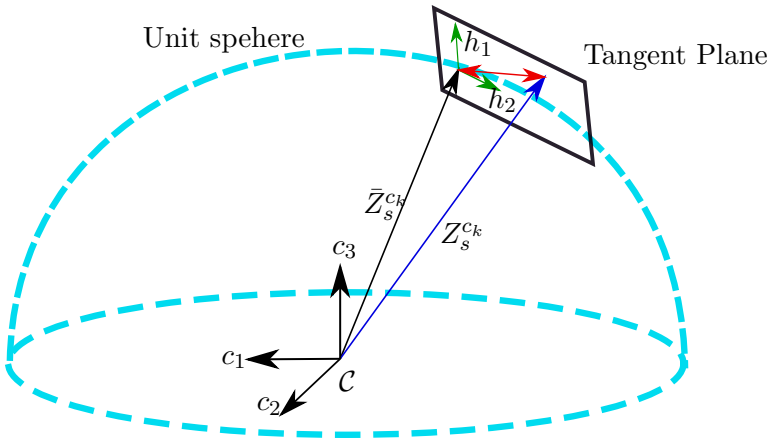


Figure 4.4: Vision residual on unit sphere.

Algorithm 1: Tightly-Coupled Visual-Inertial-Actuator State Estimation

Result: $\chi = [{x_0}^\top, {d^b_{f_0}}^\top, {x_1}^\top, {d^b_{f_1}}^\top, ..., {x_{n-1}}^\top, {d^b_{f_{n-1}}}^\top, {x_n}^\top, {l_1}^\top, ..., {l_m}^\top]^\top$

1-Measurement preprocessing:

for *every new image* **do**

-Existing features are tracked and new features are detected to keep a minimum number of features;

-Check for being keyframe or not;

-IMU and thrust measurements preintegrations (4.14), (4.42) and covariance propagation (4.18), (4.48);

end

2-Initialization;

while *Not initialized* **do**

-Up-to-scale vision-only pose estimation using the five-point algorithm and PnP method;

-Loosely-coupled alignment of vision-only estimates with preintegrated IMUs → velocity, gravity and scale;

-World frame is established and all variables are translated to the world frame;

end

3-Tightly-Coupled Optimization:

for *every new image* **do**

The cost function (4.6) is formed;

for *every optimization iteration* **do**

Update the solution by Quasi-Newton method;

if $\delta d^b_{f_k}$, δb_{ω_k}, δb_{a_k} *are below a certain threshold* **then**

Update the preintegrations with their 1st order approximation (4.19), (4.49);

else

Repropagate (4.14), (4.42);

end

end

end

4.6 Simulation Results

In order to validate the performance we test the proposed method on public datasets and compare it with the state-of-the-art algorithms. We have implemented our method on top of VIMO [27] which itself is based on VINS [26], a widely used open-source visual inertial odometry system. We use the BlackBird Dataset [109] which is a large scale, aggressive flight dataset for a custom-built quadrotor platform. Although stereo images are provided in the BlackBird Dataset, we only use the images from the left camera as visual data to be used in our algorithm. The speed of the IMU measurements is 100 Hz, and the frequency of motor speed sensors (as the actuator data that are converted to thrust using the thrust constant $C_T = 2.03 \times 10^{-8}$ N/rpm^2) is 190 Hz. The images are provided by photorealistic virtual cameras at the frequency of 120 Hz, but in our implementation the speed is down-sampled to 30 Hz as we have the assumption that the IMU data are much faster than camera measurements. A ground truth for the pose of the quadrotor is provided at the rate of 360Hz with millimeter accuracy. The testing is run on a desktop computer with 16GB of RAM and an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz. The estimates are published at 10 Hz. The algorithm parameters are considered as follows: image size: 1024 px $\times$ 768 px, projection parameters: $f_x = 665.108$ mm, $f_y = 665.108$ mm, $c_x = 511.5$ px, $c_y = 383.5$ px. Maximum number of features: 150, minimum distance between two features: 30, maximum number of optimization iteration 10. Gyroscope measurement noise standard deviation: $\sigma_\omega = 0.1 I_3$ rad/s $\sqrt{\text{Hz}}$, accelerometer bias random work noise standard deviation: $\sigma_a = 0.001 I_3$ m/s^2 $\sqrt{\text{Hz}}$, gyroscope bias random work noise standard deviation: $\sigma_{b_\omega} = 0.003 I_3$ rad/s^2 $\sqrt{\text{Hz}}$, mass normalized collective thrust in the body z axis noise standard deviation $\eta_{T3} = 0.01$ N/kg, mass normalized collective thrust in the body x and y axis noise standard deviation: $\eta_{T1} = \eta_{T2} = 0.005$ N/kg. No loop closure were applied for a fair evaluation of the proposed method and performance comparison between algorithms. A snapshot of the running simulation visualized in Rviz, the 3D visualization tool for ROS, is shown in Figure 4.5.

