

Autonomous Sensing and Navigation in Challenging Environments Using Unmanned Air Vehicles in Single- and Multi-Agent Settings

by

Andrew J. Torgesen

B.S., Mechanical Engineering, Brigham Young University (2019)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aerospace Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

© Massachusetts Institute of Technology 2021. All rights reserved.

Author
Department of Aeronautics and Astronautics
May 18, 2021

Certified by
Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics, MIT
Thesis Supervisor

Accepted by
Zoltan Spakovszky
Professor, Aeronautics and Astronautics
Chair, Graduate Program Committee

Autonomous Sensing and Navigation in Challenging Environments Using Unmanned Air Vehicles in Single- and Multi-Agent Settings

by

Andrew J. Torgesen

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Aerospace Engineering

Abstract

Small unmanned air systems (UAS), due to their navigational versatility and ability to operate autonomously, serve as an intriguing platform on which to carry out advanced sensing operations in otherwise untraversable or prohibitively dangerous environments. The need for UAS to be able to autonomously navigate and explore their environments with limited payload, communication, and computational capacity, however, poses its own challenges—particularly when subjected to the non-ideal environmental disturbances and feature spaces present in real-world scenarios. This thesis addresses these issues by presenting two complementary projects enabling UAS-based autonomous sensing in real-world environments using relatively low-cost and lightweight hardware. The first project presents a UAS capable of measuring air wakes while flying tethered behind a moving vessel. The unique challenges of tethered flight control and relative state estimation in a feature-starved environment are addressed with a novel planning and control architecture together with an error-state Kalman filter that achieves centimeter-level relative position accuracy. The second project presents a multi-agent UAS navigation system for GPS-denied environments that expands on the state-of-the-art in collaborative simultaneous localization and mapping (CSLAM) for the purpose of facilitating fast and accurate radiation mapping in contaminated and cluttered zones. CSLAM capabilities are made more robust to communication deficiencies through the novel incorporation of ultra-wideband range sensors into a distributed range-enhanced pose graph optimization (DRPGO) scheme. The experimental demonstrations of the two presented systems, considered in tandem to overcome hurdles to sensing from aerodynamic disturbances, feature-starved environments, and communication bandwidth limitations, strengthen the promise of small UAS as an effective tool for demanding real-world data collection applications.

Thesis Supervisor: Jonathan P. How

Title: R. C. Maclaurin Professor of Aeronautics and Astronautics, MIT

Acknowledgments

First and foremost, I wish to thank my advisor, Jon How, for giving me the opportunity to learn and conduct research at this world-class institution. Your commitment to regular guidance-filled meetings and frequent suggestions of papers to look into at any hour of the day in spite of a remarkably busy schedule were all indicative of genuine care about the success of the research as well as my own success, and this fact has not been lost on me. I would also like to thank Ben Cameron, Richard Kaszeta, and John Walthour over at Creare, LLC. for their time, professionalism, and expertise on the air wake measurement system project.

My time at MIT was rendered much more fruitful and enjoyable thanks to the support and comradeship with the other members of the Aerospace Controls Lab. In particular, I want to thank Parker Lusk, Andrea Tagliabue, Jeremy Cai, and Jesus Tordesillas for their varying degrees of both mentorship and friendship with me as a new graduate student trying to absorb and contribute as much as possible in a short two years. Additionally, Yulun Tian, Kasra Khosoussi, and Kaveh Fathian provided continuous and invaluable expertise from their active research in multi-agent distributed pose graph optimization and loop closure detection.

Andrew Fishberg, who joined the lab during my final semester, hit the ground running in helping me investigate UWB hardware. He also went out of his way to be a great help with air wake hardware testing, even when it wasn't directly related to his research. His assistance helped me finish on a good note.

The one to whom I undoubtedly owe the greatest debt of gratitude is my wonderful wife Kathleen, who offered unconditional love and support through the birth of our first child, the isolation of a global pandemic, and many instances of seemingly boundless stress. Her bravery and resilience are a huge part of what made my studies and work possible.

The completion of this work has been enabled by multiple funding sources: Creare, LLC., the Small Business Technology Transfer (STTR), the Office of Naval Research, and the Department of Energy / National Nuclear Security Administration.

Contents

1	Introduction	13
1.1	Project Overview	13
1.1.1	Air Wake Measurement Motivation and Literature Review	14
1.1.2	DRPGO Motivation and Literature Review	15
1.2	Thesis Contributions	18
1.2.1	Air Wake Measurement System	18
1.2.2	DRPGO for Radiological Mapping	19
2	Theory and Algorithms for Air Wake Measurement	21
2.1	Problem Formulation	21
2.2	System Overview	22
2.3	Tethered Flight Strategy	23
2.4	Absolute and Relative State Estimation with GPS, Infrared Beacons, and a Camera	27
2.5	Error-State Kalman Filter Equations Derivation	32
2.5.1	ESKF Background and Quaternion Conventions	32
2.5.2	Absolute Error-State Dynamics	33
2.5.3	Absolute Measurement Equations	36
2.5.4	Relative Error-State Dynamics	38
2.5.5	Relative Measurement Equations	40
2.6	Summary	42

3 Simulation and Hardware for Air Wake Measurement	43
3.1 Simulated Maritime Environment	43
3.1.1 UAS Software-in-the-Loop Autopilot and Dynamics	44
3.1.2 Aerodynamic Turbulence Effects	45
3.1.3 Naval Vessel and Tether Contact Dynamics	46
3.1.4 Simulated Sensor Suite	47
3.2 System Hardware	47
3.3 Summary	49
4 Theory and Implementation for DRPGO-Enabled Radiological Mapping	51
4.1 System Architecture	51
4.2 Distributed Pose Graph Optimization for Radiological Mapping	53
4.3 Distributed Range-Enhanced Pose Graph Optimization (DRPGO)	56
4.4 DRPGO-CSLAM Dataset Generation	59
4.5 Simulation for DRPGO Prototyping	62
4.6 DRPGO Solver Implementation	63
4.7 Summary	64
5 Testing and Results	65
5.1 Air Wake Measurement Tests	65
5.1.1 Simulation Tests	65
5.1.2 Indoor Flight Tests	72
5.1.3 Outdoor Flight Tests	77
5.2 Distributed Range-Enhanced Pose Graph Optimization Tests	82
5.2.1 DRPGO-CSLAM Dataset Tests	82
5.2.2 DRPGO Simulation Tests	88
6 Conclusion	95
6.1 Summary	95
6.2 Future Work	96

List of Figures

2-1	Problem formulation for the ship air wake measurement task.	22
2-2	Air wake project system architecture.	23
2-3	Hardware components for the ship air wake measurement UAS. . . .	24
2-4	Outer-loop controller commands.	26
2-5	IR beacon relative pose estimation.	31
3-1	Air wake measurement system simulation environment.	44
3-2	Air wake indoor flight testing platform.	47
3-3	Air wake outdoor flight testing platform.	49
4-1	Possible multi-UAS architectures for radiological mapping.	52
4-2	Dynamic Bayes net representation.	54
4-3	Illustration of the \boxplus operator.	55
4-4	AirSim Simulator.	61
4-5	Flightmare simulator.	61
4-6	PGO dataset generation process.	62
5-1	Air wake yaw estimate results in simulation.	66
5-2	Air wake y-position estimate simulation results.	67
5-3	UAS pitch tracking in simulation.	68
5-4	UAS lateral velocity tracking in simulation.	69
5-5	Sweep trajectory in simulation.	70
5-6	UAS altitude tracking in simulation.	71
5-7	External force observer simulation results.	71

5-8 Indoor flight tests setup.	73
5-9 Air wake y-position estimate hardware results.	75
5-10 UAS pitch tracking in hardware.	75
5-11 UAS external force observer results in hardware.	76
5-12 Sweep trajectory during indoor flight.	77
5-13 Outdoor flight tests setup.	78
5-14 Air wake outdoor flight tests.	80
5-15 Sweep trajectory for outdoor flight #1.	81
5-16 Sweep trajectory for outdoor flight # 2.	81
5-17 Sweep trajectory for outdoor flight # 3.	82
5-18 VIO-only RMSE on the EuRoC MAV dataset.	83
5-19 VIO+altimeter RMSE on the EuRoC MAV dataset.	84
5-20 VIO+altimeter+range RMSE on the EuRoC MAV dataset.	85
5-21 VIO+alt+LC RMSE on the EuRoC MAV dataset.	85
5-22 VIO+alt+LC+range RMSE on the EuRoC MAV dataset.	86
5-23 VIO-only RMSE on the feature-starved dataset.	87
5-24 VIO+alt+range RMSE on the feature-starved dataset.	87
5-25 VIO+alt+range (with anchoring) RMSE on the feature-starved dataset.	88
5-26 DRPGO RMSE with no connectivity.	89
5-27 DRPGO RMSE with no connectivity (communication visualization).	90
5-28 DRPGO RMSE with medium connectivity.	91
5-29 DRPGO RMSE with medium connectivity (communication visualization).	91
5-30 DRPGO RMSE with full connectivity.	92
5-31 DRPGO RMSE with full connectivity (communication visualization).	93
5-32 DRPGO RMSE with full connectivity and no anchor.	93
5-33 DRPGO RMSE with full connectivity and no anchor (communication visualization).	94

List of Tables

2.1	Comparison of quaternion representation conventions for two of the most commonly used conventions in robotics and aerospace [49] and the conventions used in this work.	32
3.1	All sensor hardware used for both indoor and outdoor flight testing of the ship air wake measurement system.	48
3.2	All computer hardware used for both indoor and outdoor flight testing of the ship air wake measurement system.	48
5.1	Physical parameters for the presented air wake measurement system indoor hardware flight tests.	74

Chapter 1

Introduction

1.1 Project Overview

Small unmanned air systems (UAS), in single-agent or multi-agent settings, can be used in such disparate sensing applications as obtaining precise air wake velocity measurements behind a moving maritime vessel, or mapping a cluttered area bombarded with dangerous levels of gamma radiation. Such varied mission environments exhibit their own unique challenges, which generally fall into the categories of dynamic disturbances, sensor ineffectiveness, environmental clutter, time constraints, or computational and communication constraints. To be useful for real-world data collection, UAS must be able to autonomously navigate and explore their environments with limited payload, communication, and computational capacity.

This thesis presents UAS-based solutions to two representative problems in real-world sensing that account for and directly address these challenges to varying degrees. The first project is referred to as the *Air Wake Measurement System*, and is a UAS designed to autonomously measure air wakes while flying tethered behind a moving maritime vessel and maintaining an accurate relative state estimate between the UAS and the vessel. The second project is referred to as *Distributed Range-enhanced Pose Graph Optimization* (DRPGO), and its purpose is to expand the ability of a team of UAS to explore environments for facilitating radiological mapping in post-disaster zones too dangerous for human traversal. In such scenarios, which can include explo-

ration inside of possibly damaged buildings, GPS-based localization and unlimited communication range/bandwidth cannot be assumed, and thus require a distributed navigation solution that is compatible with an ad hoc networking paradigm.

For each project, the underlying theory and algorithms are presented, followed by the simulation and hardware tools used for implementation and testing. Simulation and hardware-based results are presented, validating each project’s approach to its respective environmental sensing problem.

1.1.1 Air Wake Measurement Motivation and Literature Review

One considerable challenge posed to the takeoff and landing of maritime aircraft on a moving ship stems from having to stabilize in the midst of a wake of turbulent airflow traversing the flight deck area. This region of turbulent air wake flow arises in the lee of the ship’s superstructure as it interacts with a relative headwind. Because of the inherent risk that the air wake poses to stability during close-proximity flight, ongoing efforts are being made to use Computational Fluid Dynamics (CFD) models to accurately characterize the air wake profile for different classes of navy vessels [65, 11, 61, 56] as well as to quantitatively determine the effect of air wakes on aircraft dynamics [38, 47, 46]. Despite their prevalence, these models based on CFD analysis require experimental validation through in-situ air wake velocity measurement.

Small unmanned air vehicles (UAV) for air wake measurement on naval vessels represent an attractive alternative to deck-mounted anemometers and scale-model wind tunnel testing due to their ability to operate in the vicinity of the ship while also capturing air wake effects behind the ship and over the water. These vehicles require relatively little setup and interference with existing naval vessel configurations which are already designed to accommodate aircraft.

Flight experiments with remotely piloted air vehicles in the wake of maritime vessels such as [34] have provided a proof of concept for the usage of such vehicles as air wake measurement devices.

In recent years, there have been several attempts to obtain experimental measurements of the spatially varying air wake velocity profile using small UAV for the purpose of validating CFD simulations. Phelps et al. [37] employ the use of a small RC plane with known aerodynamic coefficients and a small sensor suite consisting of an inertial measurement unit (IMU), GPS, and a GPS mounted on the ship. While their experiments demonstrate that air wake measurements could be obtained with an inexpensive platform, the inherent limitations in the sensor suite and ad-hoc nature of the RC flight control limit the resolution and breadth of their results. Mallon et al. [32] utilize a quadrotor platform and obtain a slightly improved resolution of measurements through the use of two anemometers onboard the UAV. However, they are only able to obtain measurements directly behind the ship, unable to traverse into the free-stream region of air flow due to the difficulty of maintaining stability with RC control. Gamagedara et al. [16] improve on this design by adding a camera to the ship deck, performing optical flow to improve the relative state estimation of the UAV. While the spatial association of air wake measurements is improved, the approach still suffers from the same stability issues with RC flight and the flight envelope was also limited to a very small region of air wake measurement. Kumar et al. [28] have provided the most spatially extensive set of air wake measurements with a small RC helicopter that uses an artificial neural network to associate measured angular rate disturbances on the vehicle with the turbulent component of air wake velocities. However, their platform requires a significant vehicle-specific calibration process to train their neural network and is unable to measure the steady-state velocity component of the air wake. Further, the employment of an artificial neural network, crucial to their air wake measurement process, represents a black box whose results are difficult to verify in the presence of unfamiliar vehicle flight behavior or confounding RC pilot inputs.

1.1.2 DRPGO Motivation and Literature Review

Remote radiological source localization and mapping is needed in first-response and disaster prevention scenarios in areas containing one or more radiation sources. UAS

provide ideal platforms for traversing arbitrary terrain when equipped with specialized, lightweight radiation sensors. Recent findings show that the resolution and mapping speed of radiological sensing on a mobile platform with radiation count sensors is greatly improved when fused with pose estimates and 3D map information obtained through simultaneous localization and mapping (SLAM) [35].

Beyond simulation-based studies in source term estimation for radiological search [22, 23], there have been a handful of hardware-based studies that achieve autonomous radiological search and mapping on a single-agent platform such as LAMP [4, 36, 35] or even in a supervised swarm setting [8]. While these methods impressively demonstrate radiological source term localization in the real world, the environments they traverse are either mapped out a priori or characterized by wide open spaces, mapped by a single agent equipped with a Velodyne LiDAR [59] sensor. While LiDAR sensing is accurate at long ranges in a wide variety of sensing scenarios, its cost and weight make the prospect of multiple airborne LiDAR mapping agents a difficult proposition. Moreover, single-agent LiDAR mapping is limited in its exploratory capabilities, especially in cluttered environments where long sensing range offers little advantage.

An alternative to single-agent LiDAR mapping is found in the CSLAM literature, in which multiple agents seek to collaboratively construct a globally consistent map by fusing together relative odometry measurements as well as intra- and inter-agent loop closure detections using pose graph optimization and bundle adjustment. This is typically done either in a centralized manner , in which a central server collects the necessary information from the agents to perform this fusion, or using a distributed approach, where agents share information among themselves, individually and incrementally fusing the information that is currently and locally available [67, 42].

The state-of-the-art in centralized collaborative SLAM from Schmuck et al. [44] has each mapping agent (or “client”) equipped with a camera and performing its own visual odometry for both local navigation and keyframe generation. Select keyframes are shared with a central server that performs pose graph optimization and bundle adjustment to construct a global map that is shared back with the clients. On separate

threads, the server also seeks loop closures within and between the individual agent trajectories in order to increase the global consistency and accuracy of the constructed map. Centralized methods sidestep the complexities of distributed PGO, outlier rejection, and loop closure detection, but face challenges of scalability to more than a handful of agents.

Distributed collaborative SLAM is very much an active area of research, with state-of-the-art formulations focusing on specific elements of SLAM such as distributed pose graph optimization [55] and distributed data association [17, 54] with provable convergence properties. These problems are shown to require significant inter-agent communication to ensure good performance, prompting other work to focus on methods for efficient object representations [64, 43] as well as lightweight data packaging and transfer within a decentralized CSLAM framework [7, 58, 13].

In the last few years, increased experimentation has been performed with ultra-wideband (UWB)-based relative localization in multi-agent settings. A UWB-based approach is attractive for scenarios such as formation flying because it requires less inter-agent communication for relative localization than CSLAM-based methods. For example, modern UWB ranging hardware has been proven to be effective for aiding dead-reckoning-based relative localization for swarm behavior, achieving accuracies from centimeters to the sub-meter level [9, 29, 19, 66].

Recently, UWB has also been incorporated into more sophisticated sensor fusion schemes with multiple sensing modalities to achieve impressive levels of relative localization accuracy. For example, Xu et al. fuse inter-agent UWB range measurements with visual inertial odometry (VIO) estimates and inter-agent detections with depth cameras to achieve centimeter-level relative localization between three agents in an indoor space [63]. However, though UWB-based range measurements can facilitate high-accuracy relative position and bearing estimates, unless an additional exteroceptive sensing modality is used to “anchor” the constructed global map to the world, the global map runs the risk of drifting, as observed by Ellingson et al. when they attempt to improve the absolute localization of several fixed-wing aircraft with relative range measurements [14].

There are a small number of works that address the UWB-based global map anchoring issues by incorporating exteroceptive sensing. Queralta et al. run experiments with a heterogeneous robot team, combining ground-based LiDARs with UWB-enabled UAS to map out an indoor space [40]. Wen et al. actually incorporate UWB range measurements into full-fledged CSLAM with GPS and loop closure detections for multiple cars driving through a city [62]. These methods, while effective in anchoring the agents' global pose estimates, rely on sensing modalities that are not always available in settings like radiological mapping or otherwise cumbersome to incorporate into a multi-agent setting with only small UAS.

1.2 Thesis Contributions

1.2.1 Air Wake Measurement System

We present a UAS for in-situ air wake measurement that is capable of providing high-resolution measurements of both turbulent and steady-state flow with a spatial breadth sufficient to characterize a significant portion of the ship air wake profile. This task is accomplished through a novel flight control configuration where a small UAS flies autonomously while attached to the end of a tether, resulting in stable flight trajectories capable of entering the free stream region of the ship air wake. State feedback control for the UAS is afforded through the use of a standard nonlinear state estimator. Precise relative state estimates between the ship deck and UAS are obtained via a separate relative state estimator that fuses differential GPS [3] and vision-based pose measurements from active infrared (IR) beacons mounted on the ship deck. The vision-based pose measurement system, in line with recent efforts to achieve reliable relative navigation using active features in mobile environments [26, 25], is one of the primary facilitators of an accurate and robust relative estimation scheme that functions in a variety of weather conditions and at a large range of distances from the ship deck.

Our contributions for the measurement system are concisely stated as follows:

- A novel tethered flight control strategy for autonomous flight capable of full space coverage of the air wake region behind a moving naval vessel at significant relative wind speeds.
- An accurate and robust relative state estimation scheme based on the sensor fusion of vision-based pose measurements with active IR beacons and differential GPS.
- Demonstration of the robustness of the proposed autonomous control and state estimation strategies to high wind levels and sensor noise in a high-fidelity simulation environment.
- Validation of the flight and estimation performance results obtained in simulation with indoor and outdoor flight testing.

1.2.2 DRPGO for Radiological Mapping

This thesis presents the groundwork and preliminary results for an enabling technology for the expansion of UAS-based, single-agent radiological search and mapping platforms to a multi-agent setting. Such an expansion entails a multi-agent 3D mapping capability that is robust to a large variety of environments, from wide outdoor spaces to cluttered indoor spaces, and that also promotes a high degree of exploration while also maintaining accuracy in time-sensitive scenarios. Moreover, the mapping strategy can benefit from a distributed architecture that does not assume full connectivity and unlimited communication bandwidth between all mapping agents.

To this end, we present a distributed pose graph optimization scheme that incorporates ultra-wideband (UWB)-based inter-agent range measurements. The incorporation of inter-agent range measurements in DRPGO, when combined with a simple heterogeneous task allocation strategy, facilitates a desirable trade-off in principle between swarm mapping exploration (for coverage versus time) and exploitation (for maintaining global accuracy), which is acknowledged to be a key challenge in the active SLAM literature [31]. The DRPGO method is meant to expand current state-

of-the-art capabilities in collaborative SLAM (CSLAM) by facilitating this desirable coverage behavior while also minimizing inter-agent communication requirements.

Our contributions for the DRPGO formulation are concisely stated as follows:

- Presentation of DRPGO, which incorporates UWB-based range measurements into a standard distributed pose graph optimization scheme and achieves global as well as relative consistency and accuracy.
- DRPGO experiments on created simulated and real-world CSLAM datasets.
- Demonstration, through simulation studies, of conditions under which global accuracy can be expected with DRPGO.

Chapter 2

Theory and Algorithms for Air Wake Measurement

2.1 Problem Formulation

The analysis given in this chapter is an expansion on previously published work [57]. Figure 2-1 shows the various reference frames used in the UAS control, estimation, and measurement subsystems of the air wake measurement system, as well as a simple illustration of the flight scheme. The UAS is tethered to the ship deck with a line with controllable length, maintaining the aircraft within a finite range of the moving vessel. The ship frame, denoted S , has its origin fixed to the ship deck. The inertial North-East-Down (NED) frame is denoted as I . The S and I frames differ in orientation principally by a yaw angle, assuming no significant heaving by the ship. The UAS body frame B is defined as having its origin fixed to the aircraft center-of-mass (COM) with the x-axis \mathbf{b}_x coming out the front of the aircraft, the y-axis \mathbf{b}_y pointing to the right, and the z-axis \mathbf{b}_z pointing down. Velocities, forces, and torques are generally expressed in B , whereas positions are either expressed in S or I . The UAS camera frame C is rigidly mounted to B with its z-axis \mathbf{c}_z pointed straight out of the camera lens, its x-axis \mathbf{c}_x pointed to the right, and its y-axis \mathbf{c}_y pointed down.

