ABSTRACT

Title of Dissertation: Applied Aerial Autonomy for Reliable Indoor Flight and 3D Mapping

Animesh K. Shastry
Doctor of Philosophy, 2024

Dissertation Directed by: Professor Derek A. Paley
Department of Aerospace Engineering

Uncrewed Aerial Systems (UAS) are essential for safely exploring indoor environments damaged by shelling, fire, floods, and structural collapse. These systems can gather critical visual and locational data, aiding in hazard assessment and rescue planning without risking human lives. Reliable UAS deployments requires advanced sensors and robust algorithms for real-time data processing and safe navigation, even in GPS-denied and windy conditions. This dissertation details three research projects to improve UAS performance: (1) in-flight calibration to improve estimation and control, (2) system identification for wind rejection, and (3) indoor aerial 3D mapping.

The dissertation begins with introducing a comprehensive nonlinear filtering framework for UAV parameter estimation, which considers factors such as external wind, drag coefficients, IMU bias, and center of pressure. Additionally, it establishes optimized flight trajectories for parameter estimation through empirical observability. Moreover, an estimation and control framework is implemented, utilizing the mean of state and parameter estimates to generate suitable control

inputs for vehicle actuators. By employing a square-root unscented Kalman filter (sq-UKF), this framework can derive a 23-dimensional state vector from 9-dimensional sensor data and 4-dimensional control inputs. Numerical results demonstrate enhanced tracking performance through the integration of the estimation framework with a conventional model-based controller. The estimation of unsteady winds results in improved gust rejection capabilities of the onboard controller as well.

Closely related to parameter estimation is system identification. Combining with the previous work a comprehensive system identification framework with both linear offline and nonlinear online methods is introduced. Inertial parameters are estimated using frequency-domain linear system identification, incorporating control data from motor-speed sensing and state estimates from automated frequency sweep maneuvers. Additionally, drag-force coefficients and external wind are recursively estimated during flight using a sq-UKF. A custom flight controller is developed to manage the computational demands of online estimation and control. Flight experiments demonstrate the tracking performance of the nonlinear controller and its improved capability in rejecting gust disturbances.

Aside from wind rejection, aerial indoor 3D mapping is also required for indoor navigation, and therefore, the dissertation introduces a comprehensive pipeline for real-time mapping and target detection in indoor environments with limited network access. Seeking a best-in-class UAS design, it provides detailed analysis and evaluation of both hardware and software components. Experimental testing across various indoor settings demonstrates the system's efficacy in producing high-quality maps and detecting targets.

Applied Aerial Autonomy for Reliable Indoor Flight and 3D Mapping

by

Animesh K. Shastry

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Derek A. Paley, Chair/Advisor
Professor Dinesh Manocha
Professor Inderjit Chopra
Professor Michael W. Otte
Professor Robert M. Sanner

Dedication

To my wife, Raquel, and our beloved dog, Louis, your warmth and companionship made this journey so much more rewarding. Raquel, your unwavering love and support gave me strength during the tough times and joy in the good moments. Louis, your playful presence brought needed relaxation and happiness. I am deeply grateful to both of you for making this experience truly special.

Acknowledgments

I would like to express my profound gratitude to my advisor, Prof. Derek Paley, for granting me the opportunity to pursue a Ph.D. in aerospace engineering and for imparting a wealth of knowledge in dynamics, control theory, and autonomous systems. His expert guidance has been instrumental in my development as a researcher, helping me hone my ability to present complex ideas clearly and engagingly to a diverse audience. Dr. Paley's mentorship not only sharpened my technical skills but also broadened my intellectual horizons. He inspired me to delve deeper into the theoretical underpinnings of engineered systems, beyond the practical aspects of building robotic systems and conducting experiments.

I also extend my heartfelt thanks to my dissertation committee for their invaluable time, insightful feedback, and the profound lessons they have imparted throughout my Ph.D. journey. Prof. Inderjit Chopra, Prof. Robert Sanner, Prof. Michael Otte, and Prof. Dinesh Manocha have provided guidance and support that have been crucial to the success of this work. Their expertise and constructive critiques have significantly enriched my research.

I am deeply grateful to the Collective Dynamics and Control Laboratory (CDCL) and my CDCL labmates for their unwavering support, assistance, and the engaging discussions that have greatly contributed to my research experience. The collaborative environment and intellectual stimulation within the lab have been vital to my academic growth. Additionally, I am thankful to the Maryland Robotics Center (MRC) for providing essential equipment, access to advanced lab facilities, and logistical support, all of which have been pivotal in ensuring the smooth conduct of my research.

My sincere thanks also go to the Autonomous Micro Aerial Vehicle (AMAV) team. The opportunity to lead and collaborate with such a talented group over the years has been a highlight of my Ph.D. journey. Working with the team has not only been an enjoyable experience but has also provided me with valuable leadership and teamwork skills. I am particularly grateful to my collaborators, including Wei Cui, Ahmed Ashry, and Srijal Poojari, whose contributions were crucial to the successful completion of this dissertation. Their dedication, insights, and expertise have been invaluable.

I would like to express my deepest gratitude to my family and friends for their unwavering support throughout my Ph.D. journey. Your encouragement, understanding, and patience have been a source of strength and motivation. To my family, thank you for your companionship, whether celebrating small victories or offering a listening ear during tough moments. Special thanks to Abhishek and Lea, your constant love and belief in me have kept me grounded and inspired, especially during the most challenging times. Your support has been indispensable, and this achievement would not have been possible without all of you. I am forever grateful.

Finally, I would like to express my gratitude to the sponsors of this research. The financial support from the Vertical Lift Research Center of Excellence (VLRCOE 2017–2021, grant number W911W61720004), the Artificial Intelligence and Autonomy for Multi-Agent Systems (ArtIAMAS 2022, Army cooperative agreement number W911NF2120076), and the Vertical Lift Research Center of Excellence (VLRCOE 2023–2024, grant number W911W62120003) made the projects discussed in this dissertation possible. Their support has been instrumental in advancing my research.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

This chapter serves as an introduction to the dissertation, beginning with an exploration of the motivation driving the research. It then provides a comprehensive review of prior works, positioning them in relation to current state-of-the-art developments. Following this, the chapter outlines the key contributions of the research. It concludes with an overview of the structure and content of the subsequent chapters.

1.1 Motivation

Uncrewed Aerial Systems (UAS) are a valuable tool in exploring indoor environments that have been compromised by severe damage due to shelling, fire, flood, structural collapse, and other potentially dangerous situations. These environments often pose significant risks to human responders, making it crucial to gather detailed visual and locational information before commencing rescue operations. This information is vital for assessing the condition of the environment, identifying potential hazards, and planning the safest and most effective approach for rescue teams without physically entering the compromised area.

Deploying UAS in such dangerous situations necessitates the careful identification, design, and integration of advanced sensors and algorithms. These systems must enable the UAS to safely navigate and operate in complex and unpredictable conditions. The integration of high-resolution cameras, thermal imaging sensors, LiDAR, and other advanced technologies allows the UAS to capture comprehensive data, even in environments with low visibility and multiple obstructions. Additionally, the algorithms must be sophisticated enough to process this data in real-time, providing immediate feedback and actionable insights to rescue teams.

Figure 1.1: In disaster scenarios before commencing rescue operations a UAS can enter dangerous damaged structures and provide valuable information about the environment leading to efficient and safe rescue. (Left: Associated press photo taken by Pavel Dorogoy. Middle: Artistic rendering of UMD Intrigue UAS. Right: Associated press photo taken by Andrew Marienko)

The UAS must be capable of handling various environmental challenges to ensure reliable performance. These challenges include but are not limited to wind gusts that can destabilize the aircraft, smoke that can obscure visual sensors, and obstructed pathways that require advanced navigation capabilities. Moreover, in many scenarios, lighting may be severely limited or entirely absent, necessitating the use of infrared or other non-visual sensors to maintain situational awareness. Operating in a GPS-denied infrastructure also demands robust autonomous navigation systems that can function independently of external positioning signals.

Overall, the successful deployment of UAS in hazardous indoor environments hinges on the seamless integration of cutting-edge technologies and intelligent design to overcome the myriad challenges posed by such conditions. This approach not only enhances the safety and effectiveness of rescue operations but also significantly reduces the risk to human life. This dissertation describes three applied research ventures that were undertaken to increase the resilience, reliability and performance of indoor aerial systems: (1) improving quadrotor control performance via in-flight calibration, (2) system identification for wind rejection, and (3) indoor aerial 3D mapping.

1.2 Relation to the State of the Art

The work in this dissertation is built on similar work done by others. In this section, the relation of this work to prior research is highlighted, and how this work pushes the envelope further is described. Firstly, prior work on techniques for improving quadrotor control performance via in-flight calibration is described. Secondly, a discussion on the various system identification techniques that have been developed in the literature and is currently in use in industry is presented. Finally, the advancements in 3D Mapping and autonomous technologies performed in academia are presented.

1.2.1 Improving Quadrotor Control Performance via In-flight Calibration

Unmanned aerial vehicles (UAVs) heavily rely on specialized estimation and control systems designed to precisely understand the vehicle's condition while ensuring stability and responsiveness to user commands. Linear estimation and control frameworks, though effective at handling external disruptions and uncertainties within the system, can struggle in situations requiring precise tracking due to their reliance on linear assumptions. Conversely, nonlinear model-based frameworks offer more flexibility, yet their effectiveness is contingent upon the accuracy of the dynamic model employed. Regardless of the approach, identifying system parameters, including calibrating sensors, is imperative. These processes ensure that the UAV operates optimally, delivering reliable performance and accurate responses in various conditions and tasks, from aerial surveillance to package delivery.

Conventional linear-system identification for quadrotors [1] relies on chirped frequency inputs in hover and batch post-processing. Furthermore, not all parameters/states are observable in hover, and stability derivatives are identified in stages. In contrast, a nonlinear estimation approach [2] indirectly selects frequencies based on parametrized trajectories, uses recursive estimation to identify all parameters simultaneously, and achieves weak observability almost globally. The system actually loses observability in hover.

Utilizing onboard sensor data for parameter estimation has been studied using an unscented Kalman filter (UKF) [3], but only the vehicle's inertia was estimated. In [2], estimation of 14 parameters was performed using a UKF in real time, but the estimation accuracies of only mass and inertia were analyzed. In [4], a genetic algorithm (GA) was used to estimate the center of mass in an online fashion. In [5], a GA was used to estimate the mass, inertia, thrust, and torque coefficients offline. In [6], the mass estimation performance of least-squares and extended Kalman filters and instrumental-variable algorithms were investigated in simulation.

Most prior works does not explicitly handle the low observability of UAV parameter estimation. In [7], the authors developed an observability-aware trajectory-optimization framework that produces optimal self-calibration trajectories. The novelty of the work described here is that the calibration trajectories are optimized based on observability, which enables numerically stable estimation of more parameters.

An alternate strategy to handle parameter estimation is using an adaptive control framework capable of handling structured and unstructured parameter uncertainties. $\mathcal{L}_1$-adaptive control is a recently developed controller [8] that requires the plant model to be in a certain specific form and, therefore, does not directly apply to the nonlinear dynamical model of UAVs. Moreover, researchers have pointed out several issues with the controller [9, 10]. Model Reference Adaptive Control (MRAC) is a popular adaptive controller with a rich literature [11, 12, 13]. However, the traditional methodology requires a strict matching condition between the structure of the plant dynamics and the model used for control derivation, which may be difficult to obtain *a priori*.

1.2.2 System Identification for Wind Rejection

A UAV relies on the performance of its flight controller's estimation and control algorithms to reconstruct its state and track reference trajectories. The conventional approach for implementing stable and reliable controllers for a UAV involves a trial-and-error procedure, where control gains are manually adjusted based on flight performance and the pilot's experience. Linear controllers such as the proportional–integral–derivative controller (PID) are almost universally used in robotics and aerospace control systems [14, 15]. Commercially available quadrotors and flight control firmware [16] such as PX4, Ardupilot, and Betaflight implement cascaded PID control for stabilization. However, there are two major drawbacks to linear control: (1) performance degrades in some situations where tracking precision is required due to the linearity assumption; and (2) gain tuning can be time-consuming and requires the presence of an experienced pilot. Furthermore, it is impractical to tune PID controllers for UAVs that pick up and carry payloads or packages of unknown size and weight.

A modern approach for UAV control utilizes model-based nonlinear control laws to ensure global stability of the vehicle. Feedback linearization controllers, also known as dynamic inversion controllers, achieve high control performance and have been extensively researched [17, 18].

The state-of-the-art for nonlinear quadrotor control are adaptive control [19] and geometric control [20], which have been widely used to demonstrate aggressive and agile flights. However, a major drawback to a model-based nonlinear control is that its performance depends on the accuracy of the model parameters [18]. The same problem affects UAV state estimation. For example, the Kalman filter is the state-of-the-art, but as a model-based framework, its performance is also limited by the accuracy of the dynamic model [21]. Widely accepted solutions to parameter uncertainty are to either directly measure system parameters before flight or use a system identification process in a controlled setting. These existing solutions can often be costly, risky, and/or require specialized equipment and experiments [22, 23], making them undesirable for rapid development of high-performance UAVs.

The system identification process involves utilizing flight testing data to develop a dynamic model of the aircraft. Linear methods involve measuring sensor outputs of the aircraft in response to inputs and computing the state-space representation of the aircraft. For a typical UAV, the inputs are the commands generated by the controller and the outputs are obtained from the state estimator. Using commanded control inputs is beneficial in settings where inertial characteristics of the system do not need to be precisely estimated. If the end goal is the implementation of a linear controller, then knowledge of the state-space model is more useful than inertial parameters. On the contrary, if a nonlinear framework is desired, then inertial parameters need to be estimated, which is difficult to extract from the identified model since it contains the inertial characteristics mixed in with the speed controller's response and structural characteristics of the rotor [24]. The state-of-the-art industry-standard in rotorcraft frequency-domain linear system identification is the Comprehensive Identification from FrEquency Response (CIFER®) [25] program developed by the Ames Research Center. The CIFER® tool is widely used for aircraft and rotorcraft system identification [26], including quadrotors; however, it is almost exclusively used for linear controller implementations [27, 28]. One novelty of this article is that it improves the inertial parameter estimation accuracy of the linear system identification technique by directly sensing the individual motor revolutions per minute (RPM) to get better values for the control inputs than a traditional approach. The nonlinear dynamic model parameters are extracted from the resulting linearized model and used for the nonlinear control implementation. We augment offline linear system identification of parameters with online recursive estimation of them.

Nonlinear approaches towards system identification rely on parameter estimation models that indirectly select frequencies based on parametrized trajectories, use recursive estimation to identify all parameters simultaneously, and achieve weak observability almost globally. Notably, a multi-rotor often loses observability in hover, a flight condition typically used in system identification. Recursive estimation of mass and inertia parameters have been demonstrated using only onboard sensor data with an unscented Kalman filter (UKF) [29], but the estimates in our testing were observed to be less accurate than frequency-domain system identification methods. Parameters related to aerodynamic effects have higher observability when the aircraft operates away from hover conditions, such as high-speed maneuvers and aerobatic flight, especially using onboard sensing of inertial velocity via visual odometry. Alternate approaches for parameter estimation include adaptive control [13] and probabilistic optimization [5]. Notable developments include recent developments of adaptive nonlinear dynamic inversion control [30, 31, 32] and adaptive recursive orthogonal least-squares [33] frameworks.

1.2.3 Indoor 3D Mapping & Autonomy

In today's rapidly evolving landscape of emergency response, the demand for enhanced tools to support first responders in navigating hazardous indoor environments is more pressing than ever. Incidents such as fires, floods, and structural collapses frequently compromise the safety and practicality of traditional rescue operations, necessitating innovative solutions that can provide real-time situational awareness without exposing personnel to undue risk. UAS have emerged as a pivotal technology in this regard, offering the capability to swiftly gather and relay critical information to incident command teams during high-stakes operations.

Despite significant advancements in robotics and UAS technology, the full potential of UAS as an invaluable tool for first responders remains largely untapped. Current applications have demonstrated the ability of UAS to operate in complex and dangerous environments, yet there is still a considerable gap in the widespread integration and utilization of these systems in everyday emergency response protocols. This manuscript presents a low-cost system for indoor 3D mapping and target localization designed to be used by first responders. This system competed in the National Institute of Standards and Technology (NIST) 2023 First Responder UAS 3D Mapping Challenge[34] and won third place overall, in addition to awards for Best-in-Class Bill of Materials Total Cost, Best-in-Class Map Data Acquisition Speed and Best-in-Class Blue/Green UAS Capable.

Several commercial UAS are currently used in the public safety and search and rescue industry. This paper compares the specifications of our solution against these commercial options. Many low-cost UAS lack real-time mapping, target detection, and target localization capabilities. While flagship and enterprise-level UAS in the industry meet some first responder requirements, they are generally larger and unsuitable for indoor environments. Additionally, their higher cost makes them less appealing to public safety departments with budget constraints.

In academia, various technical manuscripts describe or implement specific capabilities desirable in UAS for search and rescue applications. Typically, the focus is on individual software frameworks for mapping, object detection, object localization, communication design, or UAS hardware design. However, a systematic approach to integrating all these capabilities into a single UAS is lacking. Karam et al. [35] utilized a microdrone equipped with six laser rangefinders (1D scanners) and an optical sensor for mapping and positioning, employing graph SLAM for loop closure detection. Despite this, the resulting maps are uncolored point clouds and textureless, limiting their usefulness for first responders. Caballero et al. [36] presented a mapping system for first responders that creates 2D maps using UAVs, but their design and implementation are geared towards outdoor use cases. Generally, indoor mapping and SLAM have recently seen numerous implementations. Otero et al. [37] provided an interesting analysis and comparison of the different mobile indoor mapping options available on the market. Their review includes handheld scanners, backpack devices, and trolley configurations, which are mostly suited for ground robots rather than UAS. Placed et al. [38] conducted a comprehensive survey of the state-of-the-art in active SLAM, highlighting how the disparity and lack of unification in the literature have hindered the development of cohesive frameworks, mature algorithms, and their transition to practical applications. Kolhatkar et al. [39] reviewed various techniques used in mapping and localization of mobile robots and the design of low-cost mobile platforms with sensors, focusing on LiDAR and RGB-D Camera technology—the latter utilized in this manuscript. Additionally, LiDAR and IMU-based mapping technologies have seen widespread use, from pipe-inspecting UAVs to self-driving cars. Kumar et al. [40] introduced a method to efficiently determine the planar position of UAVs via a point-to-point scan matching algorithm, leveraging data from a horizontally scanning primary LiDAR. The UAV's altitude relative to the ground was estimated using a vertically scanning secondary LiDAR mounted orthogonally to the primary LiDAR, with a Kalman filter integrating data from both LiDARs to calculate the 3D position. However, the resulting map lacks texture, making it challenging for human interpretation, and finer details necessary for object classification are absent, diminishing its suitability for search and rescue mapping applications. Similarly, Opromolla et al. [41] presented an approach combining point clouds from LiDAR with inertial data. However, their implementation was confined to 2D environments, unsuitable for mapping intricate 3D spaces. Chan et al. [42] employed a LiDAR approach to demonstrate and assess state-of-the-art SLAM algorithms, LeGO-LOAM [43] and LIO-SAM [44], in simulated indoor environments using ground robots. Results revealed that even advanced LiDAR odometry and mapping methods can experience significant drift due to the absence of features. Cvisic et al. [45] introduced a stereo vision SLAM method that employs separate localization and mapping threads. Localization relies on visual feature matching, while mapping utilizes depth information generated using the semi-global matching algorithm [46]. However, while the approach excels in stereo localization, the resulting maps lack texture and fidelity, limiting their utility for first responders. Labbe et al. [47] presented a comprehensive mapping software package supporting both visual and LiDAR SLAM. Their graph-based SLAM approach is adopted here due to its versatility in accommodating various sensors, including unsynchronized RGB-D systems. Recent developments include generic multi-robot and multi-modal mapping frameworks, such as the one by Cramariuc et al. [48], capable of integrating multiple robots, visual landmarks, and LiDAR scans. Additionally, there's ongoing research in utilizing UAVs equipped with neural networks for mapping tasks [49, 50]. However, these advanced frameworks are yet to attain the technical readiness level required for reliable UAS mapping performance.

Recent research has been dedicated to exploring the potential of UAS in target-tracking missions, spanning both indoor and outdoor environments. Alhafnawi et al. [51] conducted a survey investigating UAV-based target tracking and monitoring across diverse settings. Their study delved into the deployment scenarios of these systems, offering detailed characterization and analysis. Cui et al. [52] introduced an end-to-end search and revisit framework designed for small UAVs to detect and localize targets using a single onboard camera module. This framework is well-suited for outdoor applications. In another study, Wang et al. [53] proposed a real-time multi-target localization scheme employing an electro-optical stabilized imaging system. However, the system's output of geodetic coordinates limits its applicability to outdoor settings. On a different note, Unlu et al. [54] introduced a comprehensive end-to-end framework for mapping, detecting, and extinguishing fire targets in indoor environments. However, the framework relied on a fiducial marker to simulate fire and was solely demonstrated in simulated environments.

1.3 Contributions

This dissertation advances the state-of-the-art in aerial robotics by integrating concepts, tools, and techniques from various disciplines, including Aerospace Engineering, Computer Science, Robotics, and Estimation and Control Theory, as illustrated in Figure 1.2. System identification tools were developed by researchers in rotorcraft and aircraft systems, while Kalman Filters and Nonlinear SE(3) Controllers were created by experts in estimation and control. Visual Inertial Odometry techniques emerged from computer scientists specializing in computer vision, and Pose-Graph Optimization was introduced by roboticists to overcome the limitations of using

Kalman Filters for the localization and mapping of mobile robots. This dissertation leverages these interdisciplinary tools to bridge gaps between fields and address significant challenges in aerial robotics. The resulting contributions have been published in peer-reviewed journals or are currently under review [52, 55, 56, 57].

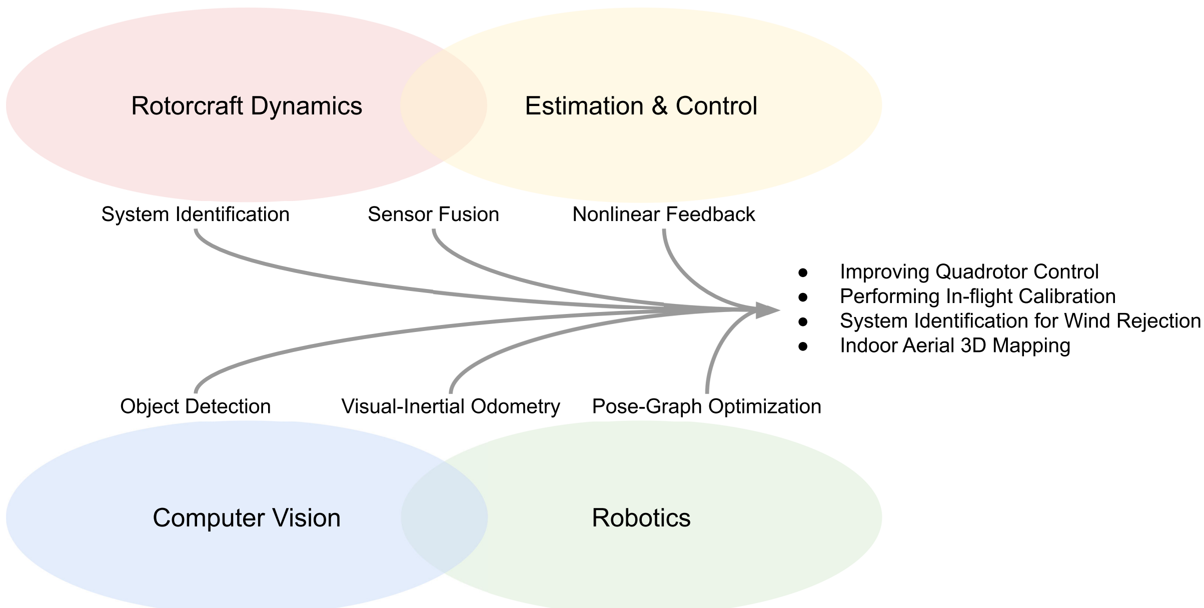


Figure 1.2: This work applies concepts from multiple disciplines to solve Indoor Aerial Autonomy Challenges.

1.3.1 Self-Calibrated Nonlinear Filtering and Control Framework for UAVs

A comprehensive nonlinear filtering framework for UAV parameter estimation is introduced, encompassing factors such as external wind, drag coefficients, IMU bias, and center of pressure. Additionally, optimized flight trajectories for parameter estimation through empirical observability is established. Futhermore, an estimation and control framework that leverages the mean of state and parameter estimates to generate appropriate control inputs for vehicle actuators is implemented. Utilizing a square-root unscented Kalman filter (sq-UKF), the framework can deduce a 23-dimensional state vector from 9-dimensional sensor data and 4-dimensional control inputs. This innovative, self-calibrated approach is the first to effectively integrate estimation and control for UAVs in windy conditions, significantly enhancing the capability of unmanned flight vehicles to operate autonomously without the need for manual intervention or tuning.

1.3.2 Advanced Integrated UAV System Identification for Wind Rejection

Linear and nonlinear estimation methods from robotics, rotorcraft, and flight dynamics is integrated to accurately estimate model parameters such as mass, inertia, drag coefficients, and external wind. The accuracy of inertial parameter estimation in linear systems is enhanced through direct sensing of control inputs. Additionally, a custom, low-cost flight controller capable of managing the high computational demands of nonlinear control and high-dimensional estimation is developed. Using a vision-based localization camera removes the need for external positioning, reducing both time and cost without compromising accuracy. These advancements enable the rapid deployment of nonlinear frameworks for high-performance UAVs, eliminating the need for manual tuning and experienced pilots.

1.3.3 Reliable Real-Time Indoor Aerial 3D Mapping

A comprehensive pipeline for real-time mapping and target detection in indoor environments with limited network infrastructure is introduced. This work is not pursuing just a proof-of-concept but a best-in-class UAS design and hence a detailed description, comparison, and thorough evaluation of the system's hardware and software components is provided. Experimental validation is conducted in various indoor settings which demonstrates the system's effectiveness in producing real-time, high-quality maps and tagging locations of targets. Additionally, it is being planned to make the code repositories publicly available, which is valuable for researchers and first responders. This work enables the development of low-cost, highly capable, and reliable unmanned aerial systems (UAS) using accessible open-source software tools and hardware designs.

1.4 Dissertation Outline

This dissertation is outlined as follows.

Chapter 2 develops a framework for online state, parameter, and wind estimation for a UAV equipped with an IMU and a ground-velocity sensor, such as visual- or lidar-based odometry. Thrust and moment control inputs are used to steer the process model and the ground-velocity and IMU measurements are assimilated by a square-root unscented Kalman filter containing 23 states, 12 of which are constant parameters. Additionally, the system's observability is characterized and optimal calibration trajectories are designed to maximize observability of the model parameters. Simulations show the improvements obtained in tracking performance by coupling the estimation framework with a standard model-based controller. By estimating unsteady winds, the onboard controller's gust rejection improves.

Chapter 3 describes and experimentally evaluates a comprehensive system identification framework for high-performance UAV control in wind. The framework incorporates both linear offline and nonlinear online methods to estimate model parameters in support of a nonlinear model-based control implementation. Inertial parameters of the UAV are estimated using a frequency-domain linear system identification program by incorporating control data obtained from motor-speed sensing along with state estimates from an automated frequency sweep maneuver. The drag-force coefficients and external wind are estimated recursively in flight with a square-root unscented Kalman filter. A custom flight controller is developed to handle the computational demand of the online estimation and control. Flight experiments illustrate the nonlinear controller's tracking performance and enhanced gust rejection capability.

Chapter 4 presents the systematic design of an advanced and feature-rich UAS tailored for first responders in search and rescue operations. It details the creation and implementation of a 3D mapping, target detection, and localization framework alongside the specific hardware and software components essential for enhanced reliability. Utilizing a distributed computing approach, the UAS offloads intensive computations to an offboard computer, resulting in improved real-time mapping performance and system stability. The cost-effective UAS participated in the National Institute of Standards and Technology (NIST) 2023 First Responder UAS 3D Mapping Challenge, securing third place overall, along with awards for Best-in-Class Bill of Materials Total Cost, Best-in-Class Map Data Acquisition Speed, and Best-in-Class Blue/Green UAS Capability. Competition results and independent testing demonstrate the system's reliable performance in diverse scenarios.

Finally, Chapter 5 concludes the dissertation by summarizing the key findings and discussing the implications of the research. It highlights the advancements made in enhancing aerial autonomy and the integration of various interdisciplinary tools and techniques. The chapter also addresses any challenges or limitations encountered during the research and outlines ongoing work aimed at further improving aerial robotics capabilities. Future directions for research are suggested, focusing on overcoming identified issues and exploring new avenues to advance the field.

Chapter 2: Quadrotor Modeling for Improved Control Performance via In-flight Calibration

Unmanned aerial vehicles (UAVs) demand precise estimation and control systems to maintain stability and fulfill user requirements. Traditional linear estimation and control frameworks, while robust against disturbances and uncertainties, struggle with precision due to their inherent linearity assumptions. Conversely, nonlinear model-based frameworks are hindered by the accuracy of the dynamic models. Therefore, accurate system parameter identification, including sensor calibration, is essential. Conventional methods for quadrotor parameter identification rely on frequency inputs in hover and batch post-processing, yet they face observability issues. A nonlinear estimation approach using recursive estimation improves parameter identification but loses observability in hover. Prior studies employing unscented Kalman filters and genetic algorithms have estimated parameters like mass and inertia, but often fail to address low observability in UAVs. This chapter introduces a novel approach that optimizes calibration trajectories based on observability, enhancing parameter estimation accuracy. The presented work offers improvements in UAV control by developing a nonlinear filtering framework, optimizing flight trajectories for parameter estimation, and implementing an estimation and control framework that integrates these estimates to stabilize the vehicle. This self-calibrated system, employing a square-root unscented Kalman filter is tested in simulations and shows improved performance especially under challenging conditions like high winds.