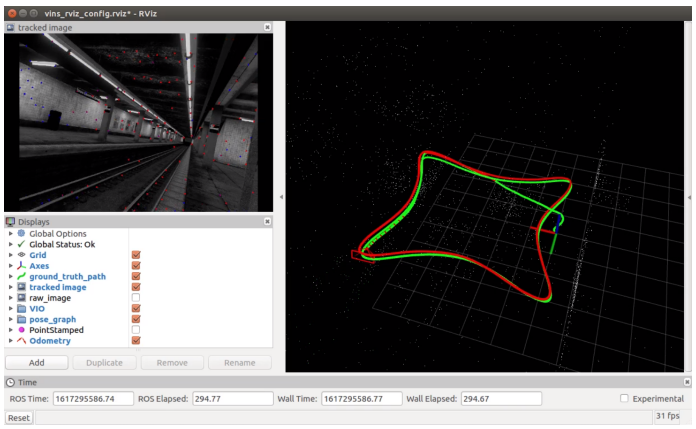
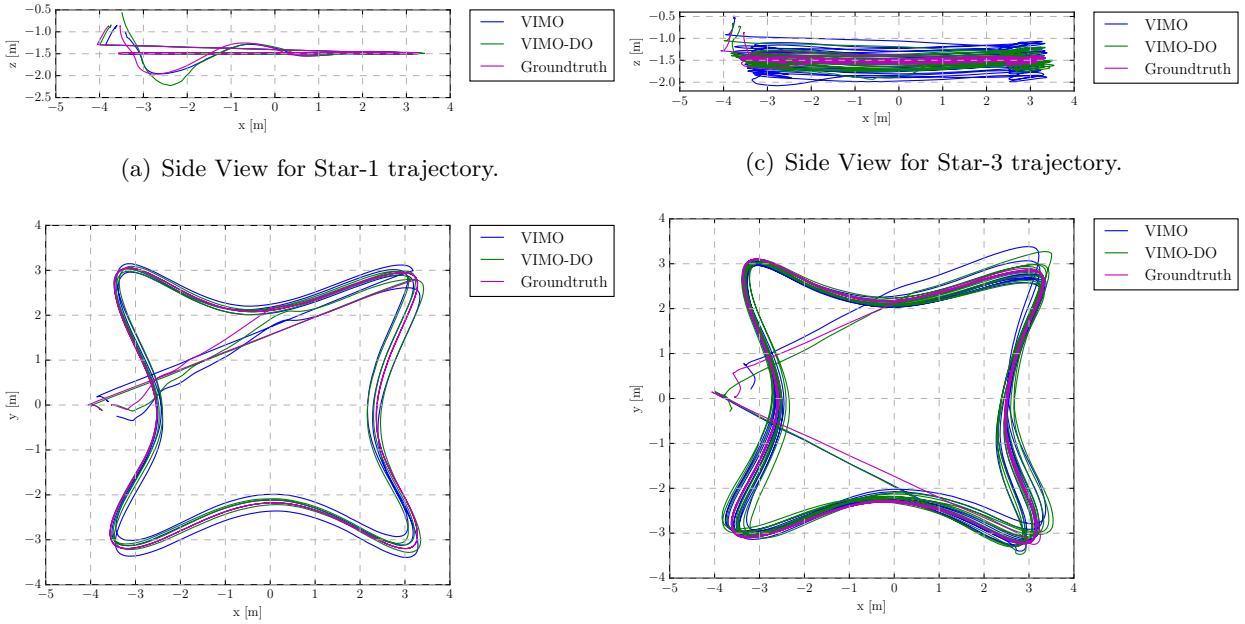


Figure 4.5: Snapshot of the simulation running on the Blackbird datasets.

The dataset used for the evaluation are the star trajectories with forward yaw (the yaw angle is

aligned with the tangent of the trajectory) and the maximum speed of $1\,\mathrm{ms}^{-1}$ (referred to as Star-1) and $3\,\mathrm{ms}^{-1}$ (referred to as Star-3). The parameter $d_{11} = 1\ \mathrm{s}^{-1}$ was hand-tuned after repeating the simulation and checking for performance improvement. However, after the initial tuning the parameters were kept the same for all other experiments. We repeated the simulations 10 times for each trajectory and logged the data. For analysing the data and comparing the performance with the ground truth, we used the trajectory evaluation toolbox [110]. We use the absolute trajectory error (ATE) as a criteria for performance evaluation and compare our method against VIMO [27]. ATE is defined as the RMS value of the alignment error between the trajectory estimate and the ground truth. Figure 4.6 shows the side and top views of the ground truth, VIMO and our proposed method. From the top views it can be seen that our method outputs a trajectory closer to the ground truth than VIMO. Also, not obvious in the Star-1 trajectory, but it can be easily seen in the side view of Star-3 trajectory that our method has a better performance with respect to the VIMO estimating in the z component of the position. These results are consistent with the numerical results presented in Table 4.1, which shows our proposed has improved the performance by 23.8% for Star-1 and 18% for Star-3 trajectories.