In this work, there are two types of states referenced: an absolute state \mathbf{x}^I and a relative state \mathbf{x}^S . While the exact components of the absolute and relative states

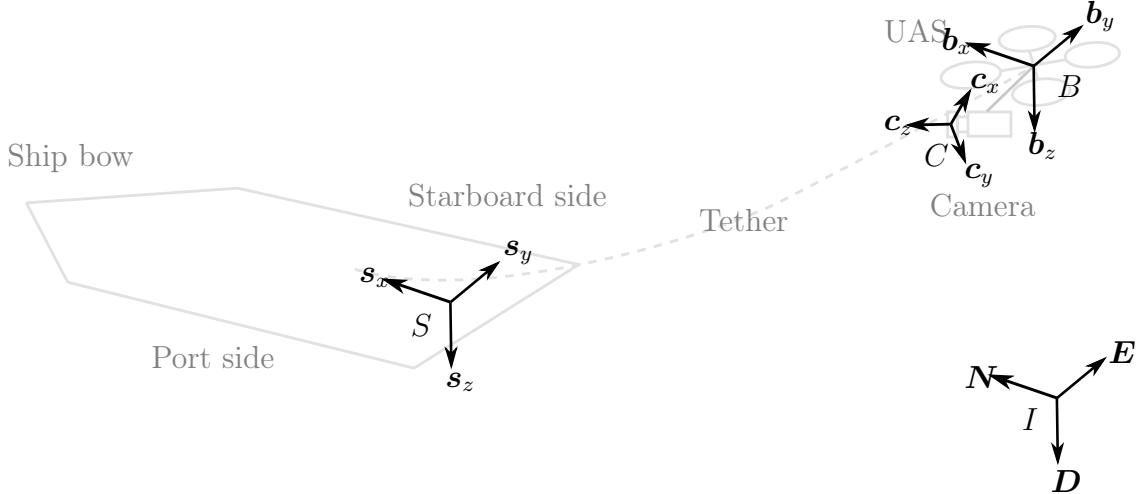


Figure 2-1: Problem formulation for the ship air wake measurement task. Principal frames and associated hardware components are marked for clarity.

in the context of filtering are explained in Section 2.4, the absolute state refers to the state of B relative to the inertial frame I and the relative state also refers to the state of B , but this time relative to S . The absolute state is used primarily for UAS flight control and the relative state is used primarily for giving spatial context to the obtained air wake velocity measurements.

2.2 System Overview

Figure 2-2 provides a high-level overview of the system architecture, including the major hardware and software components both onboard the ship deck and onboard the UAS. Communication between the ship and the UAS is facilitated with a set of WiFi antennas, and all algorithms pertaining to autonomous control and state estimation are implemented with the Robot Operating System (ROS) [51] and run on an ODROID XU-4 computer onboard the UAS for robustness in the event of a connection dropout. A tether reel controller box and user-operated ground station computer reside on the ship deck, providing the software and hardware links to command the aircraft from the ship. Sensors are present both on the ship as well as onboard the aircraft to aid in both inertial-frame and relative-frame navigation, as explained in

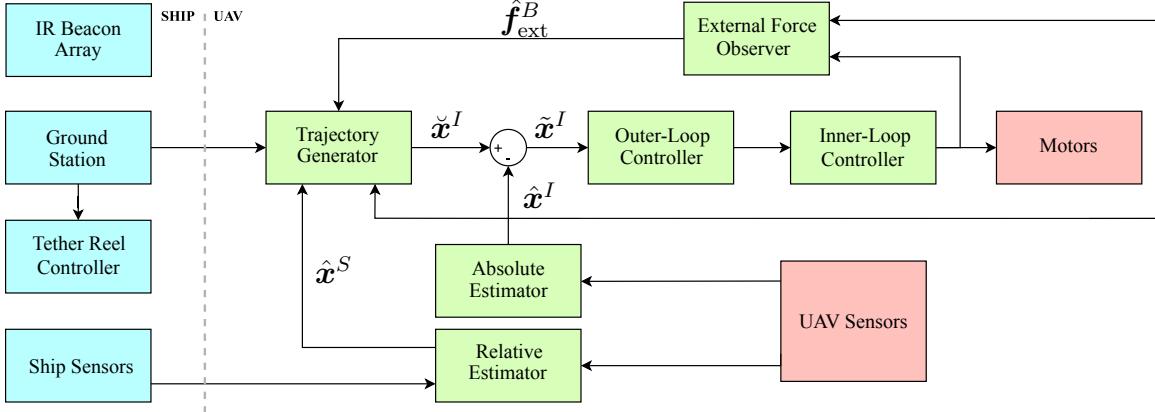


Figure 2-2: Autonomous flight control system architecture, demarcating systems present onboard the UAS during flight versus systems which remain on the ship deck. UAS plant components are colored red, UAS flight control components are colored green, and components mounted on the ship are colored blue.

detail in Section 2.4. Additional context is provided by Fig. 2-3a, which shows the UAS measurement platform, and Fig. 2-3b, which depicts both the tether reel control box and the IR beacon array in their intended placement on the deck.

The following sections detail the relevant design aspects for the control, estimation, and measurement subsystems, explaining how they have been tailored for the specific task of reliable high-resolution air wake measurement behind a moving naval vessel.

2.3 Tethered Flight Strategy

One of the distinguishing characteristics of this work is the fact that the UAS completes the entirety of its mission autonomously. The automated nature of the flight control and corresponding commanded flight trajectories adds an element of systematic regularity and repeatability to the experiment. Additionally, the autonomous flight control removes the need to account for idiosyncrasies of a remote pilot when assessing the level of bias in the measurements. Referencing Fig. 2-2, the autonomous flight software for the UAS consists of the ground station and tether reel controller on the ship and external force observer, trajectory generator, outer-loop controller, and inner-loop autopilot onboard the aircraft. The high-level functionality of each component can be summarized as:



(a) UAS platform with attached sensors. Onboard computer and flight control board are housed inside a protective water-proof case, and the camera is likewise shielded from environmental or water damage.

(b) UAS hardware present on ship deck during air wake measurement flights.

Figure 2-3: Hardware components for the ship air wake measurement UAS.

- The *ground station* allows a user on the ship to interface with the UAS remotely via a WiFi connection and send commands such as takeoff, land, maintain altitude, collect data, etc.
- The *tether reel controller* uses a tension sensor to alter the length of the tether as commanded by the ground station in such a way as to prevent the tether tension from varying widely and causing too much slack with feedback control.
- The *external force observer* uses the current absolute state estimate and motor commands to estimate the aggregate external force acting on the UAS in real time. From this estimate, the level of tether tautness can be deduced under nominal conditions, as explained below.
- The *trajectory generator* listens to commands from the ground station and to estimates of the current absolute state, relative state, and external force. It generates reference velocity and altitude commands for the outer-loop controller.
- The *outer-loop controller* uses proportional-integral-derivative (PID) control loops to generate reference attitude and yaw rate commands for the inner-loop autopilot.

- The *inner-loop autopilot* performs attitude control, also with PID, to generate motor PWM commands.

After takeoff, the UAS flies into the air wake flow region over the water and behind the ship, orienting itself so that its x-axis \mathbf{b}_x , coinciding with its front, is pointed directly at the origin of \mathcal{S} . Once it has reached this state, system enters the air wake measurement phase, with the air probe sampling air velocities at 70 Hz.

The interplay between the external force observer, trajectory generator, and outer-loop controller during the air wake measurement phase requires further discussion to shed some light on the unique aspects of the control strategy for the proposed system. During the air wake measurement phase, the trajectory generator proceeds to command velocities aligned with the y-axis \mathbf{b}_y of the UAS body frame at discrete altitudes so that the air probe can collect data at points on semi-circular arcs, as shown in Fig. 2-4.

By commanding various tether lengths and altitudes, a cylindrical grid of air wake measurements is amassed. Tracking body-frame lateral velocities with the outer-loop controller instead of position commands in the inertial frame simplifies the controller objective and avoids integrator wind-up issues that would arise from attempting to arrive at a point in inertial space rendered unreachable by the tether constraint. However, it is also necessary for the tether to be taut in order for lateral velocity commands to result in the desired semi-circular arc trajectories. Tether tautness has the additional desirable effect of driving the aircraft to be constantly yawed towards the tether anchor point, effectively passively stabilizing the longitudinal and yaw dynamics. For these reasons, the control strategy can be thought to work *with*, and not against, the tether constraint, taking advantage of the tether's passive stabilization effects.

Because tether tautness is so important to the effectiveness of the air wake measurement phase control strategy, the trajectory generator is also constantly monitoring the output of the external force observer during this phase to determine whether or not the tether is taut. Tether tautness is determined using the binary criterion of comparing the component of $\hat{\mathbf{f}}_{\text{ext}}^B$ aligned with \mathbf{b}_x against a certain threshold value,

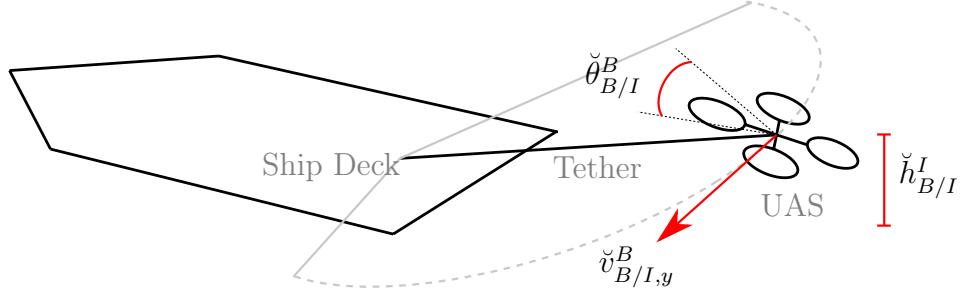


Figure 2-4: Illustration of commanded states passed to the outer-loop controller: UAS pitch angle ($\check{\theta}_{B/I}^B \geq 0$), altitude ($\check{h}_{B/I}^I$), and lateral velocity ($\check{v}_{B/I,y}^B$). With a taut tether, commands of this form result in semi-circular sweeping trajectories behind the moving ship.

$f_{\text{ext,thresh}}^B$. The observer is an unscented Kalman filter (UKF) proposed by [53] to deduce the generalized external forces acting on a small multi-rotor air vehicle given the motor speed commands ω_i and estimated state. The UKF model for translational dynamics is given by

$$\dot{\mathbf{v}}_{B/I}^B = \frac{1}{m} \left(\mathbf{f}_{\text{ext}}^B + \begin{bmatrix} 0 \\ 0 \\ f_{\text{thrust}}^B \end{bmatrix} + \mathbf{f}_{\text{aero}}^B \right) - \mathbf{R}(\mathbf{q}_I^B) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}. \quad (2.1)$$

The corresponding aerodynamic model

$$\mathbf{f}_{\text{aero}}^B = \mu \sum_{i=1}^N |\omega_i| \mathbf{v}_{B/I}^B, \quad (2.2)$$

which is a function of individual rotor speeds ω_i over N rotors and linear drag constant μ , allows the filter to distinguish between aerodynamic and non-aerodynamic (e.g., tether) forces assuming zero wind velocity. However, in an air wake measurement scenario where relative wind velocities are commonly in the vicinity of 6-7 m/s, the UKF filter output $\hat{\mathbf{f}}_{\text{ext}}^B$ invariably consists of forces imposed on the UAS by the tether as well as coupled aerodynamic drag effects.

The consequence of the inability of the UKF to distinguish between tether and aerodynamic forces is that an effort must be made to actively fly against the tether.

This is because during flight, the tether and aerodynamic forces from a relative head wind are never aligned and, in fact, are normally acting in opposing directions. Thus, actively pulling against the tether will result in a taut tether force far outweighing the magnitude of the aerodynamic forces, such that it will be easy to distinguish between a taut and slackened tether by monitoring $\hat{\mathbf{f}}_{\text{ext}}^B$. To pull against the tether, the trajectory generator commands a positive pitch angle, saturated at 15 degrees, throughout the air wake measurement phase. At that level of pitch, the tether force is easily distinguishable from aerodynamic forces under almost any wind condition. The only exception arises when there is a non-negligible tailwind that actually pushes the UAS back towards the ship. The significant tailwind scenario is rare, known mostly to happen when there is a strong recirculation effect close to the stern of the ship, and can be compensated for with an intervention from a safety pilot if necessary. When the trajectory generator detects a significant, sustained tether force indicative of tether tautness, then it allows the semi-circular sweeping trajectory to continue.

Note that, while both relative and absolute state estimates are available in real-time, outer-loop control is performed off of the absolute state only for reasons explained in Section 2.4. The relative state is leveraged in the trajectory generator as it is useful for operations such as takeoff, landing, and avoiding flying too close to the ship during the air wake measurement phase. Inner-loop attitude stabilization and control for the aircraft is handled by the onboard autopilot, which sits on a dedicated flight control board. The ROSflight autopilot [24] was chosen for this platform because of its lightweight and configurable code architecture, as well as its built-in compatibility with ROS.

2.4 Absolute and Relative State Estimation with GPS, Infrared Beacons, and a Camera

Two separate state estimators are run in parallel during flight: an absolute state estimator and a relative state estimator. The absolute state estimator is used for

robust UAS flight control and the relative state estimator is primarily for facilitating accurate spatial characterization of measured air wake velocities. In essence, the absolute estimator tracks the UAS pose with respect to the inertial frame, whereas the relative estimator tracks the UAS pose with respect to the moving ship frame. Both estimators estimate the UAS velocity with respect to the inertial frame, but the relative estimator also estimates the inertial velocity of the ship and passive rotation of the ship frame w.r.t. the inertial frame, which allows the estimated UAS inertial velocity to contribute to the dynamics of the relative UAS pose. Both estimators observe IMU and barometer biases. The absolute and relative states are:

$$\boldsymbol{x}^I \triangleq \left[\boldsymbol{p}_{B/I}^I \quad \boldsymbol{q}_I^B \quad \boldsymbol{v}_{B/I}^B \quad \boldsymbol{a}_b \quad \boldsymbol{\omega}_b \quad b_b \quad h_{\text{ref}} \right]^\top \in \mathbb{R}^{14} \times \mathbb{S}^3. \quad (2.3)$$

$$\boldsymbol{x}^S \triangleq \left[\boldsymbol{p}_{B/S}^S \quad \boldsymbol{q}_I^B \quad \boldsymbol{v}_{B/I}^B \quad \boldsymbol{a}_b \quad \boldsymbol{\omega}_b \quad b_b \quad h_{\text{ref}} \quad \boldsymbol{q}_I^S \quad \boldsymbol{v}_{S/I}^I \right]^\top \in \mathbb{R}^{17} \times \mathbb{S}^3 \times \mathbb{S}^3. \quad (2.4)$$

For both the absolute and relative case, estimation is performed with the indirect (or “error-state”) form of the extended Kalman filter (EKF), referred to as the error-state Kalman filter (ESKF). The ESKF formulation is useful because it decomposes a state estimate $\hat{\boldsymbol{x}}$ into two components: a nominal state term \boldsymbol{x} and an error-state term $\tilde{\boldsymbol{x}}$ such that $\hat{\boldsymbol{x}} \triangleq \boldsymbol{x} + \tilde{\boldsymbol{x}}$. Whereas the dynamics of the nominal state are fast with possibly highly nonlinear terms, the dynamics of the error-state are comparatively slow and are much better approximated as linear as long as the error-state remains small. Thus, because the ESKF propagates the nominal state directly through integration and only treats the error-state as a stochastic process subject to innovation corrections, its numerical properties are generally more desirable than its direct counterpart. Moreover, when $\hat{\boldsymbol{x}}$ also contains a component that cannot be expressed as a vector (such as attitude), the ESKF has the additional advantage of having all of its vector operations performed in manifold tangent spaces. When the state estimate contains a nonlinear manifold component, the definition of $\hat{\boldsymbol{x}}$ should be augmented to $\hat{\boldsymbol{x}} \triangleq \boldsymbol{x} \boxplus \tilde{\boldsymbol{x}}$, where \boxplus is the operator that increments members of a manifold (to the left of the operator) with a vector term belonging to a vector space tangent to the manifold (to the right of the operator) [48].

The principal difference between the ESKF and the EKF is the need to derive the error-state dynamics and their corresponding Jacobians on the manifold. A full set of derivations for the error-state dynamics and measurement model Jacobians for the absolute and relative estimators are given in Sec. 2.5.

The filter propagation and update steps are defined by slight modifications to the standard EKF equations, which are

$$\Phi_k = \frac{\partial}{\partial \mathbf{x}} f_k(\mathbf{x}, \mathbf{u}_k) |_{\mathbf{x}=\hat{\mathbf{x}}_k^+} \quad (2.5)$$

$$\hat{\mathbf{x}}_{k+1}^- = f_k(\hat{\mathbf{x}}_k^+, \mathbf{u}_k) \quad (2.6)$$

$$\mathbf{Q}_{k+1}^- = \Phi_k \mathbf{Q}_k^+ \Phi_k^\top + \mathbf{W}_k \quad (2.7)$$

$$\mathbf{r}_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_k^-) \quad (2.8)$$

$$\mathbf{C}_k = \frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}) |_{\mathbf{x}=\hat{\mathbf{x}}_k^+} \quad (2.9)$$

$$\mathbf{S}_k = \mathbf{V}_k + \mathbf{C}_k \mathbf{Q}_k^- \mathbf{C}_k^\top \quad (2.10)$$

$$\mathbf{Q}_k^+ = \mathbf{Q}_k^- - \mathbf{Q}_k^- \mathbf{C}_k^\top \mathbf{S}_k^{-1} \mathbf{C}_k \mathbf{Q}_k^- \quad (2.11)$$

$$\mathbf{L}_k = \mathbf{Q}_k^- \mathbf{C}_k^\top \mathbf{S}_k^{-1} \quad (2.12)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + L_k \mathbf{r}_k, \quad (2.13)$$

with $\mathbf{u}_k = [\mathbf{a}_m \ \boldsymbol{\omega}_m]^\top$ as the current IMU accelerometer and gyro measurements. At each propagation step, the error-state dynamics are linearized about the current nominal state and discretized with first-order Euler integration to obtain Φ_k with Eq. 2.5. After each update step, the error-state mean is injected into the nominal state and subsequently reset to zero with a corresponding filter covariance reset. To add robustness to outlier measurements, covariance gating [5] is also implemented for each measurement update.

The absolute estimator fuses only standard sensors onboard the UAS, including an IMU, GPS, and barometer, with the exception of vision-based attitude corrections when the IR beacon array is in view of the onboard camera. These attitude corrections allow the IMU accelerometer and gyro biases to be observable, which is important because low-cost IMU measurements (particularly gyro measurements) can

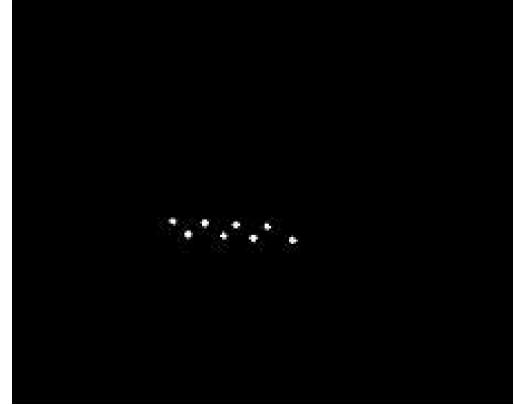
drift significantly over the course of several minutes. IMU bias observability requires inertial attitude updates.

The relative estimator, by contrast, fuses the IMU, barometer, differential GPS, full vision-based pose measurement, and a separate GPS unit mounted on the ship deck. The ship-mounted GPS provides information about the ship frame velocity and also allows for real-time kinematic (RTK) differential GPS calculations between the ship and the UAS. Because differential GPS gives relative position in the inertial frame, disparities between its and the vision pose calculator’s relative position measurements informs the estimation of \mathbf{q}_I^S . This property of iteratively estimating the passive rotation between the inertial and ship frames, however, exhibits a transient behavior that can adversely affect the accuracy of the estimated UAS velocity when the IR beacon array comes into the view of the onboard camera for the first time. Though the velocity estimates quickly re-converge within a matter of seconds, the effect is significant enough to merit controlling the UAS off of the absolute estimated state alone while using the relative estimated state in real-time only for high-level trajectory generation logic based only on relative positions.

To add robustness and accuracy to the real-time relative pose estimation between the S and B frames, a monocular pose estimation algorithm with active IR beacons based on [10] and [15] is employed. Pictured in Fig. 2-3b, two co-linear arrays of 45-watt IR beacons mounted to a light frame are fixed to the ship deck, and the position of each beacon \mathbf{p}_i^S expressed in frame S is recorded. During flight, a monocular IR camera onboard the UAS records video of the array, producing images such as the one depicted in Fig. 2-5a at 10 Hz. The size and brightness of the beacons afford them visibility during a bright, sunny day at distances well beyond 150 meters. During the air wake measurement process, the beacons are the brightest objects captured by the IR camera, making them ideal visual features for relative pose extraction. The captured camera images are passed through a filter to isolate the beacon features in the image plane and to reject any outlier features that do not conform to the co-linear beacon geometry. Figure 2-5b shows a filtered (and cropped) version of the image taken in Fig. 2-5a at a distance of 150 meters.



(a) Image captured from UAS camera of IR beacon array at a distance of 150 meters on a sunny day.



(b) Filtered camera image with programmatically isolated beacon feature points in the image plane.

Figure 2-5: Vision-based relative pose measurement system using active IR beacon features. Images obtained from [12].

The filtered image features are subsequently associated with the pre-recorded beacon positions \mathbf{p}_i^S to create eight 3D-to-2D point correspondences, leveraging the knowledge that the UAS will always be viewing the beacon array from an upright and forward-facing position. These point correspondences are used together with the previously known camera intrinsic parameters to deduce the 6-degree-of-freedom (6-DOF) transform between the ship frame S and the camera frame C using a Perspective-n-Point algorithm run in conjunction with RANSAC for outlier detection. The Perspective-n-Point algorithm uses a nonlinear optimization routine to minimize the re-projection error of each feature point correspondence by using the 6-DOF relative pose as the design variable. The transform between the UAS body frame B and the camera frame C , assumed to be known, is then used to obtain the vision-based relative pose measurement from S to B . The pose extraction algorithm, tailored to the two-row beacon array, is able to match the input image rate at a speed of 10 Hz while running on the onboard computer.

However, due to the nature of nonlinear optimization and possible image distortion that can occur with accelerations during flight, the vision system also has a final outlier detector to reject solutions which suggest that the solver has not handled a correspondence ambiguity correctly.

2.5 Error-State Kalman Filter Equations Derivation

2.5.1 ESKF Background and Quaternion Conventions

The exact process for deriving the linear form of the error-state dynamics and update equations is tightly coupled with the chosen conventions for representing rotation. Below is a table showing different convention choices for quaternions, showing two of the most popular named sets of conventions. The two sets, Hamilton and JPL, are chosen due to the vast amount of corresponding support literature available. The last column shows the conventions of the quaternion implementation used for this work, which must be taken into account in the derivations.

Table 2.1: Comparison of quaternion representation conventions for two of the most commonly used conventions in robotics and aerospace [49] and the conventions used in this work.

Convention	Hamilton	JPL	This Work
Components order	(q_w, \mathbf{q}_v)	(\mathbf{q}_v, q_w)	(q_w, \mathbf{q}_v)
Handedness	$ij = k$ (right)	$ij = -k$ (left)	$ij = k$ (right)
Function	Passive	Passive	Active
Passive Directionality	Local-to-Global	Global-to-Local	Global-to-Local

As can be seen from Table 2.1, the quaternion convention used in this work is relatively idiosyncratic compared to existing popular conventions, which makes these derivations different from many of the standard ESKF derivations in the literature (for the most part). Each convention choice carries with it implications regarding how to compute Jacobians on the manifold.