This chapter is structured as follows. Section 2.1 begins with the notation and conventions that will be used throughout this dissertation. Next, it presents the full nonlinear dynamics model of a quadrotor, followed by the control allocation model, the developed estimator model and the modified controller model. Section 2.2 describes the observability analysis and the design of an offline trajectory optimization method for optimizing estimation performance. Lastly, section 2.3 illustrates the performance of the in-flight calibration framework design through numerical simulations. It also presents the post-calibration performance of the estimation and control framework.

2.1 System Modeling

This section describes the mathematical models utilized by the UAV estimation, control, and simulation frameworks. The models depicted here are applicable to quadrotors, tail-sitters in hover, and similar vehicles whose aerodynamic effects are not negligible.

2.1.1 Notation and Conventions

A right-handed orthonormal inertial frame represented by $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ is defined such that $\mathbf{w}_3$ points opposite to the direction of gravity g. A body-fixed frame represented by $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ is attached to the vehicle, such that $\mathbf{b}_1$ points forward, $\mathbf{b}_3$ is aligned with the collective thrust, and $\mathbf{b}_2 = \mathbf{b}_3 \times \mathbf{b}_1$ completes the orthonormal triad. The orientation of the vehicle is represented by a rotation matrix $R \in \mathrm{SO}(3)$, where $[R]_{ij} = \mathbf{w}_i \cdot \mathbf{b}_j$. Additionally, the superscript notation $\mathbf{v}^{\mathcal{F}}$ is used to express a vector $\mathbf{v}$ in frame $\mathcal{F}$. Also, the convention for representing the set of standard basis vectors is $\{\mathbf{e}_i\}$, where $\mathbf{e}_i$ is a three-dimensional unit vector.

2.1.2 Aircraft Dynamics Model

The quadrotor aircraft is modeled as a rigid body with gravity, motor thrust, and aerodynamic forces acting on the aircraft's center of mass. The translational and rotational kinematics and dynamics of the aircraft are

$$\begin{aligned}
\dot{\mathbf{r}}^{\mathcal{W}} &= \mathbf{v}^{\mathcal{W}} \\
\dot{\mathbf{v}}^{\mathcal{B}} &= \frac{1}{m}\left(T\mathbf{e}_3 + \mathbf{F}_a^{\mathcal{B}}\right) - gR^{\mathsf{T}}\mathbf{e}_3 - \boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{v}^{\mathcal{B}} \\
\dot{R} &= R\left[\omega^{\mathcal{B}}\right]_{\times} \\
\dot{\boldsymbol{\omega}}^{\mathcal{B}} &= J^{-1}\left(\boldsymbol{M} - \boldsymbol{\omega}^{\mathcal{B}} \times J\boldsymbol{\omega}^{\mathcal{B}} + \boldsymbol{\tau}_a^{\mathcal{B}}\right)
\end{aligned} \tag{2.1}$$

where $\mathbf{r}^{\mathcal{W}}$ is the position of the vehicle in $\mathcal{W}$, $\mathbf{v}^{\mathcal{B}} := [v_x\ v_y\ v_z]^{\mathsf{T}} \in \mathbb{R}^3$ is the inertial velocity of the vehicle in $\mathcal{B}$, $\boldsymbol{\omega}^{\mathcal{B}} := [\omega_x\ \omega_y\ \omega_z]^{\mathsf{T}} \in \mathbb{R}^3$ is the inertial angular velocity of the vehicle in $\mathcal{B}$, m is the vehicle's mass, $J \in \mathbb{R}^{3\times3}$ is the vehicle's inertia tensor, $T \in \mathbb{R}^1$ is the thrust generated by the motors, $\boldsymbol{M} \in \mathbb{R}^3$ is the moment generated by the motors, $\mathbf{F}_a^{\mathcal{B}}$ is the aerodynamic force experienced by the vehicle in $\mathcal{B}$, and $\boldsymbol{\tau}_a^{\mathcal{B}}$ is the aerodynamic torque experienced by the vehicle in $\mathcal{B}$.

The aerodynamic force components $\mathbf{F}_{a,i}^{\mathcal{B}}\ \forall\, i \in \{1,2,3\}$ are modeled as

$$\mathbf{F}_{a,i}^{\mathcal{B}} = -C_{Di}\left|\left(\mathbf{v}^{\mathcal{B}} - \mathbf{v}_a^{\mathcal{B}}\right)\cdot\mathbf{e}_i\right|\left(\mathbf{v}^{\mathcal{B}} - \mathbf{v}_a^{\mathcal{B}}\right)\cdot\mathbf{e}_i, \tag{2.2}$$

where $\mathbf{C}_D^{\mathcal{B}} := [C_{D1}\ C_{D2}\ C_{D3}]^{\mathsf{T}} \in \mathbb{R}^3$ is the vector of body drag force coefficients, and $\mathbf{v}_a^{\mathcal{B}}$ is the wind velocity in $\mathcal{B}$. We are focused on the parasitic drag component only and hence, the drag expression is quadratic with relative wind. The wind is assumed to be static in $\mathcal{W}$ and, hence, its evolution in $\mathcal{B}$ is described by

$$\dot{\mathbf{v}}_a^{\mathcal{B}} = -\boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{v}_a^{\mathcal{B}} \tag{2.3}$$

The aerodynamic torque components $\boldsymbol{\tau}_{a,i}^{\mathcal{B}}\ \forall\, i \in \{1, 2, 3\}$ are modeled as

$$\boldsymbol{\tau}_{a,i}^{\mathcal{B}} = -C_{Mi}\left(\mathbf{b}_3 \times \left(\mathbf{v}^{\mathcal{B}} - \mathbf{v}_a^{\mathcal{B}}\right)\right) \cdot \mathbf{e}_i, \tag{2.4}$$

where $\mathbf{C}_M^{\mathcal{B}} := [C_{M1}\ C_{M2}\ 0]^{\mathsf{T}} \in \mathbb{R}^3$ is the vector of body drag moment coefficients.

2.1.3 Control Allocation Model

The thrust and moment forces generated by the motors for a quadrotor in squashed-X configuration are modeled as [58]

$$\begin{bmatrix} T \\ \boldsymbol{M} \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ k_f L_y & k_f L_y & -k_f L_y & -k_f L_y \\ -k_f L_x & k_f L_x & k_f L_x & -k_f L_x \\ -k_m & k_m & -k_m & k_m \end{bmatrix} \begin{bmatrix} u_1^2 \\ u_2^2 \\ u_3^2 \\ u_4^2 \end{bmatrix}, \tag{2.5}$$

where k_f is the thrust coefficient for the propellers, k_m is the moment coefficient for the propellers, L_x and L_y are the moment arm lengths, and u_i is the speed of the ith motor. Assume that both the forces and moments generated by the motors are proportional to the square of the speed of the motors.

2.1.4 Estimator Model

The state vector used in the estimator's process model is

$$\mathbf{x} = [\mathbf{s}^\mathsf{T}\ \mathbf{v}^{\mathcal{B}\mathsf{T}}\ \boldsymbol{\omega}^{\mathcal{B}\mathsf{T}}\ \mathbf{v}_a^{\mathcal{B}\mathsf{T}}\ \mathbf{C}_D^{\mathcal{B}\mathsf{T}}\ \mathbf{r}_{CP}^{\mathcal{B}\mathsf{T}}\ \mathbf{b}^{\mathcal{B}\mathsf{T}}]^\mathsf{T} \tag{2.6}$$

where $\mathbf{s} := [s_1\ s_2]^\mathsf{T} \in \mathbf{S}^2$ is the tilt of the quadrotor expressed in stereographic coordinates, $\mathbf{v}^{\mathcal{B}} \in \mathbb{R}^3$ is the inertial velocity of the vehicle in the body frame, $\boldsymbol{\omega}^{\mathcal{B}} := [\omega_1\ \omega_2\ \omega_3]^\mathsf{T} \in \mathbb{R}^3$ is the angular velocity of the vehicle in $\mathcal{B}$, $\mathbf{v}_a^{\mathcal{B}} \in \mathbb{R}^3$ is the velocity of the external wind acting on the vehicle in $\mathcal{B}$, $\mathbf{C}_D^{\mathcal{B}} := [C_{D1}\ C_{D2}\ C_{D3}]^\mathsf{T} \in \mathbb{R}^3$ is the vector of square-roots of the body drag coefficients, $\mathbf{r}_{CP}^{\mathcal{B}} := [x_{CP}\ y_{CP}\ z_{CP}]^\mathsf{T} \in \mathbb{R}^3$ is the position of the center of pressure relative to the center of mass in the body-fixed frame, and $\mathbf{b}^{\mathcal{B}} = [\mathbf{b}_a^{\mathcal{B}\mathsf{T}}\ \mathbf{b}_g^{\mathcal{B}\mathsf{T}}]^\mathsf{T} \in \mathbb{R}^6$ is the IMU bias vector containing the accelerometer bias $\mathbf{b}_a^{\mathcal{B}} \in \mathbb{R}^3$ and the gyroscope bias $\mathbf{b}_g^{\mathcal{B}} \in \mathbb{R}^3$.

The evolution of the state vector is [2, 58]

$$\begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{\mathbf{v}}^{\mathcal{B}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{B}} \\ \dot{\mathbf{v}}_a^{\mathcal{B}} \\ \dot{\mathbf{C}}_D^{\mathcal{B}} \\ \dot{\mathbf{r}}_{CP}^{\mathcal{B}} \\ \dot{\mathbf{b}}^{\mathcal{B}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\left(\omega_2(1+s_1^2-s_2^2)+2\omega_3 s_2-2\omega_1 s_1 s_2\right) \\ \frac{1}{2}\left(\omega_1(s_1^2-s_2^2-1)-2\omega_3 s_1+2\omega_2 s_1 s_2\right) \\ \frac{1}{m}\left(T_d\mathbf{e}_3+\mathbf{F}_a^{\mathcal{B}}\right)-gR^\mathsf{T}\mathbf{e}_3-\boldsymbol{\omega}^{\mathcal{B}}\times\mathbf{v}^{\mathcal{B}} \\ J^{-1}\left(\boldsymbol{M}_d-\boldsymbol{\omega}^{\mathcal{B}}\times J\boldsymbol{\omega}^{\mathcal{B}}+\mathbf{r}_{CP}^{\mathcal{B}}\times\mathbf{F}_a^{\mathcal{B}}\right) \\ -\boldsymbol{\omega}^{\mathcal{B}}\times\mathbf{v}_a^{\mathcal{B}} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{6\times1} \end{bmatrix} \tag{2.7}$$

where m is the vehicle's mass, $J \in \mathbb{R}^{3\times3}$ is the vehicle's inertia tensor, T_d is the thrust control input, $\boldsymbol{M}_d \in \mathbb{R}^3$ is the moment control input, and $\mathbf{F}_a^{\mathcal{B}}$ is the aerodynamic force experienced by the vehicle in the body frame, whose components $\mathbf{F}_{a,i}^{\mathcal{B}}\ \forall\, i \in \{1,2,3\}$ are modeled as [58, 59]

$$\mathbf{F}_{a,i}^{\mathcal{B}} = -{C_D}_i^2 \left|\left(\mathbf{v}^{\mathcal{B}} - \mathbf{v}_a^{\mathcal{B}}\right) \cdot \mathbf{e}_i\right| \left(\mathbf{v}^{\mathcal{B}} - \mathbf{v}_a^{\mathcal{B}}\right) \cdot \mathbf{e}_i \tag{2.8}$$

Using Equation (2.7), the process model for the kth step with time interval dt is

$$F(\mathbf{x}_k, T_{d,k}, \boldsymbol{M}_{d,k}) = \mathbf{x}_k + \dot{\mathbf{x}}_k dt \tag{2.9}$$

The measurement model for the square-root unscented Kalman filter represents the biased measurements obtained from the IMU and ground-velocity sensing as a function of the state and control:

$$\mathbf{y}_k = H(\mathbf{x}_k, T_{d,k}, \boldsymbol{M}_{d,k}) \tag{2.10}$$

Dropping the subscript k for simplicity, the sensor outputs are modeled as

$$\mathbf{y} = \begin{bmatrix} \frac{1}{m}\left(T_d \mathbf{e}_3 + \mathbf{F}_a^{\mathcal{B}}\right) - \boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{v}^{\mathcal{B}} + \mathbf{a}_{IMU}^{\mathcal{B}} + \mathbf{b}_a^{\mathcal{B}} \\ \boldsymbol{\omega}^{\mathcal{B}} + \mathbf{b}_g^{\mathcal{B}} \\ \mathbf{v}^{\mathcal{B}} \end{bmatrix} \tag{2.11}$$

where

$$\mathbf{a}_{IMU}^{\mathcal{B}} = \boldsymbol{\omega}^{\mathcal{B}} \times \left(\boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{r}_{IMU}^{\mathcal{B}}\right) + \dot{\boldsymbol{\omega}}^{\mathcal{B}} \times \mathbf{r}_{IMU}^{\mathcal{B}} \tag{2.12}$$

and $\mathbf{r}_{IMU}^{\mathcal{B}}$ is the IMU's position relative to center of mass. Popular frameworks to sense ground-velocity $\mathbf{v}^{\mathcal{B}}$ are visual-inertial odometry [60] and lidar-based odometry [61].

2.1.5 Controller Model

A standard model-based controller operating on the Special Euclidean Group SE(3) [20] modified to include aerodynamic interactions is used for evaluating the effect of the estimator framework's parameter update on the quadrotor's tracking performance. Aerodynamic forces and moments are treated as external disturbances so that the controller's exponential stability is preserved. The thrust control T_d and the moment control $\boldsymbol{M}_d$ expressions are [20]

$$\begin{aligned} T_d &= m(-k_x \boldsymbol{e}_x^{\mathcal{W}} - k_v \boldsymbol{e}_v^{\mathcal{W}} - g\mathbf{e}_3 + \ddot{\boldsymbol{x}}_d^{\mathcal{W}} - R\mathbf{F}_a^{\mathcal{B}}/m) \cdot R\mathbf{e}_3 \\ \boldsymbol{M}_d &= J\left(-k_R \boldsymbol{e}_R^{\mathcal{B}} - k_\omega \boldsymbol{e}_\omega^{\mathcal{B}}\right) + \boldsymbol{\omega}^{\mathcal{B}} \times J\boldsymbol{\omega}^{\mathcal{B}} \\ &\quad - J\left(\boldsymbol{\omega}^{\mathcal{B}} \times R^\top R_d \boldsymbol{\omega}_d^{\mathcal{B}} - R^\top R_d \dot{\boldsymbol{\omega}}_d^{\mathcal{B}}\right) - \mathbf{r}_{CP}^{\mathcal{B}} \times \mathbf{F}_a^{\mathcal{B}} \end{aligned} \tag{2.13}$$

where k_x, k_v, k_R, k_ω are positive constants and the expression for desired orientation R_d is [20]

$$\begin{aligned} R_d &= \begin{bmatrix} \mathbf{y}_B \times \mathbf{z}_B, & \mathbf{y}_B, & \mathbf{z}_B \end{bmatrix} \\ \mathbf{z}_B &= \frac{-k_x \boldsymbol{e}_x^{\mathcal{W}} - k_v \boldsymbol{e}_v^{\mathcal{W}} - g\mathbf{e}_3 + \ddot{\boldsymbol{x}}_d^{\mathcal{W}} - R\mathbf{F}_a^{\mathcal{B}}/m}{\|-k_x \boldsymbol{e}_x^{\mathcal{W}} - k_v \boldsymbol{e}_v^{\mathcal{W}} - g\mathbf{e}_3 + \ddot{\boldsymbol{x}}_d^{\mathcal{W}} - R\mathbf{F}_a^{\mathcal{B}}/m\|} \\ \mathbf{y}_B &= \frac{\mathbf{z}_B \times \mathbf{x}_C}{\|\mathbf{z}_B \times \mathbf{x}_C\|}, \quad \mathbf{x}_C = [\cos\psi_d^{\mathcal{W}},\ \sin\psi_d^{\mathcal{W}},\ 0]^\top \end{aligned} \tag{2.14}$$

The tracking errors in position $\boldsymbol{e}_x^{\mathcal{W}}$, velocity $\boldsymbol{e}_v^{\mathcal{W}}$, orientation $\boldsymbol{e}_R^{\mathcal{B}}$, and angular velocity $\boldsymbol{e}_\omega^{\mathcal{B}}$ are [20]

$$\begin{aligned} \boldsymbol{e}_x^{\mathcal{W}} &= \boldsymbol{x}^{\mathcal{W}} - \boldsymbol{x}_d^{\mathcal{W}}, & \boldsymbol{e}_v^{\mathcal{W}} &= R\boldsymbol{v}^{\mathcal{B}} - \dot{\boldsymbol{x}}_d^{\mathcal{W}} \\ \boldsymbol{e}_R^{\mathcal{B}} &= \frac{1}{2}(R_d^\top R - R^\top R_d)^\vee, & \boldsymbol{e}_\omega^{\mathcal{B}} &= \boldsymbol{\omega}^{\mathcal{B}} - R^\top R_d \boldsymbol{\omega}_d^{\mathcal{B}} \end{aligned} \tag{2.15}$$

where the orientation R is obtained from heading ψ and the estimation of stereographic tilt (s_1, s_2). The expression for the orientation R is

$$\begin{aligned} R =& R_\psi R_{tilt} \\ =& \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_{3z} + \frac{b_{3y}^2}{1+b3z} & \frac{-b_{3x}b_{3y}}{1+b3z} & b_{3x} \\ \frac{-b_{3x}b_{3y}}{1+b3z} & 1 - \frac{b_{3y}^2}{1+b3z} & b_{3y} \\ -b_{3x} & -b_{3y} & b_{3z} \end{bmatrix} \end{aligned} \tag{2.16}$$

where

$$b_{3x} = \frac{2s_1}{1+s_1^2+s_2^2},\; b_{3y} = \frac{2s_2}{1+s_1^2+s_2^2},\; b_{3z} = \frac{1-s_1^2-s_2^2}{1+s_1^2+s_2^2} \tag{2.17}$$

Note that the position $\boldsymbol{x}^{\mathcal{W}}$ and heading ψ are not estimated by the framework and their ground truth values are assumed to be known.

2.2 Observability Analysis and Optimization

This section analyzes the observability of the UAV estimator framework and describes the trajectory optimization process for self-calibration. The optimization process consists of establishing the trajectory parameters, formulating the observability gramian, designing the cost function, defining the problem, and finally choosing the appropriate optimization technique. The goal of optimizing the calibration trajectory is to ensure that the vehicle operates in the best observability conditions with feasible control efforts.

2.2.1 Observability Analysis

The observer dynamics (2.7) are expressed in control-affine form as

$$\dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) + \begin{bmatrix} \mathbf{f}_1(\mathbf{x}) & \mathbf{f}_2(\mathbf{x}) & \mathbf{f}_3(\mathbf{x}) & \mathbf{f}_4(\mathbf{x}) \end{bmatrix} \boldsymbol{\eta} \tag{2.18}$$

where $\boldsymbol{\eta} = [T_d,\ \boldsymbol{M}_d]^\mathsf{T}$ and the vector fields $\mathbf{f}_i(\mathbf{x})$ are

$$\mathbf{f}_0(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}\left(\omega_2(1+s_1^2-s_2^2)+2\omega_3 s_2 - 2\omega_1 s_1 s_2\right) \\ \frac{1}{2}\left(\omega_1(s_1^2-s_2^2-1)-2\omega_3 s_1 + 2\omega_2 s_1 s_2\right) \\ m^{-1}\mathbf{F}_a^{\mathcal{B}} - gR^\mathsf{T}\mathbf{e}_3 - \boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{v}^{\mathcal{B}} \\ J^{-1}\left\{\mathbf{r}_{CP}^{\mathcal{B}} \times \mathbf{F}_a^{\mathcal{B}} - \boldsymbol{\omega}^{\mathcal{B}} \times J\boldsymbol{\omega}^{\mathcal{B}}\right\} \\ -\boldsymbol{\omega}^{\mathcal{B}} \times \mathbf{v}_a^{\mathcal{B}} \\ \mathbf{0}_{12\times 1} \end{bmatrix} \tag{2.19}$$

$$\mathbf{f}_1(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{2\times 1} \\ m^{-1}\mathbf{e}_3 \\ \mathbf{0}_{18\times 1} \end{bmatrix},\ \mathbf{f}_i(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{5\times 1} \\ J^{-1}\mathbf{e}_{i-1} \\ \mathbf{0}_{15\times 1} \end{bmatrix} \forall\, i \in \{2,3,4\}$$

The output vector of the system is given in (2.11). Using Lie derivatives of the output vector, a 39-dimensional observable vector space $\mathcal{O}$ is constructed as [62]

$$\mathcal{O} = \begin{bmatrix} \mathbf{y}^\mathsf{T} & \mathbf{L}(\mathbf{y})^\mathsf{T} & \mathbf{L}\left(\mathbf{L}(\mathbf{y})\right)^\mathsf{T} \end{bmatrix}^\mathsf{T} \tag{2.20}$$

where

$$\mathbf{L}(\mathbf{y}) = \begin{bmatrix} \mathcal{L}_{\mathbf{f}_0}\mathbf{y}^\top & \mathcal{L}_{\mathbf{f}_1}\mathbf{y}^\top & \mathcal{L}_{\mathbf{f}_2}\mathbf{y}^\top & \mathcal{L}_{\mathbf{f}_3}\mathbf{y}^\top & \mathcal{L}_{\mathbf{f}_4}\mathbf{y}^\top \end{bmatrix}^\top \tag{2.21}$$

Using MATLAB's symbolic toolbox, the column rank of the 39×23-dimensional observability co-distribution matrix $\nabla_x\mathcal{O}$ is 23, except at certain points in the state space that causes degeneracy, such as hover. Hence, the system has the property of weak local observability almost globally [62].

2.2.2 Calibration Trajectory Optimization Process

Periodicity is a desirable property of any self-calibration trajectory. Hence the class of Lissajous trajectories is selected, following [2]. To increase the observability by exciting the states at different frequencies while keeping the control efforts low, the superposition of two Lissajous trajectories is chosen. Thus, the trajectory parameter vector $\boldsymbol{P}$ along with the trajectory's position $\boldsymbol{x}_d$ and heading ψ_d component is represented as

$$\begin{aligned}
&\boldsymbol{P} := \begin{bmatrix} \begin{bmatrix} A_{1x} & A_{1y} & A_{1z} & A_{1\psi} & n_{1x} & n_{1y} & n_{1z} & n_{1\psi} \end{bmatrix}^\top \\ \begin{bmatrix} A_{2x} & A_{2y} & A_{2z} & A_{2\psi} & n_{2x} & n_{2y} & n_{2z} & n_{2\psi} \end{bmatrix}^\top \end{bmatrix} \\
&\boldsymbol{x}_d^{\mathcal{W}}(\boldsymbol{P}, t) = \\
&\begin{bmatrix} A_{1x}\left(1-\cos\left(2\pi n_{1x}t\right)\right) + A_{2x}\left(1-\cos\left(2\pi n_{2x}t\right)\right) \\ A_{1y}\sin\left(2\pi n_{1y}t\right) + A_{2y}\sin\left(2\pi n_{2y}t\right) \\ A_{1z}\sin\left(2\pi n_{1z}t\right) + A_{2z}\sin\left(2\pi n_{2z}t\right) \end{bmatrix} \\
&\psi_d^{\mathcal{W}}(\boldsymbol{P}, t) = A_{1\psi}\sin\left(2\pi n_{1\psi}t\right) + A_{2\psi}\sin\left(2\pi n_{2\psi}t\right)
\end{aligned} \tag{2.22}$$

where t is time in seconds.

Computation of the empirical observability gramian involves simulating the dynamics of the state-space model represented by $\dot{x} = f(x, u)$ [63], where the signal u is an open-loop control to make the observability measure independent of the choice of the control law. However, it is challenging to determine the open-loop control signal from the reference trajectory. Additionally, numerical-integration techniques can also introduce large errors between the simulated and the reference trajectory. Hence, the state vector is divided into two components: reference states extracted from the trajectory and parameter states that do not evolve or are independent of the trajectory. Performing simulation and numerical integration is not required with this method, assuming that the control law is ideal with zero tracking error. Hence, the state vector composition for the observability gramian computation is

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_r^{\mathsf{T}} & \mathbf{x}_p^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \tag{2.23}$$

where $\mathbf{x}_r$ is the part of the state vector that is directly computed from the trajectory parameters and $\mathbf{x}_p$ contains the parameter states with zero dynamics. The expressions for $\mathbf{x}_r$ and $\mathbf{x}_p$ are

$$\begin{aligned} \mathbf{x}_r &= \begin{bmatrix} \mathbf{s}^{\mathsf{T}} & \mathbf{v}^{\mathcal{B}\mathsf{T}} & \boldsymbol{\omega}^{\mathcal{B}\mathsf{T}} & \mathbf{v}_a^{\mathcal{B}\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \\ \mathbf{x}_p &= \begin{bmatrix} \mathbf{C}_D^{\mathcal{B}\mathsf{T}} & \mathbf{r}_{CP}^{\mathcal{B}\mathsf{T}} & \mathbf{b}^{\mathcal{B}\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \end{aligned} \tag{2.24}$$

where $\mathbf{v}_a^{\mathcal{B}} = \mathbf{0}_{3\times1}$, because no wind is the nominal condition.

Let $\mathbf{S}_\mathbf{x}$ be a scaling vector with dimension equal to that of state $\mathbf{x}_p$. Given small, scaled displacements $\epsilon_i = \epsilon\mathbf{S}_\mathbf{x} \cdot \mathbf{e}_i$ of state $\mathbf{x}_p$, let $\mathbf{x}_p^{\pm i} = \mathbf{x}_{p0} \pm \epsilon_i\mathbf{e}_i$ and $\mathbf{y}^{\pm i}$ be the corresponding output, with $\mathbf{e}_i$ as the i^{th} unit vector in $\mathbb{R}^{12}$. The empirical local observability gramian at $\mathbf{x}_{p0}$ is the 12×12 matrix $\mathcal{W}_{\mathcal{O}}(\mathbf{x}_{p0})$ whose ijth entry is [64]

$$[\mathcal{W}_{\mathcal{O}}]_{ij} := \frac{1}{4\epsilon^2}\int_0^{\mathbb{T}} \big(\mathbf{y}^{+i}(t) - \mathbf{y}^{-i}(t)\big)^{\mathsf{T}} \big(\mathbf{y}^{+j}(t) - \mathbf{y}^{-j}(t)\big)\, dt \tag{2.25}$$

where the duration $\mathbb{T}$ must be chosen sufficiently large to encompass the longest period of the trajectory. Let $\mathbb{T} = 60$ seconds and $\epsilon = 0.1$. Additionally, the numerical value for the scaling vector is

$$\mathbf{S}_{\mathbf{x}} = [0.01,\ 0.01,\ 0.02,\ (\text{diag}\{0.1\}\mathbf{1}_{3\times 1})^{\mathsf{T}},\ \mathbf{1}_{6\times 1}^{\mathsf{T}}]^{\mathsf{T}}$$

The optimization cost function J favors trajectories with higher observability while penalizing control efforts. A valid and stable unobservability measure is the reciprocal of the smallest singular value λ_{min} of the observability gramian [64]. To ensure that the trajectory is feasible, a quadratic control and control rate cost is also added, resulting in the following cost expression:

$$J(\boldsymbol{P}) := \frac{1}{\lambda_{\text{min}}(\mathcal{W}_{\mathcal{O}})} + \int_0^T \left(Q_u \left\| \frac{\boldsymbol{u}}{u_{\text{max}}} \right\|^2 + Q_{\dot{u}} \left\| \frac{\dot{\boldsymbol{u}}}{u_{\text{max}}} \right\|^2 \right) dt \tag{2.26}$$

where $\boldsymbol{u}$ is the control input vector, u_{max} defines the control saturation limit, and Q_u and $Q_{\dot{u}}$ are the weights on the normalized control and control-rate actions, respectively. For a quadrotor $\boldsymbol{u} \in \mathbb{R}^4$ contains the motor speeds, and the numerical values of the weights are chosen as $Q_u = 50$ and $Q_{\dot{u}} = 10$.