(a) Side View for Star-1 trajectory.

(c) Side View for Star-3 trajectory.

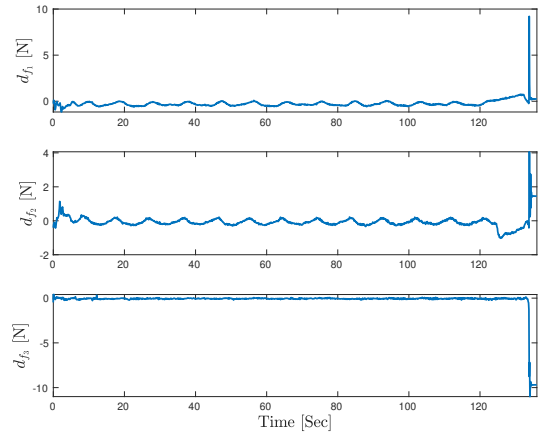
(b) Top View for Star-1 trajectory.

(d) Top View for Star-3 trajectory.

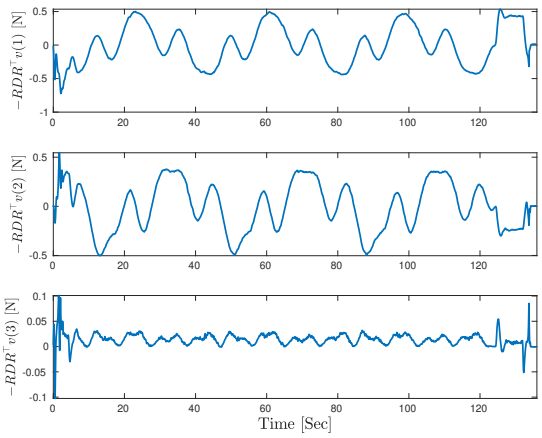
Figure 4.6: Side view and top view of the ground truth, trajectory estimation by VIMO and our proposed method.

Table 4.1: Average ATE for star trajectory (10 runs)

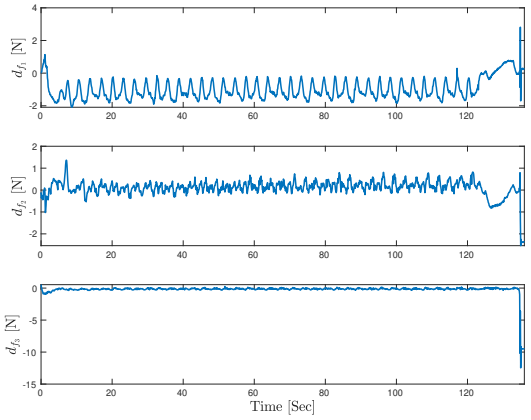
Traj.	VIMO (tran.)	VIMO-DO (tran.)	% Increase	VIMO (rot.)	VIMO-DO (rot.)	% Increase
Star-1	0.180 m	0.137 m	23.8%	0.142 deg	0.150 deg	−5.6%
Star-3	0.343 m	0.280 m	18%	0.620 deg	0.526 deg	15.1%



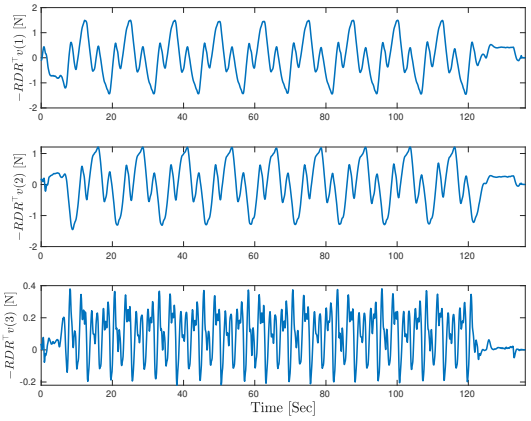
(a) Force disturbance estimate for Star-1 trajectory.



(b) Drag force for Star-1 trajectory.



(c) Force disturbance estimate Star-3 trajectory.



(d) Drag force Star-3 trajectory.

Figure 4.7: Force disturbance estimate and drag force for the proposed method.