To assist in the computation of the measurement model Jacobians, the following $SO(3)$ Jacobian blocks are computed for the quaternion conventions in this work:

$$\frac{\partial}{\partial \tilde{\theta}} \mathbf{R} \mathbf{v} = [\mathbf{R} \mathbf{v}]_{\times} \quad (2.14)$$

$$\frac{\partial}{\partial \tilde{\theta}} \mathbf{R}^T \mathbf{v} = -\mathbf{R}^T [\mathbf{v}]_{\times}, \quad (2.15)$$

$$\frac{\partial}{\partial \tilde{\theta}} \mathbf{C} \mathbf{R} \mathbf{v} = \mathbf{C} [\mathbf{R} \mathbf{v}]_{\times}, \quad (2.16)$$

$$\frac{\partial}{\partial \tilde{\theta}} \mathbf{R} \mathbf{C} = \mathbf{I}, \quad (2.17)$$

$$\frac{\partial}{\partial \tilde{\theta}} \mathbf{C} \mathbf{R}^{\top} = -\mathbf{C} \mathbf{R}^{\top}, \quad (2.18)$$

$$\frac{\partial}{\partial \tilde{\theta}} \mathbf{C} \mathbf{R}^{\top} \mathbf{v} = -\mathbf{C} \mathbf{R}^{\top} [\mathbf{v}]_{\times}, \quad (2.19)$$

assuming that $\mathbf{R} \in SO(3)$ is the quantity that varies with $\tilde{\theta}$.

2.5.2 Absolute Error-State Dynamics

The nominal dynamics corresponding to the absolute state (Eq. 2.3) are

$$\dot{\mathbf{p}}_{B/I}^I = (\mathbf{R}_I^B)^{\top} \mathbf{v}_{B/I}^B, \quad (2.20)$$

$$\dot{\mathbf{q}}_I^B = \frac{1}{2} \mathbf{q}_I^B \otimes \boldsymbol{\omega}, \quad (2.21)$$

$$\dot{\mathbf{v}}_{B/I}^B = \mathbf{a} + \mathbf{R}_I^B \mathbf{g}^I - \boldsymbol{\omega} \times \mathbf{v}_{B/I}^B, \quad (2.22)$$

where $\boldsymbol{\omega} \triangleq \boldsymbol{\omega}_m - \boldsymbol{\omega}_b$ and $\mathbf{a} \triangleq \mathbf{a}_m - \mathbf{a}_b$. Note that the equation for $\dot{\mathbf{q}}_I^B$ has its multiplication order determined by handedness and, in practice, is implemented with first-order Euler integration as $\dot{\mathbf{q}}_I^B = \boldsymbol{\omega}$.

The error-state is defined (in terms of *local* perturbations to the nominal state) as

$$\tilde{\mathbf{x}} = \left[\tilde{\mathbf{p}}_{B_t/B}^I \quad \tilde{\theta}_B^{B_t} \quad \tilde{\mathbf{v}}_{B_t/B}^B \quad \tilde{\mathbf{a}}_b \quad \tilde{\boldsymbol{\omega}}_b \quad \tilde{\mathbf{b}}_b \quad \tilde{h}_{\text{ref}} \right]^{\top}. \quad (2.23)$$

The Rodrigues form of the error-state vector $\tilde{\theta}$ is used for generality and for straightforward conversions to the quaternion and rotation matrix forms for convenience in derivations.

The position error state $\dot{\tilde{\mathbf{p}}}_{B_t/B}^I$ dynamics are derived with left and right expansions of $\dot{\mathbf{p}}_{B_t/B}^I$ that are then equated with each other. Linearization approximations are applied in all right expansions where necessary to derive a linear model.

The left expansion is given as

$$\dot{\mathbf{p}}_{B_t/I}^I = (\dot{\mathbf{p}}_{B/I}^I + \dot{\tilde{\mathbf{p}}}_{B_t/B}^I) \quad (2.24)$$

$$= (\mathbf{R}_I^B)^\top \mathbf{v}_{B/I}^B + \dot{\tilde{\mathbf{p}}}_{B_t/B}^I, \quad (2.25)$$

and the right expansion as

$$\dot{\mathbf{p}}_{B_t/I}^I = \left(\text{Exp} \left(\tilde{\boldsymbol{\theta}}_B^{B_t} \right) \mathbf{R}_I^B \right)^\top (\mathbf{v}_{B/I}^B + \tilde{\mathbf{v}}_{B_t/B}^B) \quad (2.26)$$

$$\approx (\mathbf{R}_I^B)^\top (I + \lfloor \tilde{\boldsymbol{\theta}}_B^{B_t} \rfloor_\times) (\mathbf{v}_{B/I}^B + \tilde{\mathbf{v}}_{B_t/B}^B) \quad (2.27)$$

$$\approx (\mathbf{R}_I^B)^\top \mathbf{v}_{B/I}^B + (\mathbf{R}_I^B)^\top \tilde{\mathbf{v}}_{B_t/B}^B + (\mathbf{R}_I^B)^\top \lfloor \tilde{\boldsymbol{\theta}}_B^{B_t} \rfloor_\times \mathbf{v}_{B/I}^B. \quad (2.28)$$

Equating the two expansions yields

$$\dot{\tilde{\mathbf{p}}}_{B_t/B}^I = (\mathbf{R}_I^B)^\top \tilde{\mathbf{v}}_{B_t/B}^B - (\mathbf{R}_I^B)^\top \lfloor \mathbf{v}_{B/I}^B \rfloor_\times \tilde{\boldsymbol{\theta}}_B^{B_t}. \quad (2.29)$$

In the derivation of the orientation error-state $\dot{\tilde{\boldsymbol{\theta}}}_B^{B_t}$ dynamics, the chosen quaternion conventions for handedness, function, and passive directionality (see Table 2.1) all matter. As it turns out, the conventions used in this work for function and passive directionality work together for this derivation in the exact same way as with the Hamilton convention. The derivation is then identical to what can be found in other works using the popular Hamilton convention, and thus is omitted from this section. The interested reader may find a derivation on pages 58-59 of [48]. The final result is given by

$$\dot{\tilde{\boldsymbol{\theta}}}_B^{B_t} = \tilde{\boldsymbol{\omega}} - \lfloor \boldsymbol{\omega} \rfloor_\times \tilde{\boldsymbol{\theta}}_B^{B_t}. \quad (2.30)$$

The velocity error-state $\dot{\tilde{\mathbf{v}}}_{B_t/B}^B$ dynamics are derived from the left expansion

$$\dot{\mathbf{v}}_{B_t/I}^B = \dot{\mathbf{v}}_{B/I}^B + \dot{\tilde{\mathbf{v}}}_{B_t/B}^B \quad (2.31)$$

$$= \mathbf{a} + \mathbf{R}_I^B \mathbf{g}^I - \lfloor \boldsymbol{\omega} \rfloor_\times \mathbf{v}_{B/I}^B + \dot{\tilde{\mathbf{v}}}_{B_t/B}^B, \quad (2.32)$$

and right expansion (applying linearizing assumptions and omitting higher-order

terms)

$$\dot{\tilde{\mathbf{v}}}^B_{B_t/I} = \tilde{\mathbf{a}} + \text{Exp}\left(\tilde{\boldsymbol{\theta}}_B^{B_t}\right) \mathbf{R}_I^B \mathbf{g}^I - [\boldsymbol{\omega} + \tilde{\boldsymbol{\omega}}]_\times (\mathbf{v}_{B/I}^B + \tilde{\mathbf{v}}_{B_t/B}^B) \quad (2.33)$$

$$\approx \tilde{\mathbf{a}} + \left(\mathbf{I} - [\tilde{\boldsymbol{\theta}}_B^{B_t}]_\times\right) \mathbf{R}_I^B \mathbf{g}^I - [\boldsymbol{\omega}]_\times \mathbf{v}_{B/I}^B - [\boldsymbol{\omega}]_\times \tilde{\mathbf{v}}_{B_t/B}^B - [\tilde{\boldsymbol{\omega}}]_\times \mathbf{v}_{B/I}^B \quad (2.34)$$

$$\begin{aligned} &\approx \tilde{\mathbf{a}} + \mathbf{R}_I^B \mathbf{g}^I - [\tilde{\boldsymbol{\theta}}_B^{B_t}]_\times \mathbf{R}_I^B \mathbf{g}^I - [\boldsymbol{\omega}]_\times \mathbf{v}_{B/I}^B - [\boldsymbol{\omega}]_\times \tilde{\mathbf{v}}_{B_t/B}^B \\ &\quad - [\tilde{\boldsymbol{\omega}}]_\times \mathbf{v}_{B/I}^B. \end{aligned} \quad (2.35)$$

Equating the two sides gives

$$\dot{\tilde{\mathbf{v}}}^B_{B_t/B} = \tilde{\mathbf{a}} + [\mathbf{R}_I^B \mathbf{g}^I]_\times \tilde{\boldsymbol{\theta}}_B^{B_t} - [\boldsymbol{\omega}]_\times \tilde{\mathbf{v}}_{B_t/B}^B + [\mathbf{v}_{B/I}^B]_\times \tilde{\boldsymbol{\omega}}. \quad (2.36)$$

The linearized error-state equations are combined in state-space form to define the full Jacobian matrices for the error-state dynamics, given by

$$\dot{\tilde{\mathbf{x}}}^I = \begin{bmatrix} \dot{\tilde{\mathbf{p}}}_{B_t/B}^I \\ \dot{\tilde{\boldsymbol{\theta}}}_B^{B_t} \\ \dot{\tilde{\mathbf{v}}}_{B_t/B}^B \\ \dot{\tilde{\mathbf{a}}}_b \\ \dot{\tilde{\boldsymbol{\omega}}}_b \\ \dot{\tilde{b}}_b \\ \dot{\tilde{h}}_{\text{ref}} \end{bmatrix} = \mathbf{A}_{\text{abs}} \tilde{\mathbf{x}}^I + \mathbf{B}_{\text{abs}} \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix}, \quad (2.37)$$

$$\mathbf{A}_{\text{abs}} = \begin{bmatrix} \mathbf{0} & -(\mathbf{R}_I^B)^\top [\mathbf{v}_{B/I}^B]_\times & (\mathbf{R}_I^B)^\top & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & -[\boldsymbol{\omega}]_\times & \mathbf{0} & \mathbf{0} & -\mathbf{I} & 0 & 0 \\ \mathbf{0} & [\mathbf{R}_I^B \mathbf{g}^I]_\times & -[\boldsymbol{\omega}]_\times & -\mathbf{I} & -[\mathbf{v}_{B/I}^B]_\times & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.38)$$

$$\mathbf{B}_{\text{abs}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ \mathbf{I} & [\mathbf{v}_{B/I}^B]_{\times} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (2.39)$$

2.5.3 Absolute Measurement Equations

The nonlinear measurement model $\bar{\mathbf{z}} = h(\mathbf{x}^I)$ and corresponding error-state-linearized model $\bar{\mathbf{z}} \approx \frac{\partial h}{\partial \tilde{\mathbf{x}}^I} \tilde{\mathbf{x}}^I = \mathbf{H} \tilde{\mathbf{x}}^I$ is now presented for each sensor incorporated into the absolute estimator, as explained in Section 2.4. The linearizations rely on the quaternion convention-specific matrix derivative rules (Eq. 2.14-2.19).

The barometer sensor output varies with ground pressure P_{ground} , atmospheric density ρ , and gravitational acceleration g as

$$\bar{\mathbf{z}} = P_{\text{ground}} + \rho g \mathbf{p}_{B/I,z}^I + b_b, \quad (2.40)$$

and the linearized model is

$$\mathbf{H} = \left[\begin{bmatrix} 0 & 0 & \rho g \end{bmatrix} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ 1 \ 0 \right]. \quad (2.41)$$

The high-rate velocity measurements from the UAS-mounted GPS module are described by the measurement model

$$\bar{\mathbf{z}} = (\mathbf{R}_I^B)^T \mathbf{v}_{B/I}^B, \quad (2.42)$$

$$\mathbf{H} = \left[\mathbf{0} \ -(\mathbf{R}_I^B)^T [\mathbf{v}_{B/I}^B]_{\times} \ (\mathbf{R}_I^B)^T \ \mathbf{0} \ \mathbf{0} \ 0 \ 0 \right]. \quad (2.43)$$

The GPS module also reports position and velocity in a format that directly relates to the Earth-centered, Earth-fixed (ECEF) frame. Thus, the incorporation of

the GPS fix measurements requires the rotation between the inertial NED and ECEF frames, \mathbf{R}_E^I , which is a function of the reference latitude, longitude, and altitude for the chosen NED coordinate frame origin. With this rotation in-hand, the GPS measurement models are

$$\bar{\mathbf{z}} = \begin{bmatrix} (\mathbf{R}_E^I)^\top \mathbf{p}_{B/I}^I + {}^E \mathbf{p}_{I/E} \\ (\mathbf{R}_E^I)^\top (\mathbf{R}_I^B)^\top \mathbf{v}_{B/I}^B \end{bmatrix}, \quad (2.44)$$

$$\mathbf{H} = \begin{bmatrix} (\mathbf{R}_E^I)^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -(\mathbf{R}_E^I)^\top (\mathbf{R}_I^B)^\top [\mathbf{v}_{B/I}^B]_\times & (\mathbf{R}_E^I)^\top (\mathbf{R}_I^B)^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (2.45)$$

Attitude corrections received from the relative estimator are described by the simple measurement models

$$\bar{\mathbf{z}} = \mathbf{q}_I^B, \quad (2.46)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (2.47)$$

Before takeoff, it is useful to incorporate a pseudo-measurement that prevents the estimated altitude and attitude of the UAS from changing (lateral position is allowed to vary, as the UAS would be on a moving ship). The corresponding measurement models are

$$\bar{\mathbf{z}} = \begin{bmatrix} \mathbf{q}_I^B \\ \mathbf{p}_{B/I,z}^I \end{bmatrix}, \quad (2.48)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ [0 \ 0 \ 1] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (2.49)$$

Finally, motion capture-based true pose measurements can be optionally incorporated into the estimator with the models

$$\bar{\mathbf{z}} = \begin{bmatrix} \mathbf{p}_{B/I}^I \\ \mathbf{q}_I^B \end{bmatrix}, \quad (2.50)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (2.51)$$

2.5.4 Relative Error-State Dynamics

The nominal dynamics corresponding to the relative state (Eq. 2.4) are

$$\dot{\mathbf{p}}_{B/S}^S = \mathbf{R}_I^S (\mathbf{R}_I^B)^\top \mathbf{v}_{B/I}^B - \mathbf{R}_I^S \mathbf{v}_{S/I}^I, \quad (2.52)$$

$$\dot{\mathbf{q}}_I^B = \frac{1}{2} \mathbf{q}_I^B \otimes \boldsymbol{\omega}, \quad (2.53)$$

$$\dot{\mathbf{v}}_{B/I}^B = \mathbf{a} + \mathbf{R}_I^B \mathbf{g}^I - \boldsymbol{\omega} \times \mathbf{v}_{B/I}^B. \quad (2.54)$$

For the air wake measurement system, the ship's orientation \mathbf{q}_I^S and velocity $\mathbf{v}_{S/I}^I$ are assumed to be quasi-static quantities. The local-perturbation error-state is defined as

$$\tilde{\mathbf{x}}^S = \left[\tilde{\mathbf{p}}_{B_t/B}^S \quad \tilde{\boldsymbol{\theta}}_B^{B_t} \quad \tilde{\mathbf{v}}_{B_t/B}^B \quad \tilde{\mathbf{a}}_b \quad \tilde{\boldsymbol{\omega}}_b \quad \tilde{b}_b \quad \tilde{h}_{\text{ref}} \quad \tilde{\boldsymbol{\theta}}_S^{S_t} \quad \tilde{\mathbf{v}}_{S_t/S}^I \right]^\top. \quad (2.55)$$

All error-state derivations from Section 2.5.2 can be re-used here, with the exception of the updated relative position error-state $\dot{\tilde{\mathbf{p}}}_{B_t/B}^S$ dynamics. The left and right expansions of the relative position error-state dynamics are given as

$$\dot{\tilde{\mathbf{p}}}_{B_t/S}^S = \dot{\mathbf{p}}_{B/S}^S + \dot{\tilde{\mathbf{p}}}_{B_t/B}^S \quad (2.56)$$

$$= \mathbf{R}_I^S \left((\mathbf{R}_I^B)^\top \mathbf{v}_{B/I}^B - \mathbf{v}_{S/I}^I \right) + \dot{\tilde{\mathbf{p}}}_{B_t/B}^S, \quad (2.57)$$

and

$$\begin{aligned} \dot{\tilde{\mathbf{p}}}_{B_t/S}^S &= \left(\text{Exp} \left(\tilde{\boldsymbol{\theta}}_S^{S_t} \right) \mathbf{R}_I^S \right) \left(\text{Exp} \left(\tilde{\boldsymbol{\theta}}_B^{B_t} \right) \mathbf{R}_I^B \right)^\top (\mathbf{v}_{B/I}^B + \tilde{\mathbf{v}}_{B/I}^B) \\ &\quad - \text{Exp} \left(\tilde{\boldsymbol{\theta}}_S^{S_t} \right) \mathbf{R}_I^S (\mathbf{v}_{S/I}^I + \tilde{\mathbf{v}}_{S/I}^I) \\ &\approx \left(\mathbf{I} - \lfloor \tilde{\boldsymbol{\theta}}_S^{S_t} \rfloor_\times \right) \mathbf{R}_I^S (\mathbf{R}_I^B)^\top \left(\mathbf{I} + \lfloor \tilde{\boldsymbol{\theta}}_B^{B_t} \rfloor_\times \right) (\mathbf{v}_{B/I}^B + \tilde{\mathbf{v}}_{B/I}^B) \end{aligned} \quad (2.58)$$

$$-\left(\mathbf{I} - \lfloor \tilde{\boldsymbol{\theta}}_S^{S_t} \rfloor_{\times}\right) \mathbf{R}_I^S (\mathbf{v}_{S/I}^I + \tilde{\mathbf{v}}_{S/I}^I) \quad (2.59)$$

$$\begin{aligned} &\approx \left(\mathbf{R}_B^S - \lfloor \tilde{\boldsymbol{\theta}}_S^{S_t} \rfloor_{\times} \mathbf{R}_B^S\right) \left(\mathbf{I} + \lfloor \tilde{\boldsymbol{\theta}}_B^{B_t} \rfloor_{\times}\right) (\mathbf{v}_{B/I}^B + \tilde{\mathbf{v}}_{B/I}^B) \\ &\quad - \left(\mathbf{I} - \lfloor \tilde{\boldsymbol{\theta}}_S^{S_t} \rfloor_{\times}\right) \mathbf{R}_I^S (\mathbf{v}_{S/I}^I + \tilde{\mathbf{v}}_{S/I}^I) \end{aligned} \quad (2.60)$$

$$\begin{aligned} &\approx \mathbf{R}_B^S \mathbf{v}_{B/I}^B + \mathbf{R}_B^S \tilde{\mathbf{v}}_{B/I}^B + \mathbf{R}_B^S \lfloor \tilde{\boldsymbol{\theta}}_B^{B_t} \rfloor_{\times} \mathbf{v}_{B/I}^B - \lfloor \tilde{\boldsymbol{\theta}}_S^{S_t} \rfloor_{\times} \mathbf{R}_B^S \mathbf{v}_{B/I}^B - \mathbf{R}_I^S \mathbf{v}_{S/I}^I \\ &\quad - \mathbf{R}_I^S \tilde{\mathbf{v}}_{S/I}^I + \lfloor \tilde{\boldsymbol{\theta}}_S^{S_t} \rfloor_{\times} \mathbf{R}_I^S \mathbf{v}_{S/I}^I. \end{aligned} \quad (2.61)$$

Equating the left and right expansions yields

$$\dot{\tilde{\mathbf{p}}}_{B_t/B}^S = -\mathbf{R}_B^S \lfloor \mathbf{v}_{B/I}^B \rfloor_{\times} \tilde{\boldsymbol{\theta}}_B^{B_t} + \mathbf{R}_B^S \tilde{\mathbf{v}}_{B/I}^B + \lfloor \mathbf{R}_B^S \mathbf{v}_{B/I}^B - \mathbf{R}_I^S \mathbf{v}_{S/I}^I \rfloor_{\times} \tilde{\boldsymbol{\theta}}_S^{S_t} - \mathbf{R}_I^S \tilde{\mathbf{v}}_{S/I}^I. \quad (2.62)$$

The final linearized error-state equations, then, are

$$\dot{\tilde{\mathbf{x}}}^S = \begin{bmatrix} \dot{\tilde{\mathbf{p}}}_{B_t/B}^S \\ \dot{\tilde{\boldsymbol{\theta}}}_B^{B_t} \\ \dot{\tilde{\mathbf{v}}}_{B_t/B}^B \\ \dot{\tilde{\mathbf{a}}}_b \\ \dot{\tilde{\boldsymbol{\omega}}}_b \\ \dot{\tilde{b}}_b \\ \dot{\tilde{h}}_{\text{ref}} \\ \dot{\tilde{\boldsymbol{\theta}}}_S^{S_t} \\ \dot{\tilde{\mathbf{v}}}_{S_t/S}^I \end{bmatrix} = \mathbf{A}_{\text{rel}} \tilde{\mathbf{x}}^S + \mathbf{B}_{\text{rel}} \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\boldsymbol{\omega}} \end{bmatrix},$$

$$\begin{aligned}
A_{\text{rel}} &= \begin{bmatrix} 0 & -\mathbf{R}_B^S [\mathbf{v}_{B/I}^B]_\times & \mathbf{R}_B^S & 0 & 0 & 0 & 0 & [\mathbf{R}_B^S \mathbf{v}_{B/I}^B - \mathbf{R}_I^S \mathbf{v}_{S/I}^I]_\times & -\mathbf{R}_I^S \\ 0 & -[\boldsymbol{\omega}]_\times & 0 & 0 & -\mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & [\mathbf{R}_I^B \mathbf{g}^I]_\times & -[\boldsymbol{\omega}]_\times & -\mathbf{I} & -[\mathbf{v}_{B/I}^B]_\times & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\
B_{\text{rel}} &= \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{I} \\ \mathbf{I} & [\mathbf{v}_{B/I}^B]_\times \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.
\end{aligned}$$

2.5.5 Relative Measurement Equations

Similar to the model presentation in Section 2.5.3, this section presents the measurement models for the sensors incorporated into the relative estimator, as described in Section 2.4.

The IR vision system provides a direct relative pose measurement between the UAS and the ship, modeled as

$$\bar{\mathbf{z}} = \begin{bmatrix} \mathbf{p}_{B/S}^S \\ \mathbf{R}_I^B (\mathbf{R}_I^S)^\top \sim (\mathbf{q}_I^S)^* \otimes \mathbf{q}_I^B \end{bmatrix}, \quad (2.63)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{R}_I^B (\mathbf{R}_I^S)^\top & \mathbf{0} \end{bmatrix}. \quad (2.64)$$

Note the reversed order of the quaternion composition when compared with the rotation matrix composition. Although the right-handed form of \otimes normally corresponds to the same kind of composition as with rotation matrices, the fact that quaternions in this convention are actually active necessitates the reversed order.