Given an initial state $\mathbf{x}_0 \in \mathbb{R}^{32}$ and perturbation ϵ, find a trajectory parametrization $\boldsymbol{P} \in$

$\mathbb{R}^{16}$ such that

$$\begin{aligned} &\min_{\boldsymbol{P}} \; J(\boldsymbol{P}) \\ &s.t. \quad \boldsymbol{P}_{lb} \leq \boldsymbol{P} \leq \boldsymbol{P}_{ub} \end{aligned} \tag{2.27}$$

where $\boldsymbol{P}_{lb}$ and $\boldsymbol{P}_{ub}$ define the lower and upper bounds of the parameter search space, and their numerical values are chosen as

$$\begin{aligned} \boldsymbol{P}_{lb} &= \begin{bmatrix} [1,\ 1,\ 1,\ \pi/2,\ 0,\ 0,\ 0,\ 0]^{\mathsf{T}} \\ [0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0.1]^{\mathsf{T}} \end{bmatrix} \\ \boldsymbol{P}_{ub} &= \begin{bmatrix} [2,\ 2,\ 2,\ \pi,\ 0.5,\ 0.5,\ 0.5,\ 0.05]^{\mathsf{T}} \\ [0.1,\ 0.1,\ 0.1,\ \pi/10,\ 2,\ 2,\ 2,\ 0.2]^{\mathsf{T}} \end{bmatrix} \end{aligned} \tag{2.28}$$

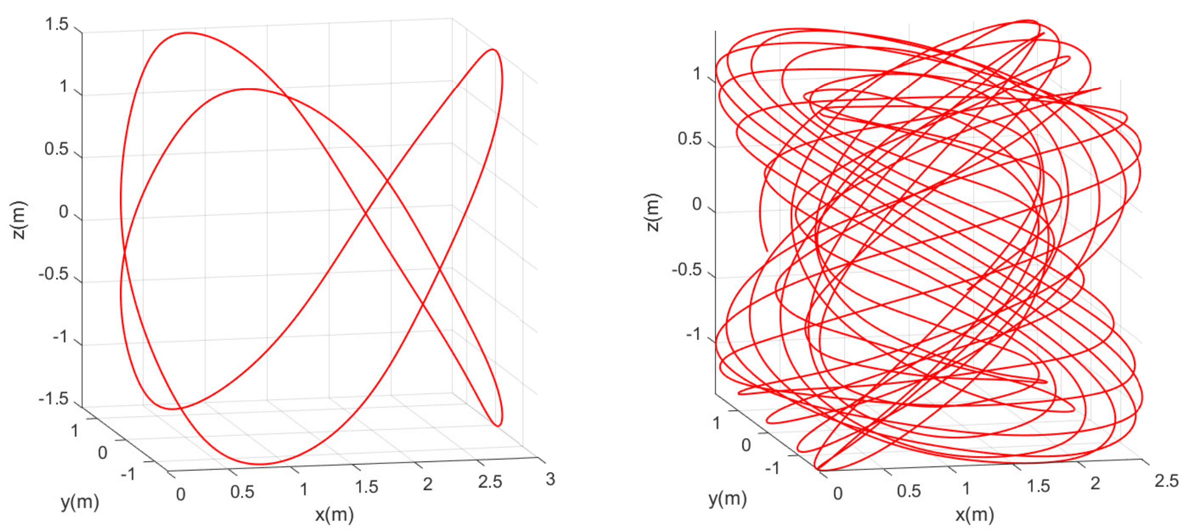
The optimization problem has multiple local minima and, hence, a group of global optimization solvers was selected. Additionally, the cost function evaluation is computationally expensive; it requires the calculation of the observability gramian, which involves simulating the dynamics repeatedly. Therefore, MATLAB's Surrogate Optimization tool was chosen for solving the optimization problem.

2.3 Numerical Results

The sq-UKF estimation framework, coupled with the modified SE(3) controller (2.13) was simulated in MATLAB, with the controller using the mean values of the state feedback, as well as the mean external wind, drag coefficients, and center of pressure estimates. The control updates, along with the sensor feedback and sq-UKF estimates, were simulated at 100Hz for a duration of 2 minutes. The true values for the system parameters utilized for simulating the vehicle with onboard sensors are shown in Table 2.1. For the observability calculation, the nominal values for the external wind, drag coefficients, center of pressure, and IMU biases are zero, whereas the true values of the remaining parameters are known. The self-calibration trajectory obtained via the optimization method of Section 2.2 is given in Equation (2.29). The manually selected trajectory [2] given in Equation (2.30) is used as a baseline to compare the optimized trajectory's performance. Figure 2.1 shows a side by side 3D view comparison of the manual and optimized calibration trajectories.

$$\boldsymbol{P}^{*}=\begin{bmatrix}[1.25,\ 1.75,\ 1.38,\ 2.49,\ .23,\ .22,\ .12,\ .04]^{\mathsf{T}}\\ [0,\ .0034,\ .006,\ .043,\ 1.16,\ 1.17,\ 1.2,\ .2]^{\mathsf{T}}\end{bmatrix} \tag{2.29}$$

$$\boldsymbol{P}^{b}=\begin{bmatrix}[1.5,\ 1.5,\ 1.5,\ \pi,\ 0.2,\ 0.2,\ 0.3,\ 0.0133]^{\mathsf{T}}\\ [0.02,\ 0.02,\ 0,\ 0.5,\ 1.2,\ 1.2,\ 1.8,\ 0.0796]^{\mathsf{T}}\end{bmatrix} \tag{2.30}$$



(a) Manual Trajectory.

(b) Optimized Trajectory.

Figure 2.1: Manual vs Optimized Calibration Trajectory.

Table 2.1: System parameters

Name	Symbol	Value	Units
mass	m	227	g
x moment of inertia	J_{11}	0.644	g·m^2
y moment of inertia	J_{22}	0.739	g·m^2
z moment of inertia	J_{33}	0.903	g·m^2
x position of IMU	x_{IMU}	50	mm
y position of IMU	y_{IMU}	50	mm
z position of IMU	z_{IMU}	50	mm
x position of CP	x_{CP}	0	mm
y position of CP	y_{CP}	0	mm
z position of CP	z_{CP}	-50	mm
x body drag coeff	C_{D1}^2	0.0123	N·s^2/m^2
y body drag coeff	C_{D2}^2	0.0123	N·s^2/m^2
z body drag coeff	C_{D3}^2	0.0243	N·s^2/m^2
motor torque coeff	k_τ	2.47×10^{-5}	N·m·s
rotor inertia	j_b	6.96×10^{-4}	g·m^2
rotor thrust coeff	k_f	1.63×10^{-7}	N/(rad^2·s^4)
rotor moment coeff	k_m	8.74×10^{-10}	N·m/(rad^2·s^4)
motor speed sat	u_{max}	3200	rad/s
arm length	L	125	mm
accel bias	$\mathbf{b}_a^{\mathcal{B}}$	diag$\{0.5\}\mathbf{1}_{3\times1}$	m/s^2
gyro bias	$\mathbf{b}_a^{\mathcal{B}}$	diag$\{0.5\}\mathbf{1}_{3\times1}$	rad/s
accel noise SD	$\sigma(\mathcal{N}_a)$	0.2	m/s^2
gyro noise SD	$\sigma(\mathcal{N}_g)$	0.2	rad/s
velocity noise SD	$\sigma(\mathcal{N}_v)$	0.2	m/s

2.3.1 Self-calibration

For self-calibration, simulated unsteady wind is generated using the following expression:

$$\mathbf{v}_a^{\mathcal{W}} = \sqrt{3}\begin{bmatrix}\cos\left(\frac{2\pi t}{10}\right) & \sin\left(\frac{2\pi t}{10}\right) & \sin\left(\frac{2\pi t}{10}\right)\end{bmatrix}^{\mathsf{T}} \tag{2.31}$$

To measure the estimation and control tracking performance three scaled error functions are introduced given by the following expression:

$$
\begin{aligned}
E_{\text{state}} &= \left\| \left(\mathbf{x}_r^{\text{est}} - \mathbf{x}_r^{\text{true}}\right)^{\mathsf{T}} \operatorname{diag}\{\mathbf{S}_{\text{state}}\} \right\|_2 \\
E_{\text{param}} &= \left\| \left(\mathbf{x}_p^{\text{est}} - \mathbf{x}_p^{\text{true}}\right)^{\mathsf{T}} \operatorname{diag}\{\mathbf{S}_{\text{param}}\} \right\|_2 \\
E_{\text{track}} &= \left\| \left[\boldsymbol{e}_x^{\mathsf{T}}\ \boldsymbol{e}_v^{\mathsf{T}}\ \boldsymbol{e}_R^{\mathsf{T}}\right]^{\mathsf{T}} \operatorname{diag}\{\mathbf{S}_{\text{track}}\} \right\|_2
\end{aligned}
\tag{2.32}
$$

where $\mathbf{x}_r^{\text{est}}$ is the vector of estimated state values, $\mathbf{x}_p^{\text{est}}$ is the vector of estimated parameter values, $\mathbf{x}_r^{\text{true}}$ is the vector of ground truth state values, and $\mathbf{x}_p^{\text{true}}$ is the vector of ground truth parameter values. Also, $\mathbf{S}_{\text{state}}$, $\mathbf{S}_{\text{param}}$, and $\mathbf{S}_{\text{track}}$ are scaling vectors whose numerical values are

$$
\begin{aligned}
\mathbf{S}_{\text{state}} &= \left[0.1,\ 0.1,\ \mathbf{1}_{3\times1}^{\mathsf{T}},\ \left(\operatorname{diag}\{0.1\}\mathbf{1}_{3\times1}\right)^{\mathsf{T}},\ \mathbf{1}_{3\times1}^{\mathsf{T}}\right]^{\mathsf{T}} \\
\mathbf{S}_{\text{param}} &= \left[C_{D1}^{2},\ C_{D2}^{2},\ C_{D3}^{2},\ \mathbf{1}_{6\times1}^{\mathsf{T}},\ \left(\operatorname{diag}\{0.1\}\mathbf{1}_{3\times1}\right)^{\mathsf{T}}\right]^{\mathsf{T}} \\
\mathbf{S}_{\text{track}} &= \mathbf{1}_{9\times1}
\end{aligned}
\tag{2.33}
$$

The scaled $\mathcal{L}_2$-norm of the estimation and tracking errors computed by running 100 Monte Carlo simulations using the optimized calibration trajectory are shown and compared with the manually selected trajectory in Figure 2.2. The optimized trajectory has better averaged estimation and tracking RMSE. The framework's chances of failure due to estimator divergence or losing positive-definiteness of the covariance matrix is approximately 92% with the manual trajectory and 4% with the optimized trajectory. Figure 2.3 shows the optimized trajectory's position and the framework's tracking improvement with time for a single realization. The estimation of sensor biases, external wind, and drag parameters, as shown in Figures 2.4, 2.5, and 2.6, respectively, provides evidence for the framework's improvements in tracking and estimation performance. Note that the z-components of accelerometer bias, wind, and drag coefficients have lower estimation performance than other states. A possible cause is that the rotor dynamics of (2.5) are not present in the estimator's process model because the desired control efforts generated by the controller is used as input. Also, this model mismatch is more pronounced in the z-direction as the thrust term directly affects the z-component of the accelerometer measurements.

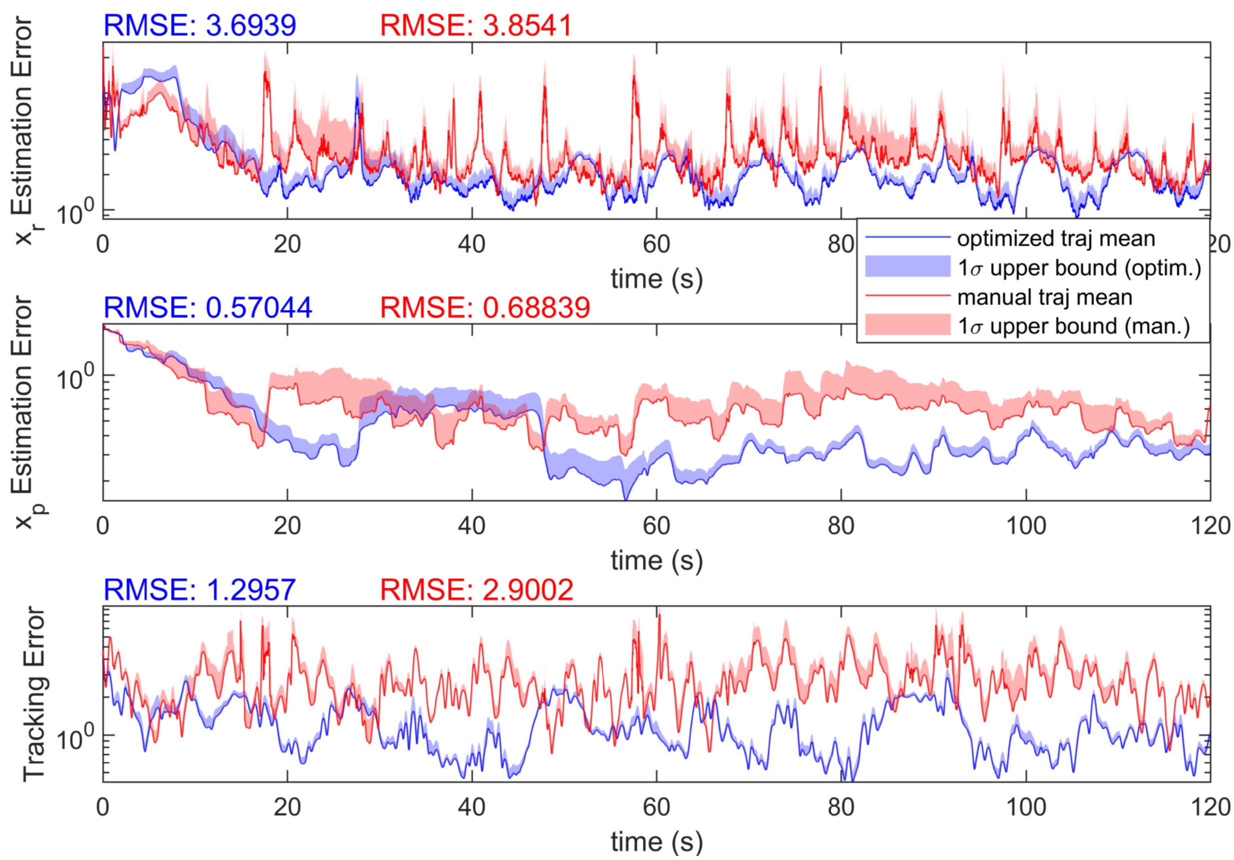


Figure 2.2: Estimation and tracking performance comparison of optimized and manually selected calibration trajectories computed on a set of 100 Monte Carlo simulations. Performance is quantified by scaled $\mathcal{L}_2$-norm of the error vectors and averaged RMSE scores are shown at the top of each subplot.

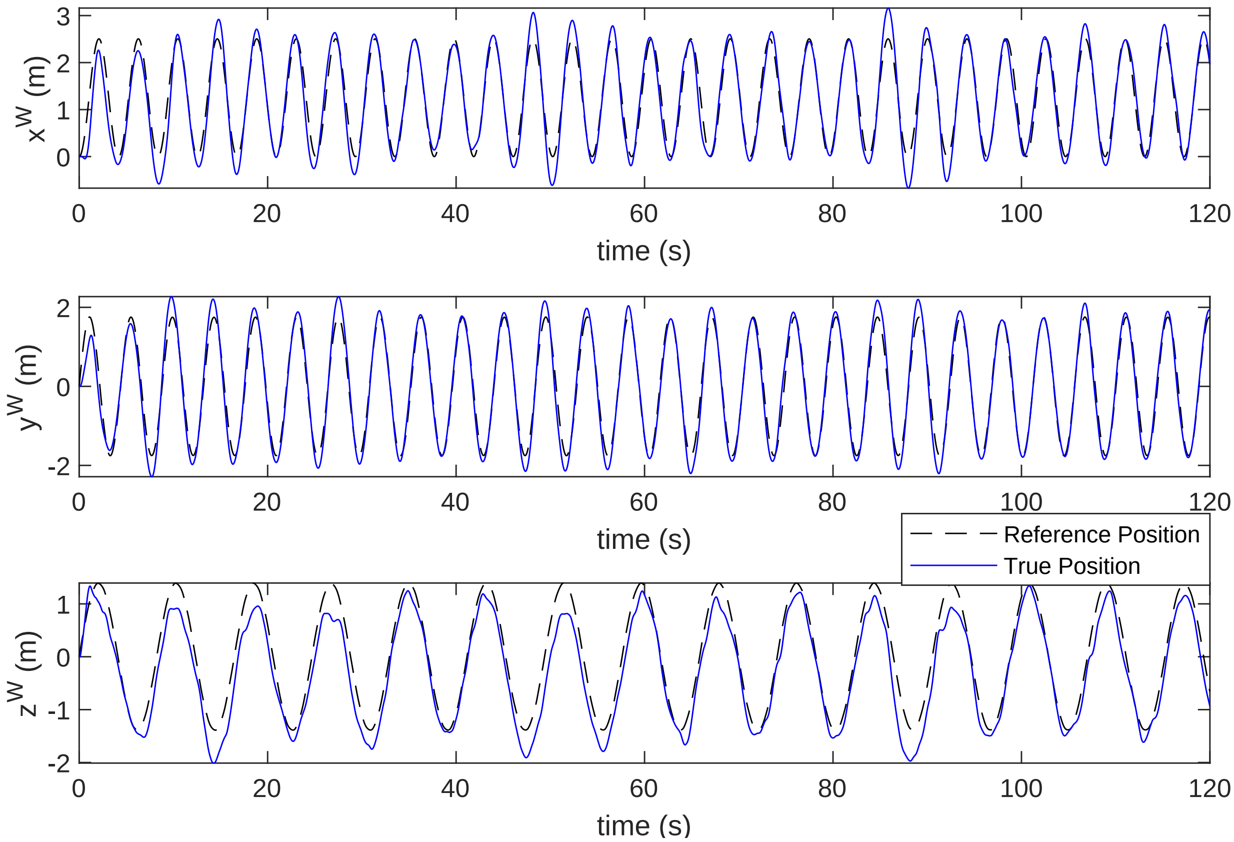


Figure 2.3: Position tracking of the optimized calibration trajectory.

2.3.2 Post-calibration Estimation and Control Performance

For gust rejection simulation, unsteady wind is generated using the following expression;

$$\mathbf{v}_a^{\mathcal{W}}=\frac{6}{\sqrt{3}}\begin{bmatrix}\cos\left(\frac{2\pi t}{10}\right)\\ \sin\left(\frac{2\pi t}{10}\right)\\ \sin\left(\frac{2\pi t}{10}\right)\end{bmatrix}+\frac{3}{\sqrt{3}}\begin{bmatrix}\cos\left(\frac{2\pi t}{5}\right)\\ \sin\left(\frac{2\pi t}{5}\right)\\ \sin\left(\frac{2\pi t}{5}\right)\end{bmatrix}+\frac{1}{\sqrt{3}}\begin{bmatrix}\cos\left(2\pi t\right)\\ \sin\left(2\pi t\right)\\ \sin\left(2\pi t\right)\end{bmatrix} \tag{2.34}$$

In this simulation, the final estimated values of the vehicle's parameters $\mathbf{x}_p$ are taken from one of the calibration flights and are kept fixed by removing it from the sq-UKF's state. The remnant of the state $\mathbf{x}_r$ is estimated by the sq-UKF. The post-calibration framework is able to estimate and

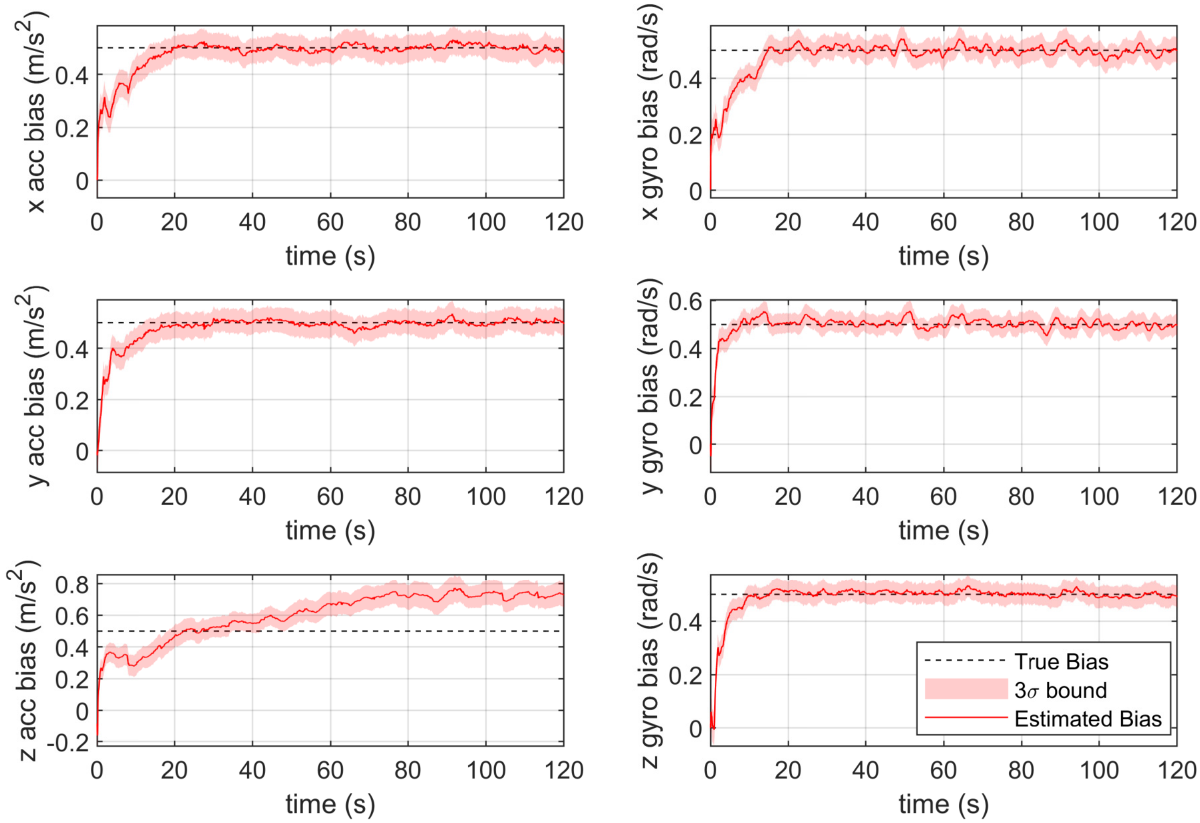


Figure 2.4: IMU bias estimation of sq-UKF.

reject the unsteady wind (2.34). A Monte Carlo simulation over 100 trials resulted in averaged estimation and tracking RMSE scores of 3.5136 and 0.75906, respectively. The simulated step-response with position-hold and wind-estimation performance for a single realization are shown in Figure 2.7 and Figure 2.8, respectively. The results show the framework estimating and rejecting unsteady winds; the performance in the z-direction is again worse than x, y, possibly due to the same modeling mismatch reason as described above.

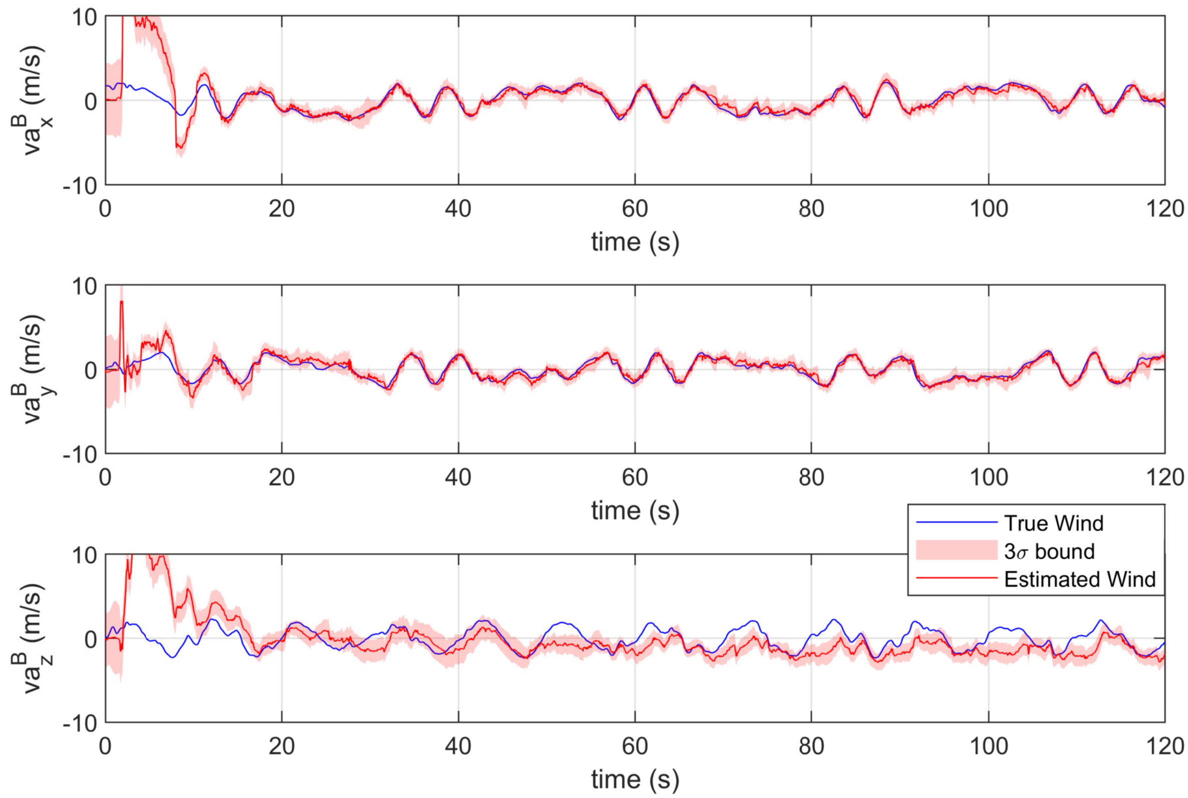


Figure 2.5: Unsteady wind estimation of sq-UKF.

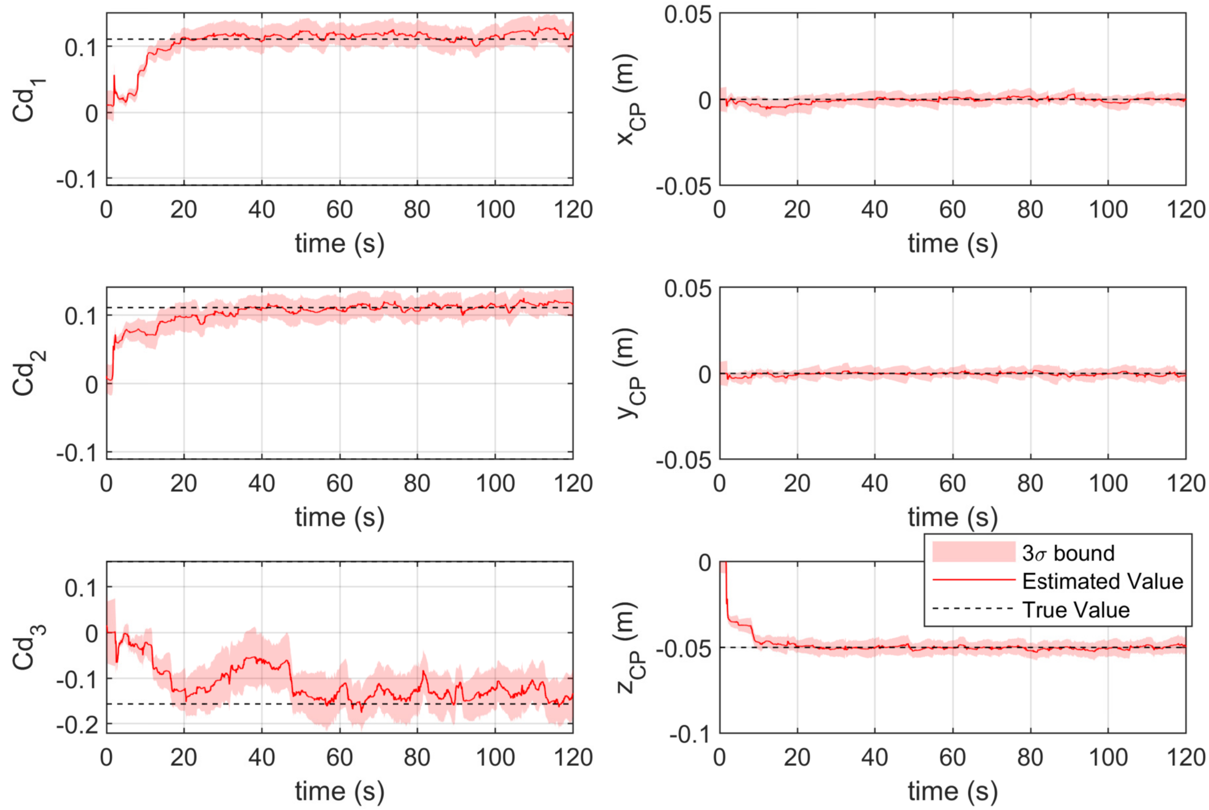


Figure 2.6: Drag coefficients and center of pressure estimation of sq-UKF.