The plots of the estimated force disturbance and rotor drag is also shown in Figure 4.7. The reason we selected the star trajectory is that it was the only trajectory that had an acceptable performance while the loop closure was disabled, which was necessary for a fair comparison with VIMO for the purpose of evaluation of our contribution in the preintergation. Also the average computation time of our method was similar to VIMO method.

One drawback of the Blackbird Dataset is it has no sequences where the multirotor is subject to an actual external force disturbance (e.g., wind-gust or interaction with the environment) so that we can better highlight the advantage of our method. As mentioned in [27], in the case where the multirotor is subject to a an external force that has a constant or slowly time-varying component, the VIMO estimates the constant part as the accelerometer bias which is not ideal, specifically in the applications where an accurate estimate of the external force is needed (e.g., maintenance of infrastructure using an aerial manipulator). Also, the other drawback is that the Blackbird

Dataset provides the commanded rotor speeds which are different from the actual rotor speeds that affect the vehicle dynamics. This difference which is detected as force disturbance can hinder the accuracy of the estimation, however it can be mitigated in flight testing by employing the actual rotor speeds which can be obtained from certain electronic speed controllers (ESC).

4.7 Summary

In this chapter we investigated the problem of multirotor state, external force and map estimation using visual, inertial and actuator data. We extended a VIO system by incorporating a dynamic residual formed from a revisited model of a multirotor dynamics subject to rotor drag and actuator data. Adding dynamic residual enables the estimator to estimate the difference between the dynamic and the inertial residuals as external force which is useful in applications where an external force estimate is needed and also can increase the odometry accuracy. We also considered a more realistic assumption of constant or slowly time-varying force disturbance in the problem formulation instead of a traditional model of zero mean Gaussian noise model which is not a proper representation for the disturbances arising from wind-gust or interaction with the environment. This assumption is implemented by deriving a residual term from a disturbance observer proposed for UAVs and including it into the optimization. This enables the estimator to differentiate between the drag force and the external disturbance forces which is important for applications where an exact estimate of the external force is needed. The proposed method was implemented on top of an open source VIO system VIMO [27] and was tested on benchmark dataset for performance evaluation, showing a significant improvement in the estimation accuracy with respect to VIMO.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis we investigated the problem of robust motion control design for the position and yaw tracking control for a multirotor UAV in the presence of external force and torque disturbances. First we presented the quadrotor modelling and derived an expression for rotor drag force which is proven to be important. Then we presented the ANCL quadrotor platform that employed for testing and evaluation of the proposed motion controls.

In this thesis we proposed motion control based on backstepping design that were using estimates from nonlinear disturbance observers to compensate for the effect of disturbances. The observer was coupled to the backstepping controller by adding extra terms to its dynamics to improve the transient response. The coupled structure in conjunction with the concept of tuning function lead to a rigorous exponential stability readily deducted from Lyapunov theorem for the case of constant disturbances. Ultimate boundedness is shown when disturbances are time-varying. The proposed controller was evaluated in Matlab simulation however control expressions were too complicated for on-board implementation.

In order to simplify the control expressions we employed a decoupled structure. Due to the cascade structure, convergence of the tracking error is not readily implied from the derivative of the Lyapunov function which the common procedure for the stability proof in backstepping method. However, by checking the system dynamics it can be seen that the decoupled structure converts the tracking error dynamics to a LTV system with the disturbance estimate error acting as a system input. Using LTV system stability analysis, exponential stability is proven in the presence of constant disturbances. The method is validated in SITL simulation and flight experiment using the commonly used PX4 autopilot firmware. It was also compared with a stock PX4 inner outer loop control and a traditional backstepping method demonstrating improved tracking performance and robustness relative to these controllers.

In this thesis we employed a more accurate modelling for the multirotor UAV by considering the effect of rotor drag in the translational dynamics and compensating its effect in the motion control

by decomposing it into two components. The first one is independent of the vehicle attitude and can be readily compensated by an appropriate choice of virtual control in the backstepping procedure. The second component of the drag force is in the direction of thrust and can be compensated by an appropriate choice for desired thrust during backstepping. Also integral augmentation were employed for increasing the robustness to external disturbances and model uncertainties. Exponential stability of the closed loop is proven assuming constant wind velocity and constant disturbances. SITL simulations were presented demonstrating improved controller performance of the method, and the importance of rotor drag compensation. The simulations illustrate the approach can compensate for time-varying disturbances and parametric uncertainty.