The high-rate NED-frame velocity measurements from the UAS-mounted GPS are modeled as

$$\bar{\mathbf{z}} = (\mathbf{R}_I^B)^\top \mathbf{v}_{B/I}^B, \quad (2.65)$$

$$\mathbf{H} = \left[\mathbf{0} \quad -(\mathbf{R}_I^B)^\top [\mathbf{v}_{B/I}^B]_\times \quad (\mathbf{R}_I^B)^\top \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \right]. \quad (2.66)$$

The RTK GPS relative position measurements, given in the NED frame, are described by the measurement models

$$\bar{\mathbf{z}} = (\mathbf{R}_I^S)^\top \mathbf{p}_{B/S}^S, \quad (2.67)$$

$$\mathbf{H} = \left[(\mathbf{R}_I^S)^\top \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad -(\mathbf{R}_I^S)^\top [\mathbf{p}_{B/S}^S]_\times \quad \mathbf{0} \right]. \quad (2.68)$$

In the relative scheme, the ship has its own GPS module affixed to it, reporting NED-frame velocities described with

$$\bar{\mathbf{z}} = \mathbf{v}_{S/I}^I, \quad (2.69)$$

$$\mathbf{H} = \left[\mathbf{0} \quad \mathbf{I} \right]. \quad (2.70)$$

Optionally, a magnetometer can also be mounted on the ship deck, and its magnetic field measurements are incorporated into the relative estimator with

$$\bar{\mathbf{z}} = \mathbf{R}_S^M \mathbf{R}_I^S \hat{\mathbf{m}}^I, \quad (2.71)$$

$$\mathbf{H} = \left[\mathbf{0} \quad \mathbf{R}_S^M [\mathbf{R}_I^S \hat{\mathbf{m}}^I]_\times \quad \mathbf{0} \right]. \quad (2.72)$$

Motion capture measurements, which must provide information about both the UAS and ship frames, can be used in lieu of the aforementioned sensors with the measurement models

$$\bar{\mathbf{z}} = \left[(\mathbf{R}_I^S)^\top \ \mathbf{p}_{B/S}^S \ \mathbf{q}_I^B \ \mathbf{q}_I^S \ \mathbf{v}_{S/I}^I \right]^\top, \quad (2.73)$$

$$\mathbf{H} = \begin{bmatrix} (\mathbf{R}_I^S)^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -(\mathbf{R}_I^S)^\top [\mathbf{p}_{B/S}^S]_\times & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (2.74)$$

2.6 Summary

This chapter presents the tailored control strategy for tethered flight employed by the UAS for the air wake measurement system. A description of the sensing modalities and estimation schemes for absolute and relative state estimation is given, together with derivations for the underlying error-state dynamic and measurement models. Chapter 3 presents the simulation environment and hardware used to test and evaluate the control and estimation algorithms subject to representative disturbances and environmental conditions.

Chapter 3

Simulation and Hardware for Air Wake Measurement

3.1 Simulated Maritime Environment

The use of an autonomous control strategy coupled with real-time sensor fusion in-the-loop necessitates a high-fidelity test bed for development and validation of the flight-critical algorithms. To partially fulfill this need, a simulator built in ROS has been developed to model the 6-degree-of-freedom (DOF) UAS dynamics, ship motion, tether contact dynamics, sensor noise models, and aerodynamic drag effects present in a mobile, maritime environment with significant wind. Figure 3-1 shows sample views of the simulation environment, depicting the naval vessel, UAS, and tether. The simulator runs all onboard algorithms described in Figure 2-2 as hardware-in-the-loop (HIL) with the UAS ODROID-XU4 computer and runs through all phases of the air wake measurement process, including takeoff, measurement acquisition, return-to-home, and land. An identical ground station interface to the one used in field testing is also provided for the user to practice sending commands to the UAS and tether reel controller.

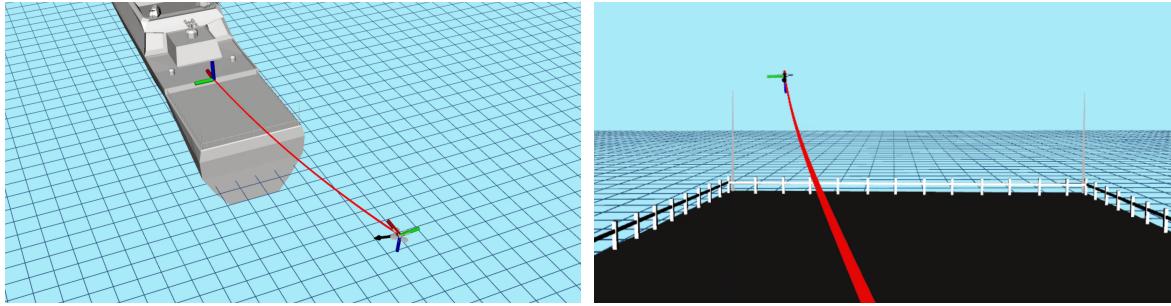


Figure 3-1: Simulation environment for the air wake measurement system, used for testing all phases of the measurement flight in the presence of configurable wind disturbances and at varying levels of sensor noise and bias. The simulated tether line, colored red, keeps the airborne UAS anchored to a naval vessel moving forward at a steady velocity.

3.1.1 UAS Software-in-the-Loop Autopilot and Dynamics

To establish the simulation environment as a useful measure of UAS suitability for ship air wake measurement, the parameters defining UAS dynamic characteristics and aerodynamic susceptibility are validated against measured behavior in hardware. The UAS motor parameters are measured directly using the Dynamometer Series 1580 thrust stand, mass is measured on a scale, and aerodynamic parameters for the rotor blade flapping effect are estimated using least-squares optimization with hardware data from several motion capture flights. Sensor noise and bias parameters are validated against recorded sensor data on the hardware platform, determining covariance levels through obtaining Gaussian fits of the data. As a final validation test, the simulated UAS has been found to exhibit the same qualitative flight behavior as the hardware platform as measured with a motion capture system when provided identical inputs to the inner-loop autopilot. With validated dynamic behavior, the simulation environment provides a suitable test bed for evaluating control and estimation algorithm performance before executing a mission on an actual naval vessel.

3.1.2 Aerodynamic Turbulence Effects

Environmental wind conditions are simulated using a Dryden gust stochastic process model [30] in its pseudo-spectral form given by

$$\mathbf{G}(s) = \sigma \sqrt{\frac{2L}{\pi V}} \frac{1 + \frac{2\sqrt{3}L}{V}s}{\left(1 + \frac{2L}{V}s\right)^2} \mathbf{U}(s), \quad (3.1)$$

where V is the effective velocity of the UAS through the air, L is the turbulence length scale, and σ is the turbulence intensity.

At non-negligible relative wind speeds, aerodynamic effects begin to impose significant moment and force disturbances on small aircraft. These effects are modeled in the simulator through the environmental implementation of the rotor blade flapping effect [20], which occurs due to both linear drag on the aircraft frame as well as differential moments that arise across the individual rotors when the half of the rotor blade moving into the wind generates more lift than the half moving with the wind. For example, in the scenario where the UAS is flying directly behind the ship and facing the air wake stream, a significant differential moment is generated about the UAS pitch axis, described by

$$\mathbf{M}_y^B = \sum_i^N (f_{i,\text{thrust}} h_i \sin(k_f v_{\text{wind},b_x}) + k_\beta k_f v_{\text{wind},b_x}), \quad (3.2)$$

where v_{wind,b_x} is the component of relative wind velocity aligned with the UAS body-x-axis, h_i is the height of rotor blade i with respect to the UAS COM, k_β is the rotor blade stiffness, and k_f is a constant providing a linear approximation to Eq. 8 on page 7 of [20], summed over N rotor blades. The linear approximation is explained preceding Eq. 14 on page 4 of [21].

With a relative headwind, the rotor blade flapping effect generates a positive moment about the pitch axis, resulting in a steady-state pitch disturbance through the duration of the flight. The effects are not limited to moments about the pitch axis, and must be reckoned with in a robust tethered flight control strategy.

In addition to the generated moments, the total force on the quadrotor from the

rotor blade flapping effect is approximated by

$$\mathbf{F}_{flap}^I = -\lambda_1 \sum_i \omega_i \mathbf{V}_\perp, \quad (3.3)$$

where ω_i is the rotational speed of each rotor, \mathbf{V}_\perp is the projection of $\mathbf{v}_{B/I}^B$ onto the propeller plane, and λ_1 is the rotor drag coefficient, usually empirically derived for a specific platform as on page 37 of [1]. Alternatively to estimating λ_1 directly, Abeywardena et al. estimate the aggregate parameter

$$k_1 = \lambda_1 \sum_i \omega_i. \quad (3.4)$$

The relationship between autopilot motor speed commands and individual motor thrust must also be modeled. Motor thrust is related to total angular rotation rate by the following relation [1]:

$$\mathbf{F}_{thrust}^B = k_F \sum_i \omega_i^2, \quad (3.5)$$

where k_F is the thrust coefficient of the propellers. A similarly-defined torque coefficient k_T is used to determine resulting motor torques:

$$\mathbf{T}_{thrust}^B = k_T \sum_i \omega_i^2. \quad (3.6)$$

3.1.3 Naval Vessel and Tether Contact Dynamics

In the simulation, the ship is configured to move forward at a constant velocity, with optional sinusoidal rolling, pitching, and yawing. Tether forces between the moving tether mount point and the UAS are modeled using the geometry of a catenary curve with spring-like contact dynamics, following the approach outlined in [27].



Figure 3-2: Indoor flight testing platform for the air wake measurement system, pictured within the Vicon motion capture space used to spoof GPS sensing.

3.1.4 Simulated Sensor Suite

All ship-based and UAS sensors involved in both absolute and relative state estimation are implemented with noise and random walk bias levels commensurate with empirically gathered data. The input sensor data, consisting of the IMU, barometer, camera, and GPS modules are modeled with added Gaussian white noise and random walk. The GPS measurements from the modules on the ship and onboard the UAS are modeled as Gauss-Markov processes, following the strategy outlined in [41] to model GPS error from ephemeris data, satellite clock, ionosphere, and troposphere-related effects.

3.2 System Hardware

Two different UAS hardware platforms are used for real-world flight testing of the autonomous control and estimation algorithms. The indoor flight test platform, together with a 1:35 scale model of the IR beacon array and tether reel controller, is pictured in Fig. 3-2.

Table 3.1: All sensor hardware used for both indoor and outdoor flight testing of the ship air wake measurement system.

Sensor	Placement	Product	Notes
IR Camera	UAS	Pointgrey Chameleon 3	4×4 binning, shutter speed: 0.005, exposure: -7.5, gain: -2.5
IMU	UAS	MPU-6000	Mounted to Openpilot Revolution board.
Barometer	UAS	MS5611	Mounted to Openpilot Revolution board.
Rover GPS	UAS	ublox ZED-F9P	Configured as a rover in “moving base” mode.
Base GPS	Ship	ublox ZED-F9P	Configured as a base in “moving base” mode.

Table 3.2: All computer hardware used for both indoor and outdoor flight testing of the ship air wake measurement system.

Function	Machine	Running Processes
Tether reel control	Raspberry Pi 3	Tether motor controller, reel HTTP server
Ground station	Dell XPS 15	Mission control, reel HTTP client
UAS autonomy	ODROID XU4	Control, absolute and relative estimation
UAS autopilot	STM32F4 chip	ROSflight autopilot

All sensors described in Section 2.4 are listed with their actual hardware product names in Table 3.1. During indoor flight testing, all actual sensors are used, with the exception of GPS, which is spoofed from the Vicon [60, 33] motion capture system. Similarly, all computers running the various described control and estimation algorithms are given in Table 3.2.

For outdoor flight testing, a larger airframe equipped with all of the same sensors and onboard computers is pictured in Fig. 3-3. The outdoor flight testing platform is also equipped with an additional Raspberry Pi 4 computer, whose sole purpose is to log time-stamped data from the omnidirectional air probe used to collect the actual air wake measurements. In Fig. 3-3, a probe placeholder is attached to mimic the sensor weight distribution during the testing of the autonomy algorithms, which is what this thesis focuses on.

Although both the indoor and outdoor UAS platforms are equipped with identical

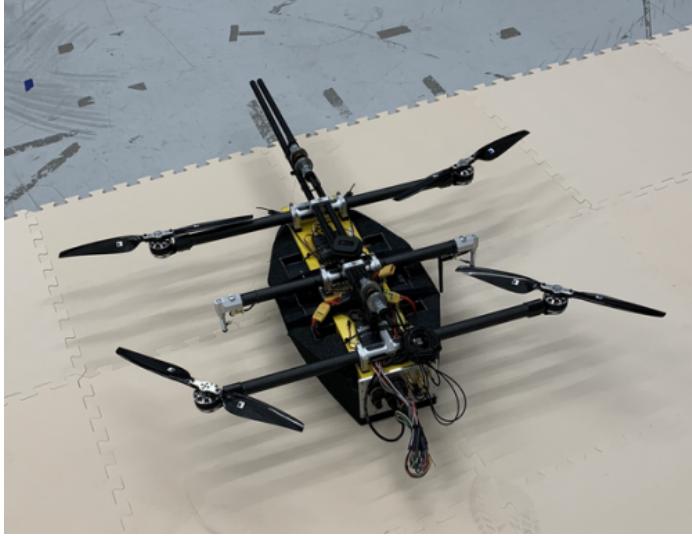


Figure 3-3: Outdoor flight testing platform for the air wake measurement system. An air probe placeholder is shown extending out of the front of the airframe, and the tether attachment points protrude out of the airframe’s sides. The yellow housing, which contains all onboard computers and all sensors except for the GPS antenna and air probe, is surrounded by a black foam casing for protection.

sensors and computers, the outdoor platform handles differently and is significantly heavier than the indoor platform at 7.6 kilograms versus 2.1 kilograms. This difference necessitates the tuning of separate controller gains for each platform.

The tether reel controller box, pictured between the miniature IR beacon array and UAS platform in Fig. 3-2, is commanded by the ground station operator via HTTP requests to set the tether at specified lengths during the course of the flight for varied sweep geometries. The reel line is handled with PID control and a tension feedback sensor to ensure that line is not doled out too quickly, which assists with flight stability.

3.3 Summary

This chapter presents the simulation environment, physical models, and hardware used to develop, test, and evaluate the control and estimation algorithms for the air wake measurement system project. The simulation and hardware components are designed to mirror each other and provide sufficient testing of all system modes

to accurately assess the system's suitability for maritime operation. The associated results for simulation and indoor/outdoor testing are presented in Sections 5.1.1-5.1.3.

Chapter 4

Theory and Implementation for DRPGO-Enabled Radiological Mapping

4.1 System Architecture

With their ability to traverse arbitrary terrain, UAS provide an ideal platform for performing radiological mapping, which entails covering dangerous areas containing one or more radiation sources of unknown location and magnitude in a time-sensitive manner. Previous attempts at remote sensing for radiological search and mapping with air vehicles have had success with single-agent architectures, such as the Localization and Mapping Platform (LAMP) [35]. There is currently interest in improving the coverage ability of platforms like LAMP by expanding to multiple agents. Moreover, amid the target applications of first-response, facilities surveillance, and disaster prevention, it is not always reasonable to assume that absolute sensing such as GPS will be available or of sufficient quality to reliably localize all agents during the mapping process. The requirements for radiological mapping thus entail a multi-agent architecture capable of joint 3D mapping and radiological sensing.

Possible multi-agent radiological mapping architectures are shown in Fig. 4-1. It

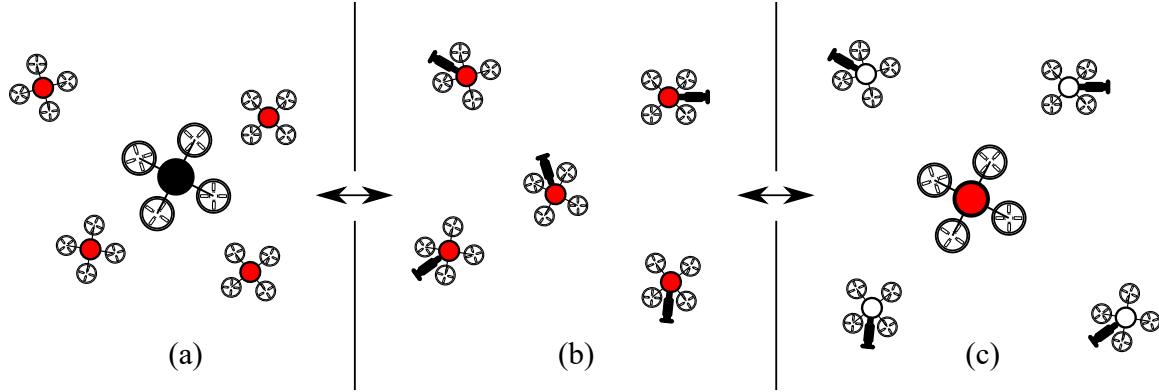


Figure 4-1: Possible multi-agent UAS architectures for radiological mapping. *Left:* A single, larger UAS generates 3D maps with LiDAR and several smaller UAS exclusively relay radiological sensing data to the central drone. *Middle:* Several smaller UAS equipped with radiological and RGBD sensors relay radiation and 3D map data either to each other or to a central server. *Right:* A single, larger UAS measures radiation data, while several smaller UAS exclusively focus on constructing a 3D map to be shared with the group.

is conceivable that in a multi-agent setting, a high-resolution sensor such as LiDAR could provide the needed 3D mapping, though its very high weight and cost would likely limit its ability to be placed on multiple vehicles, leading to dependence on the coverage ability of a single agent and a single failure point (Fig. 4-1a). This work aims to explore alternative architectures (Fig. 4-1b-c) and expand remote sensing capabilities for radiological source localization by laying the groundwork for a multi-agent collaborative SLAM solution utilizing lightweight sensing and a sparse communication network. The decentralization of perception capability facilitates greater robustness, more scalability, more rapid and widespread mapping ability, and applicability to a larger variety of real-world environments.

For the system architectures pictured in Fig. 4-1b-c, the CSLAM-enabled UAS agents are posited to be equipped with color and depth (RGBD) cameras as well as UWB ranging modules. The RGBD cameras allow for the agents to estimate their ego-motion, e.g., using visual inertial odometry (VIO), which forms the backbone for a distributed pose graph optimization solution. The UWB modules facilitate not only inter-agent ranging but also ad hoc inter-agent networking for sharing the minimum required information to obtain globally accurate trajectory estimates of the swarm.

Under the proposed mapping architecture, radiological mapping can be performed in a greater variety of environments, including cluttered and indoor spaces.

4.2 Distributed Pose Graph Optimization for Radiological Mapping

Because CSLAM methods rely on globally consistent multi-agent trajectory estimation to construct an accurate 3D map [67], this work focuses on increasing the robustness and lowering the communication requirements for distributed PGO. A brief explanation of PGO is presented here to give context to the DRPGO formulation.

Conceptually, PGO can be thought of as a maximum likelihood optimization over a Bayes net (Fig. 4-2), which incorporates a chronological sequence of state beliefs $P(X_k = \mathbf{x}_k)$ and conditional probability distributions from state transition models $P(X_k|X_{k-1} = \mathbf{x}_{k-1})$ and measurement models $P(\mathbf{x}_k|Y_k = \mathbf{y}_k)$ to maximize the joint probability distribution

$$\max_{\mathbf{x}_k \forall k \in 0, \dots, N} P(X_k, \dots | Y_k = \mathbf{y}_k, \dots, X_{k-1} = \mathbf{x}_{k-1}, \dots) \quad (4.1)$$

in a process known as smoothing. If all probability distributions in the smoothing problem are assumed to be Gaussian, then Eq. 4.1 can be reformulated as the unconstrained optimization problem

$$\min_{\mathbf{x}_k \forall k \in 0, \dots, N} \sum_{k=0}^N (\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}))^\top \mathbf{Q}_k (\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})) \quad (4.2)$$

$$+ (\mathbf{y}_k - h(\mathbf{x}_k))^\top \mathbf{R}_k (\mathbf{y}_k - h(\mathbf{x}_k)), \quad (4.3)$$

where $f(\cdot)$ and \mathbf{Q}_k represent the state transition model and covariance, and $h(\cdot)$ and \mathbf{R}_k represent the measurement model and covariance.

In the classical PGO problem, all states are 6-DOF poses:

$$\mathbf{x}_k = \mathbf{T}_k \in SE(3), \quad (4.4)$$

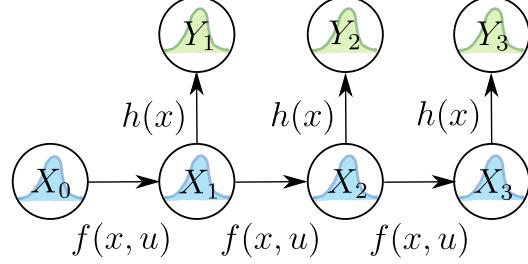


Figure 4-2: Dynamic Bayes net representation. The nodes in the graph represent random variables, corresponding either to robot states X_k or measurements Y_k through time. The edges in the graph represent conditional probability relationships governed by either a measurement model $h(\cdot)$ or state transition model $f(\cdot)$.

and only the state transition models $f(\cdot)$ are utilized, taking the form

$$\mathbf{T}_k = f(\mathbf{T}_{k-1}, \mathbf{T}_{k-1,k}) = \mathbf{T}_{k-1} \mathbf{T}_{k-1,k}, \quad (4.5)$$

where $\mathbf{T}_{k-1,k}$ is a measured relative pose coming from e.g., VIO. In addition to chronologically consecutive relative poses, temporally disjoint relative poses can also be incorporated into the optimization, as with the relative pose measurements provided by loop closures. Because of this, the PGO problem with N poses is expressed with the general formulation

$$\min_{\mathbf{T}_k \forall k \in 0, \dots, N} \sum_{(i,j) \in \mathcal{E}} ((\mathbf{T}_j^{-1} \mathbf{T}_i) \boxminus \mathbf{T}_{i,j})^\top \mathbf{Q} ((\mathbf{T}_j^{-1} \mathbf{T}_i) \boxminus \mathbf{T}_{i,j}), \quad (4.6)$$

where the edges in the pose graph $(i,j) \in \mathcal{E}$ and corresponding relative pose measurements $\mathbf{T}_{i,j}$ are not constrained to the strictly chronological structure shown in Fig. 4-2. In general, chronologically consecutive pose graph edges are constructed with VIO, and non-consecutive edges are constructed with loop closure measurements.