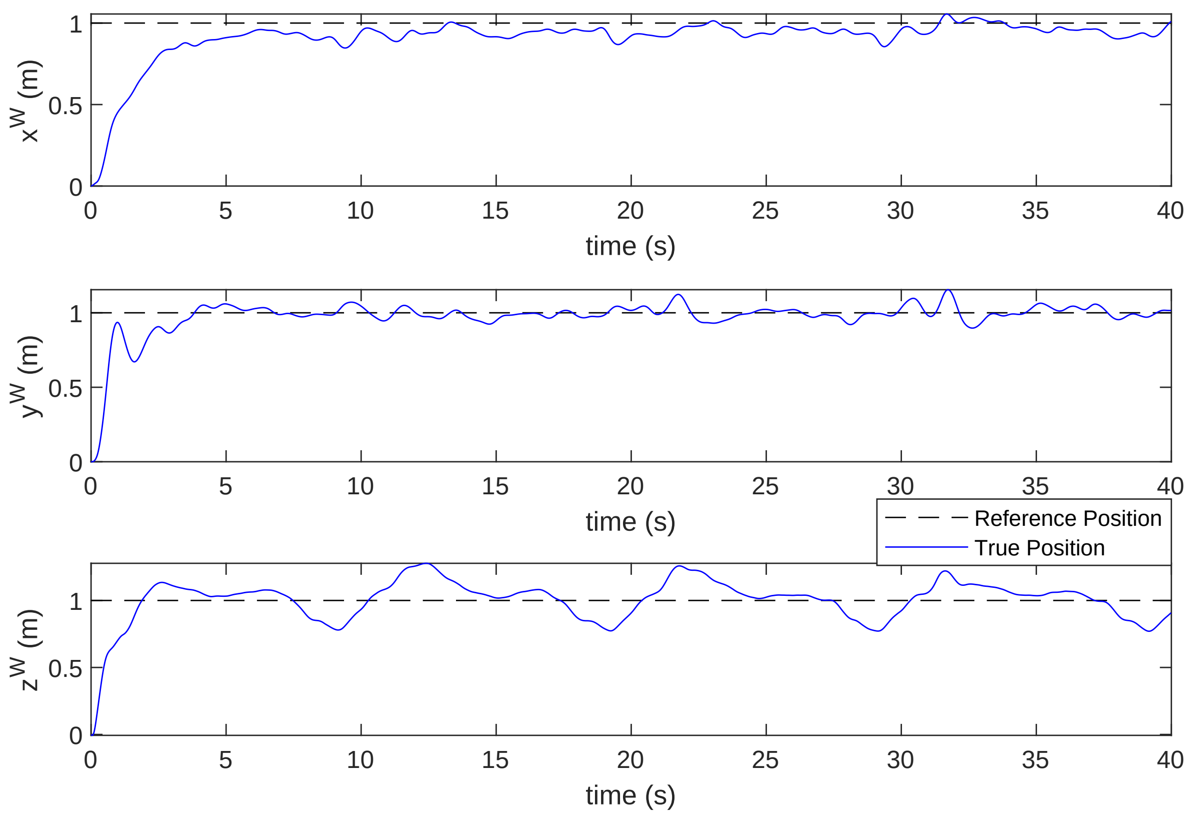


Figure 2.7: Post-calibration position-hold performance in unsteady wind.

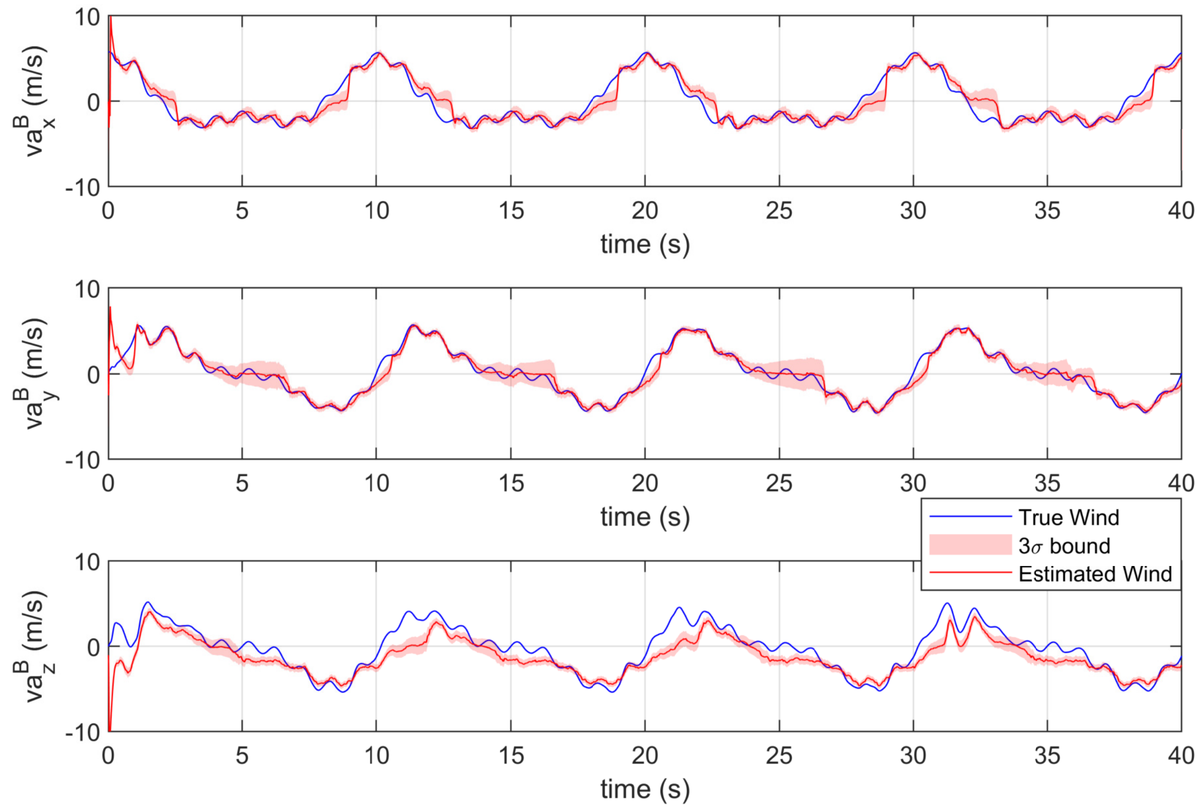


Figure 2.8: Post-calibration unsteady wind estimation performance.

Chapter 3: Experimental Implementation of System Identification for Gust Rejection

This chapter builds upon the work performed in Chapter 2. It utilizes the state and nonlinear parameter estimation algorithm that was introduced in the previous chapter. The work done in Chapter 2 was primarily theoretical and included simulation results only, whereas the work described in this chapter builds upon it and demonstrates its feasibility through real-world experiments. Control inputs steered the process model and the ground-velocity and IMU measurements updated the state and parameter estimates. This chapter demonstrates experimentally the estimation performance of a subset of the estimator's state vector. Additionally, the aerodynamic drag torque expression is simplified, assuming zero sensor bias drift. A model-based feedback linearization controller operates on the Special Euclidean Group $\mathrm{SE}(3)$ [20] with necessary modifications to include aerodynamic interactions [55]. Nominal values for mass and inertia are obtained from a weight scale and a CAD model, respectively. Instead of manually tuning the position and orientation control gains, they were obtained by setting the damping ratio to 0.75 and natural frequency to 2 for translation control and 13 for orientation control. The gain values in simulations of prior work [55] also follow the same method, reinforcing the idea that manual tuning is not required if inertial parameters are known. Inertial parameters are estimated offline with a linear system identification process, whereas the aerodynamic parameters including drag force coefficients and wind are estimated online in a recursive manner with sq-UKF. The resulting parameter estimates are utilized by the nonlinear controller to achieve high control performance. The controller assumes zero wind and drag force coefficients to generate flight data for the linear system identification process. For nonlinear estimation and control experiments, the controller uses the drag force coefficients and wind estimates generated recursively by the sq-UKF estimator in flight.

This chapter is structured as follows. Section 3.1 describes the aircraft and corresponding hardware and software components that enable real-time implementation of nonlinear estimation and control. Section 3.2 presents the lower level propulsion dynamics model along with a summary of parameters and their estimation methods. Section 3.3 describes the linearization of the aircraft dynamics and presents the estimates of the model parameters extracted from frequency-domain linear system identification. Finally, section 3.4 illustrates the recursive, in-flight estimation of drag force coefficients and wind gusts through the use of a nonlinear state and parameter estimation and control framework.

3.1 Experimental Testbed Development

A quadrotor with customized hardware and software was used in this research. This section presents an overview of the aircraft's hardware components, avionics, and software development.

3.1.1 Hardware Overview

The aircraft shown in Figure 3.1 is a squashed-X quadrotor with a 230 mm wheelbase. A dummy payload in the shape of a cuboid is attached to the bottom to introduce unmodeled

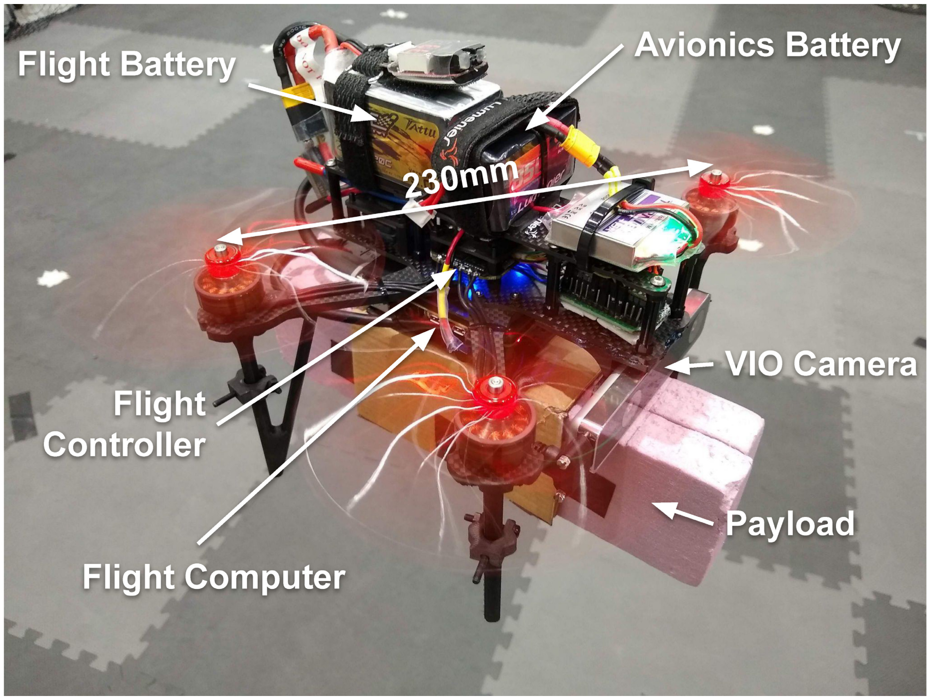


Figure 3.1: The quadrotor platform.

dynamic effects and additional disturbance so that the proposed framework's performance can be evaluated in a worst-case scenario. Propulsion and avionics components are assembled onto a freestyle carbon fiber airframe. A pair of clockwise and counter-clockwise rotating Gemfan 5043 propellers are driven by Lumenier ZIP 2207 2450kV brushless direct-drive electric motors, each with 12 electromagnets on the stator and 14 N52SH curved neodymium permanent magnets on the rotor. The motors are controlled by electronic speed controllers (ESC) running KISS firmware, and communication with the ESC is established using DShot600 digital signals generated using Direct Memory Access (DMA). The ESC is powered by a 4 cell (14.8V) 2000mAh lithium polymer battery. Avionics include a custom-built flight controller board, Intel Realsense T265 Tracking Camera, and a Raspberry Pi 4B flight computer. The T265 Tracking Camera enables vision-based localization for the quadrotor; eliminates the requirement for a motion-capture system or a Global Positioning System (GPS) for localization, thus reducing the system's cost, complexity, and setup time.

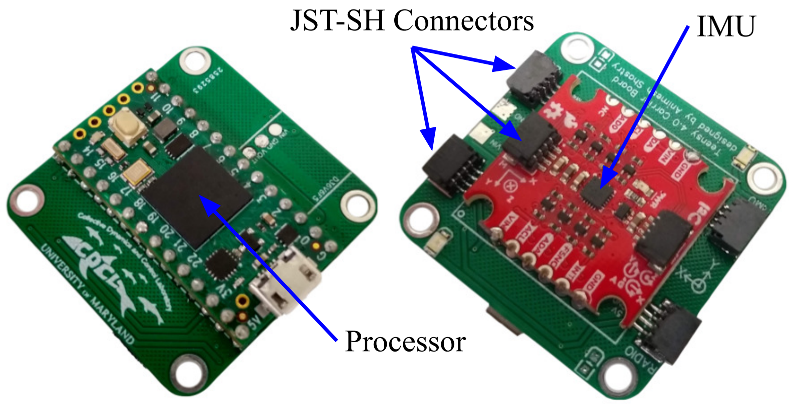


Figure 3.2: The custom-made flight controller module. (Left) the top side; (right) the bottom. The module measures 37 $\times$ 36 $\times$ 6 mm and has the standardized 30.5 $\times$ 30.5 mm mounting with JST-SH connectors.

Commercially available, off-the-shelf, open-source flight controllers [16] such as Pixhawk, PixRacer, Sparky2, Chimera, Atom, APM 2.8, FlyMaple, PXFMini, etc., have insufficient computing power for implementation of modern nonlinear control and estimation algorithms that are increasingly using sampling and optimization-based techniques. The recursive online and onboard estimation scheme described in [55] is high-dimensional and takes more computational load as compared to the controller. A backup estimator also needs to run in parallel so that in case the framework diverges, the software will choose the estimates provided by the backup estimator. The framework also needs to support a high receiving rate of visual-inertial odometry data; hence, the hardware will need higher cache memory. Additionally, the framework developed in this article required testing of many different motor speed measurement techniques and utilizing off-the-shelf flight controllers will restrict the freedom of choice as they haven't been designed for research or experimentation. Moreover, modifying the estimator framework of open-source firmware such as PX4 and Ardupilot comes with a very steep learning curve. Hence, seeking a custom approach, we designed a flight controller board with a focus on cost-effectiveness, high performance, crash resistance, and modularity, making it suitable for a wide class of UAVs.

Figure 3.2 shows the custom flight controller module, which carries the Teensy 4.0 Micro-Controller Unit (MCU) on the top and the ICM-20948 Inertial Measurement Unit (IMU) on the bottom. The Teensy 4.0 is presently the fastest available MCU; it contains an NXP IMXRT1062 processor with a 32-bit ARM Cortex-M7 CPU core, making it capable of executing instructions at 600 MHz. The high computing power of the MCU allows for the implementation of computationally demanding nonlinear estimation and control laws such as the high-dimensional sq-UKF and SE(3) control. The flight control software, when configured with the estimation and control laws mentioned in subsections 2.1.4 and 2.1.5 respectively, runs seamlessly at 120 Hz on the MCU. Higher computational speeds are also possible by using faster communication protocols and by overclocking the CPU to a maximum speed of 1 GHz. The design flies of the flight controller [65] have been made open-source by the authors.

3.1.2 Software Overview

The Teensy 4.0 MCU is integrated with the Arduino Integrated Development Environment (IDE), which facilitated the development of a user-friendly flight control software geared towards students and researchers. Through the use of C++ templates and matrix software utilities, standard operators are overridden to be used naturally in algebraic expressions, which results in simplistic Matlab-type syntaxes. Figure 3.3 shows a simplified block diagram of the overall software architecture. The system is designed to work fully onboard with negligible intervention from the operator. The flight control software [66] has been made open-source by the authors.

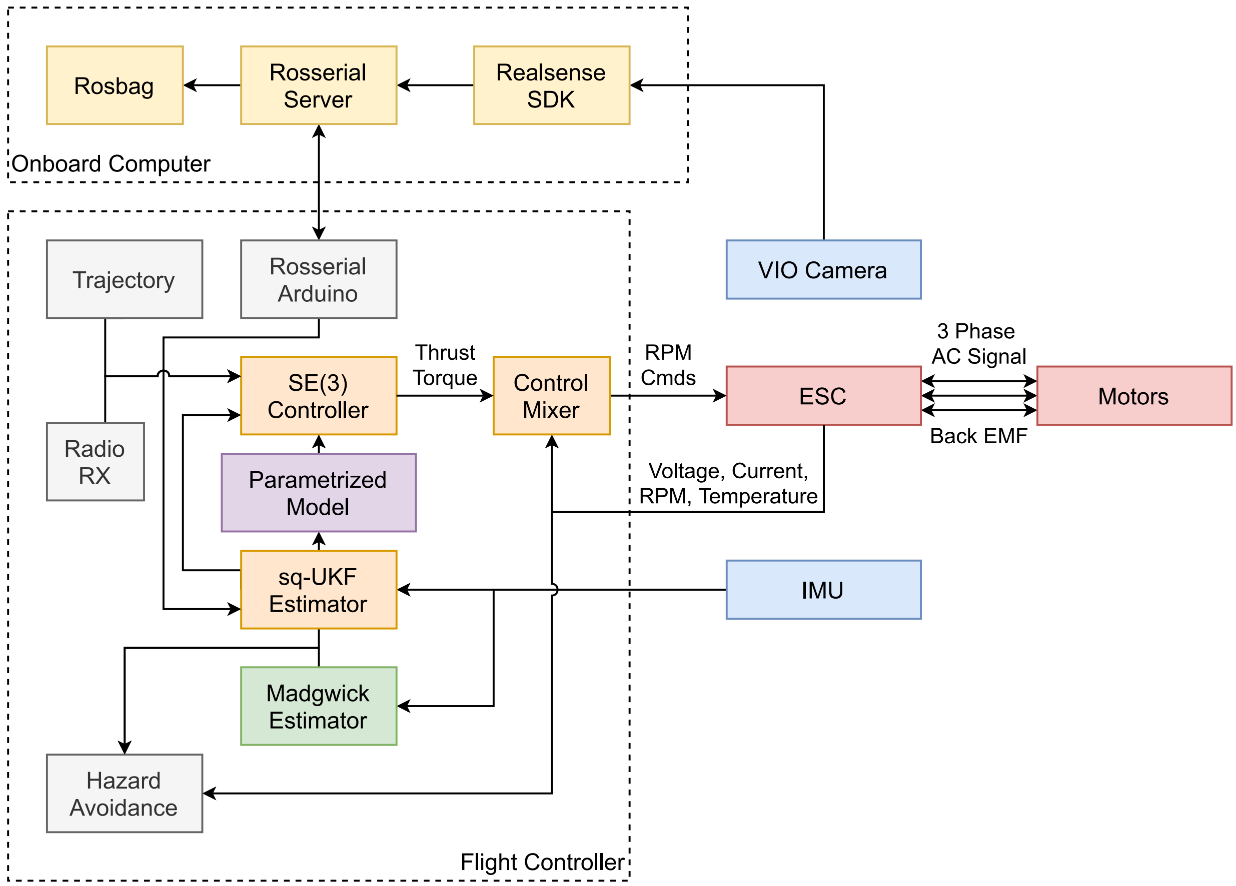


Figure 3.3: Software architecture overview

3.2 Rotor Dynamics Modeling and Calibration

3.2.1 Propulsion Model

The quadrotor's electric brushless motors are controlled by varying the voltage amplitude of the active phases of the individual coils. The rotor torque τ_{z_i} and the applied voltage V_i of the ith motor is modeled by [67]

$$\begin{aligned} V_i &= K_e u_i + R_a I_i \\ \tau_{z_i} &= K_q I_i, \end{aligned} \tag{3.1}$$

where I_i is the current through the ith motor, R_a is its electrical resistance, and K_e and K_q are motor parameters. Since the torque is assumed to be proportional to the square of the rotor speed, the expression for the applied voltage as a function of motor speed is

$$V_i = \frac{R_a k_m}{K_q} u_i^2 + K_e u_i \tag{3.2}$$

To compensate for the motor friction, a constant voltage V_0 is added and the final expression for the applied voltage is

$$V_i = V_{calib} \left(K_2 u_i^2 + K_1 u_i \right) + V_0, \tag{3.3}$$

where V_{calib} is the calibration voltage at which the constants $K_2 = \frac{R_a k_m}{K_q V_{calib}}$ and $K_1 = \frac{K_e}{V_{calib}}$ are calculated. The parameters V_0, K_1, and K_2 can be obtained either from manufacturer data for common motor-propeller setups or from calibration experiments such as a motor bench test for custom setups. Finally, as the electric motors are controlled by Pulse-Width Modulation (PWM) signals, their values can be obtained by

$$PWM_i = \frac{V_i}{V_{battery}} \tag{3.4}$$

where PWM_i is the PWM setting for the ith motor and $V_{battery}$ is the total voltage of the battery pack. Typically, voltages of lithium-based batteries and other battery types decrease as the battery drains and (3.4) will maintain the motor speed at desired levels by increasing the PWM setting, thus ensuring consistent flight performance. Additionally, to minimize the influence of sharp voltage changes and sensing noise on the vehicle's control performance, a digital low-pass filter is added to the battery voltage sensor readings.

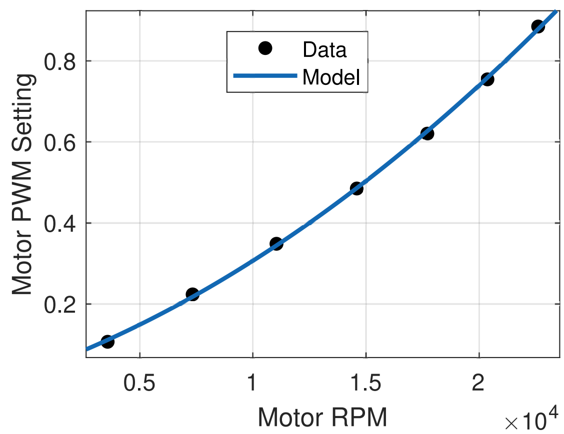
3.2.2 Model Parameter Estimation

The parameters of the aircraft dynamic model (2.1) are estimated collectively from linear system identification and nonlinear parameter estimation techniques. The wind is treated as a time-varying parameter. For verification purposes, beliefs of true values for the parameters were obtained from independent experiments, CAD modeling, and specialized tools. Table 3.1 lists the parameters used in the aircraft dynamics model and their respective estimation method(s). The parameters used in the control allocation and propulsion model were obtained by mounting the motor-propeller setup on a thrust stand and measuring the thrust, torque, and RPM values at various PWM settings. Figure 3.4 shows the experimental data from the thrust stand and fitted propulsion and control allocation models. The calibration voltage is 15 V, and each data point is recorded by setting the PWM at a fixed value and averaging 500 measurements after the motor RPM reaches a steady state. Table 3.2 lists the parameters used in the control allocation and propulsion model along with their measured values.

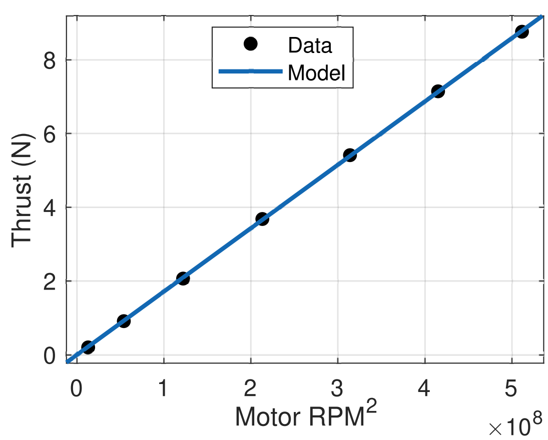
Table 3.1: Estimation techniques for the parameters of the aircraft dynamics model

Parameter Name	Symbol	Units	Linear System ID	Nonlinear Estimation	True Value Source
Mass	m	kg	✓		Weight Scale, CAD
Inertia	J	kg·m^2	✓		CAD
Drag Moment Coefficient	$\mathbf{C}_M^{\mathcal{B}}$	N·s	✓		—
Drag Force Coefficient	$\mathbf{C}_D^{\mathcal{B}}$	N·s^2/m^2	✓	✓	Gust Generation System
Wind	$\mathbf{v}_a^{\mathcal{B}}$	m/s		✓	Anemometer

$$PWM_i = K_2 u_i^2 + K_1 u_i + \frac{V_0}{V_{calib}}.$$



$$T_i = k_f u_i^2.$$



$$\tau_{z_i} = k_m u_i^2$$

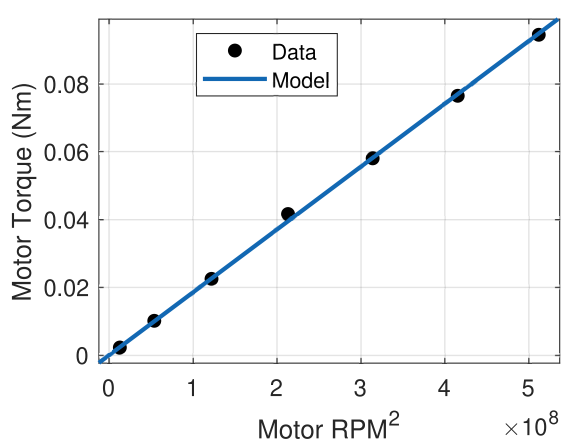


Figure 3.4: Propulsion and control allocation models extracted from thrust-stand data.

Table 3.2: Control allocation and propulsion model parameters

Parameter Name	Symbol	Value	Units
Propeller Thrust Coefficient	k_f	1.717×10^{-8}	N/RPM^2
Propeller Moment Coefficient	k_m	1.845×10^{-10}	N·m/RPM^2
Motor Constant Voltage	V_0	0.4575	V
Motor Model Linear Coefficient	K_1	1.975×10^{-5}	RPM^{-1}
Motor Model Quadratic Coefficient	K_2	7.835×10^{-10}	RPM^{-2}

3.3 Linear System Identification

This section linearizes the nonlinear dynamic model about hover and performs frequency-domain system identification. Two sets of control data using different methods are collected for each flight, and the resulting model-parameter accuracies are compared.

3.3.1 Model Description

Before deriving the linearized model of the aircraft, certain assumptions are made so that the final state-space model resembles a standard rotorcraft model. Assumptions include negligible cross diagonal inertia terms, no wind, i.e., $\mathbf{v}_a^{\mathcal{W}} = 0$, and a simplified linear drag force model. As a consequence, the models presented in (2.2) and (2.4) become

$$\begin{aligned}\mathbf{F}_{a,i}^{\mathcal{B}} &= -\bar{C}_{Di}\mathbf{v}^{\mathcal{B}} \cdot \mathbf{e}_i \\ \boldsymbol{\tau}_{a,i}^{\mathcal{B}} &= -C_{Mi}\left(\mathbf{b}_3 \times \mathbf{v}^{\mathcal{B}}\right) \cdot \mathbf{e}_i\end{aligned} \tag{3.5}$$

where $\bar{C}_{Di}$ are the body-drag force coefficients for the simplified linear aerodynamic force model.

The nonlinear aircraft model presented in Section 2.1.2 is linearized about hover and the resulting quadrotor dynamics decomposed into longitudinal, lateral, heave, and yaw degrees of freedom. Additionally, the orientation representation of the vehicle is changed to a Z-Y-X Euler angles set. The resulting linearized dynamics model is

$$
\begin{aligned}
\dot{v}_x &= -\frac{\bar{C}_{D1}}{m}v_x + g\theta & \dot{\phi} &= \omega_x & \dot{\omega}_x &= -\frac{C_{M1}}{J_{xx}}v_y + \frac{1}{J_{xx}}\tau_x \\
\dot{v}_y &= -\frac{\bar{C}_{D2}}{m}v_y - g\phi \quad , & \dot{\theta} &= \omega_y \quad , & \dot{\omega}_y &= -\frac{C_{M2}}{J_{yy}}v_x + \frac{1}{J_{yy}}\tau_y \\
\dot{v}_z &= -\frac{\bar{C}_{D3}}{m}v_z + \frac{1}{m}T & \dot{\psi} &= \omega_z & \dot{\omega}_z &= \frac{1}{J_{zz}}\tau_z
\end{aligned}
\tag{3.6}
$$

Converting (3.6) to state-space representation and adding time-delay parameters to the control inputs, the resulting state-space gray-box models are

$$
\begin{bmatrix} \dot{v}_x \\ \dot{\omega}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_{v_x} & 0 & g \\ M_{v_x} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ \omega_y \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ M_{\tau_y} \\ 0 \end{bmatrix} \tau_y(t - \Delta t_1) \tag{3.7}
$$

$$
\begin{bmatrix} \dot{v}_y \\ \dot{\omega}_x \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} X_{v_y} & 0 & -g \\ M_{v_y} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_y \\ \omega_x \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ M_{\tau_x} \\ 0 \end{bmatrix} \tau_x(t - \Delta t_2) \tag{3.8}
$$

$$
\begin{bmatrix} \dot{\omega}_z \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_z \\ \psi \end{bmatrix} + \begin{bmatrix} M_{\tau_z} \\ 0 \end{bmatrix} \tau_z(t - \Delta t_3) \tag{3.9}
$$

$$
\dot{v}_z = X_{v_z}v_z + X_T T(t - \Delta t_4), \tag{3.10}
$$

where $X_{v_x} = -m^{-1}\bar{C}_{D1}$, $X_{v_y} = -m^{-1}\bar{C}_{D2}$, $X_{v_z} = -m^{-1}\bar{C}_{D3}$, $M_{v_x} = -J_{yy}^{-1}C_{M2}$, $M_{v_y} =$

$-J_{xx}^{-1}C_{M1}$, $M_{\tau_x} = J_{xx}^{-1}$, $M_{\tau_y} = J_{yy}^{-1}$, $M_{\tau_z} = J_{zz}^{-1}$, $X_T = m^{-1}$, and $\Delta t_i \forall\ i \in \{1, 2, 3, 4\}$ are the unknown stability derivatives and model parameters.

3.3.2 Model Parameter Identification

The Comprehensive Identification from FrEquency Response (CIFER®) [25] program was utilized to identify the parameters of the state-space model representing the quadrotor dynamics linearized about hover. The frequency-domain system identification procedure is more suitable than time-domain techniques for an unstable vehicle like the quadrotor. It avoids divergence due to time-domain integration of the equations of motion while simultaneously minimizing errors associated with bias effects and processing noise.

To perform frequency-domain system identification, individual degrees of freedom of the quadrotor are excited using an automated frequency sweep maneuver. The frequency sweep involves a sinusoidal trajectory with frequency increasing exponentially in time from 0.1 rad/s to 50 rad/s. The amplitude is automatically adjusted within manually specified bounds to ensure the control commands do not saturate and the current consumption is below a set threshold. In addition to the controller-generated commands, individual motor RPMs were also estimated using each motor's measured back Electromotive force (EMF) and supplied to the control allocation model to obtain more accurate albeit noisy control input data. Traditionally, system identification is performed with controller-generated commands, and this procedure is labeled as Method 1. An alternate procedure involving control input measurements obtained from motor RPM data is also investigated and labeled Method 2.