In this thesis we extended a visual-inertial odometry system by incorporating a dynamic residual formed from a revisited model of a multirotor dynamics subject to rotor drag and actuator data. Adding dynamic residual enabled the estimator to estimate the difference between the dynamic and the inertial residuals as external force which is useful in applications where an external force estimate is needed and also can increase the odometry accuracy. We also considered a more realistic assumption of constant or slowly time-varying force disturbance in the problem formulation instead of a traditional model of zero mean Gaussian noise model which is not a proper representation for the disturbances arising from wind-gust or interaction with the environment. We implemented this assumption by deriving a residual term from a disturbance observer proposed for UAVs and including that residual into the optimization. This enabled the estimator to differentiate between the drag force and the external disturbance forces which is important for applications where an exact estimate of the external force is needed. The proposed method is implemented on top of an open source VIO system VIMO [27] and have been tested on benchmark dataset for performance evaluation, showing a significant improvement in the estimation accuracy with respect to VIMO.

5.2 Future work

For the future work, the decoupled control-observer design with integral augmentation and rotor drag compensation (Section 3.4) can be experimentally tested on the ANCL-Q3 platform under disturbances exerted from an industrial fan. Also there is still a lot of research than can improve the design performance. For example although it has been shown the proposed method is robust with respect to model uncertainty, the design can be modified by incorporation of adaption laws for mass and inertia in order to increase the performance.

Also dataset recording is another step that we are planning for by flying the ANCL-Q3 withing the lab environment and collecting visual data, IMU measurements and rotor speeds data. The dataset can be used to test our proposed algorithm and evaluate its performance. The advantage of such a dataset is that we can exert external disturbances with known values as use them as groundthruth to evaluate the accuracy of the force disturbance estimate form the proposed method. This is a significant advantage as the dataset that we used for evaluation [109] does not include any trajectory where the multirotor is under external disturbances. Also the actual rotor speeds, which

are different from the commonly used commanded speeds, can be used as actuator measurements.

The ultimate goal is to run the proposed motion control using the estimates from the proposed state estimation algorithm (rather than MCS) and evaluate the closed-loop performance as this was the ultimate goal of this research: increasing the autonomy and reliability of multirotor UAVs by developing robust controller and estimation algorithms relying on onboard sensors.

Bibliography

[1] D. Cabecinhas, R. Cunha, and C. Silverstre, "A nonlinear quadrotor trajectory tracking controller with disturbance rejection," *Control Eng. Practice*, vol. 26, pp. 1–10, 2014.

[2] S. Tang and V. Kumar, "Autonomous flight," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 29–52, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105149

[3] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315–378, March/April 2012.

[4] S. Paschall and J. Rose, "Fast, lightweight autonomy through an unknown cluttered environment: Distribution statement: A — approved for public release; distribution unlimited," in *2017 IEEE Aerospace Conference*, 2017, pp. 1–8.

[5] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles.* Springer, 2015, vol. 2077.

[6] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE Contr. Sys. Mag.*, vol. 33, no. 1, pp. 61–75, 2013.

[7] N. Cao and A. F. Lynch, "Inner-outer loop control with constraints for rotary-wing UAVs," in *Proc. IEEE Int. Conf. on Unmanned Aircraft Systems*, Denver, CO, June 2015, pp. 294–302.

[8] G. Antonelli, E. Cataldi, F. Arrichiello, P. R. Giordano, S. Chiaverini, and A. Franchi, "Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances," *IEEE Trans. Contr. Syst. Technol.*, vol. 26, no. 1, pp. 248–254, January 2018.

[9] W. Dong, G.-Y. Gu, X. Zhu, and H. Ding, "High-performance trajectory tracking control of a quadrotor with disturbance observer," *Sens. Actuators, A*, vol. 211, pp. 67–77, 2014.

[10] S. Dai, T. Lee, and D. S. Bernstein, "Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6149–6154.

[11] J. X. J. Bannwarth, Z. J. Chen, K. A. Stol, and B. A. MacDonald, "Disturbance accomodation control for wind rejection of a quadcopter," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 695–701.

[12] S. Kim, S. Choi, H. Kim, J. Shin, H. Shim, and H. J. Kim, "Robust control of an equipment-added multirotor using disturbance observer," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1524–1531, 2018.

[13] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, July 2018.

[14] C. Wang, B. Song, P. Huang, and C. Tang, "Trajectory tracking control for quadrotor robot subject to payload variation and wind gust disturbance," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 2, pp. 315–333, 2016.

[15] M. Faessler, A. Franchi, and D. Scaramuzza, "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.

[16] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8189 – 8195, 2017, 20th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896317317822

[17] B. Herisse, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow," *IEEE Trans. Robot.*, 2011, early Access.

[18] B. Ludington, E. Johnson, and G. Vachtsevanos, "Augmenting UAV autonomy," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 63–71, 2006.

[19] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I: The first 30 years and fundamentals," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, December 2011.