The PGO problem (Eq. 4.6) is generally solved by a variation on Gauss-Newton local search, which relies on computing the Jacobian of the problem residuals

$$r(\mathbf{T}_i, \mathbf{T}_j, \mathbf{T}_{i,j}) \triangleq (\mathbf{T}_j^{-1} \mathbf{T}_i) \boxminus \mathbf{T}_{i,j} \in \mathbb{R}^6, \quad (4.7)$$

which evolves on a manifold, illustrated in Fig. 4-3 with $\mathcal{M} = SE(3)$. Because the

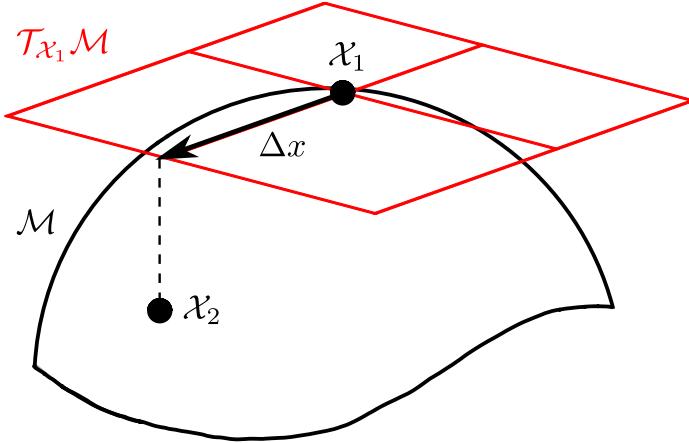


Figure 4-3: Illustration of the operation of adding the tangent space vector $\Delta x \in T_{\mathcal{X}_1}\mathcal{M}$ to the manifold quantity \mathcal{X}_1 to obtain the new quantity $\mathcal{X}_2 = \mathcal{X}_1 \boxplus \Delta x$ on the manifold. \mathcal{X}_1 does *not* live in a vector space, but the incremental quantity Δx lives in a tangent vector space which is tangent to \mathcal{M} precisely at \mathcal{X}_1 , and thus is still a vector.

residuals evolve on a manifold, their Jacobians are calculated using the retraction \boxplus over decision variables $\mathbf{T} = (\mathbf{T}_i, \mathbf{T}_j) \in SE(3) \times SE(3)$:

$$\frac{\mathbf{T}^T \partial r(\mathbf{T})}{\partial \mathbf{T}} \triangleq \lim_{\boldsymbol{\tau} \rightarrow 0} \frac{r(\mathbf{T} \boxplus \boldsymbol{\tau}) - r(\mathbf{T})}{\boldsymbol{\tau}} \in \mathbb{R}^{6 \times 12}, \quad (4.8)$$

$$\boldsymbol{\tau} \in \mathbb{R}^{12} \cong \mathfrak{se}(3) \times \mathfrak{se}(3), \quad (4.9)$$

$$(\mathbf{T} \boxplus \boldsymbol{\tau}) \in SE(3) \times SE(3). \quad (4.10)$$

To solve the PGO in a distributed setting, each agent constructs its own pose graph, which includes select shared poses with other agents only if an inter-agent loop closure is found. Otherwise, the process of incorporating VIO and intra-agent loop closures to iteratively solve for the most likely sequence of poses \mathbf{T}_k resembles the single-agent PGO problem. When poses are shared among agents, care must be taken to ensure that each received neighbor pose is inserted into the graph at the correct point in time. One way to accomplish this is to ensure that all agents' clocks are synchronized, and that VIO-derived poses are only inserted into the graph at pre-determined and consistent intervals. With the synchronous method, only time delay associated with inter-agent communication needs to be taken into account when incorporating inter-agent measurements.

4.3 Distributed Range-Enhanced Pose Graph Optimization (DRPGO)

In order to increase the robustness in terms of global accuracy as well as communication requirements of distributed PGO, an additional UWB range sensor is incorporated into the problem. Its measurement model is

$$h(\mathbf{T}_i, \mathbf{T}_j) = \|\mathbf{t}_i - \mathbf{t}_j\|_2, \quad (4.11)$$

where $\mathbf{t} \in \mathbb{R}^3$ is the translational component of $\mathbf{T} \in SE(3)$. The UWB range measurements are incorporated into the pose graph as inter-agent measurements corresponding to two agents' poses at the same point in time. Similar to Eq. 4.7, the range measurement residual is constructed using Eq. 4.11 as

$$r(\mathbf{T}_i, \mathbf{T}_j, d_{i,j}) \triangleq \|\mathbf{t}_i - \mathbf{t}_j\|_2 - d_{i,j} \in \mathbb{R}. \quad (4.12)$$

With the incorporation of the range measurement residuals, Eq. 4.6 becomes

$$\min_{\mathbf{T}_k \forall k \in 0, \dots, N} \sum_{(i,j) \in \mathcal{E}_{\text{pose}}} \|(\mathbf{T}_j^{-1} \mathbf{T}_i) \boxminus \mathbf{T}_{i,j}\|_{\mathbf{Q}}^2 + \sum_{(i,j) \in \mathcal{E}_{\text{range}}} \|\|\mathbf{t}_i - \mathbf{t}_j\|_2 - d_{i,j}\|_{\mathbf{Q}}^2, \quad (4.13)$$

where $d_{i,j} \in \mathbb{R}$ is the measured range between pose i and pose j in the graph.

In DRPGO, the UWB range measurements take the place of inter-agent loop closure detections. An advantage of this is that significantly less information needs to be shared between agents, as the entire distributed data association problem is circumvented. The cost of this diminished communication requirement is that many more inter-agent range measurements are needed to improve the accuracy of the global pose graph compared to a single inter-agent loop closure detection. According to the observability analysis performed in [9], computing relative position between two agents ranging off of each other requires that the actual relative position vectors corresponding to the range measurements consist of at least two mutually exclusive subsets, each containing three linearly independent vectors. What this suggests is

that the agents should be engaged in constant, non-planar relative motion if the full 3D relative position estimate is to be improved. It also requires that relative range measurements be shared at a relatively high rate compared to the hypothetical frequency of shared loop closure measurements, which is in general a safe assumption.

An important caveat to the observability analysis in [9] is that it applies only to *relative* position accuracy between agents, and not global (or absolute) accuracy. While this condition often suffices for operations such as formation flying and collision avoidance, it is not acceptable for a CSLAM application, in which the goal is to construct a globally accurate and consistent map of an environment. VIO measurements suffer from this same problem in classical PGO and SLAM since they are relative measurements being used in a problem that aims for global accuracy. At the single-agent scale, the relative VIO measurement problem is usually solved through the use of loop closures, which “anchor” the entire pose graph in a concrete fashion by identifying locations that have been visited before. The corrective loop closure measurements are thus able to correct for the drifting VIO estimate to achieve accuracy in a global, or absolute, sense.

Similarly, at the multi-agent level, inter-agent range measurements prevent relative drift between individual agents’ VIO-based trajectory estimates, but they cannot prevent the entire map from drifting with respect to the global frame. Stated differently, the range measurements anchor the agents’ trajectories to each other, but not to the global map.

The proposed solution to the range anchoring problem is to distinguish two types of agent behavior under the DRPGO scheme: explore and anchor. The terminology is inspired by a central challenge in the active SLAM literature, which is finding the optimal balance between exploration (i.e., building a map as quickly and efficiently as possible) and exploitation (i.e., seek to revisit locations to find loop closures). With the explore/anchor strategy, an explorer agent focuses unilaterally on expanding the map by traversing previously unvisited areas, and an anchor agent focuses predominantly on staying within a certain vicinity to obtain *intra-agent* loop closure measurements.

The presence of high-rate inter-agent range measurements in DRPGO affords a simple method for assigning anchor agents. Returning to the anchor analogy, if any anchor at all is provided between the swarm and the global map, then any agent tethered to that anchor will also be anchored to the global map. Thus, if a *communication sub-graph* is defined as a connected graph between agents where each edge is an inter-agent range measurement, then only one agent within that sub-graph needs to be assigned as an anchor; all other agents can focus on exploration. This methodology reduces the problem of optimizing between exploration and exploitation in a multi-agent setting to the problem of ensuring that the global communication graph never becomes completely disjoint, where no agents are able to communicate with any other agent. Moreover, it simplifies the active SLAM problem while simultaneously drastically reducing the necessary bandwidth of inter-agent communication by side-stepping the need for distributed data association. Algorithm 1 gives an overview of the anchor assignment problem for a single agent in DRPGO, which includes identifying current communication sub-graphs and picking an agent to act as the anchor for that sub-graph. It provides a simplistic, minimalist implementation where the agent with the smallest ID in a sub-graph is assigned as the anchor, and can be replaced with more sophisticated consensus strategies for robustness. The intuition behind the anchor assignment strategy is corroborated by the presented simulation experiment results in Sections 5.2.1-5.2.2.

Ensuring that there is always an anchor agent within each communication sub-graph ensures that, as the sub-graph topologies evolve in an ad hoc fashion, global accuracy and consistency is maintained. The entire DRPGO strategy for improved CSLAM, then, entails anchor assignment and incremental distributed, time-synchronized pose graph optimization with inter-agent range measurements. Algorithm 2 provides the pseudo-code for a single agent running DRPGO, assuming that all agent clocks are synchronized. The sub-routine SeekLC stands in as a place-holder for executing a flight pattern that limits navigation to the already-constructed map, seeking locations known by the trajectory history to have already been visited. Likewise, the sub-routine Explore steers navigation towards the frontier of the currently-known

Algorithm 1 DRPGO Synchronous Anchor Agent Selection

Input: ID of agent ν .

Input: Anchor establishment interval Δt_{ai} .

Input: Anchor establishment window Δt_{aw} .

Output: ID of current anchor ι .

```
1: Initialize connected index set  $\Phi = \{\nu\}$ .
2: while True, Executed every  $\Delta t_{ai} + \Delta t_{aw}$  do
3:   while  $t \leq \Delta t_{aw}$  do
4:     Receive connected index list from neighbor agents  $\mathcal{I}$  asynchronously.
5:      $\Phi = \Phi \cup \mathcal{I}$ .
6:     Publish  $\Phi$  to neighbor agents.
7:   end while
8:    $\iota \leftarrow \min(\Phi)$ .
9: end while
```

map.

4.4 DRPGO-CSLAM Dataset Generation

To help validate the claims associated with DRPGO’s global accuracy and to assess its suitability for use in realistic scenarios with multiple UAS, DRPGO-CSLAM datasets are generated from several different photorealistic simulated environments and one real-world environment. Specifically, the datasets consist of the following time-stamped data:

- IMU accelerometer and gyroscopic measurements
- RGB images
- VIO pose estimates
- Altimeter measurements, calculated from ground truth with added noise
- Inter-agent UWB range measurements, calculated from ground truth with added noise
- Intra-agent loop closure detections
- Inter-agent loop closure detections

Algorithm 2 DRPGO

Input: ID of agent ν .
Input: Optimization interval Δt .
Input: Anchor index ι , supplied by Alg. 1 on a separate thread.

- 1: **Initialize** DRPGO pose graph \mathcal{P} (Eq. 4.13).
- 2: **while** True, Executed every Δt **do**
- 3: Extract VIO relative pose \mathbf{T}_{VIO} .
- 4: Add pose to graph $\mathcal{P} \leftarrow \mathbf{T}_{\text{VIO}}$ (Eq. 4.7).
- 5: Send range requests to neighbor agents, along with most recent pose $\mathbf{T}_f \in \mathcal{P}$.
- 6: **if** $\nu = \iota$ **then**
- 7: SeekLC().
- 8: **if** Loop closure found **then**
- 9: Add loop closure to graph $\mathcal{P} \leftarrow \mathbf{T}_{\text{LC}}$ (Eq. 4.7).
- 10: Optimize \mathcal{P} .
- 11: **end if**
- 12: **else**
- 13: Explore().
- 14: Add received ranges and neighbor poses to graph $\mathcal{P} \leftarrow d_{i,j}, \mathbf{T}_j$ (Eq. 4.12, 4.7).
- 15: Optimize \mathcal{P} .
- 16: **end if**
- 17: **end while**

and are derived from:

- The EuRoC MAV Dataset [6] (3 agents)
- AirSim [45] (4 agents)
- Flightmare [50] (4 agents)

The AirSim (Fig. 4-4) and Flightmare (Fig. 4-5) simulators are chosen for the photorealism of their environments as well as their suite of implemented sensors. AirSim distinguishes itself with the large amount of unique environments that are made publicly available, built using the popular Unreal Engine. Flightmare, built on Unity, has the advantage of allowing the user to easily spawn custom objects in one of their pre-built environments, also allowing for a large degree of configurability. Both simulators provide built-in RGBD sensing for their unmanned air vehicles, and although the depth component is not utilized in this work, it can potentially be

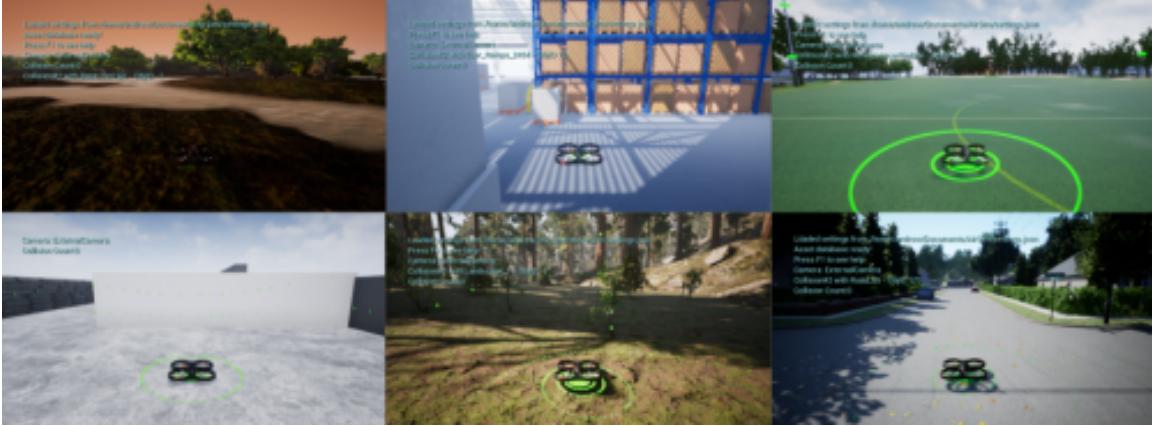


Figure 4-4: Sample environments from AirSim, an Unreal Engine-based simulator, used to generate datasets for PGO experiments.



Figure 4-5: Sample images from Flightmare, a Unity-based simulator that provides RGBD sensing and UAS dynamics simulation, used to generate datasets for PGO experiments.

useful for further study of the integration of DRPGO into a full-fledged CSLAM system. Datasets from feature-rich, feature-sparse, open, and cluttered environments are generated using the available configurations of the two simulators.

Because the DRPGO evaluations focus exclusively on pose graph optimization, the outputs of the other elements of CSLAM are added to the datasets as part of the dataset pre-processing. Figure 4-6 illustrates this process. Raw RGB, IMU, and truth data are obtained from the AirSim and Flightmare in a variety of simulated environmental conditions. For each dataset, the raw data is then processed by VINS-Mono [39] to obtain the VIO measurements for each agent and by CCM-SLAM [44] to obtain the intra- and inter-agent loop closure detections. UWB range measurements are generated to coincide with each VIO measurement, and are generated by adding zero-mean Gaussian noise with a standard deviation of 10 centimeters to the true inter-agent ranges.

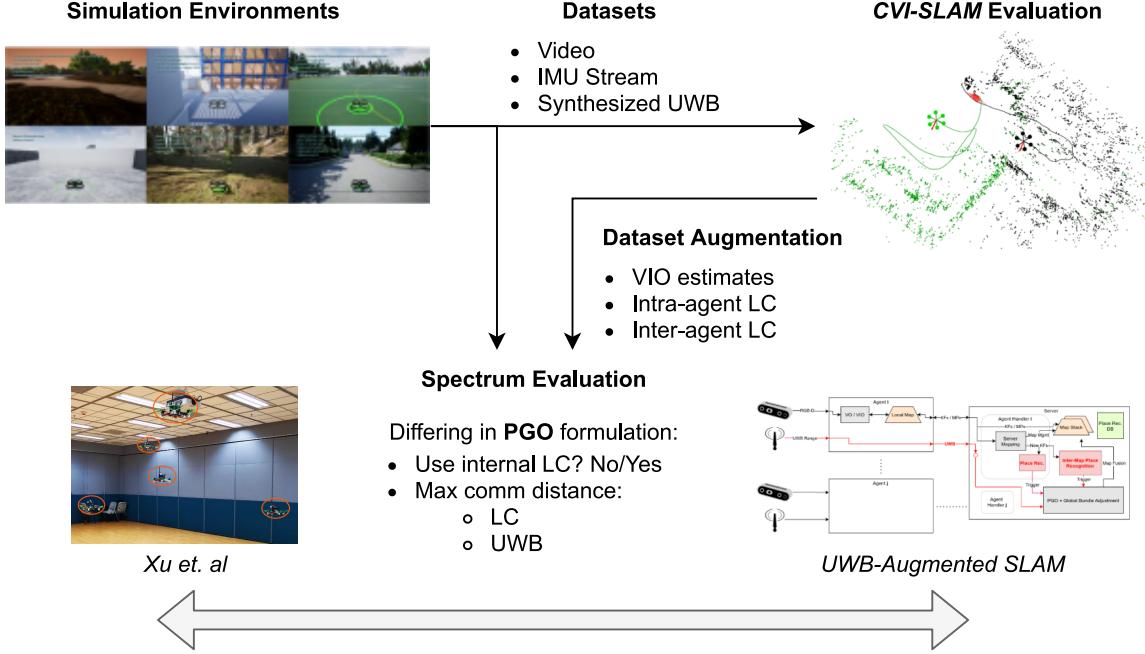


Figure 4-6: Visualization of the PGO dataset generation process. Raw sensor data is generated from either AirSim or Flightmare and subsequently processed by a centralized SLAM algorithm to generate VIO and intra-/inter-agent loop closure measurements. These measurements, along with synthesized UWB range measurements, are processed by a PGO evaluation scheme that allows for giving more importance to either loop closure or range measurements on a sliding scale.

Once the datasets are generated, they are each used to solve the distributed version of Eq. 4.13 (see Section 4.6) with configurable solver covariances \mathbf{Q} for the loop closure and range measurements. Making \mathbf{Q} configurable for the different measurement types allows for insights to be made about the relative importance of loop closure versus range measurements as they are allocated more or less importance in the solver. An analysis of these results, given in Section 5.2.1, informs the intuitions behind the value of UWB range measurements and having an anchor assignment strategy for the global accuracy of DRPGO.

4.5 Simulation for DRPGO Prototyping

After utilizing the DRPGO-CSLAM datasets to draw conclusions about the relative importance of loop closures and UWB range to the global accuracy of DRPGO, the

full DRPGO algorithm with anchor selection (Algorithm 2) is developed and tested in a dynamics-only Python simulation environment designed for rapid prototyping.

In the simulation, individual agent dynamics are run as PID-based trajectory following of a bounded random walk over the state space defined on $SE(3)$. For communication sub-graph identification and anchor agent assignment, Algorithm 1 is replaced by a breadth-first-search over an adjacency matrix constructed at each time step using the maximum communication distance parameter, outputting the communication sub-graphs and allowing for a random anchor assignment for each new sub-graph in the same fashion as with Algorithm 1.

Inter-agent sharing of range and select pose information occurs synchronously to ensure that the measurement nodes are inserted into the correct index of each agent’s pose graph. Commensurate with the procedure outlined in Algorithm 2, each explorer agent optimizes its pose graph with the range measurements synchronously, whereas each anchor agent does not incorporate the inter-agent range measurements, but only shares them along with the requested neighbor poses. Instead, an anchor agent focuses on adding intra-agent loop closures (i.e., loop closure with its own trajectory) to its own pose graph to anchor its own global accuracy, which its neighbors in the communication graph depend on.

The simulation environment allows for configuration of the individual measurement noise levels, environment size, number of agents, and max communication range. Its emphasis on the core, distributed functionality of the DRPGO algorithm and not on details such as clock synchronization allows for proof-of-concept rapid prototyping only. That said, its level of fidelity does allow for conclusions to be drawn about the computational feasibility and global accuracy/consistency expectations for DRPGO, to be corroborated by more in-depth hardware testing.

4.6 DRPGO Solver Implementation

The DRPGO solver is written using the Ceres Solver C++ library [2], which is meant for solving large-scale nonlinear least-squares problems such as Eq. 4.13. One of

Ceres Solver’s greatest strengths is an efficient auto-differentiation implementation, which suggests that Eq. 4.8 is solved “for free” as long as the user specifies a definition for \boxplus pertaining to the problem’s decision variables that live in $SE(3)$. The definition for \boxplus is provided to Ceres using the Sophus library [52], which provides C++ implementations of $SO(3)$ and $SE(3)$. The residual definitions (Eq. 4.7 and 4.12) are also provided to Ceres using Sophus and Eigen [18].

To interface the DRPGO implementation with the simulation environment outlined in Section 4.5, Python wrappers are written for the solver, $SO(3)$, $SE(3)$, and residual implementations.

4.7 Summary

This chapter presents a systems-level analysis of the 3D mapping problem for radiological search and mapping. A discussion of distributed pose graph optimization and its applicability to radiological mapping is given, leading into the presentation of a novel method for incorporating inter-agent range measurements into the pose graph. Datasets and a simulation environment are developed for testing the global accuracy of the presented DRPGO algorithm, whose implementation details are also given. The testing results from the dataset and simulation environment are given in Sections 5.2.1 and 5.2.2, respectively.

Chapter 5

Testing and Results

5.1 Air Wake Measurement Tests

5.1.1 Simulation Tests

Simulation-based results are provided that validate the performance of the absolute state estimator, the relative state estimator, and the autonomous control strategy in a simulated windy environment with implemented sensor noise and bias. Specifically, the results in this section correspond to simulation tests run in the environment presented in Section 3.1 with relative headwind speeds in the vicinity of 10 meters per second, which is in the upper range of expected average relative wind speeds to be experienced during tethered flight behind a ship. The simulated flights from which these results are extracted carry out the tasks of takeoff, establishing tether tension, and sweeping behind a moving ship at a commanded altitude, exciting all of the different flight modes of the UAS for trajectory tracking performance evaluation.

The performance of the absolute state estimator is evaluated on the basis of how well the states used in the flight control feedback loop conform to the true state values provided by the simulator. These feedback states consist the UAS altitude, body-frame lateral velocities, attitude, and angular rates.