Figure 3.5 shows the state estimates and the lateral control inputs of the automated frequency sweep maneuver of the lateral degree of freedom. The difference between the commanded torque and the torque obtained from sensing the motor RPM increases with frequency due to the influence of the ESCs' response and structural characteristics of the rotor. Figure 3.6 shows the resulting model fit to the flight data for the lateral degree of freedom. Similar results were also obtained for the other degrees of freedom. Table 3.3 presents the identified stability derivatives, as well as their Cramer-Rao (C-R) bounds and insensitivities showing the level of confidence of the identification. Some parameters have low confidence scores as it is difficult to improve coherence at lower frequencies. Table 3.4 shows a comparison of the model parameters extracted from the identified stability derivatives with true value estimates obtained from sources mentioned in Table 3.1. Method 2 provides more accurate estimates of the inertial parameters. As for the drag parameters, since their confidence scores are low, their estimates have large errors.

Table 3.3: Identified stability derivatives.
Method 1 uses controller generated command data and Method 2 uses control data obtained from RPM sensing.

Parameter	Method 1			Method 2			Confidence Rating
	Value	C-R %	Insensitivity	Value	C-R %	Insensitivity	
X_{v_x}	-0.11	158.2	72.59	-0.055	245.7	119.3	LOW
M_{v_x}	-4.27	5.91	2.55	-5.235	5.34	2.28	MED
M_{τ_y}	234.2	2.48	1.08	273.7	2.6	1.1	HIGH
Δ_{t_1}	0.051	3.2	1.59	0.022	7.11	3.54	MED
X_{v_y}	-1.32	16.09	7.74	-1.513	14.24	6.92	LOW
M_{v_y}	-1.89	9.12	4.08	-2.5	8.37	3.68	MED
M_{τ_x}	298.6	2.43	1.07	319.6	2.49	1.07	HIGH
Δ_{t_2}	0.048	3.38	1.67	0.023	7.08	3.51	MED
M_{τ_z}	241.8	2.63	1.32	241.6	2.64	1.32	HIGH
Δ_{t_3}	0.064	11.84	5.92	0.096	5.58	2.79	MED
X_{v_z}	-0.16	36.03	17.6	-0.15	39.98	19.54	LOW
X_T	0.99	3.81	1.88	1.045	3.81	1.89	HIGH
Δ_{t_4}	0.016	34.18	16.84	0.013	41.59	20.49	LOW

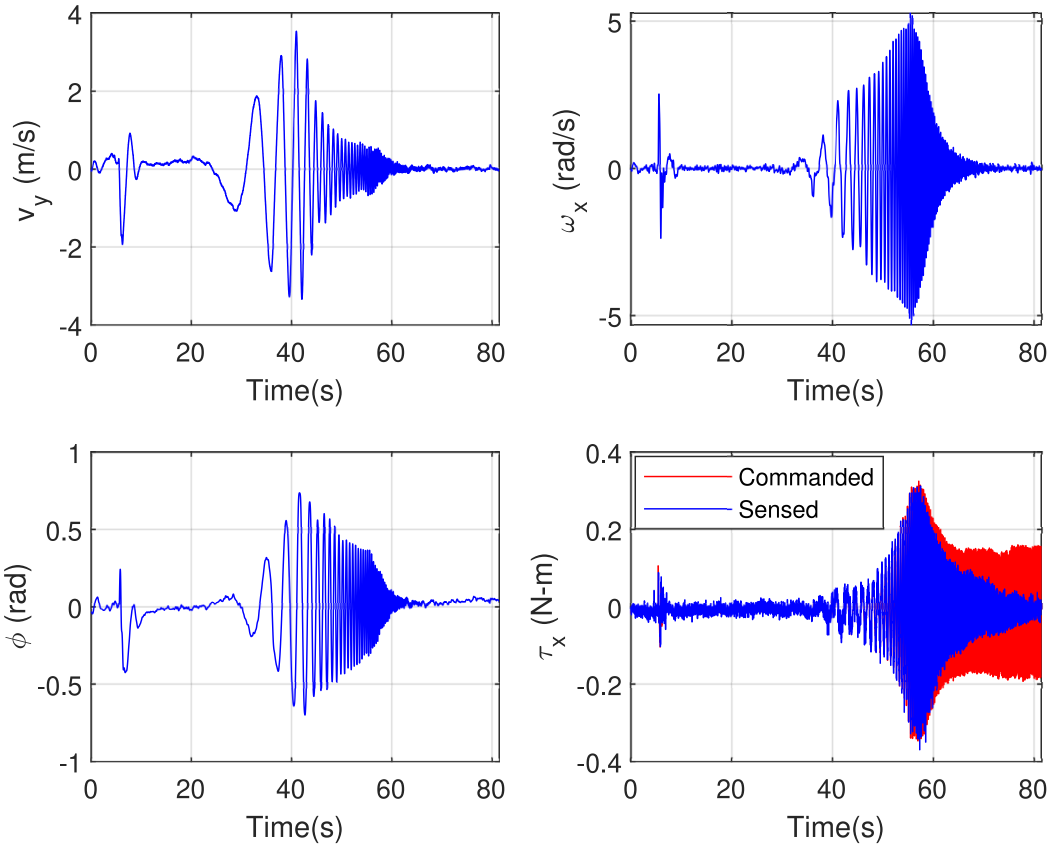


Figure 3.5: Automated frequency sweep data for lateral degree of freedom used as input to CIFER®.

3.4 Nonlinear Parameter Estimation and Control

This section presents a methodology for recursive, in-flight estimation of drag force coefficients and wind gusts. Although the nonlinear model described in Section 2.1 is used for estimation, other models can also replace it without changing the estimation methodology. Experimental results are summarized with relevant observations and discussions detailing the framework's benefits and limitations.

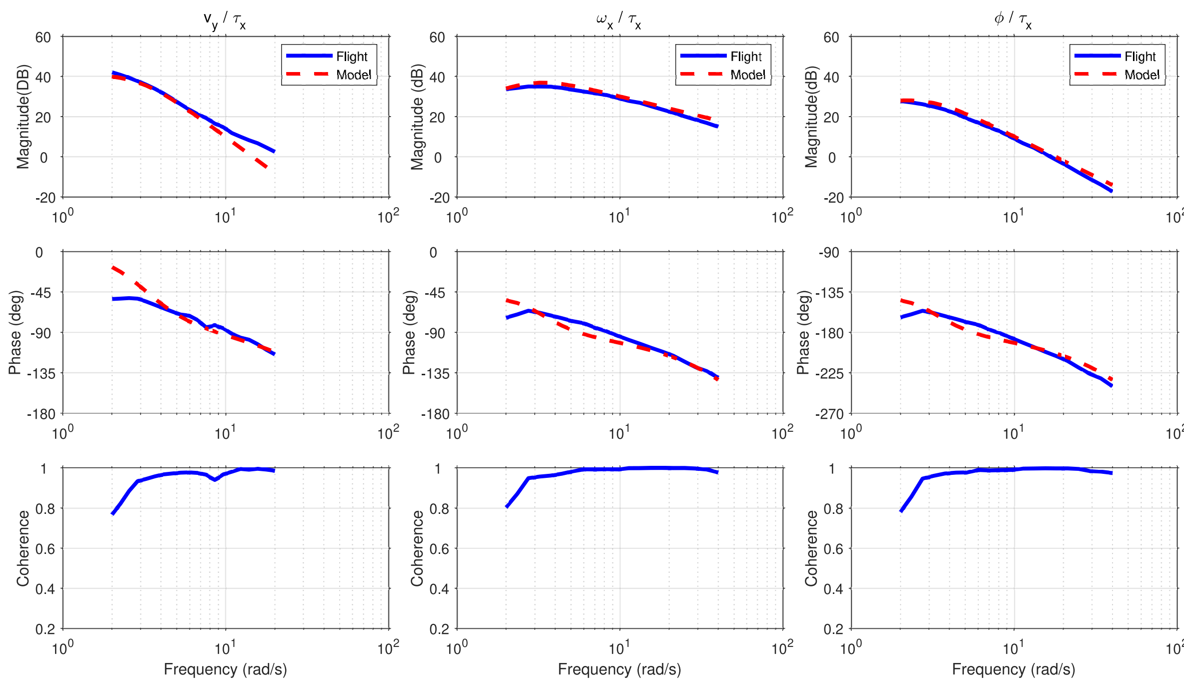


Figure 3.6: Model-data agreement using stability derivatives identified in CIFER®. Coherence represents the fractional part of the output signal power that is produced by the input at that frequency.

Table 3.4: Estimates of model parameters extracted from the identified stability derivatives Method 1 uses controller generated command data and method 2 uses control data obtained from RPM sensing.

Parameter Name	Symbol	Method 1	Method 2	True Value Belief	Units
mass	m	1.004	0.957	0.966	kg
x moment of inertia	J_{xx}	0.00335	0.00312	0.00309	kg·m^2
y moment of inertia	J_{yy}	0.00427	0.00365	0.00345	kg·m^2
z moment of inertia	J_{zz}	0.00413	0.00414	0.00423	kg·m^2
x body drag moment coefficient	C_{M1}	0.0063	0.0078	—	N·s
y body drag moment coefficient	C_{M2}	0.018	0.019	—	N·s
x body linear drag force coefficient	$\bar{C}_{D1}$	0.11	0.0526	0.21	N·s/m
y body linear drag force coefficient	$\bar{C}_{D2}$	1.325	1.45	0.42	N·s/m
z body linear drag force coefficient	$\bar{C}_{D3}$	0.16	0.143	—	N·s/m

3.4.1 Methodology

The study utilizes a coupled estimation and UAV control framework, building upon the author's previous work [55]. This framework facilitates the recursive estimation of drag force coefficients and wind parameters in separate experimental trials. Online parameter and state estimation are executed using a sq-UKF (Square-Root Unscented Kalman Filter) framework, extracting an 11-dimensional state vector from 9-dimensional sensor data and 4-dimensional control inputs.

In the wind estimation experiment, observability criteria are met during hover, enabling accurate wind parameter estimation. However, in the drag force coefficient estimation experiment, observability is compromised at zero speed. To address this, the quadrotor follows an ellipsoid trajectory, maintaining a non-zero speed to ensure consistent observability. This maneuver allows for robust drag force coefficient estimation despite the challenges posed by the experimental setup.

The state vector of sq-UKF framework includes the states and either the drag force coefficients or the wind vector, depending on the experiment. For the drag force coefficient estimation experiment, the estimator's state vector is

$$\mathbf{x} = [\mathbf{s}^{\mathsf{T}} \; \mathbf{v}^{\mathcal{B}\mathsf{T}} \; \boldsymbol{\omega}^{\mathcal{B}\mathsf{T}} \; \mathbf{C}_D^{\mathcal{B}\mathsf{T}}]^{\mathsf{T}} \tag{3.11}$$

and for the wind estimation experiment the state vector is

$$\mathbf{x} = [\mathbf{s}^{\mathsf{T}} \; \mathbf{v}^{\mathcal{B}\mathsf{T}} \; \boldsymbol{\omega}^{\mathcal{B}\mathsf{T}} \; \mathbf{v}_a^{\mathcal{B}\mathsf{T}}]^{\mathsf{T}}, \tag{3.12}$$

where $\mathbf{s} := [s_1 \ s_2]^\mathsf{T} \in \mathbf{S}^2$ is the tilt of the quadrotor expressed in stereographic coordinates. The process model of each component of the state vector and the measurement model is the same as in the original formulation [55]. The orientation estimate mean values in the stereographic coordinates are combined with the yaw value from the VIO camera and transformed to SO(3) orientation before supplying it to the SE(3) controller.

3.4.2 Drag Force Coefficient Estimation

The drag force coefficients estimation experiments are performed in an indoor flying area of approximately 4 m $\times$ 3 m and a height of 4.5 m. For safety, the flight area was fenced in the flight controller's software to 2 m $\times$ 2 m and height to 2.5 m. The quadrotor is commanded to track an ellipse-shaped trajectory as shown in Figure 3.7 whose time-parameterized expression is

$$\boldsymbol{r}_d^{\mathcal{W}}(t) = \begin{bmatrix} A_x \sin(2\pi n_x t) \\ -A_y \cos(2\pi n_y t) \\ A_z \sin(2\pi n_z t) + z_0 \end{bmatrix}, \tag{3.13}$$

where $z_0 = 1.75$ m, $n_x = n_y = n_z = \frac{1}{\pi} s^{-1}$, and the amplitudes A_x, A_y, A_z are linearly ramped up in 7 seconds from 0.25 m to 0.5 m for A_z and 0.5 m to 1 m for both A_x and A_y. The ramping up of the trajectory amplitudes is implemented to reduce the large oscillations caused by low control performance due to the inaccurate nominal values of the drag coefficients. The nominal values for the drag coefficients were selected as $\mathbf{C}_D^{\mathcal{B}} = [0.5 \ 1 \ 0]^\mathsf{T}$ to simulate a worst-case scenario where the controller's drag compensation almost eliminates the damping effects of velocity feedback, essentially starting the vehicle with an almost unstable control behavior. The z-drag coefficient

C_{D_3} is set to zero; we observed that the estimator is incapable of estimating it and, when set to a non-zero value, it can lead to divergence. Figure 3.8 shows the commanded position and the quadrotor's position-tracking performance improving with time. Figure 3.9 shows that the position and velocity tracking errors decrease rapidly in 5 seconds and oscillate around 0.2 m and 0.5 m/s, respectively. Figure 3.10 shows the drag force coefficients' estimation performance and Table 3.5 compares the final drag coefficient estimates with the true value belief. The estimates of the coefficients converge quickly to a value that is slightly higher than the value of the true coefficient belief. A possible cause is that there are other uncertainties present in the system dynamics. Also, the aerodynamic force model of (2.2) is a simplified model and does not fully capture the aerodynamic effects, thus creating residual dynamics affecting the estimation and tracking performance.

Figure 3.7: Experimental setup showing the quadrotor tracking an ellipse trajectory.

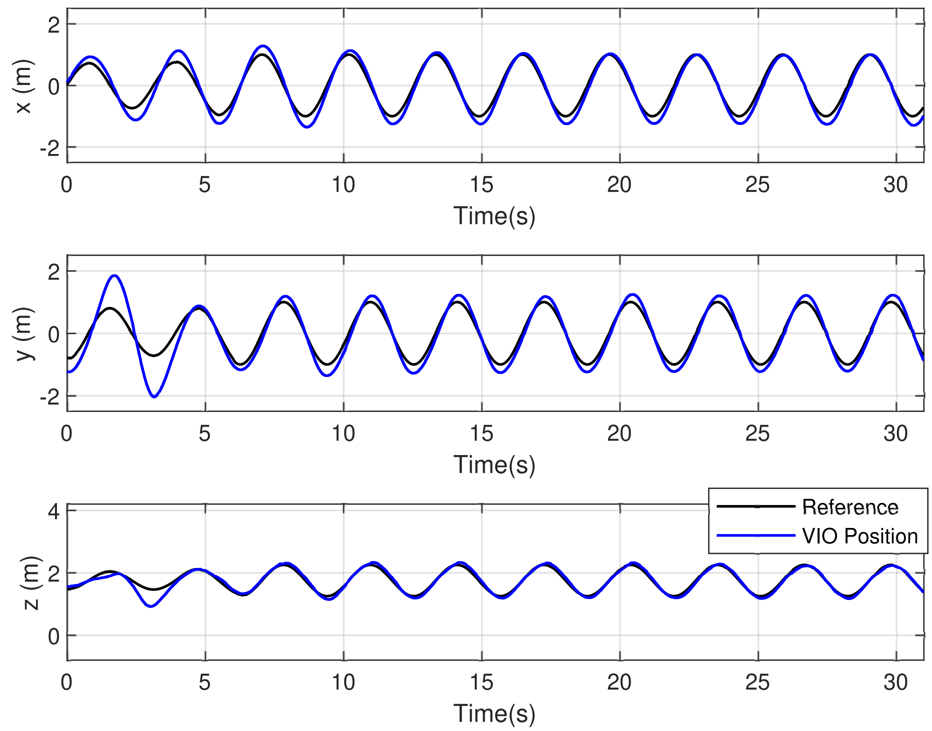


Figure 3.8: Position tracking of the ellipse trajectory.

Table 3.5: sq-UKF's final drag force coefficient estimates

Parameter Name	Symbol	sq-UKF's estimate	True Value Belief	Units
x body drag force coefficient	C_{D1}	0.13	0.07	N·s^2/m^2
y body drag force coefficient	C_{D2}	0.21	0.14	N·s^2/m^2
z body drag force coefficient	C_{D3}	0	—	N·s^2/m^2

3.4.3 Wind Estimation

The wind estimation experiments were conducted within a controlled indoor gust generation facility, as depicted in Figure 3.11. This setup features a series of eight Dyson fans strategically positioned behind servo-controlled blinds, which are intricately managed by an Arduino system. The blinds operate dynamically, automatically opening and closing to produce gusts of wind reaching velocities of 3.5 m/s, with each gust lasting for a duration of 5 seconds. To sim-

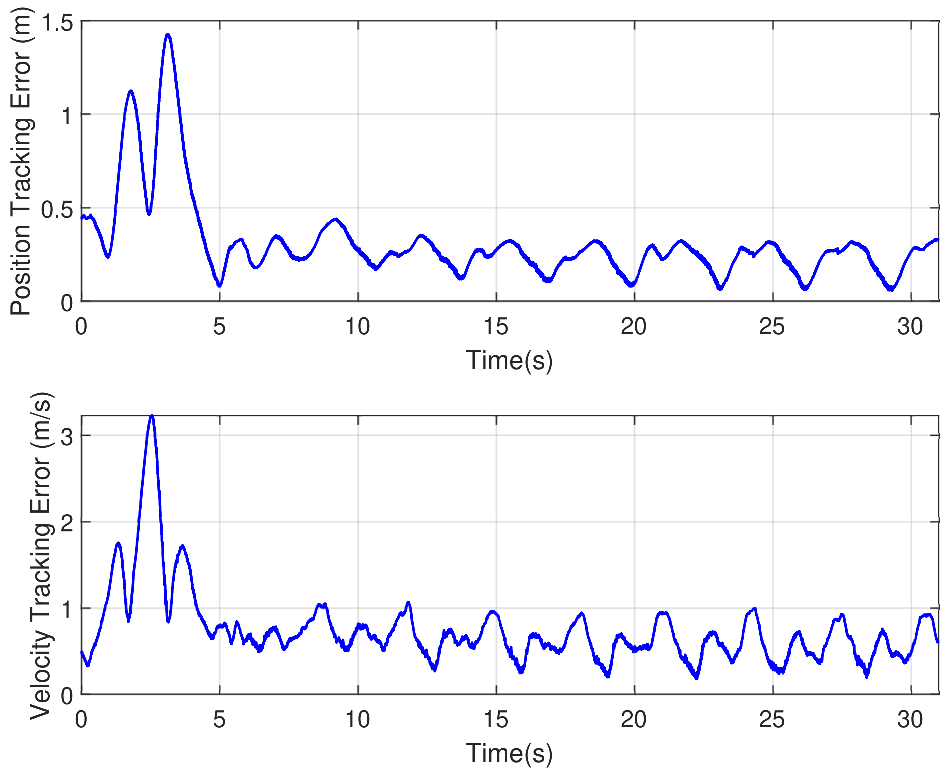


Figure 3.9: Position and velocity tracking error magnitudes.

ulate a worst-case scenario, the quadrotor's heading is deliberately aligned perpendicular to the wind direction, maximizing the surface area upon which aerodynamic forces act.

In order to gauge the efficacy of wind estimation, the control performance was meticulously assessed by comparing scenarios with and without wind estimation. This comparison was facilitated by deactivating wind estimation, effectively removing it from the estimator's state vector. Figure 3.12 provides a visual representation of the quadrotor's positional displacements induced by the wind. Notably, with wind estimation enabled, the quadrotor demonstrates enhanced resilience against wind forces, resulting in improved station-holding performance.

Furthermore, Figure 3.13 offers a comparative analysis of the wind estimates generated by the sq-UKF (Square-Root Unscented Kalman Filter) against the true wind measurements ob-

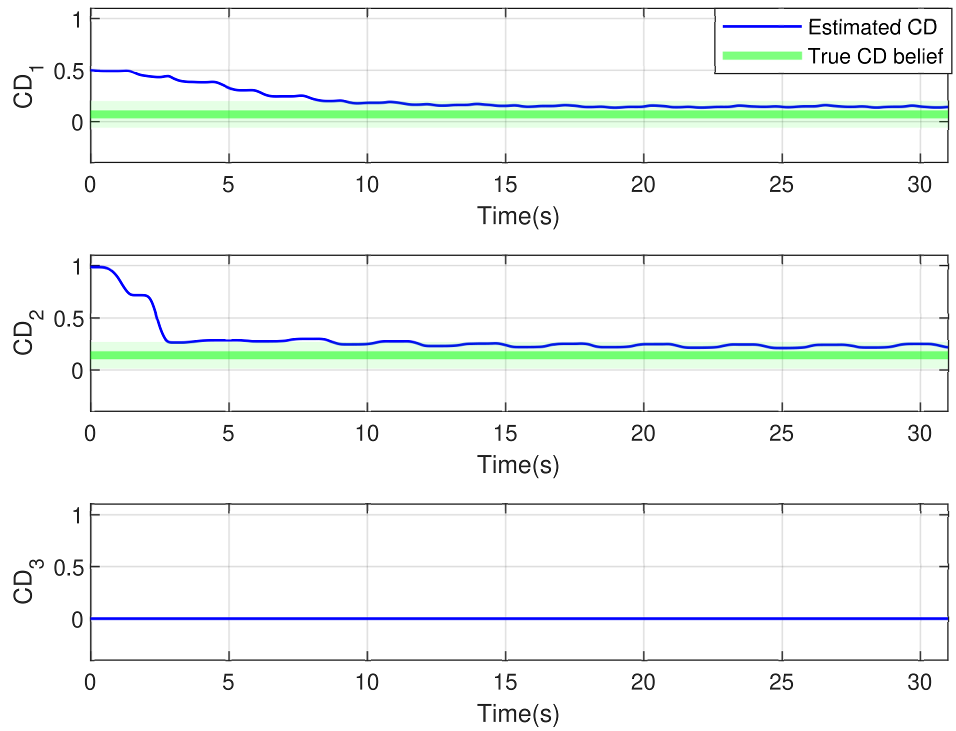


Figure 3.10: Drag force coefficient estimation performance.

tained from anemometer readings. Impressively, the estimates exhibit a high degree of correlation with the ground truth, underscoring the accuracy and reliability of the wind estimation process. It is essential to note, however, that due to the inherent nature of estimating wind based on the quadrotor's motion, a certain degree of delay is inevitable in the estimation process. Additionally, the performance of the estimator may be compromised under windy conditions, particularly during hover and no-wind scenarios, where observability is inherently diminished.

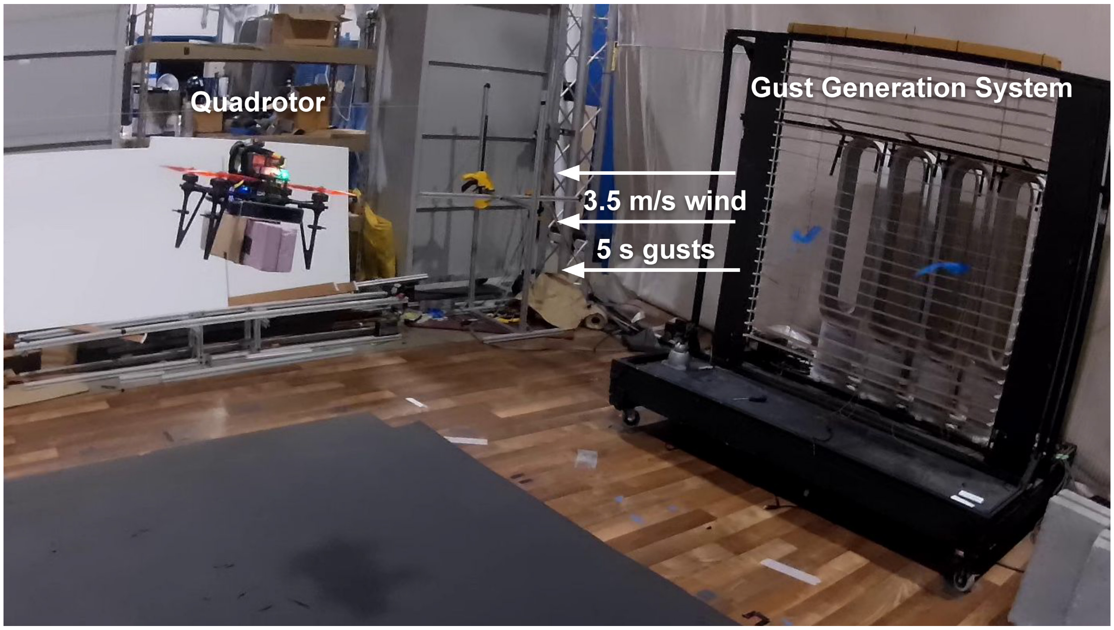


Figure 3.11: Experimental setup showing the gust generation system on the right and the quadrotor opposing the wind gust forces on the left.

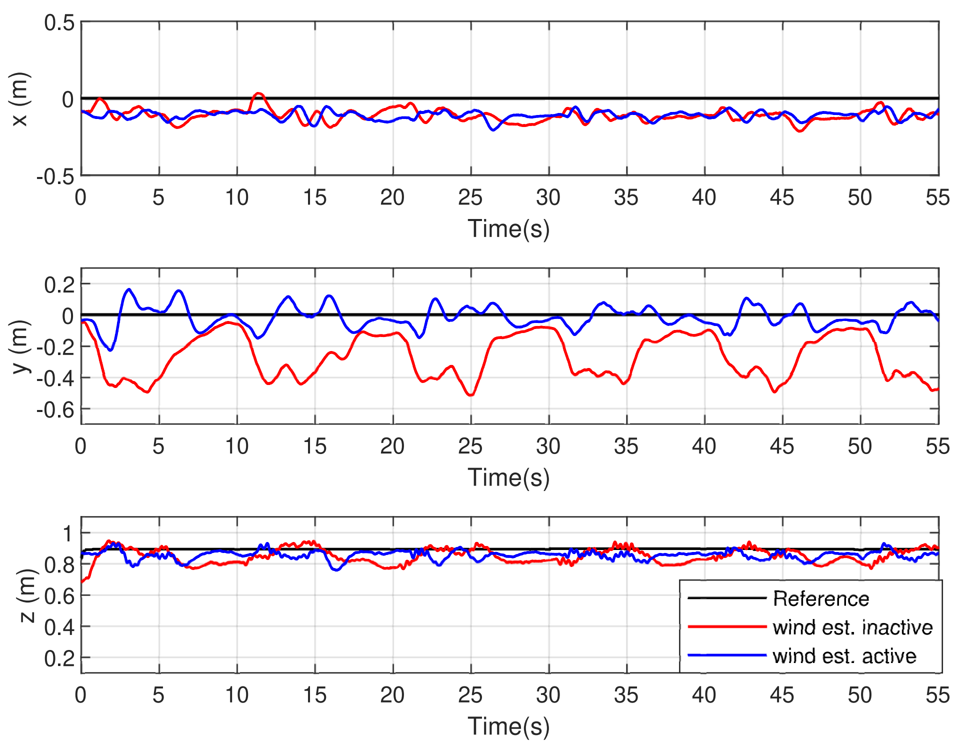


Figure 3.12: Station holding performance comparison in 5 s duration 3.5 m/s wind gusts in $-\mathrm{w}_2$ direction.

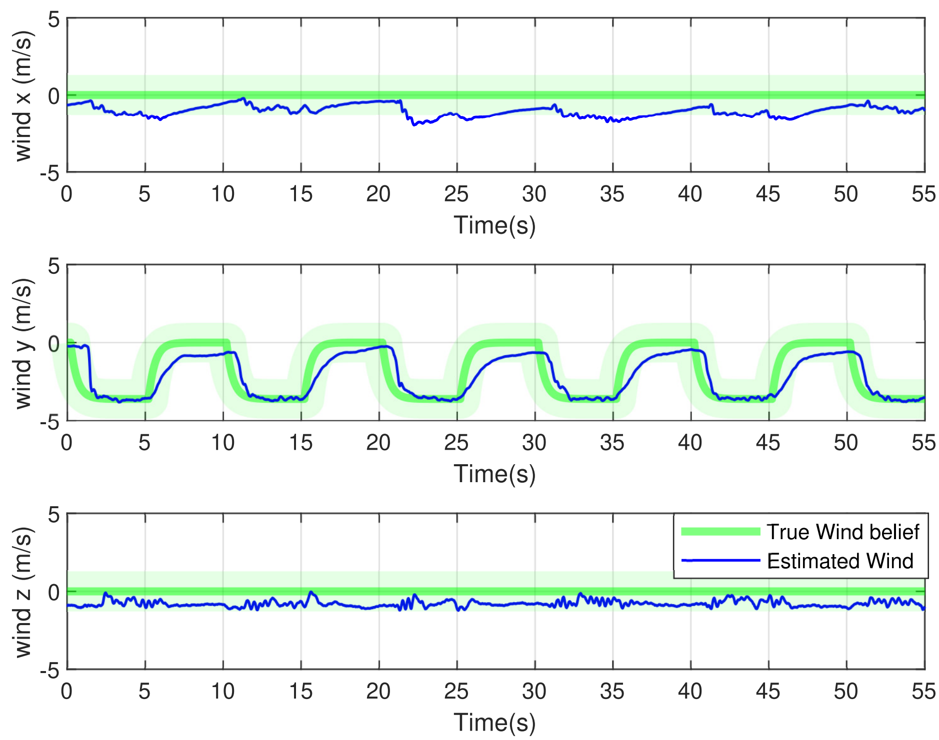


Figure 3.13: Wind estimation performance in inertial frame. Body frame wind estimates are transformed to inertial frame $\mathcal{W}$.