[20] F. Fraundorfer and D. Scaramuzza, "Visual odometry : Part II: Matching, robustness, optimization, and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 78–90, June 2012.

[21] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev*, vol. 43, no. 1, pp. 55–81, January 2015.

[22] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, ETH Zürich, Zürich, Switzerland, 2012.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORb-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, October 2015.

[24] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, April 2017.

[25] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaios, and I. Rekleitis, "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7227–7233.

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

[27] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, "Vimo: Simultaneous visual inertial model-based odometry and force estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.

[28] C. Liu, J. Pan, and Y. Chang, "PID and LQR trajectory tracking control for an unmanned quadrotor helicopter: Experimental studies," in *2016 35th Chinese Control Conference (CCC)*, July 2016, pp. 10 845–10 850.

[29] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, *Autonomous Control of a Mini Quadrotor Vehicle Using LQG Controllers*. Tokyo: Springer Japan, 2010, pp. 61–76. [Online]. Available: https://doi.org/10.1007/978-4-431-53856-1_3

[30] P. Castillo, R. Lozano, and A. Dzul, *Modelling and control of mini flying machines*. New York City, USA: Springer-Verlag, 2005.

[31] F. Chen, W. Lei, K. Zhang, G. Tao, and B. Jiang, "A novel nonlinear resilient control for a quadrotor UAV via backstepping control and nonlinear disturbance observer," *Nonlinear Dynamics*, vol. 85, no. 2, pp. 1281–1295, Jul 2016. [Online]. Available: https://doi.org/10.1007/s11071-016-2760-y

[32] T. J. Koo and S. Sastry, "Output tracking control design of a helicopter model based on approximate linearization," in *IEEE Conference on Decision and Control*, Tampa, FL, December 1998, pp. 3635–3640.

[33] A. Aboudonia, A. El-Badawy, and R. Rashad, "Disturbance observer-based feedback linearization control of an unmanned quadrotor helicopter," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 230, no. 9, pp. 877–891, 2016. [Online]. Available: https://doi.org/10.1177/0959651816656951

[34] M. J. Reinoso, L. I. Minchala, P. Ortiz, D. F. Astudillo, and D. Verdugo, "Trajectory tracking of a quadrotor using sliding mode control," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2157–2166, May 2016.

[35] T. Wang, H. Parwana, K. Umemoto, T. Endo, and F. Matsuno, "Non-cascade adaptive sliding mode control for quadrotor UAVs under parametric uncertainties and external disturbance with indoor experiments," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 8, Apr 2021. [Online]. Available: https://doi.org/10.1007/s10846-021-01351-z

[36] G. Perozzi, D. Efimov, J.-M. Biannic, and L. Planckaert, "Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains," *Journal of the Franklin Institute*, vol. 355, no. 12, pp. 4809–4838, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003218302965

[37] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, 2011.

[38] A. Poutney, C. R. Kennedy, G. M. Clayton, and H. Ashrafiuon, "Robust tracking control of quadrotors based on differential flatness: Simulations and experiments," *IEEE/ASME Trans. Mechatronics*, p. 1, 2018.

[39] L. Meier, "PX4 autopilot," Institute for Visual Computing, Swiss Federal Institute of Technology Zurich, 2018. [Online]. Available: https://px4.io/

[40] M. Labbadi and M. Cherkaoui, "Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV," *Aerospace Science and Technology*, vol. 93, p. 105306, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963819300318

[41] X. Shao, J. Liu, H. Cao, C. Shen, and H. Wang, "Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer," *Int. J. Robust Nonlin.*, vol. 28, pp. 2700–2719, 2018.

[42] W. Xie, D. Cabecinhas, R. Cunha, and C. Silvestre, "Adaptive backstepping control of a quadcopter with uncertain vehicle mass, moment of inertia, and disturbances," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021.

[43] Y. Zou, "Nonlinear robust adaptive hierarchical sliding mode control approach for quadrotors," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 6, pp. 925–941, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.3607

[44] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, pp. 1327–1361, 1995.

[45] D. Mellinger, "Trajectory generation and control for quadrotors," Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA, 2012.

[46] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor UAV on SE(3)," *Asian J. Control*, vol. 15, no. 2, pp. 391–408, 2013.

[47] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-body attitude control: Using rotation matrices for continuous, singularity-free control laws," *IEEE Contr. Sys. Mag.*, vol. 31, no. 3, pp. 30–51, June 2011.

[48] M. van Nieuwstadt and R. M. Murray, "Outer flatness: Trajectory generation for a model helicopter," in *European Control Conference*, Brussels, Belgium, July 1997, pp. 325–330.

[49] M. Kristic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Wiley and Sons, 1995.