Both outdoor and simulation testing demonstrates that while GPS provides less drift-prone velocity than position estimates, the barometer provides sufficient, al-

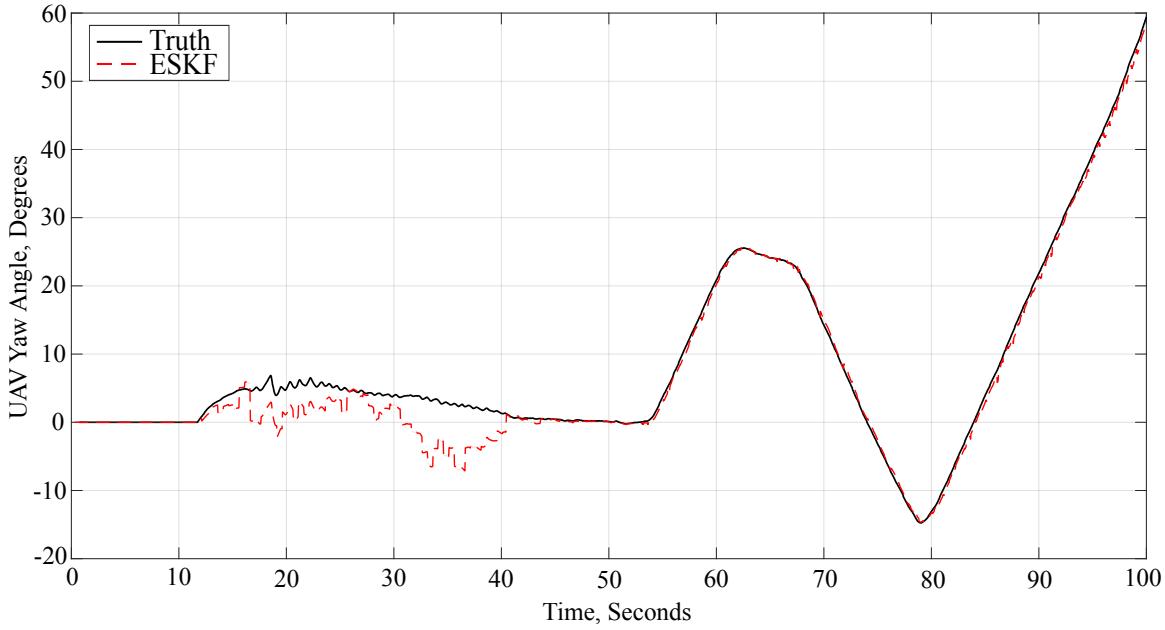


Figure 5-1: Absolute state estimate of UAS yaw with respect to the inertial frame over the course of a simulated ship air wake measurement flight. Attitude corrections in the estimate are activated once the IR beacon array comes into view 40 s into the simulation.

beit noisy, altitude stability to ensure that the absolute altitude estimate does not drift beyond approximately one meter from the true altitude. A similar effect, well-documented in small UAS applications, occurs with the estimated UAS attitude as the IMU gyro biases are corrected by the less drift-prone accelerometer measurements. The attitude component most susceptible to drift over very short time scales is the yaw angle, which relies principally on gyro signal integration in the absence of corrective vision-based attitude measurements from the IR beacon array. Figure 5-1 demonstrates the amount of drift exhibited by the yaw angle estimate until the IR beacon array comes into full view of the onboard camera 40 seconds into the simulation, subsequently providing very precise attitude estimates. Moreover, it is important to note that while no active yaw control is employed in the flight control scheme, a comparison between Fig. 5-1 and 5-2 reveals the passive yaw control effect of a taut tether, which maintains the UAS pointing towards the ship deck.

The performance of the relative state estimator is based principally on the accuracy of the estimated relative pose. Accurate estimated velocity states are also

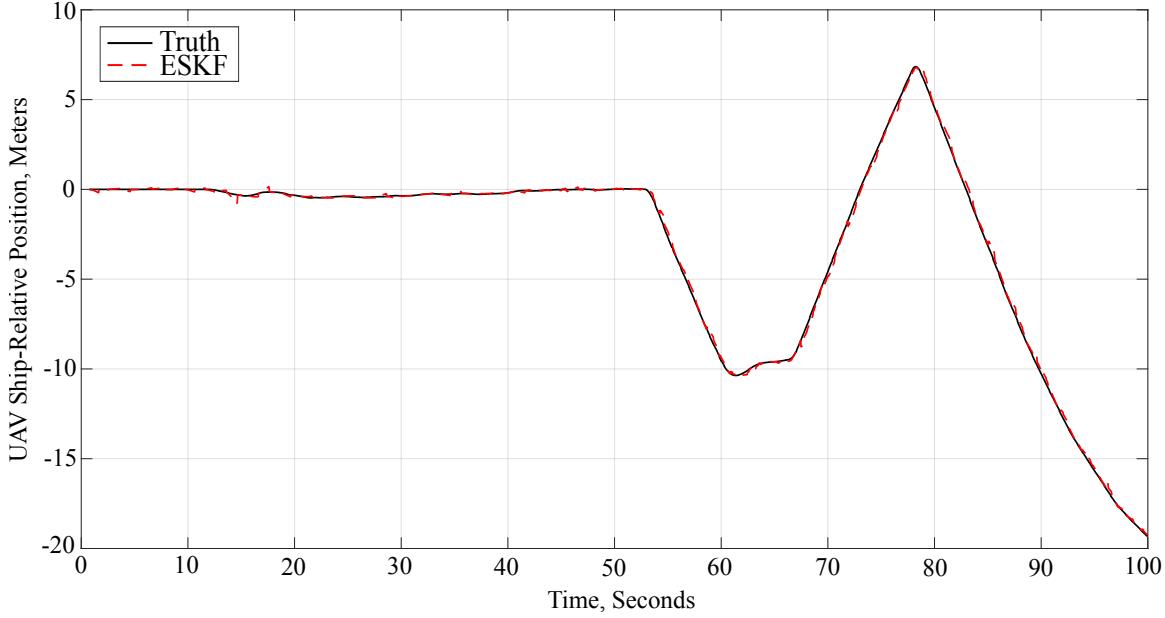


Figure 5-2: Relative state estimate of UAS y position w.r.t. and expressed in S over the course of a simulated ship air wake measurement flight. Accurate relative state position estimation is facilitated by the fusion of differential GPS and, after 40 s into the simulation, vision-based pose measurements.

important for propagating the pose state in the event of differential GPS or vision measurement dropouts as well as compensating for relative UAS velocity when post-processing the measured wind velocity vectors. The relative estimator fuses the same sensors as the absolute estimator for velocity estimation, but with the added benefit of centimeter-level accuracy in relative position and single-degree-level accuracy in attitude from the differential GPS and vision-based pose sensors. Moreover, when the estimate of \mathbf{q}_I^S converges to a steady-state value after the IR beacon array comes into view, the full relative state estimate is found to be very stable and accurate, which is to be expected given the accuracy of the relative sensors. Figure 5-2 provides insight into both the accuracy of the relative position estimate as well as the stability and coverage of the sweeping patterns necessary for generating an informative vector field of measured air wake velocities.

With stable absolute and relative state estimates, the success of the autonomous control strategy is primarily a function of how well the controller is able to track the commanded absolute UAS pitch, lateral velocity, and altitude once the passively

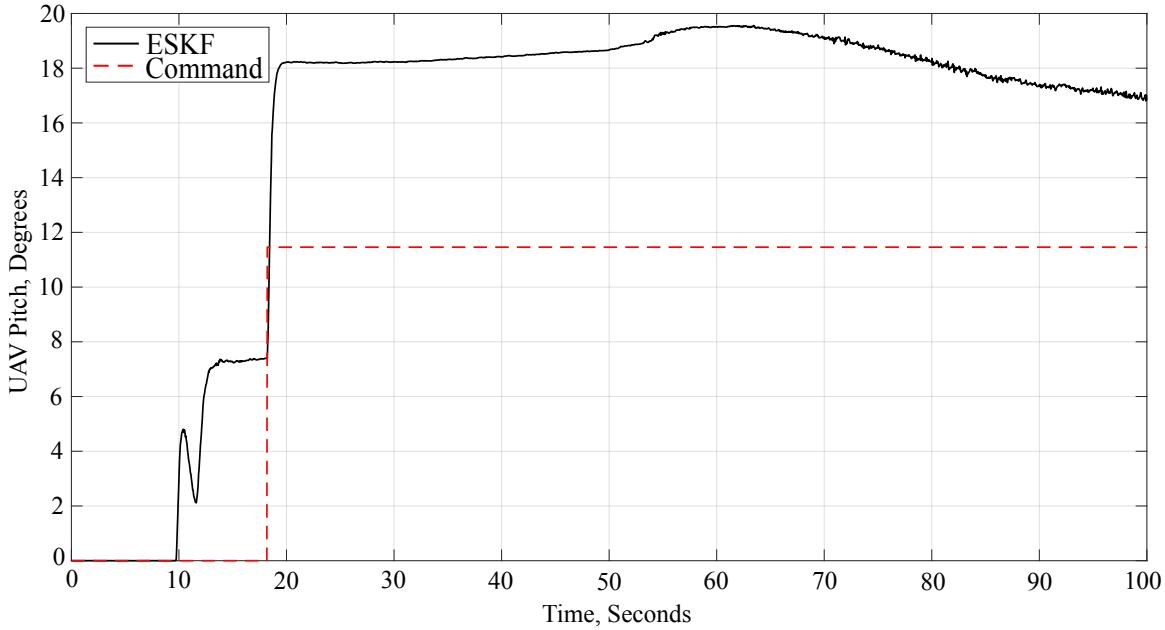


Figure 5-3: UAS pitch tracking performance throughout a simulated ship air wake measurement flight. A steady-state pitch disturbance of $\approx 7^\circ$ is induced by the rotor blade flapping effect at a relative wind speed of 10 m/s. Over all simulation trials, the disturbance is bounded and does not have any destabilizing effect on the flight trajectory due to the tether constraint.

stabilizing effect of the tether is verified. Figure 5-3 demonstrates the pitch tracking performance of the control strategy at a relative wind speed of 10 meters per second. Due to the rotor blade flapping effect, the relative headwind induces a steady-state pitch disturbance of approximately seven degrees. Through testing at various wind speed levels, it is observed that the wind-induced pitch disturbance can be expected to range from zero to ten degrees. Because the pitch disturbance is bounded and has no appreciable effect on flight stability due to the ability of the tether reel controller to maintain a relatively taut tether, no attempt is made to place integral control on pitch (which can introduce low-frequency oscillations). Additionally, no effort is made to estimate the wind vector in real time for computing feed-forward pitch control.

Though integral control is avoided in most cases, it is used in lateral velocity tracking in order to achieve the desired area coverage in the face of increasing aerodynamic drag as the UAS sweep angle increases. Because the lateral velocity vector is defined to be perpendicular to the tether, the tether does not lead to integrator

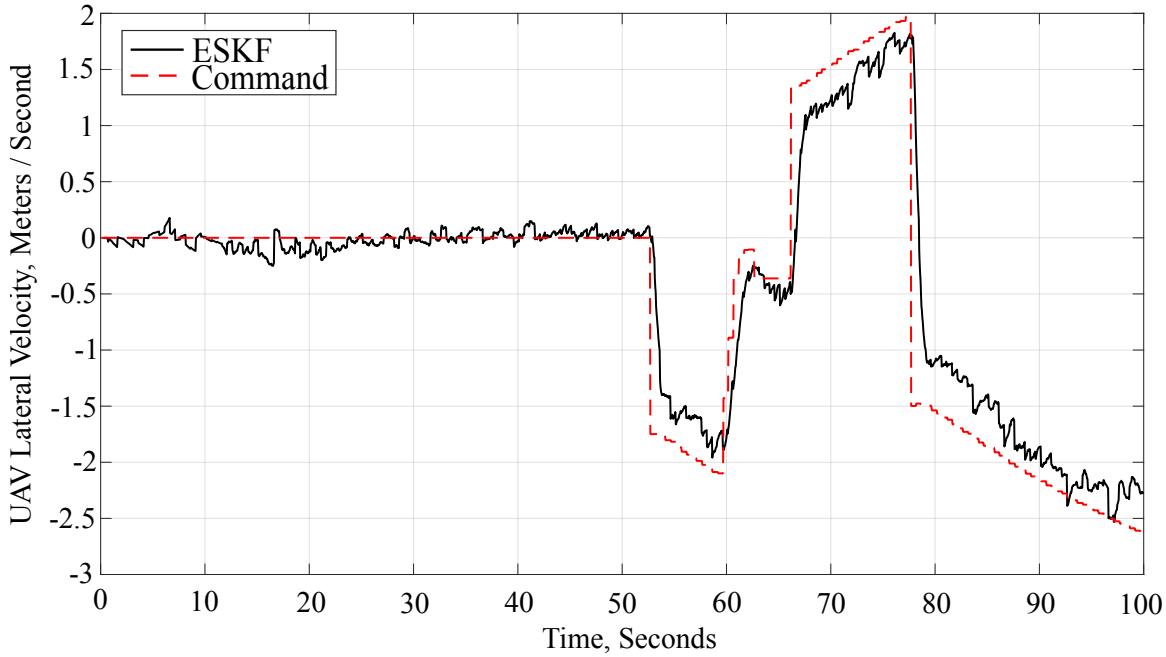


Figure 5-4: UAS lateral velocity tracking performance over the course of a simulated ship air wake measurement flight. Aerodynamic drag effects are visible after 50 s, when the UAS begins to fly out from directly behind the moving ship. Despite these drag effects, velocity tracking is observed to be tractable with PID control.

windup as would be the case with a longitudinal or position-based reference state command. Figure 5-4 demonstrates the ability of the controller to track reference lateral velocities such that the UAS is able to sweep angles up to 50-60 degrees even in the presence of heavy wind. Additionally, Fig. 5-5 provides a top-down view of one such sweeping trajectory, showing how the UAS is able to cover the region where air wake effects (defined by wind direction with respect to the ship superstructure) are present.

It is apparent from analyzing tracking performance that the greatest difficulty comes in tracking altitude, as shown in Fig. 5-6. Comparing the ESKF altitude estimate with the reference altitude command, the state exhibits a random walk centered on the commanded value. This behavior is attributable to both the rotor blade flapping effect as well as the fact that altitude controllability is weakened with a sustained non-zero aircraft pitch value. The altitude error, however, remains bounded at around half a meter over the course of all simulation trial flights with different wind speeds. This level of error is dealt with for landing operations by allowing the tether

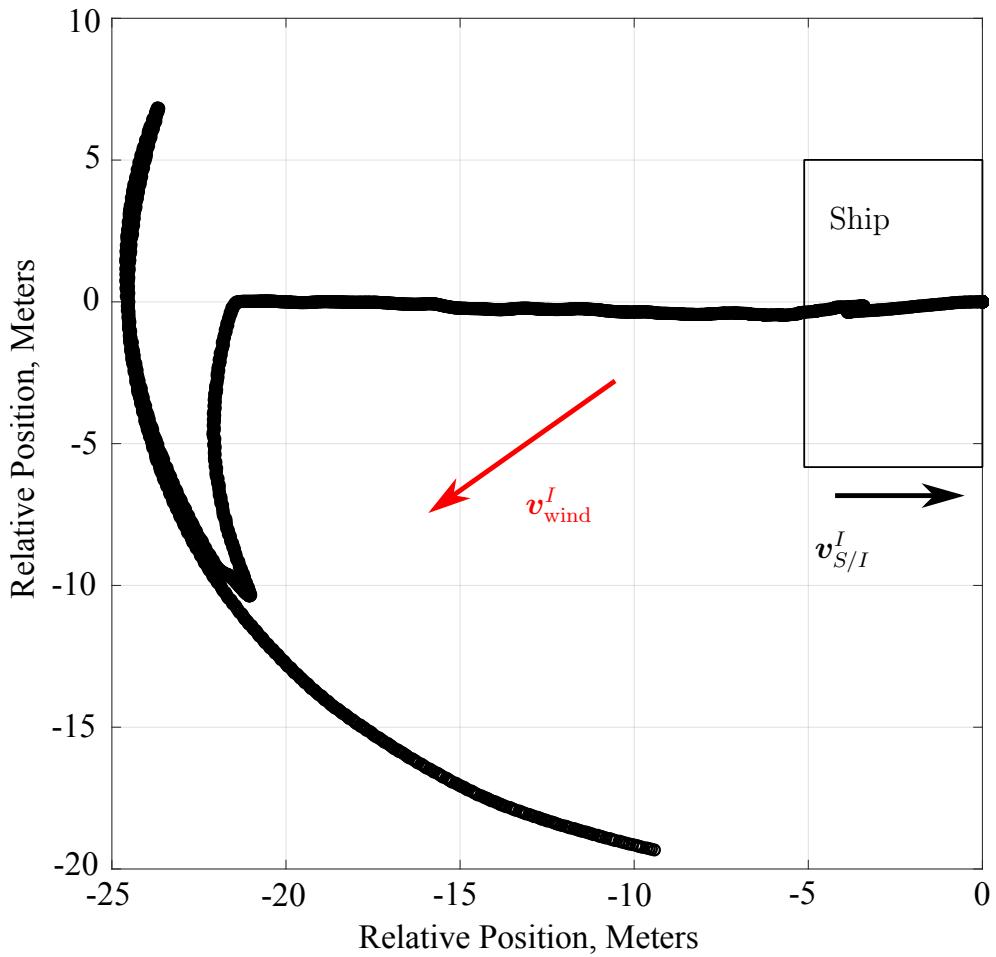


Figure 5-5: Top view of a sweep trajectory from a simulated UAS flight in the presence of 10 m/s wind. Position values expressed in the S frame. The commanded trajectory is tailored to the principal wind direction, providing symmetric coverage about the air wake generated by the wind's interaction with the ship superstructure. In order for a symmetric sweep like the one shown, the estimated wind direction must be measured and reported to the trajectory generator before flight.

to “reel in” the aircraft in the absence of integral control on altitude.

In addition to reference state tracking, the ability to automatically deduce tether tautness is important for the trajectory generator to determine when it is feasible to begin sweeping with pitch and lateral velocity commands. For simulated testing, $f_{\text{ext,thresh}}^B$ is assigned the amount of backward thrust generated with a pitch value of 10 degrees, since this would correspond to the amount of tension in a taut tether in the absence of wind and at an altitude that is level with the tether reel controller. As suggested by Fig. 5-7, this choice of $f_{\text{ext,thresh}}^B$ leaves enough buffer to make a binary

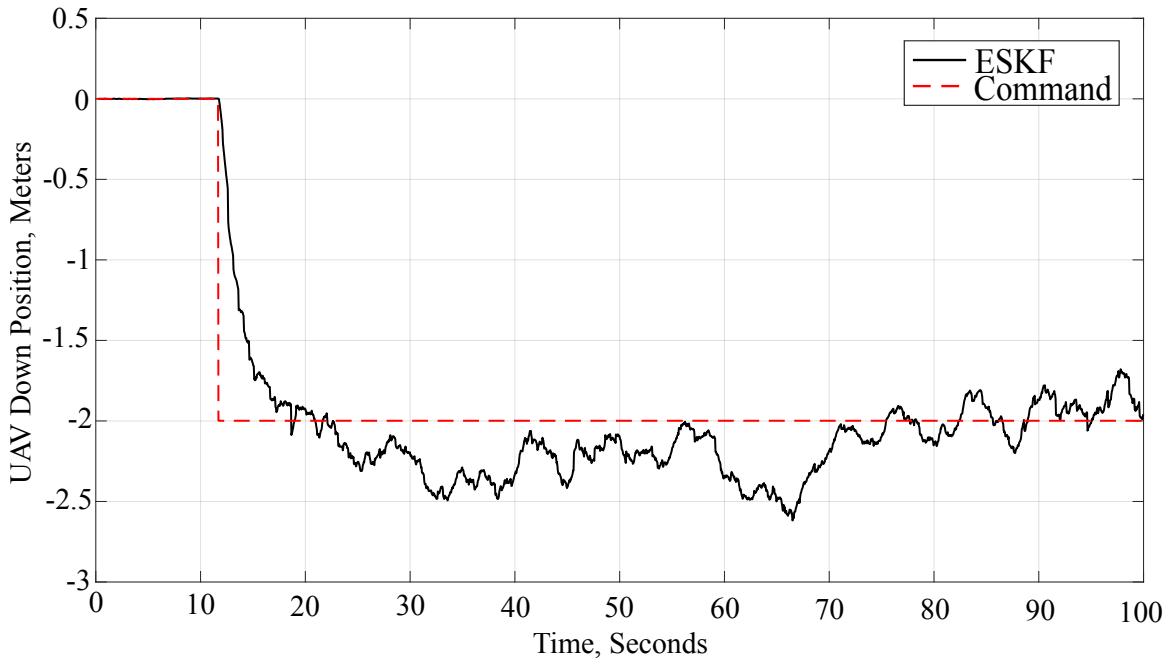


Figure 5-6: UAS absolute altitude tracking performance over the course of a simulated ship air wake measurement flight. Precise absolute altitude tracking poses the greatest difficulty for the flight control system, with bounded random walk behavior about the commanded altitude due to drag and tether effects.

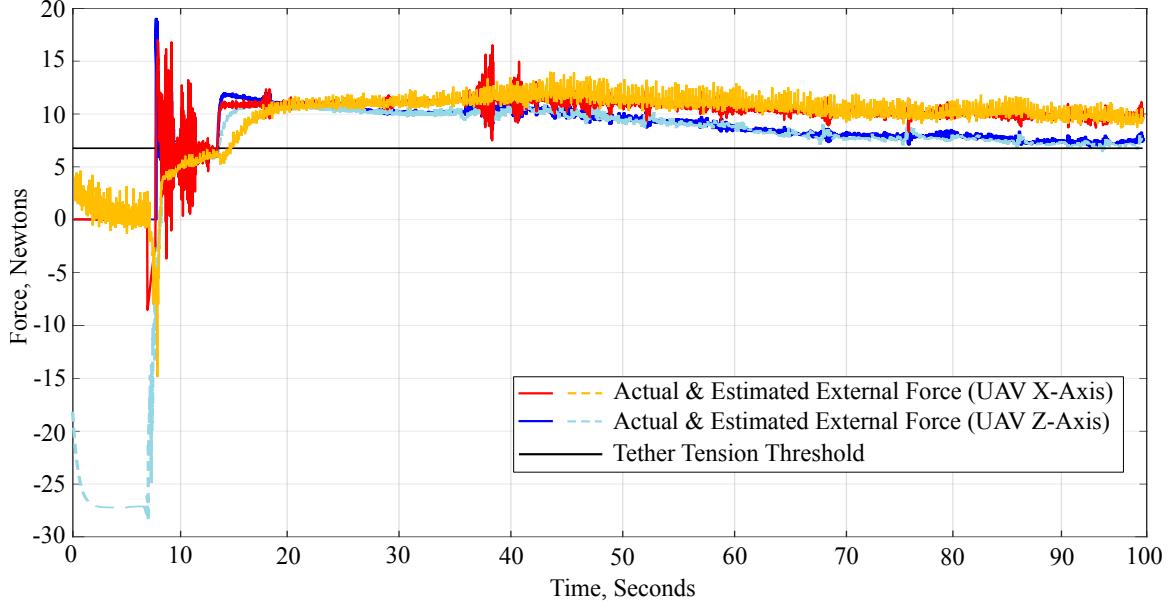


Figure 5-7: Comparison between the x and z components of $\mathbf{f}_{\text{ext}}^B$ versus $\hat{\mathbf{f}}_{\text{ext}}^B$. The estimated forces in the UAS \mathbf{b}_x , \mathbf{b}_z directions reveal the takeoff point at ≈ 8 s and the point of tether tautness at ≈ 15 s. The tether tension threshold, $f_{\text{ext},\text{thresh}}^B$, is compared with the x-component of $\hat{\mathbf{f}}_{\text{ext}}^B$ to determine tether tautness.

determination of tether tautness even with significant aerodynamic forces opposing the force of the tether. A close examination of the z-component of $\hat{\mathbf{f}}_{\text{ext}}^B$ on the plot also reveals the various stages of flight, such as takeoff at 8 seconds and tether tautness at 15 seconds. The relatively large z-component of $\hat{\mathbf{f}}_{\text{ext}}^B$ is attributable to the UAS pitch angle and the fact that the estimated external force is expressed in B .

These results over many simulated trials with significant aerodynamic disturbances, coupled with the implementation of all relevant dynamic effects and the efforts made to validate the modeled effects against empirical motion capture data, demonstrate the suitability of the presented control and estimation strategies for the air wake measurement environment. In aggregate, the body of testing work performed to date instills confidence in the ability of the UAS to perform the allotted air wake measurement task in hardware and to corroborate these presented results in a maritime environment.