Chapter 4: Indoor Aerial 3D Mapping with a Multispectral Visual-Inertial Sensor System

Imagine a scenario where responders can access a real-time 3D map of an indoor environment, pinpointing obstacles, damaged structures, and the location of individuals in distress with high accuracy. Such capabilities would revolutionize the effectiveness and safety of emergency operations, providing responders with the insights needed to make split-second decisions in high-stakes situations. Unfortunately, the current landscape presents significant barriers to the widespread adoption of such advanced UAS solutions. While the technology exists in concept, commercial availability at a low cost remains elusive, posing challenges for public safety budgets and compliance with stringent national security requirements. Initiatives like the Defense Innovation Unit's Blue UAS Cleared List and the Association for Uncrewed Vehicle Systems International's Green UAS Cleared List are instrumental in setting standards and guidelines for security clearance. However, there is still much ground to cover in terms of affordability and accessibility for emergency response agencies.

This chapter presents a systematic design of a highly capable and feature-rich UAS solution, integrating various valuable works that have been performed in academia in recent years. The aim is to bridge the gap between academia and industry by creating a comprehensive and adaptable UAS to meet the needs of first responders. This UAS will provide crucial assistance to

Incident Command, facilitating rapid assessment of damage, hazard identification, and survivor location—essential components for an effective emergency response.

This chapter is structured as follows. Section 4.1 provides a detailed description of the end-result of the UAS autonomy framework design. It describes a proposed UAS autonomy levels followed by detailed descriptions of the onboard and offboard autonomy pipelines. Section 4.2 offers a complete description of the communication system's design including hardware selection and software modification that were performed to increase the overall system's run-time reliability and performance. Section 4.3 presents an overview of the UAS hardware design process including the design objectives, custom airframe manufacturing, avionics description, and thermal management. Finally, section 4.4 presents the results from real-world testing of the developed UAS, discussing the performance, effectiveness, and any observed limitations during these trials.

4.1 Autonomy Framework Design

Achieving real-time performance of a 3D mapping framework, encompassing the intricate tasks of generating dense colored point-clouds and subsequent meshing with high-resolution texturing, presents a formidable challenge in the realm of UAV technology. This challenge is chiefly rooted in the monumental computational demands inherent to these tasks, rendering them impractical for smaller UAVs where stringent constraints on weight and size of onboard avionics reign supreme. Striking a delicate balance between computational power and the physical limitations of UAVs poses a significant obstacle, necessitating innovative solutions to optimize performance without compromising functionality or jeopardizing flight stability.

One such solution lies in the strategic implementation of distributed computing, offering a pathway to alleviate the computational load burdening the UAV while simultaneously enhancing runtime reliability and ushering advanced autonomy capabilities to lighter and smaller-sized UAVs. Through the orchestration of onboard and offboard computers, a symbiotic relationship is established wherein autonomy tasks are efficiently communicated and distributed. The onboard computer assumes the role of executing lightweight software modules tailored specifically for GPS-denied flight and obstacle avoidance, thereby ensuring swift and agile maneuvering in dynamic environments. Additionally, it assumes the responsibility of collecting data from onboard sensors and compressing the acquired data to optimize transmission efficiency before dispatching it to the offboard computer for further processing.

In contrast, the offboard computer serves as the computational powerhouse, tasked with executing resource-intensive processes integral to the generation of dense 3D maps. Leveraging its robust computing capabilities, the offboard computer meticulously crafts intricate point-clouds and meshes, while concurrently running neural network models for object detection and localization. Furthermore, it undertakes the arduous task of rendering detailed textured 3D maps in real-time, thereby facilitating swift decision-making and enhancing situational awareness during UAV operations. A block diagram of the full autonomy framework is shown in Figure 4.1. Additionally, a block diagram for the onboard part of the autonomy framework is shown in Figure 4.2.

4.1.1 Levels of Autonomy

In the design of Autonomy Frameworks for aerial vehicles, the incorporation of fallback options in the event of system failure emerges as a critical consideration. Drawing inspiration from

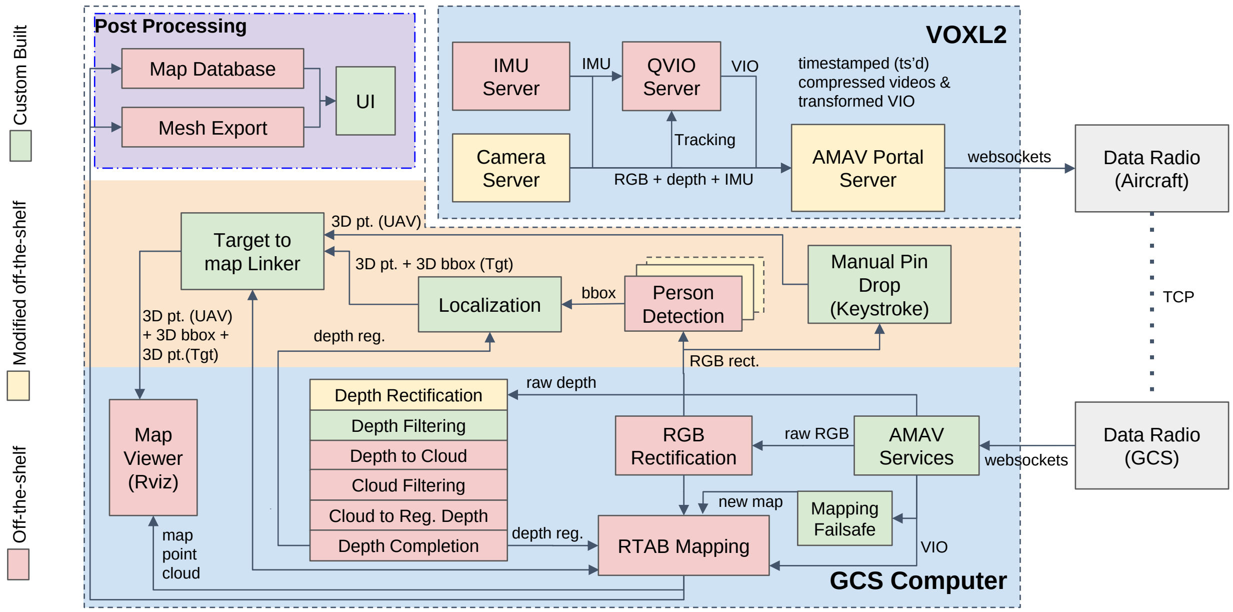


Figure 4.1: Full autonomy block diagram

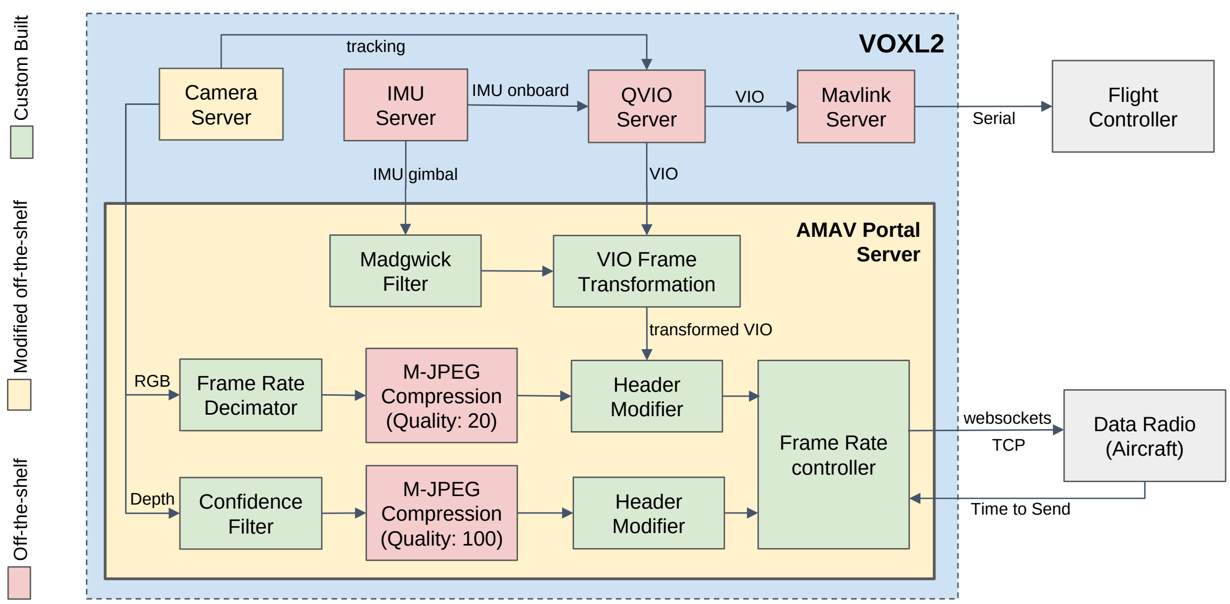


Figure 4.2: Onboard autonomy block diagram

the Autonomy Levels established for Self-Driving Vehicles, we have devised a structured hierarchy of aerial autonomy levels, shown in Figure 4.3. It is strategically engineered to minimize operator workload while maximizing operational efficiency. As the vehicle's autonomy level ascends within this framework, the reliance on human intervention diminishes progressively.

Each autonomy level is meticulously defined, with specific capabilities and design requirements delineated to ensure seamless progression through the hierarchy. At the lowest autonomy levels, human operators retain primary control, with the system serving in a supplementary capacity to assist and augment decision-making. As autonomy levels advance, the aerial vehicle assumes greater responsibility for navigation, obstacle avoidance, and mission execution, thereby reducing the cognitive burden on operators. In accordance with this framework, stringent criteria are assigned to each autonomy level, encompassing factors such as sensor redundancy, fault tolerance, and real-time decision-making capabilities. By adhering to these predetermined standards, the autonomy framework not only fosters operational safety and reliability but also fosters innovation and advancement in autonomous aerial technologies. Moreover, the delineation of clear autonomy levels serves as a roadmap for the progressive development and integration of autonomy features, guiding the evolution of aerial vehicles towards increasingly autonomous operation while ensuring compatibility with existing regulatory frameworks and industry standards.

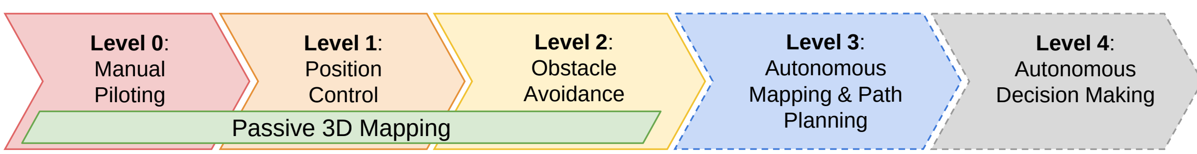


Figure 4.3: Levels of autonomy

4.1.2 Autonomy Architecture

4.1.2.1 Onboard Autonomy

An onboard computer always has computational constraints due to the aircraft's size, weight, and power requirements. To achieve real-time performance through a distributed computing approach, the onboard autonomy is designed to be extremely lightweight, containing only software modules necessary for sustaining stable flight. A combination of off-the-shelf software tools and custom-built-from-scratch software modules forms the building blocks of the autonomy framework. Some off-the-shelf modules were also modified to make them more suitable for our requirements and to resolve existing bugs that were not addressed by the developers for the specific version.

Three cameras and two IMUs are used for Localization and Mapping needs. Two mapping cameras and one IMU are present on a gimbal, while a localization/tracking camera and the other IMU are fixed to the aircraft's frame. The OV7251 global shutter camera with a fisheye lens is utilized as a tracking camera for obtaining features for performing Visual-Inertial Odometry with the help of an onboard ICM42688 IMU. The IMX412 low-light camera [68] is used as the RGB sensor for mapping purposes, capturing color images in a wide range of lighting conditions. The PMD-Tof camera [69] is used as the depth sensor for mapping, generating high-quality depth images for capturing scene structure. Finally, a secondary ICM2098 IMU on a gimbal provides inertial measurements to help in computing the gimbal orientation.

For Visual-Inertial-Odometry computation, a proprietary filter-based algorithm developed by Qualcomm is utilized. The odometry pose outputs are transformed to the gimbal frame by utilizing the gimbal's orientation estimate obtained from the gimbal IMU through the use of the Madgwick Filter. Both RGB and depth images undergo a preprocessing step. The RGB image frames are decimated to lower the frame rate to that of the depth image sensor before sending to a compression algorithm. The depth images are filtered by a confidence threshold so that lower confidence values are rejected. Confidence for depth is computed by inverting the estimated noise of the depth value of each pixel. The filtered depth frames are then sent to another instance of the compression algorithm with slightly different parameters. For compression, the M-JPEG intra-frame compression method is utilized as it is simple and fast to set up. The JPEG-quality parameter for depth is set to 100 to lower compression noise as much as possible so that the point cloud, which will be eventually computed from depth, has as few outliers as possible. Since the depth images are lower resolution 224x168 and are formatted with 8-bit monocular encoding, the resultant bandwidth consumption is low enough for reliable transmission. As for the RGB images, the JPEG-quality parameter is set to 20 in order to match the bandwidth consumption of the depth images and thereby match its transmission reliability as well. Since the bandwidth consumption of the RGB images is slightly lower than that of the depth images, the transformed VIO measurements are attached to its header along with the RGB image's acquisition timestamp. As for the depth images, only the acquisition timestamp is added to the M-JPEG frames. Both RGB and depth M-JPEG streams are then sent to a frame rate controller, which takes in the time to send each frame as feedback and switches between a low or a high frame-rate value in order to smoothly control the data packets that are being sent to the aircraft's radio through the use of WebSockets. We chose to use the Transmission Control Protocol (TCP) even though it has higher overhead since we needed the transmission to be reliable and image frames to be intact without data-loss artifacts that usually are associated with User Datagram Protocol (UDP). Using UDP will allow for lower transmission overhead, but it will destroy the map structure when there are data-loss artifacts in depth images; hence, it's important to use TCP, at least for the purpose of transmitting depth images.

4.1.2.2 Offboard Autonomy

An offboard computer, being external to the primary system, enjoys considerably looser computational constraints compared to its onboard counterpart, which is typically constrained by size, weight, and power limitations. This freedom from strict computational boundaries opens up a realm of possibilities for more complex and resource-intensive tasks. However, despite this flexibility, managing real-time performance can become a delicate balancing act, particularly when dealing with demanding processes such as running large neural networks and rendering extensive map databases concurrently. One of the primary challenges arises from the fact that while the offboard computer may have ample computational resources, ensuring seamless real-time performance remains paramount, especially in critical applications like autonomous navigation. This becomes particularly apparent when attempting to run tasks such as processing large-scale point-cloud or mesh data alongside intensive neural network computations. Additionally, the architecture of some processes may not be optimized to fully leverage the computational power of modern multi-core CPUs, further complicating the real-time performance optimization process. Therefore, when designing offboard autonomy frameworks, it becomes imperative to not only harness the computational power available but also to carefully monitor and manage memory consumption, CPU utilization, and GPU loads. Failure to do so can result in performance bottlenecks, leading to significant delays in processing and potentially compromising the real-time mapping accuracy essential for autonomous navigation systems. Despite these challenges, the relaxed computational constraints of the offboard system offer unique opportunities for integration with advanced middleware solutions such as the Robot Operating System (ROS). ROS provides a flexible and modular framework equipped with a vast array of off-the-shelf software components, enabling seamless integration and interoperability with various hardware and software components.

To bridge the gap between the onboard and offboard systems, a custom script AMAV Services [70] was developed to facilitate the transmission of data via a datalink radio. This script employs WebSockets to efficiently read and package data into individual ROS messages, effectively bringing the data into the ROS ecosystem. Within this framework, RGB and Depth image frames undergo rectification to correct for distortions caused by the intrinsic properties of their respective camera lenses. However, rectification requires different interpolation methods for RGB and depth images due to the inherent characteristics of the data. RGB images are rectified using linear interpolation, while depth images utilize nearest-neighbor pixel interpolation techniques to ensure accurate sensor data transformation. Following rectification, the RGB images are directly utilized for mapping and object detection purposes. Meanwhile, the depth images undergo a comprehensive filtering process to eliminate compression noise and outliers, ensuring the integrity of the data before transformation into a point-cloud format. Despite the diligent filtering process, the asynchronous nature of the sensors often results in discrepancies between the timestamps of RGB and depth images. To address this, a motion compensation algorithm leveraging Visual-Inertial Odometry (VIO) data is employed to align the point-cloud data with the acquisition time of RGB frames, ensuring temporal coherence. Once aligned, the point cloud is projected onto the RGB sensor frame using meticulously calibrated extrinsic transformation matrices, effectively registering the depth image onto the RGB camera's frame. This registration process yields a registered depth image, which serves as a critical component for accurate spatial perception. Following registration, an unguided depth completion process utilizing linear spatial interpolation is employed to fill in the gaps in the registered depth image, further enhancing its completeness and accuracy. The rectified RGB images, registered depth images, and timestamped VIO data are then fed into the Real-Time Appearance-Based Mapping (RTAB) [47] framework. RTAB utilizes this rich dataset to generate and store a comprehensive map representation, leveraging graph-based optimization techniques to refine the map structure whenever loop closure is detected.

In addition to the mapping pipeline, an autonomous localization framework has been developed to identify and tag objects of interest, such as humans and markers, in the environment. To achieve this, we leveraged a 2D mapping target created by the National Institute of Standards and Technology (NIST), specifically designed for evaluating camera systems' performance on UAVs. This target comprises a series of Landolt rings utilized for measuring visual acuity. Consequently, an object detection model was trained to detect these Landolt rings alongside humans to facilitate comprehensive scene understanding. For training the object detection model, we employed the You Only Look Once v8 (YOLOv8) [71] architecture. A semi-custom dataset was curated, containing samples of Landolt rings and persons extracted from the Common Objects in Context (COCO) dataset, a widely used dataset for object detection tasks. Diverse augmentation techniques were applied to the dataset to enhance the model's robustness to various environmental conditions, including changes in lighting, exposure, orientation, and occlusion. During the training process, 15% of the images in the dataset were randomly selected for validation purposes. The validation subset served to monitor the model's performance and prevent overfitting by halting the training process when the validation loss ceased to improve. This approach ensured that the model generalized well to unseen data and maintained robustness across different scenarios. At runtime, the trained model outputs 2D bounding box coordinates, detection confidence scores, and class identifiers for each detected object. Concurrently, a Multi-Object Tracking algorithm, known as ByteTrack [72], operates to assign unique IDs to individual objects in the scene, enabling robust tracking capabilities. The combined output from the object detection model and the Multi-Object Tracking algorithm is fed into the Localization module. This module employs a confidence threshold to filter out false positives and ensures that only detections with valid track IDs are considered, reducing spurious detections. Subsequently, the registered depth image is utilized to calculate the relative position of each pixel within the bounding box of the detected object. A clustering algorithm is then applied to extract foreground pixels, representing the object of interest. The mean position of these foreground pixels serves as the estimate of the object's relative position in the 3D space. Furthermore, a 3D bounding box is constructed by determining the extreme points of the foreground pixel set, providing a comprehensive spatial representation of the detected object.

4.2 Communications and Reliability Considerations

4.2.1 Radio DataLinks Hardware Investigation

Robotic systems operating in long-distance or indoor environments frequently contend with bandwidth limitations. An illustrative example involves implementing robotic solutions utilizing WiFi Halow [73] to navigate complex indoor settings, where conventional 2.4GHz or 5GHz WiFi frequencies often face obstruction from glazed glass or concrete walls commonly encountered in buildings. Wi-Fi HaLow, operating in the sub-GHz band for industrial IoT frequency range, delivers enhanced coverage and superior ability to penetrate obstacles such as walls, presenting significant advantages over traditional WiFi technologies. However, it is worth noting that one limitation of WiFi HaLow is its relatively lower data transfer rates compared to higher frequency WiFi technologies like 2.4GHz and 5GHz. The best wireless device we have identified is Microhard [74], which offers a maximum bandwidth of 8Mhz, which translates to a maximum observed throughput of 16Mbps in ideal scenarios. In most situations, we have achieved 14Mbps through careful selection of antennas. The throughput is reduced due to obstacles, range limitations, interference, antenna orientations, and antenna polarity.

Table 4.1: Performance of various Data Links tested in a semi-controlled environment. NLOS testing involved obstruction by two glazed glass walls, which contributed to signal strength attenuation by about -60 dB.

Radio / Technology	Center Frequency	Bandwidth	Transmit Power	Data rate (81ft, LOS)	Data rate (341 ft, NLOS)	Cost (2 devices)
Wifi (802.11n)	2.4 GHz	20 MHz	20 dBm (0.1 W)	$>$ 20 Mbps	0 Mbps	$<$ \$100
Wifi Halow (802.11ah)	915 MHz	1-4 MHz	20 dBm (0.1 W)	1-3 Mbps	$<$ 0.5 Mbps	\$250
Doodle Mini Mesh Rider Radio	2.4 GHz	20 MHz	30 dBm (1 W)	$>$ 20 Mbps	0 Mbps	\$3,800
Microhard Radio 2x2 MIMO pMDDL2450	2.4 GHz	8 MHz	30 dBm (1 W)	15-20 Mbps	0 Mbps	\$1,400
Microhard Radio Dual-Band pDDL900	2.4 GHz	8 MHz	30 dBm (1 W)	10-15 Mbps	$<$ 0.5 Mbps	\$1,400
Microhard Radio Dual-Band pDDL900	915 MHz	8 MHz	30 dBm (1 W)	5 Mbps	2 Mbps	\$1,400

Table 4.1 provides a comprehensive evaluation of six radio data links conducted within a semi-controlled environment. The testing setup involved placing the transmitter inside a room within a building while the receiver was positioned outdoors, replicating scenarios of Non-Line-of-Sight (NLOS) conditions typical for indoor Unmanned Aerial Systems (UAS). In this simulated environment, the challenge arose from the need for the transmitted signal to traverse two glazed glass walls upon exiting the building. These walls posed a significant obstacle, inducing

higher attenuation compared to reinforced concrete, approximately -30 dB per wall, thus resulting in a cumulative -60 dB attenuation solely from the walls themselves.

Data throughput measurements were obtained from two distinct locations: one situated at an 81 ft distance from the transmitter, benefiting from Line of Sight through a single glazed wall, and the other positioned at a 341 ft distance from the transmitter, not in Line of Sight, and contending with two glazed glass walls alongside various sparse obstacles such as trees, poles, and pedestrians. Remarkably, observations revealed the inadequacy of 2.4GHz radios, including low-power Wifi, in maintaining a stable connection at the distant 341 ft NLOS testing location. Among these, only the Microhard pDDL900 operating at 2.4 GHz managed to transmit a limited amount of data, albeit at a minimal rate. This disparity in performance was attributed to the design of radios such as Doodle Mini and Microhard 2x2 MIMO, which employ dual antennas, consequently necessitating a reduction in RF power per antenna to adhere to FCC regulations stipulating a total Equivalent Isotropic Radiated Power (EIRP) of less than 1 Watt. In contrast, radios operating at 915 MHz, namely Wifi Halow and Microhard pDDL900, exhibited the capability to sustain a robust connection since they used single antennas only. However, Wifi Halow demonstrated significantly reduced throughput owing to its lower RF output power, which is ten times less than that of Microhard and its maximum frequency bandwidth being half of that of Microhard. Notably, the pDDL900 radio, equipped with stock antennas, achieved the highest throughput of 2Mbps among all radio data links at the challenging 341 ft distance in NLOS conditions, thereby solidifying its selection for integration into the UAS design. Subsequent enhancements were made to the pDDL900 by upgrading its antennas to a superior omnidirectional variant on the aircraft and a high-gain directional antenna on the Ground Unit Radio, both boasting superior Voltage Standing Wave Ratio (VSWR) characteristics compared to the stock antennas. Additionally, adjustments were made to reduce the RF power on both the Air and Ground Unit Radio Modules to ensure compliance with FCC requirements. Further experimentation revealed that the radios perform optimally at a 27 dBm power setting, yielding a maximum throughput of around 14Mbps under favorable environmental conditions. These findings underscore the importance of meticulous evaluation and strategic optimization in enhancing the performance and reliability of radio data links in challenging operational scenarios.

4.2.2 Software Communication Configuration

ROS has gained popularity in robotics for several reasons, such as open source, modularity, and large community support. Many common packages have been developed by the community. The other key feature of ROS is its communication infrastructure, which enables different components of a robotic system to exchange data seamlessly. However, the middleware layer in ROS introduces overhead compared to direct socket communication. The overhead can impact performance, especially for real-time or latency-sensitive applications. When operating in bandwidth-limited conditions, we have noted that ROS fails to transmit any messages from VOXL2, flashed with System Image 1.1 from Modal AI, when the network bandwidth drops below 8Mbps.

To facilitate ROS-based robotics operating in bandwidth-limited environments, our approach involves having UAS directly transmit data to the ground station using socket communication. The ROS messages are reconstructed on the ground station for ROS-based applications. Table 4.2 compares the pros and cons of the socket communication and ROS. Socket communication offers a range of advantages. Firstly, its versatility transcends specific application domains, making it adaptable for various purposes beyond robotics. Secondly, developers wield precise control over the communication process, managing aspects such as data encoding, transmission, and error handling. Additionally, its language-agnostic nature facilitates interoperability across systems coded in different programming languages. Lastly, due to its direct nature, socket communication tends to operate with lower overhead than higher-level frameworks like ROS, thus proving efficient for performance-critical scenarios. Socket communication also presents several challenges. Firstly, implementing it from scratch can be intricate, particularly when dealing with low-level intricacies like network protocols and data serialization. Moreover, socket communication lacks high-level abstractions for prevalent robotic tasks such as sensor processing, robot control, and message passing, leading to increased complexity in development. Furthermore, there is a dearth of tools and libraries tailored for socket-based communication compared to ROS, potentially elongating development timelines and intensifying the effort required. Lastly, managing edge cases and error conditions in socket communication necessitates meticulous programming to uphold robustness and reliability, posing a considerable risk for errors.

	Socket Communication	**ROS**
Pros	Little overhead Low level control Efficiency	High-level abstraction Visualization and debugging features Community support
Cons	Lack of abstraction Limited features Error prone	Resource intensive Overhead Limited flexibility

Table 4.2: Performance comparison of socket communication and ROS

In environments where bandwidth is a precious resource, robotics finds a niche application in deploying indoor Unmanned Aerial Systems (UAS) for intricate 3D mapping tasks. These environments, often characterized by obstacles like glazed glass or concrete walls commonly found in buildings, present formidable challenges to establishing reliable communication networks re-

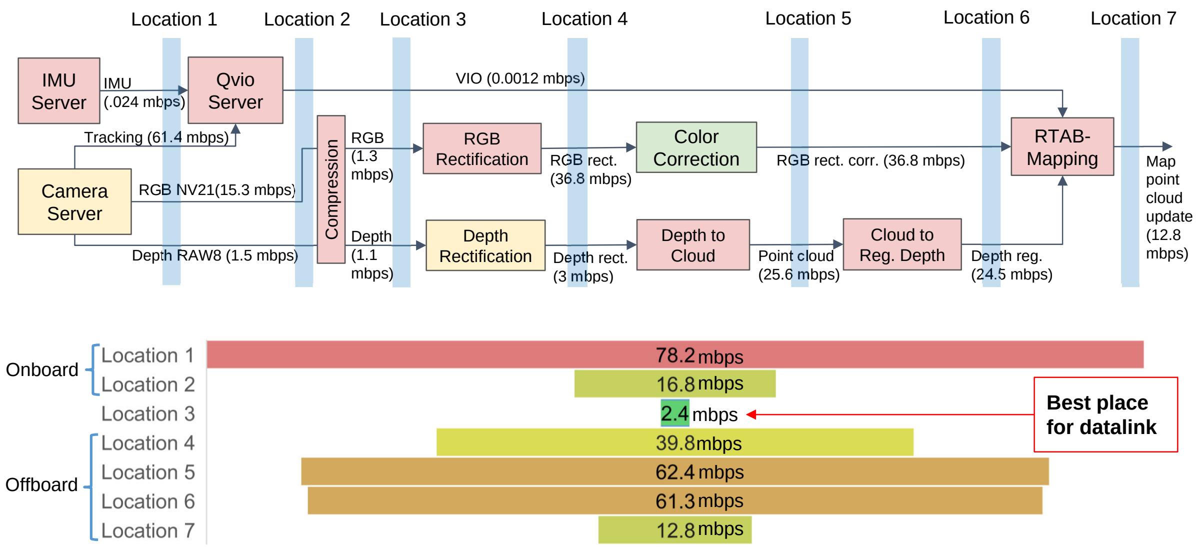


Figure 4.4: Autonomy data rate diagram

quired for mapping with an onboard and a remote ground station computer. The Microhard 915 MHz Data Link offers a promising solution in such communication-hostile environments. Operating within the sub-GHz frequency range, it provides extended coverage and improved penetration through obstructions, making it particularly well-suited for navigating the complexities of indoor environments. Despite its advantages, the Microhard 915 MHz Data Link has restricted bandwidth. This limitation necessitates a careful and strategic approach in determining which types of data should be prioritized for transmission from the onboard computer to the offboard ground station computer. Every bit of data sent must be carefully considered to ensure that essential information is conveyed efficiently while minimizing unnecessary data transfer. To aid in this decision-making process, a flattened and simplified autonomy block diagram, as depicted in Figure 4.4, serves as a visual representation of the message data rates between various blocks in the system. By analyzing the aggregated data rates at seven different locations and plotting them, as demonstrated in the figure, we can pinpoint Location 2 as having the minimum data rate. This discovery highlights Location 2 as the optimal site for establishing data link communication, as it offers a substantial bandwidth margin compared to other locations. Additionally, it inadvertently also ensures low computational load on the aircraft's onboard computer, which is beneficial for long-term reliable operations. This strategic decision significantly enhances the reliability and robustness of the overall framework, enabling seamless operations even under challenging conditions such as extreme ranges and communication-hostile environments. It ensures that the UAS can effectively navigate through complex indoor spaces, capturing and transmitting essential data for 3D mapping purposes without being hampered by bandwidth constraints. In our pursuit of accurate 3D mapping, we rely on RTAB-Map [75]. This powerful toolset requires the publication of critical data streams, including point cloud, RGB image, and camera pose data, via ROS topics, with a minimum frequency requirement of 1 Hz. The camera pose data, in particular, plays a crucial role in the mapping process. It is computed by applying a fixed transformation to the computed pose estimates from Visual-Inertial Odometry (VIO). This fixed transformation, determined during the extrinsic calibration procedure, ensures the alignment of the camera's viewpoint with the UAS's position and orientation, facilitating accurate and consistent mapping results.