[50] A. Dzul, T. Hamel, and R. Lozano, "Helicopter's nonlinear control via backstepping techniques," in *Proc. European Control Conf. (ECC)*, Porto, Portugal, September 2001, pp. 463–468.

[51] T. Hamel and R. Mahony, "Pure 2D visual servo control for a class of under-actuated dynamic systems," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, 2004, pp. 2229–2235.

[52] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an X4-Flyer," in *Proc. IFAC World Congress*, 2002, pp. 217–222.

[53] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, École Polytechnique Fédérale, Lausanne, Switzerland, 2007.

[54] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on SE(3) for agile maneuvers," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 137, no. 9, pp. 091 007–1–12, June 2015.

[55] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, "Nonlinear control of VTOL UAVs incorporating flapping dynamics," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Tokyo, November 2013, pp. 2419–2425.

[56] B. Sumantri, N. Uchiyama, S. Sano, and Y. Kawabata, "Robust tracking control of a quadrotor helicopter utilizing sliding mode control with a nonlinear sliding surface," *Journal of System Design and Dynamics*, vol. 7, no. 2, pp. 226–241, 2013.

[57] T. Madani and A. Benallegue, "Backstepping sliding mode control applied to a miniature quadrotor flying robot," in *Proc. IECON 2006 - 32nd Annual Conf. IEEE Industrial Electronics*, Paris, France, November 2006, pp. 700–705.

[58] Y. He, H. Pei, and T. Sun, "Robust tracking control of helicopters using backstepping with disturbance observers," *Asian J. of Control*, vol. 16, no. 5, pp. 1387–1402, 2014.

[59] S. J. Lee, S. Kim, K. H. Johansson, and H. J. Kim, "Robust acceleration control of a hexarotor UAV with a disturbance observer," in *Conf. Decision Control*, 2016, pp. 4166–4171.

[60] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. New York, NY: John Wiley & Sons, 1995.

[61] F. Mazenc and A. Iggidr, "Backstepping with bounded feedbacks," *Syst. Control Lett.*, vol. 51, pp. 235–245, 2004.

[62] D. Cabecinhas, R. Cunha, and C. Silvestre, "A globally stabilizing path following controller for rotorcraft with wind disturbance rejection," *IEEE Trans. Contr. Syst. Technol.*, vol. 23, pp. 708–714, 2015.

[63] J. Colmenares-Vazquez, N. Marchand, P. Castillo, J. E. Gomez-Balderas, J. U. Alvarez-Munoz, and J. J. Tellez-Guzman, "Integral backstepping control for trajectory tracking of a hybrid vehicle," in *Int. Conf. Unmanned Aircraft Systems*, 2015, pp. 209–217.

[64] A. Poultney, P. Gong, and H. Ashrafiuon, "Integral backstepping control for trajectory and yaw motion tracking of quadrotors," *Robotica*, vol. 37, no. 2, pp. 300–320, 2019.

[65] W. Jasim and D. Gu, "Integral backstepping controller for quadrotor path tracking," in *Int. Conf. Advanced Robotics*, 2015, pp. 593–598.

[66] Z. Jia, J. Yu, Y. Mei, Y. Chen, Y. Shen, and X. Ai, "Integral backstepping sliding mode control for quadrotor helicopter under external uncertain disturbances," *Aerospace Science and Technology*, vol. 68, pp. 299–307, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963817308878

[67] R. Zhang, Q. Quan, and K. Y. Cai, "Attitude control of a quadrotor aircraft subject to a class of time-varying disturbances," *IET Control Theory Appl.*, vol. 5, no. 9, pp. 1140–1146, June 2011.

[68] S. Li, J. Yang, W. H. Chen, and X. Chen, *Disturbance observer-based control: methods and applications*, 1st ed. CRC press, 2014.

[69] W.-H. Chen, "Disturbance observer based control for nonlinear systems," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 4, pp. 706–710, Dec 2004.

[70] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 45, no. 10, pp. 1893–1899, October 2000.

[71] K. Guo, J. Jia, X. Yu, L. Guo, and L. Xie, "Multiple observers based anti-disturbance control for a quadrotor UAV against payload and wind disturbances," *Control Engineering Practice*, vol. 102, p. 104560, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066120301544

[72] K. Sasaki and Z.-J. Yang, "Disturbance observer-based control of UAVs with prescribed performance," *International Journal of Systems Science*, vol. 51, no. 5, pp. 939–957, 2020. [Online]. Available: https://doi.org/10.1080/00207721.2020.1746436

[73] M. Bangura, "Aerodynamics and control of quadrotors," Ph.D. dissertation, College of Engineering and Computer Science, The Australian National University, 2017.

[74] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual SLAM," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4523–4530.

[75] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2012, pp. 957–964.

[76] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.