5.1.2 Indoor Flight Tests

Hardware flight testing in a controlled indoor environment provides further validation of the simulation results in Section 5.1.1. Each performance indicator from the simulated tests is measured in a replicated testing scenario in hardware. In the indoor environment, motion capture software is used to evaluate tracking and state estimation accuracy, as well as spoof absolute and relative GPS sensor measurements with added bias and noise. Apart from GPS, all other sensor data come from real-time hardware. The indoor flight testing results bear striking similarity to those obtained in simulation, further validating the effectiveness of the flight control software for real-world air wake measurement missions.

The simulation environment models the UAS dynamics, tether constraint, ship motion, wind/aerodynamic effects, and a full sensor suite. The indoor testing setup is designed to replicate those conditions as closely as possible, as they are meant to be representative of the real maritime mission environment. Figure 5-8 shows the indoor flight testing arrangement, which consists of the UAS hardware platform and sensor suite, tether reel controller, 1:35 scale model LED beacon array, and artificial

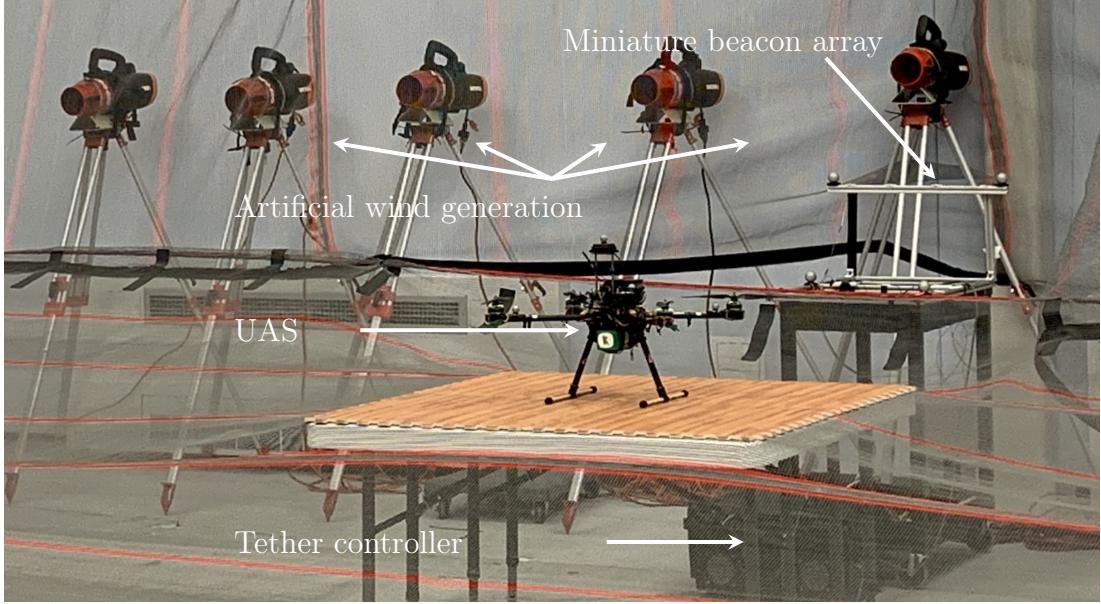


Figure 5-8: Physical setup for the indoor flight test trials. The tether length for wake survey mode is 7 meters, and a 1:35 scale beacon array is used for real-time relative pose estimation via a combination of real and spoofed sensor data. Generated wind provides realistic wind disturbances despite a static *S*-frame.

wind generators. As can be seen from the figure, reflective markers are also attached to both the UAS and beacon array so that the Vicon motion capture system is able to provide truth references for their states. These truth references are useful both for evaluating tracking and estimation performance as well as forming the basis for spoofed sensor measurements. The indoor setup is able to replicate (in miniature) the simulated and hypothetical maritime environments, as real hardware, real-time sensing, and substantial wind disturbances are used during tethered flight. In terms of real-time sensing, the following actual sensor data is used:

- Camera and corresponding relative pose estimation,
- Barometer altitude and pressure data,
- IMU accelerometer and gyro data,

and the following sensor data is spoofed from motion capture (with added noise and bias levels consistent with those described in Section 3.1):

- Absolute GPS measurements of UAS position,

- Relative GPS measurements between the tether reel controller and UAS.

The results presented in this paper correspond to the physical arrangement in Table 5.1.

Table 5.1: Physical parameters for the presented air wake measurement system indoor hardware flight tests.

Parameter	Value
Wind Speed (Furthest Point in Trajectory)	5 ± 0.5 meters/second
Wind Speed (Closest Point in Trajectory)	6 ± 0.75 meters/second
Tether Length	7 meters
Sweep Angle	40°

In total, four full flight tests, consisting of takeoff, tether tension establishment, several sweeps, and landing, have been carried out with artificial wind, with an additional four tests without wind. The results in this section present representative data indicative of the observed flight and state estimation performance over all flight trials with wind.

The relative estimator, fusing spoofed GPS and actual vision, barometer, and IMU data, achieves comparable accuracy and consistency (Fig. 5-9) to the relative estimator fusing purely simulated data (Fig. 5-2). Similarly, relative attitude estimation accuracy is comparable to simulation levels.

As with Fig. 5-3, Fig. 5-10 depicts a steady-state pitch offset induced by the 4–7 m/s wind. The offset is not present in the absence of wind, validating the simulation model for the rotor blade flapping effect. Empirically, it has been determined that sustained roll and pitch angles over 30 degrees begin to diminish the altitude control authority of the UAS. It is important to note that roll and pitch angles of that size have only been imposed artificially, and have never arisen by accident due to wind disturbances—especially when the tether is properly attached near the UAS center-of-mass. That said, the trajectory generator is designed to enact emergency return-to-home functionality should sustained, excessive pitching ever arise mid-flight.

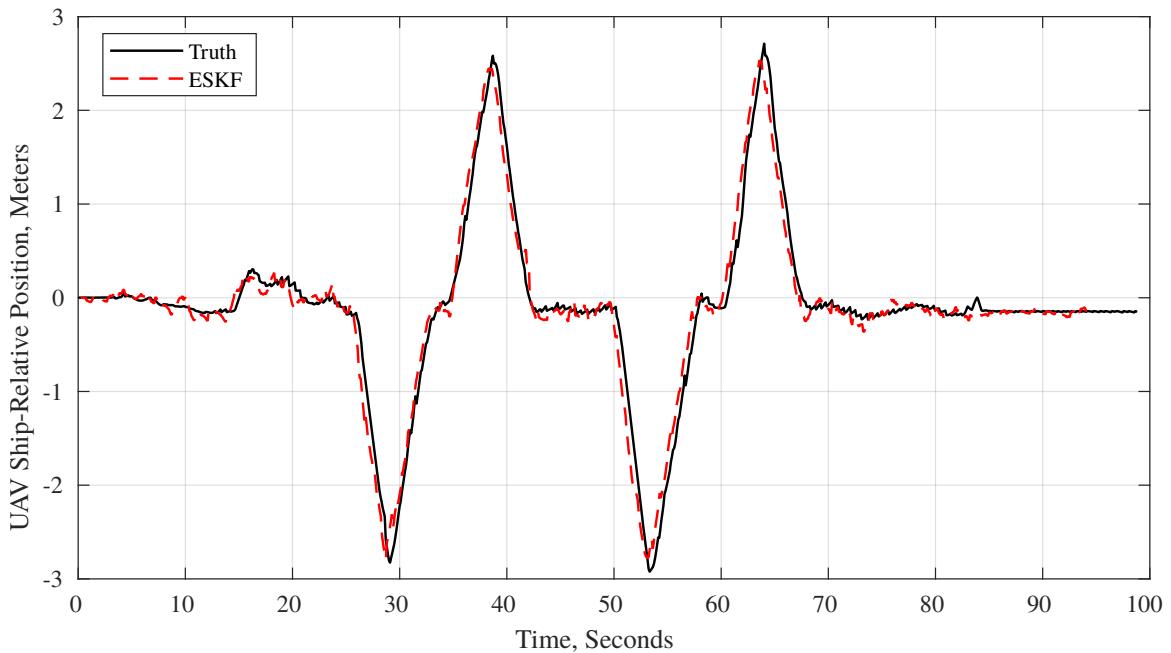


Figure 5-9: Relative state estimate of UAS y position w.r.t. and expressed in S over the course of an entire indoor hardware flight test with multiple sweeps. Accurate estimation is facilitated by the fusion of spoofed differential GPS and real vision-based pose measurements.

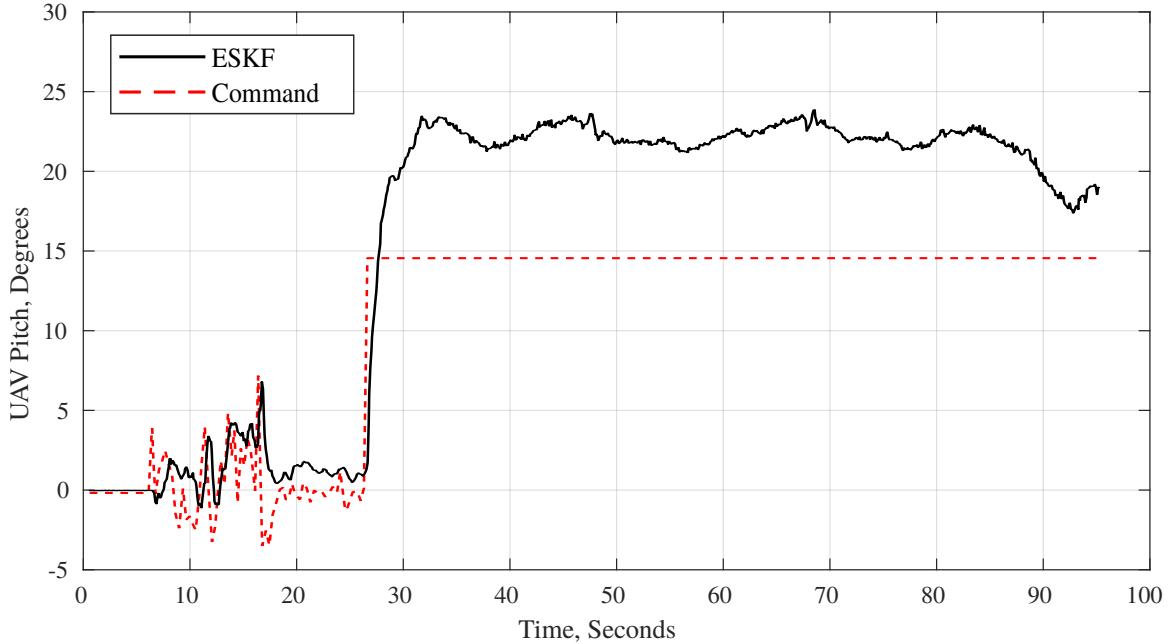


Figure 5-10: UAS pitch tracking performance throughout an entire indoor hardware flight test with multiple sweeps. A steady-state pitch disturbance between 5 and 10 degrees is induced by the rotor blade flapping effect at a relative wind speed of 4–7 m/s, very similarly to observed simulation results in Fig. 5-3.

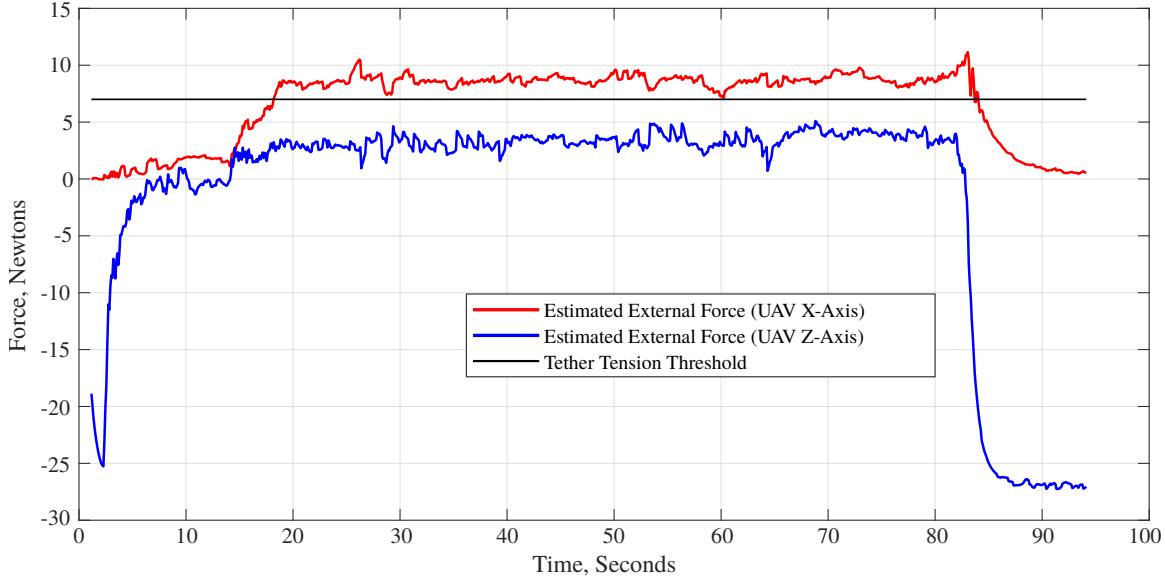


Figure 5-11: Comparison between the x and z components of $\mathbf{f}_{\text{ext}}^B$ versus $\hat{\mathbf{f}}_{\text{ext}}^B$ for an indoor hardware flight. As with simulation results in Fig. 5-7, the estimated forces in the UAS \mathbf{b}_x , \mathbf{b}_z directions reveal the takeoff point at ≈ 3 s and the point of tether tautness at ≈ 20 s.

Unlike the simulated external force estimation performance results in Fig. 5-7, the hardware results in Fig. 5-11 do not have a truth reference for the actual resultant forces caused by wind and the tether. However, a qualitative comparison between the simulation and hardware estimates points to the hardware system's ability to accurately deduce both when the tether is taut as well as when the UAS is grounded versus in the air. Moreover, no adjustment to the rough tether tautness threshold is needed due to the quality of the physical parameter matching between the simulated and hardware platforms.

Similarly to the simulation results in Fig. 5-5, Fig. 5-12 demonstrates that the hardware platform is able to perform the entire commanded 40-degree sweep angle despite wind disturbances. In the case of indoor flight testing, the limiting factor is the size of the flight space. Because the sweep operation is controlled in a low-bandwidth feedback loop by the trajectory planner using the estimated relative position, sweep angle ability is maintained in spite of wind disturbances and a lack of inner-loop attitude integrator control.

Indoor testing results with a combination of actual and spoofed sensing indi-

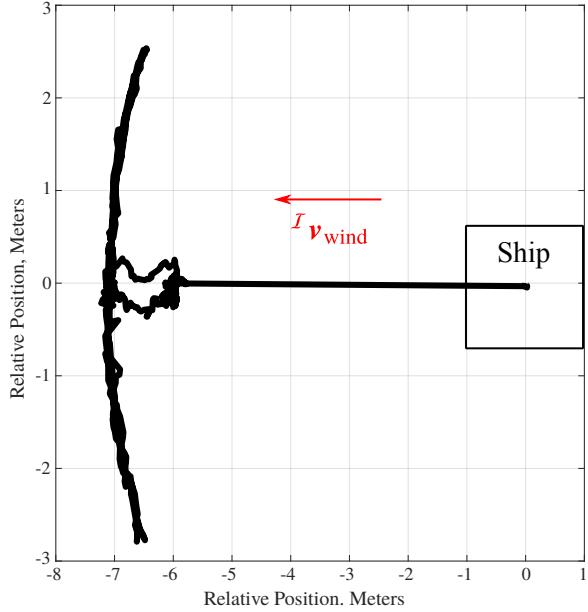


Figure 5-12: Top view of a sweep trajectory from indoor flight testing with a tether length of 7 m and wind speed of 4-7 m/s.

cate that the constructed hardware platform achieves the same level of mission performance as the previously fine-tuned simulated platform. To achieve comparable performance in hardware, controller gain and Kalman filter covariance tuning are required; however, no appreciable re-writing of the code used in simulation is necessary. With hardware performance and simulation results validated, next steps include expanding on the results presented here with a moving base station, real-time outdoor GPS sensing, and wind velocity measurement with the omnidirectional air probe.

5.1.3 Outdoor Flight Tests

In addition to the autonomy software validation from simulation and indoor flight testing, outdoor flight testing provides the opportunity to assess system integrity when subjected to real-world disturbances and real sensor noise conditions. To that end, for the outdoor flight tests, all real sensor hardware is used, and the ability of the air wake measurement system to maintain an accurate relative state estimate under sweeping conditions is assessed. Figure 5-13 depicts the outdoor testing setup. The UAS launches from a trailer bed which has mounted on it a full-scale IR beacon



Figure 5-13: Outdoor testing setup for the air wake measurement system. Pictured are the trailer bed, meant to stand in for a ship deck, with mounted IR beacons, the tether reel controller, and a base station GPS. The UAS platform is seen engaged in tethered flight as the tether line is doled out to its specified sweep length.

array, the tether reel controller, and the base station GPS. In essence, the outdoor testing setup differs from the indoor flight testing setup in that it encompasses a much larger flight area, is equipped with a full-scale beacon array, and has real, not spoofed, GPS and wind. Only the ship-mounted magnetometer is not used, leaving the ship orientation with respect to the NED frame to be estimated by fusing vision and differential GPS measurements.

Three separate outdoor flights with different tether sweep lengths (8 meters, 13 meters, 18 meters) are performed as the UAS uses the full sensor suite to estimate its relative state in real-time. Figure 5-14 visualizes this process, depicting the UAS's first-person view of the beacon array, a ground station view of the tethered UAS, and a rendering of the UAS's current estimated state at a snapshot in time for all three flights. The rendered arrows in the right portion of the figure also depict the UAS's current estimate of where inertial North is in the absence of a ship-mounted magnetometer. A qualitative viewing of the arrows indicates consistency across the three flights. This consistency is key, as it allows the relative ESKF to fuse both vision and differential GPS measurements to back out an accurate relative pose with

respect to the ship frame.

In the absence of true state data for quantitative comparison, Figures 5-15–5-17 present the estimated sweeping profiles for the three tethered flights. As with the simulation and indoor flight trials, good coverage of the hypothetical air wake region is demonstrated. In terms of relative position estimation accuracy, the fact that each sweeping arc traces out a stable radius corresponding to the correct tether length instills confidence that the differential GPS and vision measurements are being fused correctly. Only Fig. 5-17 indicates slight inconsistency in the shape of the sweep, but this is in part attributable to the fact that the tether was temporarily slackened during the third flight.

One important note that is not apparent in the sweep profile plots is that beyond a distance of approximately 11 meters, the vision pose measurements from the IR beacon array drop out. This is due to a thresholding parameter tuning issue given the bright day, and means that both the 13 meter and 18 meter flights execute their full-length sweeps without the assistance of the vision system. As can be seen from Figures 5-16–5-17, however, the relative position estimate is able to be maintained by the differential GPS measurements alone because the correct inertial North direction was previously deduced by the estimator with the help of the vision measurements toward the beginning of the flight. This observed behavior validates the ESKF’s robustness to temporary sensor dropouts, which is essential for real-world operation.

The presented sweep area and relative position estimation results from the three outdoor flight tests, taken together with the simulation-based and indoor flight tests, demonstrate that the air wake measurement system is able to handle real-world disturbances and sensor degradation to the extent that it can complete its mission of autonomously navigating the air wake zone under a variety of challenging conditions. The system is capable of the flight coverage breadth and state estimation accuracy needed for the in-situ collection of air flow data in a turbulent and feature-starved environment.

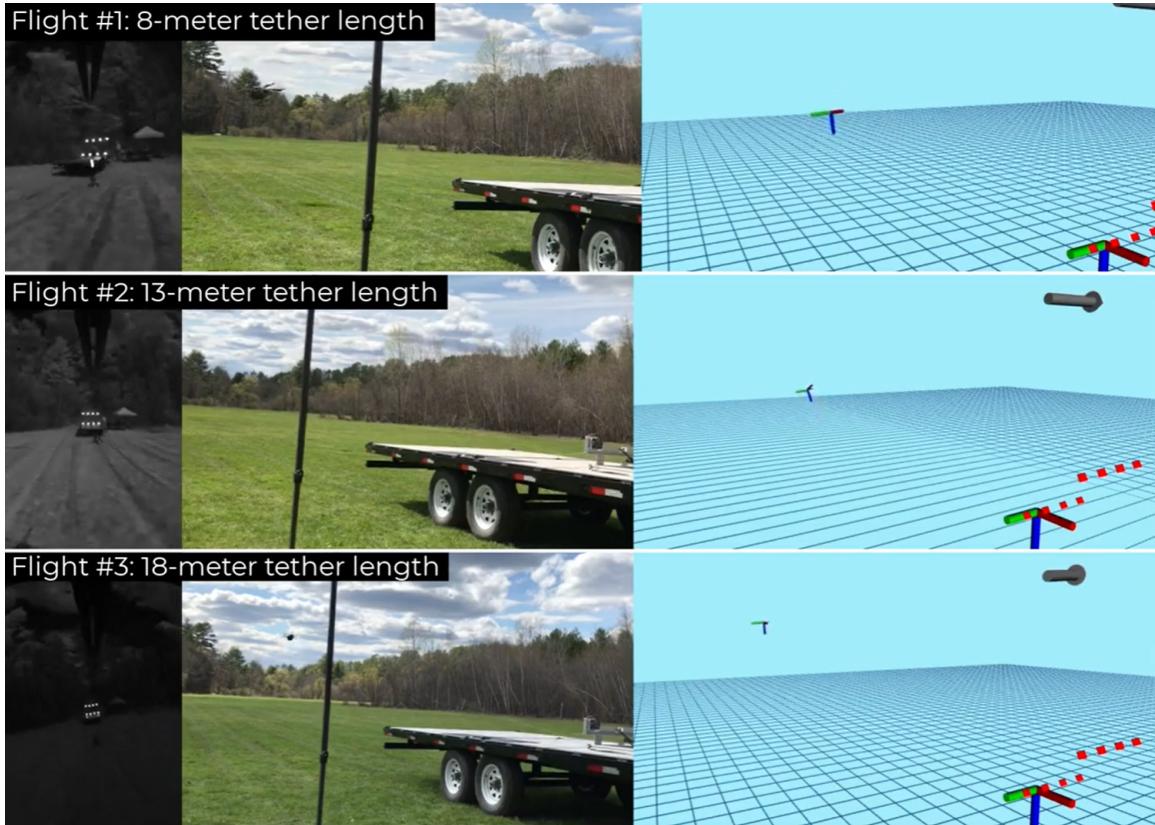


Figure 5-14: Visualization of the three presented outdoor flights of the air wake measurement system. The left panes show the first-person view of the UAS camera with the full-scale IR beacon array in frame. The middle panes give a ground station view of the UAS mid-flight at the end of a taut tether. The right panes illustrate key outputs of the relative estimator. The UAS (B) and ship (S) frames are both drawn, following the front-right-down body frame convention. The beacon array positions are also drawn as red dots for scale reference. Finally, the rendered 3D arrows in the sky point in the direction of the current estimated inertial North, whose accuracy and consistency is key to successfully fusing differential GPS and vision-based pose measurements.