Table 4.3 provides a breakdown of the topics reconstructed on the ground station, along with the associated message sizes transmitted via sockets. For optical RGB imaging, we utilize the Starvis IMX412 sensor [68]. The RGB image resolution is 640 x 480 pixels, with a raw image size of 921.6 KB, which can be reduced using M-JPEG compression. The original point cloud consists of 40,000 points encoded with mono16, requiring 2 bytes per point, resulting in approximately 80 KB per message or a transmission rate of 640 Kbps when messages are sent at 1 Hz. Alternatively, we reconstruct the point cloud from depth images captured by the Modal AI VOXL Time of Flight (ToF) Depth Sensor [69]. The depth image resolution is 224 x 171 pixels, with a raw size of 38 KB. Pose data from Visual Inertial Odometry (VIO) includes position (x, y, z) and orientation in quaternions (qx, qy, qz, qw), with each double consuming 8 bytes, totaling 56 bytes. Additionally, VIO quality, represented as an integer between 0 and 100, is included, computed using the inverse of the largest diagonal element of the covariance matrix. To ensure optimal mapping quality, all topics must be synchronized. Therefore, when transferring data from VOXL2 to the ground station via sockets, timestamps (27 bytes) for VIO, RGB image, and depth image are included. This facilitates accurate reconstruction of ROS messages on the ground station. VIO data, VIO quality, and timestamps are concatenated with the images for transmission through the same socket. Socket communication allows precise selection and control over the transmitted data, facilitating efficient transmission for 3D mapping with less than 3 Mbps bandwidth. RGB images are compressed to less than 30 KB, and depth images to less than 35 KB. Depth images are less compressed to preserve their quality, improving the fidelity of reconstructed point clouds. The source code for serializing the data on VOXL2 can be found in the repository [76], while the repository for deserializing the data to reconstruct topics on the ground station is available [70]. Despite the memory-intensive nature of 3D mapping and the tendency of existing mapping apps on smartphones to crash due to memory shortages, our communication setup enables RTAB-Map to operate on a laptop with fewer memory constraints.

4.3 Hardware Design and Evolution

4.3.1 UAS Design Objectives

The UAS was designed and manufactured with a comprehensive vision: not just to validate a newly formulated mapping framework but to elevate its utility to the level of established

Topic	Message Size	Bandwidth Usage (1 Hz)	Bandwidth Usage (5 Hz)
RGB image	15 - 30 KB	120 - 240 Kbps	0.6 - 1.2 Mbps
Depth image	15 - 35 KB	120 - 270 Kbps	0.6 - 1.35 Mbps
Odometry	56 B	448 bps	2.24 Kbps
Odometry quality	2 B	16 bps	80 bps
Timestamp	27 B	216 bps	1.08 Kbps
Total	30 - 65 KB	240 - 510 Kbps	1.2 - 2.55 Mbps

Table 4.3: Size of messages per topic for 3D mapping. Raw images undergo M-JPEG compression for size reduction.

commercial solutions for indoor Search and Rescue (SAR) operations. This ambitious goal necessitated meticulous attention to several key factors. Foremost among these was the importance of achieving a high level of reliability and real-time performance for both the mapping framework and the detection and localization of targets within indoor environments. Concurrently, the UAS has to adhere to strict size and weight limitations, ensuring nimble maneuverability within confined spaces. Complementing these core objectives was the integration of a diverse suite of onboard sensors tailored to the unique demands of indoor SAR missions. The inclusion of high-definition (HD) First Person View (FPV) video capability provides operators with clear, immersive visuals essential for navigating complex indoor layouts with precision and confidence. Meanwhile, a Night Vision-capable camera system provides critical support for operations in environments with limited or zero illumination, ensuring SAR efforts can continue unhindered by darkness. Additionally, thermal imaging technology assists in detecting human body heat signatures, facilitating swift localization of individuals in need of rescue, and aiding in the identification and characterization of fires, a common hazard in emergency situations. Beyond sensor technology, the UAS boasts a suite of advanced functionalities expressly designed to enhance its effectiveness in SAR missions. A two-way audio system enables direct communication with victims, fostering a vital lifeline of reassurance and information exchange in the midst of a crisis. The inclusion of a full 180° pitch range gimbal augments situational awareness by affording comprehensive views of the surrounding environment, enabling operators to swiftly assess and respond to dynamic circumstances. Furthermore, the incorporation of turtle/auto-flip mode ensures mission continuity in the event of an unforeseen mishap, minimizing downtime and maximizing operational efficiency. Moreover, to safeguard both the UAS and its surroundings, a simple collision avoidance capability is useful, serving as a critical safety net against accidents that could jeopardize rescue efforts. Finally, it's important to keep the total cost low and affordable for first responders and to also comply with Blue UAS requirements to make the UAS available to DoD and Federal Government partners. Keeping these objectives in focus, we developed a competitive UAS named *Intrigue*, and its specifications are presented in Figure 4.5 and are compared with those of existing top-tier commercial SAR UAVs.

Feature	Intrigue	Skydio 2+ *	Skydio X10	DJI Mini4 Pro	DJI Mavic 3T	Elios3
Size	13 x 11 x 3 in.	16 x 15 x 5 in.	31 x 26 x 6 in.	15 x 12 x 4 in.	23 x 21 x 5 in.	20 x 20 x 14 in.
Weight	2.4 lbs.	1.7 lbs.	4.7 lbs.	0.55 lbs.	2.1 lbs.	5 lbs.
Hover Flight Time	18 min	27 min	35 min	30 min	38 min	<9 min
HD Video LOS Range	4 miles	0.5 mile	3 miles	4 miles	4 miles	<0.1 mile
Real-Time Mapping	Detailed in color	Low-Polygon	Low-Polygon	No	No	Uncolored
Post-Process. Map	On GCS	In Cloud	On UAV	In Cloud	In Cloud	No
Night Vision	Yes	No	Yes	No	No	No
Gimbal Tilt Range	180°	120°	180°	150°	125°	180°
Thermal Resolution	160x120	No	640x512	No	640x512	160x120
Two-way Audio	Yes	No	Yes	No	Yes	No
Integrated Rotor Protection	Yes	No	No	No	No	Yes
Turtle Mode	Yes	No	No	No	No	Not Req.
Human Detection & Location Tagging	Autonomous	Manual	Manual	No	Manual	No
Collision Avoidance	Yes	Yes	Yes	Yes	Yes	No
Blue UAS	Capable	Compliant	Compliant	No	No	No
Cost	$6.5k	$2.2k	>$15k	$1k	$7.5k	>$70k

* No longer in Production

Figure 4.5: Comparision of Intrigue UAS with top-tier commercial search and rescue UASs

4.3.2 UAS Overview

An external overview of the final design iteration of the intrigue UAS is shown in Figure 4.6. Locations of major individual hardware components are labeled with red arrows. The UAS structure comprises two CNC-milled carbon fiber plates connected together using Aluminium standoffs and steel fasteners, creating a rigid, strong, and durable frame while keeping material and manufacturing costs low. The top plate is 3 mm thick since it's the primary load-bearing structure, and the bottom plate is 2 mm thick to aid in the airframe's rigidity. Additionally, covers to protect the electronics and also to aid in cooling were 3D printed using nylon reinforced with chopped carbon fibers. The gimbal structure was also 3D printed using the same material. Appropriate mounting holes were strategically drilled on the carbon fiber plates to securely mount the avionics components. Vibration isolation standoffs were used to mount the flight controller and the onboard computer. Antennas were strategically mounted to place them as far away from the conductive frame as possible without adding too much weight from the co-axial extension cables/rods. Additionally, appropriately rated capacitors were integrated into the power source for ESC and Microhard radio to attenuate voltage spikes and electrical noise generated from rapid motor acceleration, active motor braking, and switching regulators.

A block diagram of the complete set of avionics hardware on the Intrigue UAS is presented in Figure 4.7. The arrows represent wired connections between the components and the direction where the majority of the data goes from one component to another. The dashed lines and a wireless icon on a block represent wireless communication. The colors represent individual avionics sub-groups, i.e., mapping, flight control, FPV video, control radio, gimbal, audio, thermal, and illumination systems. The Ground Station Laptop can be any generic laptop running

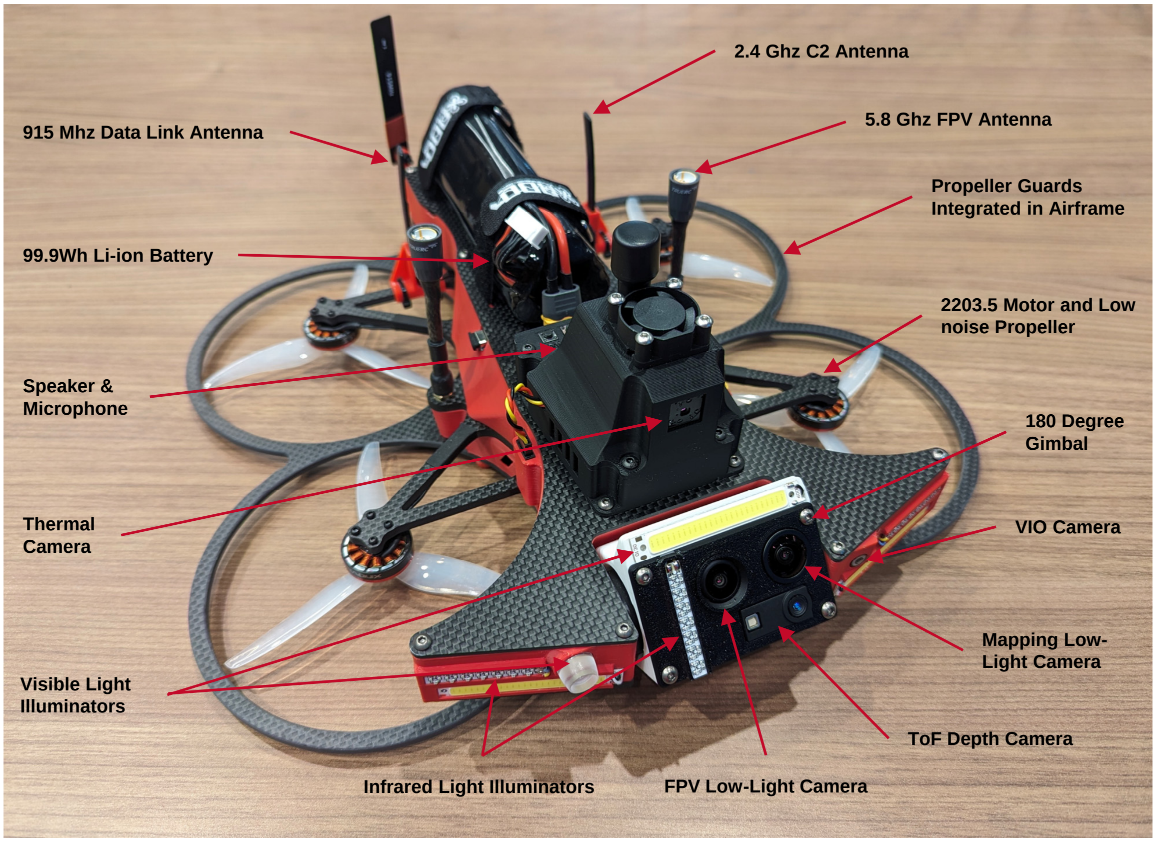


Figure 4.6: Intrigue UAS overview

Ubuntu 20.04 Operating System configured with ROS noetic, latest stable Rtabmap and corresponding AliceVision software packages. We have successfully tested the software framework on multiple laptops, but for the purpose of this manuscript, we used the Gigabyte Aero 15 XC Laptop equipped with a 10th Gen Intel Core i7-10870H (2.2GHz) CPU, 16 GB DDR4 RAM, and NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB GDDR6 VRAM.

4.3.3 Multispectral Visual-Inertial Sensor System

A gimbaled multispectral visual-inertial sensor system was developed to provide complete freedom for performing mapping from any desired pose. The gimbal provides pitch axis control

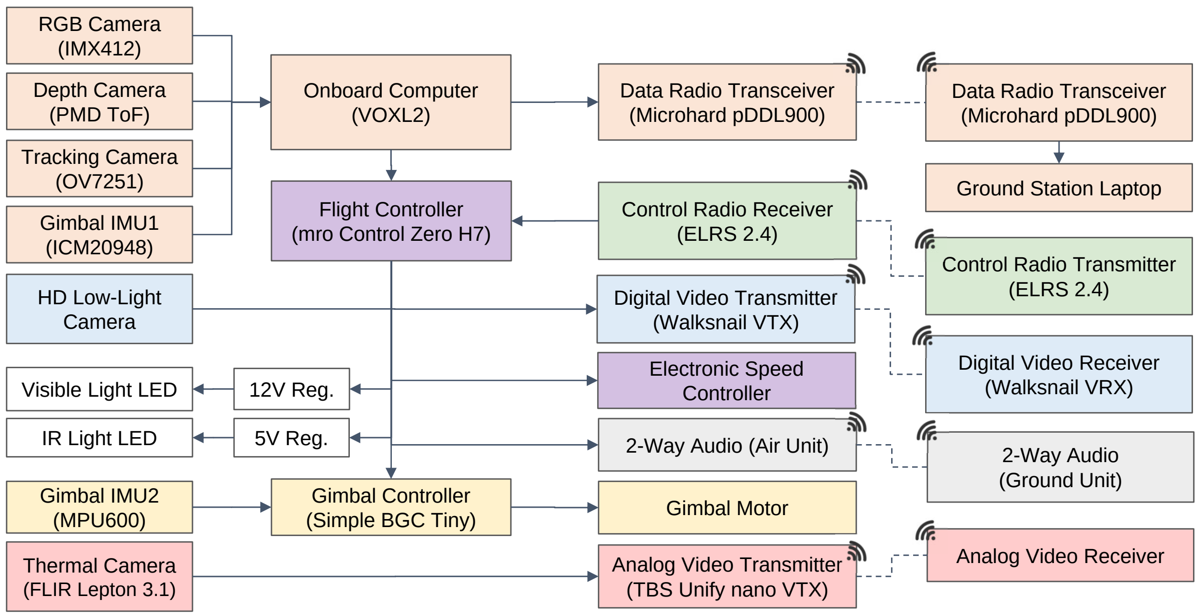
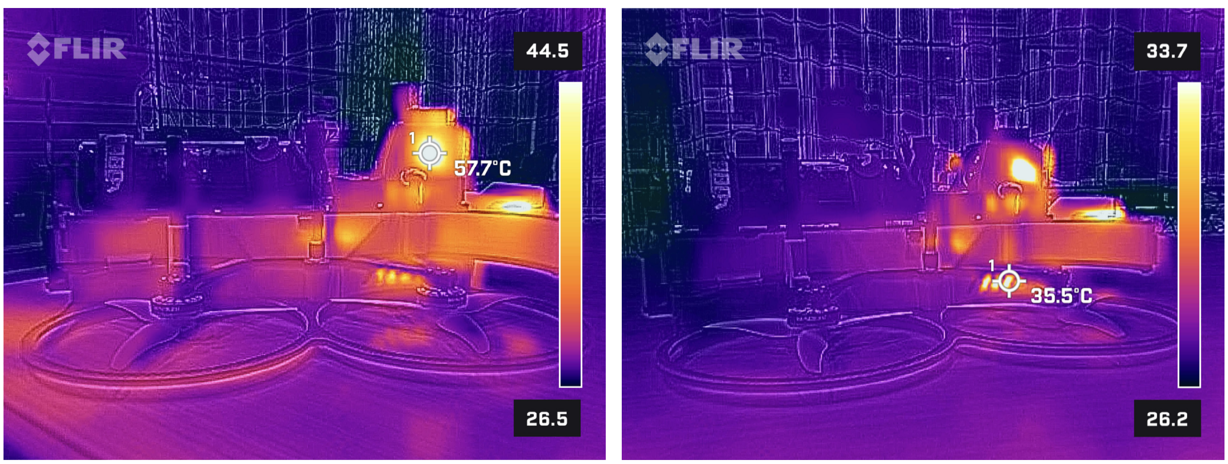


Figure 4.7: UAS avionics hardware block diagram

and houses the FPV low-light camera, RGB camera, ToF camera's IR emitter and receiver system, and two IMUs. One of the IMUs is used by the gimbal controller to stabilize and control the gimbal's pitch angle, whereas the other IMU is used by the onboard computer to calculate the poses of individual sensors in the camera system. It must be noted that the the ToF camera was chosen because of its lower size, weight, and power benefits. A LiDAR on the other hand would have been superior in terms of performance and range.

4.3.4 Avionics Cooling Considerations

Some avionics components inside the UAS generate a considerable amount of heat, and hence, it is imperative to integrate proper cooling measures in the design to prevent damage and improve runtime performance and stability. Appropriately sized aluminum heat sinks were installed on the Microhard Radio, VOXL2, and the FPV video transmitter. Ambient air is also forced inside the avionics bay with the help of a 40 mm and a 20 mm fan mounted on the back and top side of the UAS respectively. Without active cooling from fans, we observed a maximum temperature of 93°C on the VTX, 79°C on the Microhard Radio, 71°C on VOXL2, and 45°C on the Gimbal motor during benchtop testing. With active cooling from the dual fans, the temperatures were brought down to 70°C on the VTX, 42°C on the Microhard Radio, 44°C on VOXL2, and 34°C on the Gimbal motor. Figure 4.8 shows a side view of the active vs cooling performance of the UAV with a thermal heatmap overlaid. The heat from the internal avionics heats up the 3D-printed side walls, and it's the outer surface temperature that is plotted in the images.

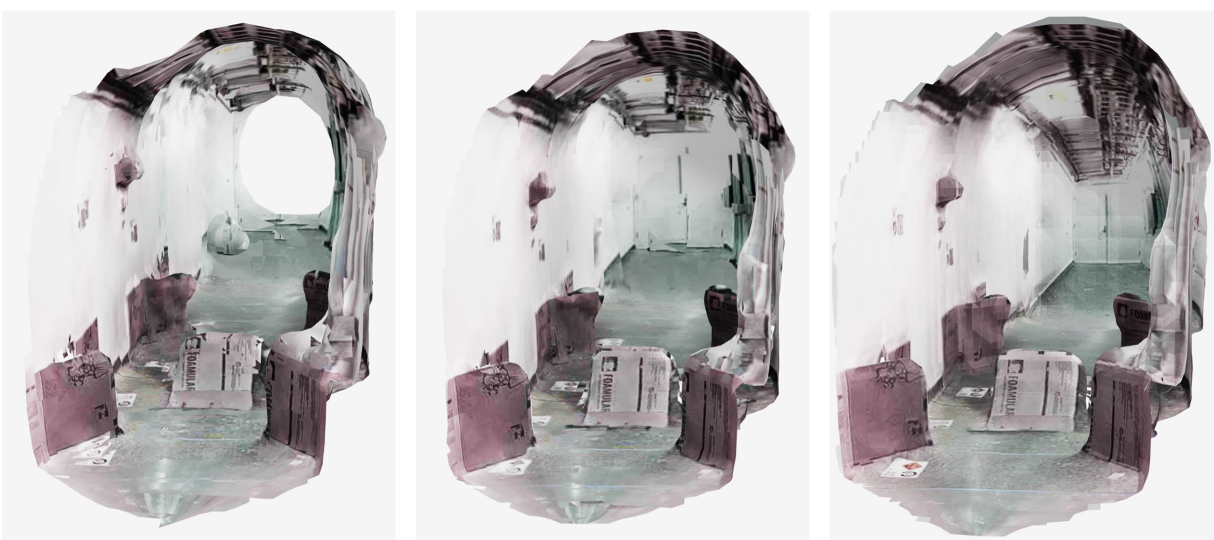


(a) Passive cooling (b) Active and passive cooling

Figure 4.8: Avionics cooling performance during benchtop tests

4.4 Experimental Testing Results

Rtabmap outputs map data into a database file (.db) format, which requires post-processing to generate the final textured map files. This includes the conversion to a wavefront geometry (.obj) file, material library (.mtl) file, and texture image (.jpg) file. Figure 4.9 illustrates the two primary post-processing steps employed to enhance the fidelity of the mesh map. For the experiment, the Intrigue UAS flew in a straight line within a narrow corridor designated for NIST's UAS 3D Mapping Lane Test. The corridor featured fiducial foam structures for measuring map dimensional accuracy, foam cubes on walls and floors to assess captured shape details, and NIST ring targets on letter paper affixed to the wall to evaluate visual acuity. Figure 4.9a displays the original mesh map without any post-processing. In contrast, Figure 4.9b exhibits the mesh map after the bundle adjustment procedure. Bundle adjustment refines the map's shape and texture by minimizing the re-projection error of landmark features across the camera pose-graph data structures. This refinement enhances overall accuracy. The final step, depicted in Figure 4.9c, involves Multiband Blending, a feature provided by AliceVision [77], an open-source Photogrammetric Computer Vision tool. This process eliminates edges between adjacent texture patches and corrects texture matching to individual mesh cells, resulting in increased sharpness and resolution across the mesh map.



(a) Before post-processing (b) After bundle adjustment (c) After multiband blending

Figure 4.9: Post-processed output map after each stage.

In industry, the most widely used technique for generated 3D maps is photogrammetry.

This technique involves capturing multiple photographs from different angles, which are then analyzed to create accurate 2D or 3D models of the subject. It is widely used in fields such as cartography, architecture, engineering, geology, and archaeology, enabling detailed documentation and analysis of structures, landscapes, and historical sites. It leverages advancements in digital imaging and computer vision to provide precise spatial data and measurements. Figure 4.10 shows the results of the industry standard Photogrammetry pipeline and presents a side by side comparison with the map output from our method. To generate the photogrammetry map the 1080p HD video from the onboard FPV camera was first extracted from the aircraft's onboard storage. Next, it was downsampled to 5 Hz and all individual frames were extracted and provided as input to the photgrammetry pipeline inside the Meshroom software [77].

Similarly, a notable recent advancement in 3D capture techniques within academia is the use of Neural Radiance Field (NeRF). NeRF is a neural network capable of reconstructing intricate three-dimensional scenes from a limited collection of two-dimensional images. NeRF works by optimizing a continuous volumetric scene function using a sparse set of input images along with their corresponding camera poses. This allows it to model the light radiance in every direction from any point in a given scene. Essentially, NeRF can synthesize new views of a scene by predicting the color and density at any point in 3D space, thereby creating a highly detailed and accurate 3D representation from just a few 2D snapshots. Figure 4.11 shows the results of training a nerfacto model using Nerfstudio software [78]. The same downsampled images as those used in the photgrammetry comparision was pre-processed by COLMAP [79, 80] and the resultant frames and poses were supplied as input to Nerfstudio. Figure 4.11a shows a snapshot of the volumetric rendering of the NeRF model. NeRF's performance heavily relies on the quality and quantity of the input images. Hence, the sparsely distributed images leads to a degradation in quality of the reconstructed 3D scene. Figure 4.11b shows the mesh output exported from the trained NeRF model. It's inferior in quality since NeRF models the scene as a continuous volumetric function, which is difficult to convert to mesh geometric representations.

The complete photogrammetry pipeline with default parameter values took about 15 minutes on the Ground Station Laptop. On the other hand the NeRF pipeline took about 10 minutes to train on a powerful desktop computer equipped with RTX4090 GPU. Table 4.4 provides an overview of the comparision of the two techniques against our method. However, it must be noted that photogrammetry and NeRF have been developed to be a general purpose mapping tool which doesn't require any camera calibration steps and it's output can be improved with careful tuning of the pipeline parameters depending on the type of environment we are trying to map. Neverthless, photogrammetry is unable to predict depth values of unfeatured surfaces which leads to large gaps in the map as is evident in Figure 4.10a. On the other hand, NeRF is unable to fully capture the scene information from the limited amount of images and camera viewpoints supplied to it. In contrast, our technique based on RGB-D mapping results in a more complete map output with less surface gaps as shown in Figure 4.10b.

Table 4.4: Comparison of pipeline properties of our mapping technique with photogrammetry tool and NeRF technique.

Property	**Photogrammetry**	**NeRF**	**Our Method**
Real-Time vs Offline	Offline	Offline	Real-time
Image format used	1080p RGB	1080p RGB	480p RGB + 168p Depth
Post-processing time of NIST 3D Mapping Lane	15 min	10 min	$<$ 1 min
Sensor Requirements	Uncalibrated RGB Camera	Uncalibrated RGB Camera	Calibrated RGB and Depth cameras
Scale Ambiguity	Present	Present	Absent

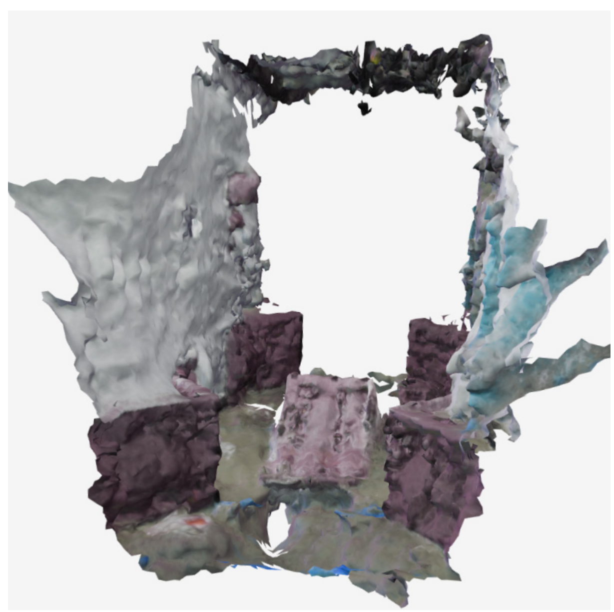
The development of the Intrigue UAS underwent meticulous testing across various sim-

(a) Map using photogrammetry.

(b) Map using our method.

Figure 4.10: Comparison of resultant maps from our mapping technique with the one from an industry standard photogrammetry tool.

(a) NeRF viewport.



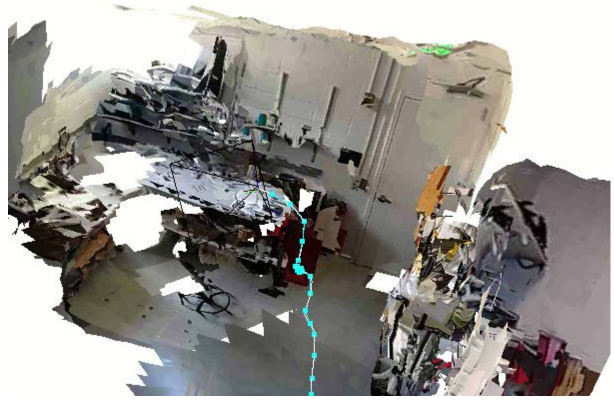
(b) NeRF mesh output.

Figure 4.11: Mapping results from NeRF technique.

ulated indoor missions to ensure optimal performance of both its hardware and software components in real-world scenarios. These missions served as vital proving grounds, enabling the team to assess the UAS's capabilities and refine its functionalities. A primary focus during these missions was achieving real-time mapping, target detection, and localization visualizations. To accomplish this, tools like Rviz and Rtabmapviz were utilized, providing valuable insights into the UAV's surroundings for swift decision-making and course corrections as needed. Furthermore, a custom-designed user interface (UI) was developed to comprehensively analyze mission outcomes. This UI served as a platform for post-processing analysis, presenting collected data in a gallery-style format. Through this interface, team members could review detailed mesh representations, pinpoint target locations, and assess the accuracy of target detections. A critical aspect of the evaluation process involved comparing different rendering techniques. Figure 4.12 illustrates one such comparison, focusing on real-time map rendering details. By presenting the same scene side by side, one with the display of point-cloud data and the other with the application of the FastMesh algorithm, the figure highlights the significant enhancement in textural detail achieved through the latter approach. The FastMesh procedure, detailed in FastMesh [81], involves connecting neighboring points (pixels in depth image space) to construct a quad mesh. This process enhances the visual fidelity of the mapped environment, providing a more detailed picture of the surroundings and objects within it. However, in our current software state, localized targets and objects in the map cannot be displayed in Rtabmapviz. Therefore, we utilize Rviz to show the map point-cloud and target locations in real-time, while the custom-developed UI displays the final post-processed mesh map along with the target locations.

The Intrigue UAS underwent a rigorous evaluation in a simulated indoor search and rescue mission orchestrated by the National Institute of Standards and Technology (NIST) as part of

(a) Real-time map rendered as a point cloud by Rviz.



(b) Real-time FastMesh generation in rtabmapviz.