[77] A. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, April 2007, pp. 3565–3572.

[78] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011. [Online]. Available: http://ijr.sagepub.com/content/30/1/56.abstract

[79] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, November 2011.

[80] D. M. W. Abeywardena and G. Dissanayake, "Tightly-coupled model aided visual-inertial fusion for quadrotor micro air vehicles," in *FSR*, ser. Springer Tracts in Advanced Robotics, vol. 105. Springer, 2013, pp. 153–166.

[81] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005. [Online]. Available: https://books.google.ca/books?id=2Zn6AQAAQBAJ

[82] D. Strelow and S. Singh, "Motion estimation from image and inertial measurements," *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1157–1195, 2004. [Online]. Available: https://doi.org/10.1177/0278364904045593

[83] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended kalman filter based SLAM," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 473–479.

[84] A. Barrau, "Non-linear state error based extended kalman filters with applications to navigation," Ph.D. dissertation, Mines Paristech, 2015.

[85] B. Bell and F. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[86] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE Int. Sym. on Mixed and Augmented Reality*, Nara, Japan, November 2007, pp. 225–234.

[87] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual SLAM: why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[88] A. Moeini, A. Lynch, and Q. Zhao, "Disturbance observer-based nonlinear control of a quadrotor UAV," *Adv. Control Appl.*, vol. 2, no. 1, pp. 1–20, 2019.

[89] A. Moeini, A. F. Lynch, and Q. Zhao, "A backstepping disturbance observer control for multirotor UAVs: theory and experiment," *International Journal of Control*, vol. 0, no. 0, pp. 1–15, 2021. [Online]. Available: https://doi.org/10.1080/00207179.2021.1912393

[90] ——, "Exponentially stable motion control for multirotor uavs with rotor drag and disturbance compensation," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 1, p. 15, Aug 2021. [Online]. Available: https://doi.org/10.1007/s10846-021-01452-9

[91] P. Martin and E. Salaün, "Design and implementation of a low-cost observer-based attitude and heading reference system," *Control Engineering Practice*, vol. 18, no. 7, pp. 712–722, July 2010.

[92] A. Moeini and A. Lynch, "Modified px4 autopilot firmware," 2020. [Online]. Available: https://github.com/ANCL/UAV_DOBBS

[93] ——, "Modified px4 autopilot firmware," 2020. [Online]. Available: https://github.com/ANCL/UAV_DOBIBS_V2

[94] P. Martin and E. Salaün, "An embedded attitude and heading reference system based on a nonlinear filter," in *Informatics in Control, Automation and Robotics*, ser. Lecture Notes in Electrical Engineering, J. Andrade-Cetto, J.-L. Ferrier, and J. Filipe, Eds. Springer Berlin Heidelberg, 2009, vol. 37, pp. 267–281.

[95] M. D. Shuster, "A survey of attitude representations," *Navigation*, vol. 41, pp. 439–517, October 1993.

[96] L. Meier, P. Tanskanen, L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[97] G. Fink, "Computer vision-based motion control and state estimation for unmanned aerial vehicles (UAVs)," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2018.

[98] L. Meier, "QGroundControl," http://www.qgroundcontrol.org/ [accessed 01 Jan 2018], Institute for Visual Computing, Swiss Federal Institute of Technology Zurich, 2017. [Online]. Available: http://www.qgroundcontrol.org/

[99] ——, "Mavlink: Micro air vehicle communication protocol," 2015. [Online]. Available: https://mavlink.io/en/

[100] H. Xie, "Dynamic visual servoing of rotary wing unmanned aerial vehicles," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2016.

[101] N. Cao, "Control of quadrotor unmanned aerial vehicles with saturation and time delay," Ph.D. dissertation, University of Alberta, Edmonton, Alberta, September 2018.

[102] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly, "A nonlinear disturbance observer for robotic manipulators," *IEEE Trans. Ind. Electr.*, vol. 47, no. 4, pp. 932–938, August 2000.

[103] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[104] A. Loría and E. Panteley, *Cascaded Nonlinear Time-Varying Systems: Analysis and Design*, F. Lamnabhi-Lagarrigue, A. Loría, and E. Panteley, Eds. London: Springer, 2005. [Online]. Available: https://doi.org/10.1007/11334774_2

[105] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, November 2013, pp. 1280–1286.

[106] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 932–945, October 2008.

[107] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.

[108] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, Jul 2008. [Online]. Available: https://doi.org/10.1007/s11263-008-0152-6

[109] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The blackbird UAV dataset," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1346–1364, 2020. [Online]. Available: https://doi.org/10.1177/0278364920908331

[110] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251. [Online]. Available: https://github.com/uzh-rpg/rpg_trajectory_evaluation