Figure 4.12: Comparison of real-time point-cloud and real-time mesh map rendering.

the NIST UAS 5.0 First Responder UAS 3D Mapping Challenge. This mission was designed to assess the UAS's mapping capabilities as well as detecting and localizing various objects within a highly cluttered and hostile environment. In Figure 4.13, a snapshot from the Rviz visualization tool showcases the UAS's autonomous detection of a person mannequin in the RGB image. Impressively, the system accurately localized the detected individual within the point-cloud map, represented by distinctive red spheres. Similarly, Figure 4.14 presents another instance of successful detection and localization, this time featuring NIST ring targets. The corresponding pink spheres in the point-cloud map precisely indicate the locations of these targets, demonstrating the UAS's ability to identify and map objects of interest in real-time. Despite encountering challenges during the mission, such as premature crashes leading to missed loop closures and odometry drift errors, the resulting map, showcased in Figure 4.15, remained remarkably useful. Despite these setbacks, the map retained numerous intricate details, as highlighted in Figure 4.16. For instance, Figure 4.16a offers a detailed view of a mapped room containing a hidden person behind a couch. The fidelity of the mapping captured the room's shape and textures in great detail, underscoring the UAS's capability to provide detailed reconstructions of complex environments. Additionally, Figure 4.16b showcases another significant observation, illustrating a narrow room featuring a NIST ring target positioned on the ceiling. The flexibility afforded by the mapping sensors mounted on a gimbal proved invaluable in capturing intricate details, particularly in scenarios requiring upward or downward detection. A combined recorded video of the GCS laptop screen and the FPV video feed for the NIST simulated indoor mission can be found at [82]. Overall, despite encountering operational challenges, the Intrigue UAS demonstrated significant capabilities in autonomous detection, localization, and detailed mapping of complex indoor environments. These capabilities hold considerable promise for applications in search and rescue operations, disaster response, and various other critical scenarios where precise spatial awareness is paramount.

Figure 4.13: Real-Time detection and localization of a person

In addition to its evaluation in simulated search and rescue scenarios, the Intrigue UAS underwent testing within a building located at the University of Maryland (UMD), where it encountered a controlled and cluttered environment containing mannequins. Figure 4.17 offers a

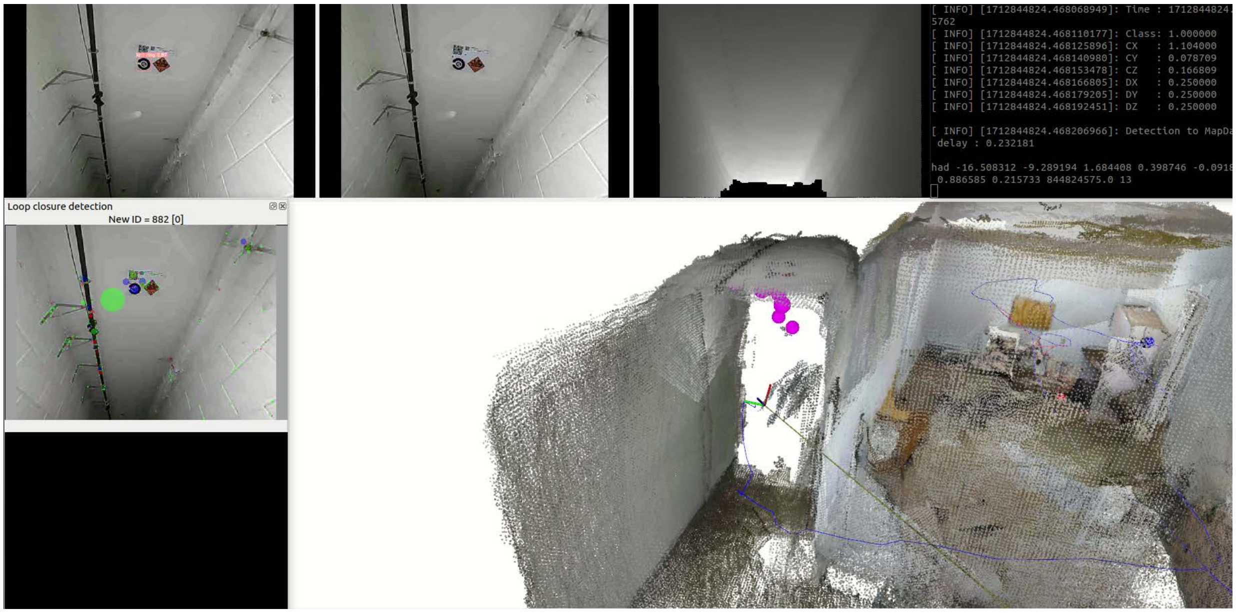
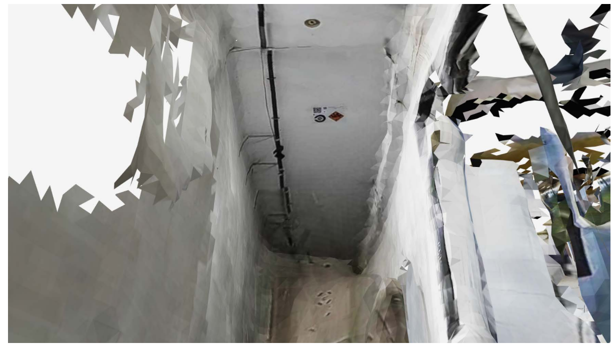


Figure 4.14: Real-Time detection and localization of a NIST ring target

Figure 4.15: The final post-processed map generated during the NIST's simulated indoor mission

glimpse into the post-processed map generated from a successful flight of the Intrigue UAS within this building. Notably, the structure of the building comprises a series of cluttered interconnected rooms along with a narrower corridor featuring considerably fewer distinctive features. During the flight, the UAS demonstrated its robustness by executing successful loop closures, a vital process that helped maintain the integrity of the map structure by rectifying Visual-Inertial Odometry

(a) A person hidden behind the couch



(b) A NIST ring target on the ceiling

Figure 4.16: Interesting viewpoints in the post-processed mesh map.

(VIO) drift errors. This ensured the accuracy and reliability of the mapped environment, even in complex and cluttered indoor settings. A closer examination of the map, as depicted in Figure 4.18, reveals the textural details that enhance the fidelity of the reconstruction. Noteworthy features such as objects positioned on desks, labels on trash cans, and even writing on a whiteboard are discernible within the map. This level of detail serves as a testament to the efficacy of both the custom-developed low-cost multispectral mapping sensor system and the associated mapping framework deployed on the Intrigue UAS. The successful capture and representation of such intricate details underscore the system's capabilities in accurately mapping indoor environments with diverse textures and objects. This achievement is particularly noteworthy given the challenges posed by cluttered spaces and varying lighting conditions often encountered in indoor settings. A combined recorded video of the GCS laptop screen and the FPV video feed for this controlled indoor mapping experiment can be found at [83]. Overall, the performance of the Intrigue UAS within the UMD building highlights its potential for a wide range of applications, including indoor surveillance, mapping, and reconnaissance tasks. Its ability to navigate and map complex indoor environments while preserving detailed textural information positions it as a valuable tool for various domains, including urban planning, facility management, and emergency response.

Figure 4.17: Detailed post-processed map of a set of cluttered interconnected rooms and a narrow corridor of a building at UMD.

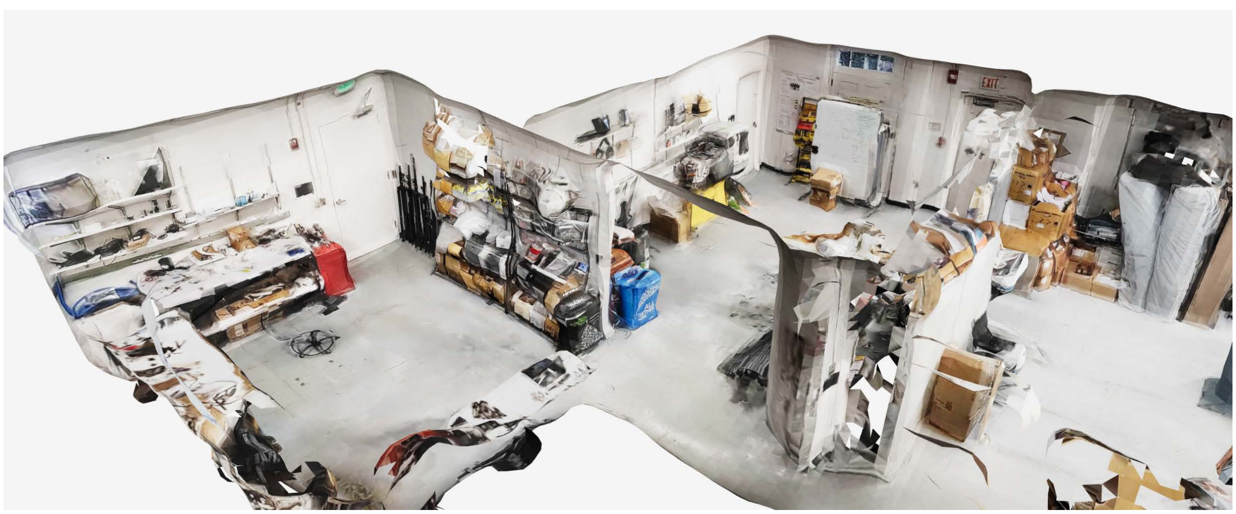


Figure 4.18: Another view of the map showing the quality of textural details captured by the custom developed low-cost multispectral mapping sensor system.

Chapter 5: Conclusion

This dissertation detailed three key research ventures aimed at improving UAS capabilities: enhancing quadrotor control performance through in-flight calibration, developing systems for wind rejection, and advancing indoor aerial 3D mapping. The first venture focused on improving the control performance of quadrotors by employing in-flight calibration techniques, ensuring more precise and stable flight dynamics. The second venture developed robust systems for wind rejection, allowing UAS to maintain performance and stability even in adverse wind conditions. The third venture advanced indoor aerial 3D mapping, enabling UAS to create detailed and accurate maps of indoor environments, which is crucial for effective navigation and situational awareness. These endeavors contribute to the overall goal of maximizing the efficacy of UAS in rescue operations, thereby enhancing the safety of human responders and increasing the success rate of missions in compromised indoor environments. The resilience, reliability, and performance of UAS can be further enhanced through ongoing research and development efforts. The dissertation research integrated interdisciplinary concepts and techniques from Aerospace Engineering, Computer Science, Robotics, and Estimation and Control Theory, addressing important challenges in aerial robotics. The integration of these diverse fields facilitated the development of sophisticated solutions and advancements, with the findings published in peer-reviewed journals, underscoring the significance and impact of the research. Chapter 2 described a UAV self-calibration framework that estimates vehicle state, external wind, drag coefficients, center of pressure, and sensor bias using IMU and ground velocity measurements, with optimized calibration trajectories enhancing tracking performance and estimation accuracy, particularly in unsteady wind conditions. Chapter 3 presented the development and experimental evaluation of a UAV system identification framework, combining rotorcraft, robotics, and nonlinear control tools to enhance parameter estimation accuracy and wind gust rejection, with improved tracking performance and computational efficiency using a custom, low-cost flight controller. Chapter 4 detailed the design and implementation of a cost-effective UAS for first responders, featuring advanced 3D mapping and target detection capabilities, validated by success and multiple awards in the NIST 2023 First Responder UAS 3D Mapping Challenge, demonstrating significant improvements in data acquisition speed and system stability.

5.1 Summary of Contributions

This section offers a succinct overview of the primary outcomes of this dissertation. Firstly, it provides a condensed summary of the findings detailed in Chapter 2, elucidating the contributions and results. Secondly, it distills the essence of Chapter 3, highlighting the improvements made over conventional methods. Lastly, key insights from Chapter 4 are encapsulated, laying a robust groundwork for development of advanced, reliable, real-time 3D mapping, and autonomy capabilities of UAS.

Estimation for Improved Control Performance via In-flight Calibration

The research showcased in Chapter 2 described a UAV self-calibration framework that estimates the vehicle state, external wind, drag coefficients, center of pressure, and sensor bias using IMU and ground velocity measurements, such as obtained via visual-inertial odometry or lidar-based odometry. The observability of the system is also analyzed and trajectory optimization performed to maximize the observability with feasible control inputs during calibration. The improvements in tracking performance due to the estimator are evaluated in simulation using a model-based controller. The self-calibration trajectory provided by the optimization compare favorably with a manually selected calibration trajectory, demonstrating the improvements in tracking and estimation performance. Optimizing the calibration trajectories makes the estimation framework more numerically stable; the parameters are calibrated faster and more accurately. Post-calibration simulation demonstrates the framework's ability to estimate unsteady winds without directly sensing the wind.

Experimental Implementation of System Identification for Wind Rejection

The study detailed within Chapter 3 presented the development and experimental evaluation of a combined approach toward system identification of a UAV by synthesizing tools from rotorcraft, robotics, and nonlinear control and estimation. The frequency-domain linear system identification tool CIFER® was used to accurately estimate the inertial parameters from an automated frequency sweep data. By sensing the motor RPM, accurate control values were obtained. Comparison with the traditional approach of using controller-generated commands revealed improvements in identification accuracy. A reduced version of the square-root unscented Kalman filter and a nonlinear model-based controller were utilized to recursively estimate the drag coefficient and external wind in separate experiments. The computational requirements of the nonlinear estimation and control frameworks were satisfied by designing a custom, low-cost flight controller. System identification accuracy of both linear and nonlinear techniques is evaluated by comparing parameter estimates with true value beliefs obtained from various sources. Experimental evaluation of the drag coefficient estimation showed rapid improvement in the controller's tracking performance. Additionally, wind estimation experiments demonstrated the framework's ability to reject wind gusts without directly sensing the wind.

Indoor Aerial 3D Mapping with a custom built visual-inertial sensor system

The investigation outlined in Chapter 4 detailed the design and implementation of a robust and cost-effective Unmanned Aerial System (UAS) specifically engineered for first responder use in search and rescue operations. The innovative 3D mapping, target detection, and localization framework, combined with a distributed computing approach, has demonstrated significant improvements in data acquisition speed and system stability. The system's performance was validated through its success in the National Institute of Standards and Technology (NIST) 2023 First Responder UAS 3D Mapping Challenge, where it earned notable accolades, including Third Place Overall and multiple Best-in-Class awards. These achievements, along with independent testing results, underscore the UAS's reliability and effectiveness in real-world scenarios.

5.2 Suggestions for Ongoing and Future Work

This section offers some forward-looking suggestions for future research trajectories.

Continuing the work in Chapter 2, one can seek to address the inaccuracies in the estimation of the z-component of certain parameters. This might involve refining the estimation framework to include additional parameters such as mass, inertia, and the center of gravity, which are critical for achieving more accurate and reliable performance of UAS. Additionally, exploring non-probabilistic approaches may offer increased numerical stability and ensure convergence, thereby enhancing the robustness of the estimation process.

Work done in Chapter 3 can be extended by demonstrating the recursive, in-flight estimation of additional model parameters. By implementing the frequency-domain system identification procedure directly on the onboard computer, it might be possible to achieve robust, accurate, and periodic updates of inertial parameters, ensuring the UAS remains calibrated and responsive to changing conditions. Further improvements in wind estimation and gust rejection performance can be pursued through several means. Upgrading the Inertial Measurement Unit (IMU) will provide more precise measurements, while reducing sensor delay will enhance the responsiveness of the system. Additionally, incorporating control delay into the estimation framework will result in a more accurate representation of the system's dynamics, leading to better control and stability in windy conditions.

Proceeding with the work in Chapter 4 research is ongoing focusing on expanding the system's autonomous capabilities, including autonomus navigation and exploration. These enhancements are aimed at providing even greater support for first responders in critical situations, enabling UAS to operate independently in complex environments and deliver vital information for rescue operations. By improving autonomous navigation, UAS can more effectively map and explore damaged indoor environments, identifying hazards and locating survivors without putting human responders at risk. Furthermore, bringing the framework in outdoors environments such as under forested canopy, over water bodies and in windy environments will be the ultimate test of the UAS's capabilities.

Overall, these efforts should be directed towards making UAS more reliable, resilient, and effective tools for rescue operations. The integration of advanced estimation techniques, improved hardware, and sophisticated control algorithms will significantly enhance the performance of UAS, ensuring they can operate effectively in the most challenging conditions. The continued research and development in this field hold promise for even greater advancements in aerial robotics, ultimately contributing to safer and more efficient rescue missions.

Bibliography

[1] W. Wei, M. Tischler, and K. Cohen, "System identification and controller optimization of a quadrotor Unmanned Aerial Vehicle in hover," *Journal of the American Helicopter Society*, vol. 62, 10 2017.

[2] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "IMU-based inertia estimation for a quadrotor using newton-euler dynamics," *IEEE Robotics and Automation Letters*, pp. 1–1, 2020.

[3] N. Abas, A. Legowo, and R. Akmeliawati, "Parameter identification of an autonomous quadrotor," in *4th International Conference on Mechatronics*, 2011, pp. 1–8.

[4] V. Khorani, N. Ajilforoushan, and A. M. Shahri, "An IMU-based system identification technique for quadrotors," in *3rd Joint Conference of AI Robotics and 5th RoboCup Iran Open International Symposium*, 2013, pp. 1–6.

[5] J. Yang, Z. Cai, Q. Lin, D. Zhang, and Y. Wang, "System identification of quadrotor UAV based on genetic algorithm," in *IEEE Chinese Guidance, Navigation and Control Conference*, 2014, pp. 2336–2340.

[6] D. Ho, J. Linder, G. Hendeby, and M. Enqvist, "Mass estimation of a quadcopter using IMU data," in *International Conference on Unmanned Aircraft Systems*, 2017, pp. 1260–1266.

[7] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to UAVs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1770–1777, 2017.

[8] B. Michini and J. How, "L1 adaptive control for indoor autonomous vehicles: design process and flight testing," in *AIAA Guidance, Navigation, and Control Conference*, 2009, pp. 5754–5768.

[9] L.-C. Fu and S. Sastry, "Slow-drift instability in model reference adaptive systems—an averaging analysis," *International Journal of Control*, vol. 45, no. 2, pp. 503–527, 1987.

[10] S. Jafari, P. A. Ioannou, and L. Rudd, "What is L1 adaptive control," in *AIAA Guidance, Navigation, and Control Conference*, 2013, p. 4513.

[11] H. Elliott and W. Wolovich, "A parameter adaptive control structure for linear multivariable systems," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 340–352, 1982.

[12] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. USA: Dover Publications, Inc., 2009, ch. 6.

[13] J. Guo, G. Tao, and Y. Liu, "A multivariable MRAC scheme with application to a nonlinear aircraft model," *Automatica*, vol. 47, no. 4, pp. 804–812, 2011.

[14] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated vtolvehicles: A review of basic control design ideas and principles," *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 61–75, 2013.

[15] T. Koo and S. Sastry, "Output tracking control design of a helicopter model based on approximate linearization," *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 4, pp. 3635–3640 vol.4, 1998.

[16] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, "A survey of open-source UAV flight controllers and flight simulators," *Microprocessors and Microsystems*, vol. 61, pp. 11–20, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141933118300930

[17] D. Enns, D. Bugajski, R. Hendrick, and G. Stein, "Dynamic inversion: An evolving methodology for flight control design," *International Journal of Control*, vol. 59, no. 1, pp. 71–91, 1994.

[18] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, 2009.

[19] N. Hovakimyan and C. Cao, *L1 adaptive control theory: Guaranteed robustness with fast adaptation*. Society for Industrial and Applied Mathematics, 2010.

[20] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conference on Decision and Control*, 2010, pp. 5420–5425.

[21] Meda-Campaña and A. Jesús, "On the estimation and control of nonlinear systems with parametric uncertainties and noisy outputs," *IEEE Access*, vol. 6, pp. 31 968–31 973, 2018.

[22] Jardin, M. R, and E. R. Mueller, "Optimized measurements of unmanned-air-vehicle mass moment of inertia with a bifilar pendulum," *Journal of Aircraft*, vol. 46, no. 3, pp. 763–775, 2009.

[23] N. Anton, R. M. Botez, and D. Popescu, "Stability derivatives for a delta-wing X-31 aircraft validated using wind tunnel test data," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 4, pp. 403–416, 2011.

[24] W. Wei, K. Cohen, and M. B. Tischler, "System identification and controller optimization of a quadrotor UAV," *Proceedings of the AHS International's 71st Annual Forum and Technology Display*, 2015.

[25] J. F. Horn, "Aircraft and rotorcraft system identification: engineering methods with flight test examples, second edition," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 117–119, 2016.

[26] J. W. Harding, S. J. Moody, G. J. Jeram, H. Mansur, and M. B. Tischler, "Development of modern control laws for the AH-64D in hover/low speed flight," *Annual Forum Proceedings-American Helicopter Society*, vol. 62, no. 3, 2006.

[27] M. Tischler, "System identification methods for aircraft flight control development and validation," in *Advances in Aircraft Flight Control*. Taylor & Francis, 10 1995, pp. 35–69.

[28] W. Craig, J. Lewis, D. A. Paley, and W. H. Young Jr, "Stabilization of a quadrotor in wind with flow sensing: Linear modeling and control for attitude and position hold," *VFS Autonomous VTOL Technical Meeting and eVTOL Symposium*, pp. 1–10, 2019.

[29] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "IMU-based inertia estimation for a quadrotor using Newton-Euler dynamics," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.

[30] A. K. Shastry, M. T. Bhargavapuri, M. Kothari, and S. R. Sahoo, "Quaternion based adaptive control for package delivery using variable-pitch quadrotors," in *2018 Indian control conference (ICC)*. IEEE, 2018, pp. 340–345.

[31] M. Bhargavapuri, A. K. Shastry, H. Sinha, S. R. Sahoo, and M. Kothari, "Vision-based autonomous tracking and landing of a fully-actuated rotorcraft," *Control Engineering Practice*, vol. 89, pp. 113–129, 2019.

[32] E. H. Heim, E. Viken, J. M. Brandon, and M. Croom, "NASA's learn-to-fly project overview," *2018 Atmospheric Flight Mechanics Conference*, 2018.

[33] T. Lombaerts, E. R. van Oort, Q. Chu, J. A. Mulder, and D. Joosten, "Online aerodynamic model structure selection and parameter estimation for fault tolerant control," *Journal of Guidance Control and Dynamics*, vol. 33, pp. 707–723, 2010.

[34] NIST, "NIST First Responder UAS Challenge Home," https://firstresponderuaschallenge.org/uas5/, 2023, online; accessed May 2024.

[35] S. Karam, F. Nex, B. T. Chidura, and N. Kerle, "Microdrone-based indoor mapping with graph SLAM," *Drones*, vol. 6, no. 11, 2022.

[36] E. Caballero, A. Madridano, D. Sainidis, K. Konstantoudakis, P. Daras, and P. Flores, "An automated UAV-assisted 2D mapping system for First Responders," in *18th International Conference on Information Systems for Crisis Response and Management. Virginia Tech, Blacksburg, VA*, 2021, pp. 890–902.

[37] R. Otero, S. Lagüela, I. Garrido, and P. Arias, "Mobile indoor mapping technologies: A review," *Automation in Construction*, vol. 120, p. 103399, 2020.

[38] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, 2023.

[39] C. Kolhatkar and K. Wagle, "Review of SLAM algorithms for indoor mobile robot with LIDAR and RGB-D camera technology," in *Innovations in Electrical and Electronic Engineering*. Springer Singapore, 2021, pp. 397–409.

[40] G. A. Kumar, A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai, "A LiDAR and IMU integrated indoor navigation system for UAVs and Its Application in Real-Time Pipeline Classification," *Sensors*, vol. 17, no. 6, 2017.

[41] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "LIDAR-inertial integration for UAV localization and mapping in complex environments," in *2016 International Conference on Unmanned Aircraft Systems*, 2016, pp. 649–656.

[42] T. H. Chan, H. Hesse, and S. G. Ho, "LiDAR-based 3D SLAM for indoor mapping," in *2021 7th International Conference on Control, Automation and Robotics*, 2021, pp. 285–289.

[43] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 4758–4765.

[44] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5135–5142.

[45] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *Journal of Field Robotics*, vol. 35, no. 4, pp. 578–595, 2018.

[46] H. Hirschmüller, "Semi-global matching-motivation, developments and applications," *Photogrammetric Week 11*, pp. 173–184, 2011.

[47] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[48] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "maplab 2.0 – A Modular and Multi-Modal Mapping Framework," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 520–527, 2023.

[49] A. Steenbeek and F. Nex, "CNN-based dense monocular visual SLAM for real-time UAV exploration in emergency conditions," *Drones*, vol. 6, no. 3, p. 79, 2022.

[50] P. Chhikara, R. Tekchandani, N. Kumar, V. Chamola, and M. Guizani, "DCNN-GA: A deep neural net architecture for navigation of UAV in indoor environment," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4448–4460, 2020.

[51] M. Alhafnawi, H. A. Bany Salameh, A. Masadeh, H. Al-Obiedollah, M. Ayyash, R. El-Khazali, and H. Elgala, "A Survey of indoor and outdoor UAV-based target tracking systems: Current status, challenges, technologies, and future directions," *IEEE Access*, vol. 11, pp. 68 324–68 339, 2023.

[52] W. Cui, A. Shastry, D. A. Paley, and S. Nogar, "Autonomous Aerial Search and Revisit Behavior for Communication Limited Environments," in *AIAA SCITECH 2024 Forum*, 2024, p. 1170.

[53] X. Wang, J. Liu, and Q. Zhou, "Real-time multi-target localization from Unmanned Aerial Vehicles," *Sensors*, vol. 17, no. 1, 2017.

[54] H. U. Unlu, D. Chaikalis, A. Tsoukalas, and A. Tzes, "UAV indoor exploration for fire-target detection and extinguishing," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, p. 54, 2023.

[55] A. Shastry and D. A. Paley, "UAV state and parameter estimation in wind using calibration trajectories optimized for observability," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1801–1806, 2021.

[56] A. K. Shastry and D. A. Paley, "System identification for high-performance UAV control in wind," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 17, pp. 10 451–10 467, 2023.

[57] A. K. Shastry, W. Cui, S. Abdi, S. Poojari, A. Ahmed, Q. Wei, A. Luterman, V. Ved, and D. A. Paley, "Indoor aerial 3D mapping and target localization with a custom-built multi-spectral visual-inertial sensor system," in *Submitted to AIAA SCITECH 2025 Forum*, 2025.

[58] N. A. Johnson and W. Singhose, "Dynamics and modeling of a quadrotor with a suspended payload," in *Applied Aerodynamics Conference*, 2018.

[59] S. Allison, H. Bai, and B. Jayaraman, "Wind estimation using quadcopter motion: A machine learning approach," *Aerospace Science and Technology*, vol. 98, p. 105699, 2020.

[60] D. Scaramuzza and Z. Zhang, "Visual-Inertial Odometry of Aerial Robots," *arXiv:1906.03289 [cs.RO]*, 2019.

[61] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014.

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

[63] N. D. Powel and K. A. Morgansen, "Empirical observability Gramian rank condition for weak observability of nonlinear systems with control," in *54th IEEE Conference on Decision and Control*, 2015, pp. 6342–6348.

[64] A. J. Krener and K. Ide, "Measures of unobservability," in *48h IEEE Conference on Decision and Control held jointly with 28th Chinese Control Conference*, 2009, pp. 6401–6406.

[65] A. Shastry, "Teensy4.0-flight-controller," https://github.com/animeshshastry/Teensy4.0-Flight-Controller, 2022, online; accessed May 2024.

[66] A. Shastry, "quadx_software," https://github.com/animeshshastry/quadx_software, 2022, online; accessed May 2024.

[67] M. Bangura and R. E. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," *Australasian Conference on Robotics and Automation*, 2012.

[68] Modal AI, "Image sensor 4k high-resolution, low-light sensor for VOXL (Starvis IMX412 w/ M12-style Lens)," https://www.modalai.com/products/mdk-m0061-1, 2023, online; accessed Apr 2024.

[69] ModalAI, "VOXL Time of Flight (ToF) depth sensor," https://www.modalai.com/products/voxl-dk-tof, 2011, online; accessed Apr 2024.

[70] UMD AMAV Team, "AMAV Services," https://github.com/UMD-AMAV/amav_services_nist5, 2024, online; accessed Apr 2024.

[71] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," https://github.com/ultralytics/ultralytics, 2023, online; accessed Apr 2024.

[72] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European Conference on Computer Vision*. Springer, 2022, pp. 1–21.

[73] ALFA Network, "HaLow-U," https://www.alfa.com.tw/products/halow-u?variant=39467228758088, 2020, online; accessed May 2024.

[74] Microhard, "pDDL900 - Dual Frequency OEM Ethernet & Serial Digital Data Link," https://www.microhardcorp.com/pDDL900.php, 2020, online; accessed Apr 2024.

[75] Interdisciplinary Robot Lab, Université de Sherbrooke, "RTAB-Map," https://github.com/introlab/rtabmap, 2011, online; accessed Apr 2024.

[76] UMD AMAV Team and Modal AI, "AMAV Portal," https://github.com/UMD-AMAV/amav-portal, 2024, online; accessed Apr 2024.

[77] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. D. Lillo, and Y. Lanthony, "AliceVision Meshroom: An open-source 3D reconstruction pipeline," in *Proceedings of the 12th ACM Multimedia Systems Conference - MMSys '21*. ACM Press, 2021.

[78] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A Modular Framework for Neural Radiance Field Development," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23, 2023.

[79] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[80] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise View Selection for Unstructured Multi-View Stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[81] D. Holz and S. Behnke, “Fast range image segmentation and smoothing using approximate surface reconstruction and region growing,” in *Intelligent Autonomous Systems 12: Volume 2 Proceedings of the 12th International Conference*. Springer, 2013, pp. 61–73.

[82] A. Shastry, “Real-time mapping NIST indoor simulated mission,” https://youtu.be/LKVjx_fZacU, 2024, online; accessed May 2024.

[83] A. Shastry, “Real-time mapping at UMD engineering annex,” https://youtu.be/6Ib0U-VxAXU, 2024, online; accessed May 2024.