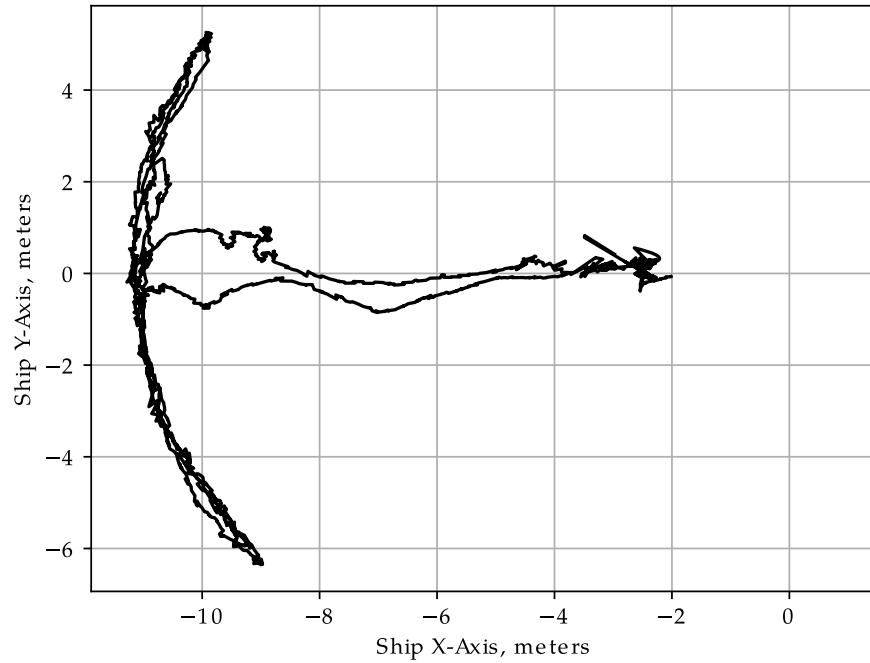


Figure 5-15: Sweep flight profile for outdoor flight test with a tether length of 8 meters.

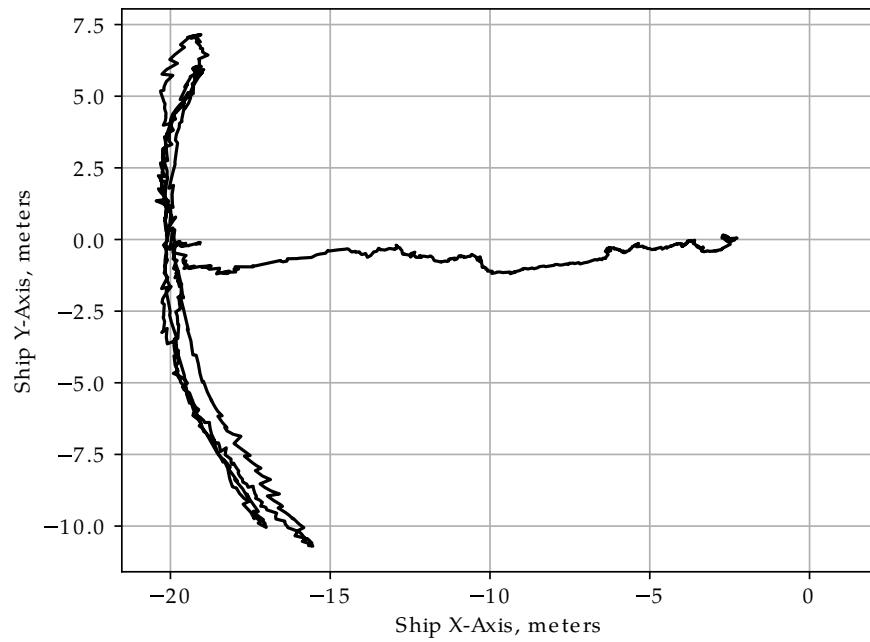


Figure 5-16: Sweep flight profile for outdoor flight test with a tether length of 13 meters.

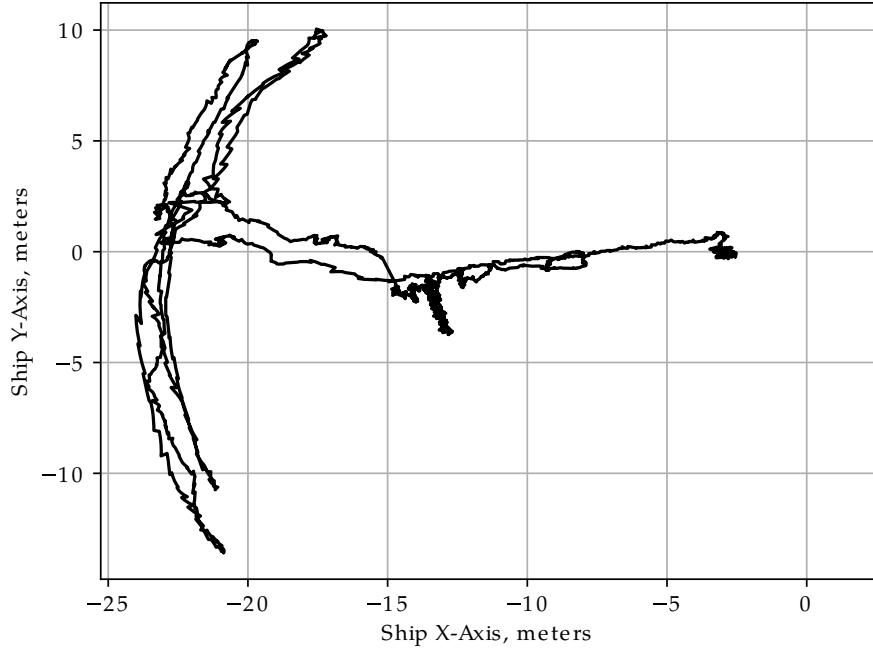


Figure 5-17: Sweep flight profile for outdoor flight test with a tether length of 18 meters.

5.2 Distributed Range-Enhanced Pose Graph Optimization Tests

5.2.1 DRPGO-CSLAM Dataset Tests

The performed experiments on the DRPGO-CSLAM datasets explained in Section 4.4 highlight the relative importance between loop closure measurements and inter-agent range measurements to global accuracy in distributed pose graph optimization. Furthermore, they establish the observations about the importance of anchoring that lay the foundation for the DRPGO experiments presented in Section 5.2.2.

The first set of presented results (Fig. 5-18-5-22) focuses on the EuRoC-derived DRPGO-CSLAM dataset. Coming from visual-inertial data collected in a feature-rich indoor space, the initial VIO estimates are reasonably accurate on their own. Figure 5-18 gives the root-mean-squared error (RMSE) for each of the three translational and three rotational degrees of freedom of each agent’s entire estimated trajectory, using VIO alone, as a function of number of poses added to the graph. While the

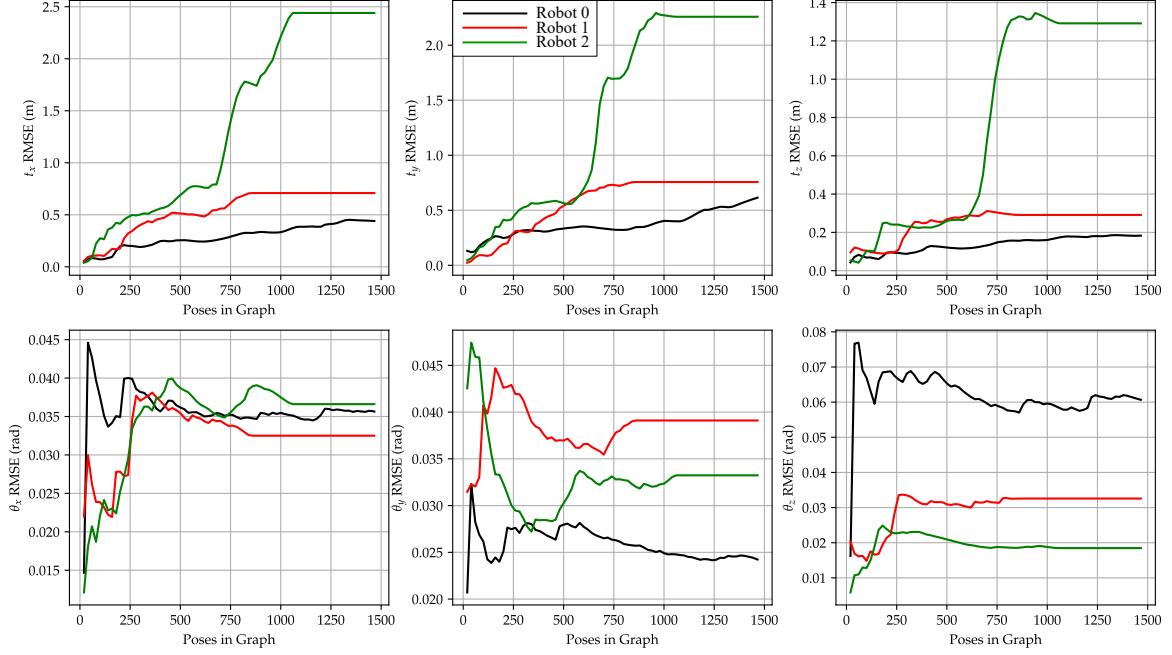


Figure 5-18: Cumulative global RMSE of the 6-DOF estimated trajectories of the three agents from the EuRoC MAV dataset, relying on VIO alone.

translational components of Robot 2’s trajectory estimate are less accurate due to a faster and jerkier trajectory, the trajectory errors for Robot 0 and 1 are maintained at the sub-meter level. Moreover, the rotational errors are consistently hover around only 2.5 degrees.

The global accuracy of the 6-DOF estimates for each robot is observed to improve intuitively as additional sensing modalities are added. Figure 5-19, which depicts the RMSE results after incorporating noisy altimeter measurements, shows a nominal increase in the estimation accuracy of the global translational z-component t_z , while all other degrees of freedom exhibit the same overall accuracy levels as with Fig. 5-18.

Inter-agent range measurements are incorporated for the first time in Fig. 5-20, which gives RMSE results for VIO with altimeter and range measurements added with each new pose in the graph. From the translation error plots, it is apparent that Robot 2’s position estimate improves while that of Robot 0 and 1 remain mostly unaffected. Differential roll and pitch is completely untouched compared to the results from Fig. 5-19; the only change in rotation accuracy is in Robot 0’s yaw estimate, which worsens compared to the VIO-only solution. These results demonstrate that

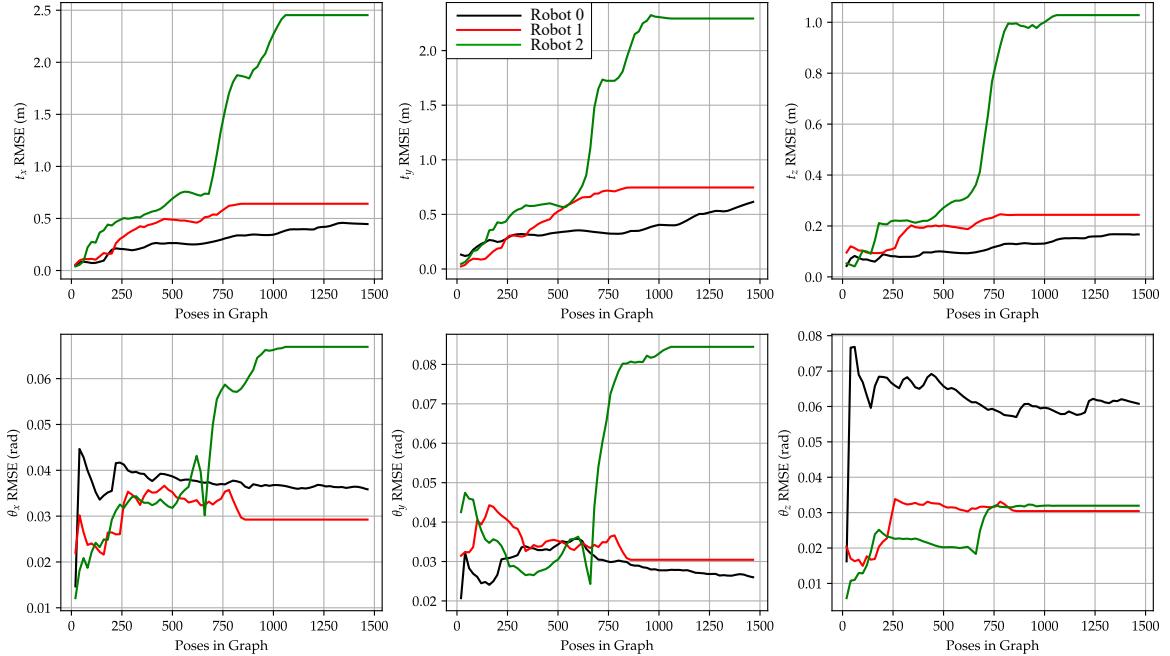


Figure 5-19: Cumulative global RMSE of the 6-DOF estimated trajectories of the three agents from the EuRoC MAV dataset, fusing VIO and altimeter data using PGO.

a naive incorporation of inter-agent range measurements will not necessarily improve all agents' state estimates across the board.

Alternatively to inter-agent range measurement incorporation, Fig. 5-21 gives error plots with loop closure measurements prioritized. Analyzing the steady-state RMSE values, the global accuracy of all agents' estimates is almost universally improved. It is interesting to note the error spikes that occur e.g., at 750 poses. These are attributable to the distributed nature of the optimization; when an inter-agent pose constraint is imposed, an error in one agent's trajectory will affect the accuracy of the agent with a shared pose, and the accuracy of each trajectory will incrementally improve in a leap-frog fashion. Nevertheless, the errors even at the spikes are comparable to or less than the errors without incorporating loop closures. Additionally, comparing Fig. 5-21 and 5-22, it is apparent that the effect of adding loop closures dominates the accuracy of the entire solution with all possible sensors incorporated.

It is informative to re-run a subset of the experiments above on a different DRPGO-CSLAM dataset that results in a different baseline solution quality. Figure 5-23 gives

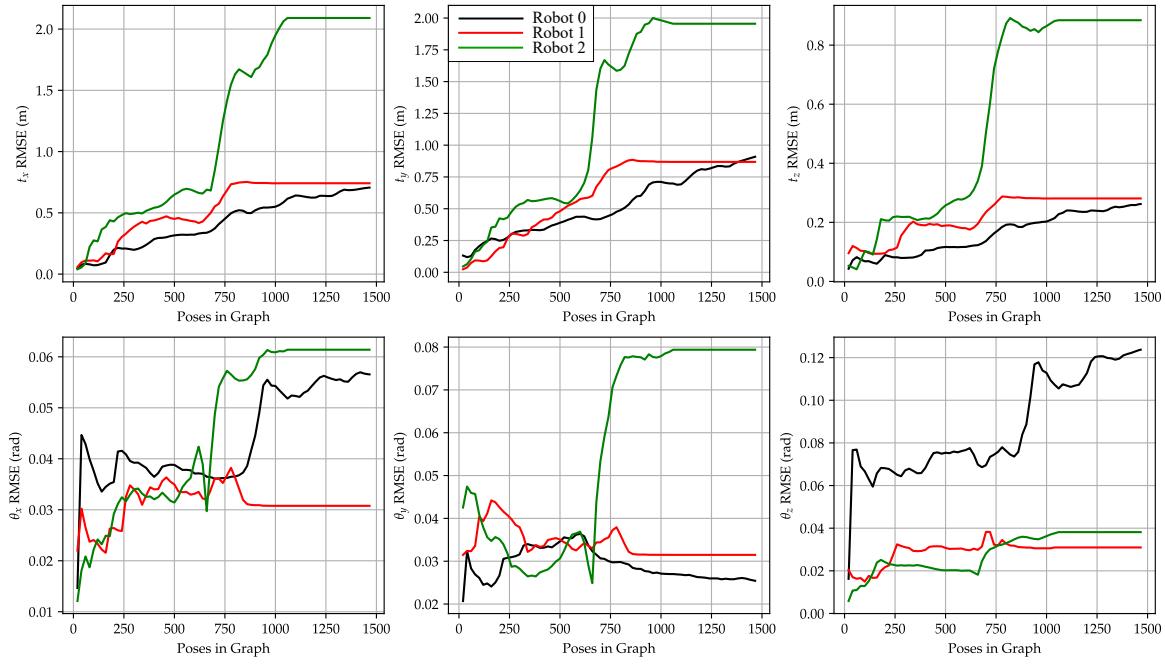


Figure 5-20: Cumulative global RMSE of the 6-DOF estimated trajectories of the three agents from the EuRoC MAV dataset, fusing VIO, altimeter, and inter-agent range data using PGO.

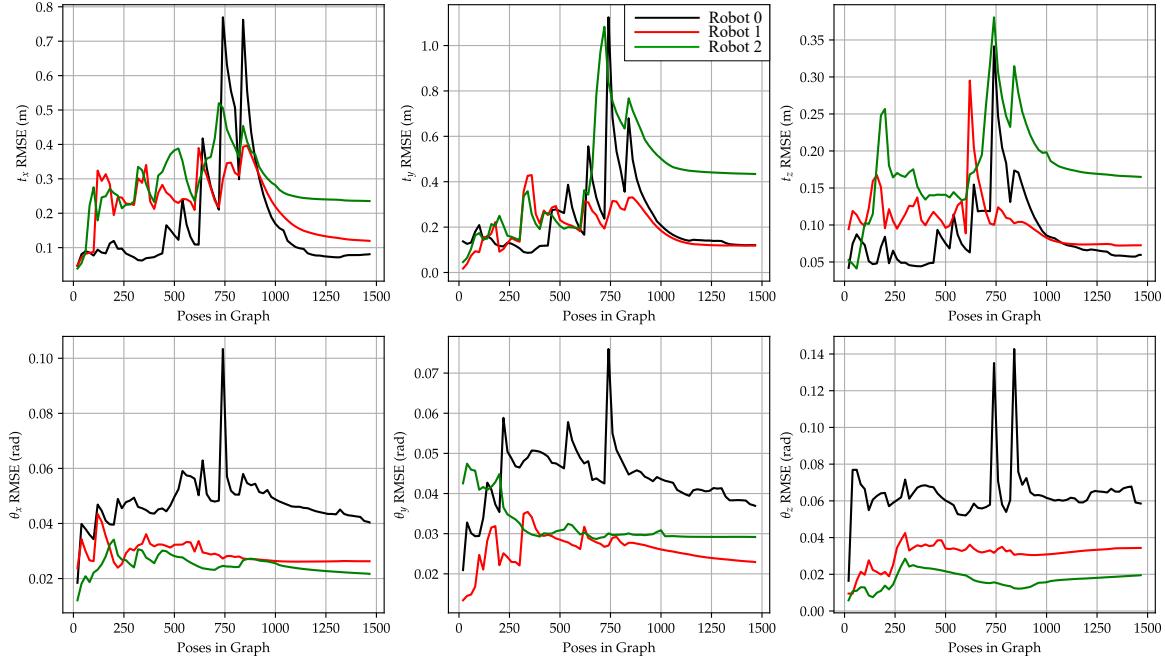


Figure 5-21: Cumulative global RMSE of the 6-DOF estimated trajectories of the three agents from the EuRoC MAV dataset, fusing VIO, altimeter, and intra-/inter-agent loop closures using PGO.

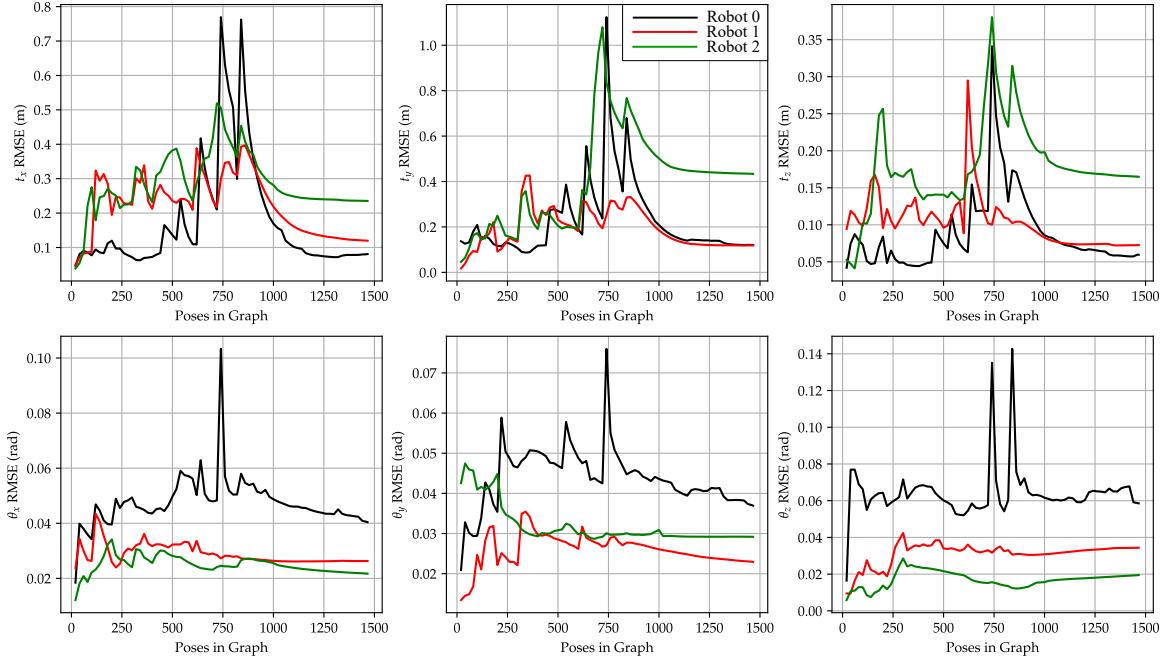


Figure 5-22: Cumulative global RMSE of the 6-DOF estimated trajectories of the three agents from the EuRoC MAV dataset, fusing VIO, altimeter, intra-/inter-agent loop closures, and inter-agent range data using PGO.

the RMSE for the state estimates of a four-agent DRPGO-CSLAM dataset with a feature-starved environment. The degraded feature quality directly results in a heavily degraded translation solution for all agents with a VIO-only estimation scheme.

With a lower-accuracy VIO baseline solution, the mixed results of incorporating inter-agent range information become more pronounced. Figure 5-24 gives the RMSE with VIO, altimeter, and range data only. Compared to the translation plots in Fig. 5-23, it is apparent that the inter-agent range measurements cause the higher-accuracy estimates to degrade as the lower-accuracy estimates improve, leading all agents' accuracies to gravitate toward a common mean. This behavior is the result of the agents' relative accuracy improving, but with global drift in the overall multi-agent trajectory estimate, as discussed in Section 4.3.

As a brute-force attempt to address the inconsistent accuracy improvements in Fig. 5-24, the state estimates of Robot 2 are artificially replaced with the true global pose estimate, and the resulting errors for the remaining agents are plotted in Fig. 5-25. The improvement in accuracy for the remaining agents is stark, improving an

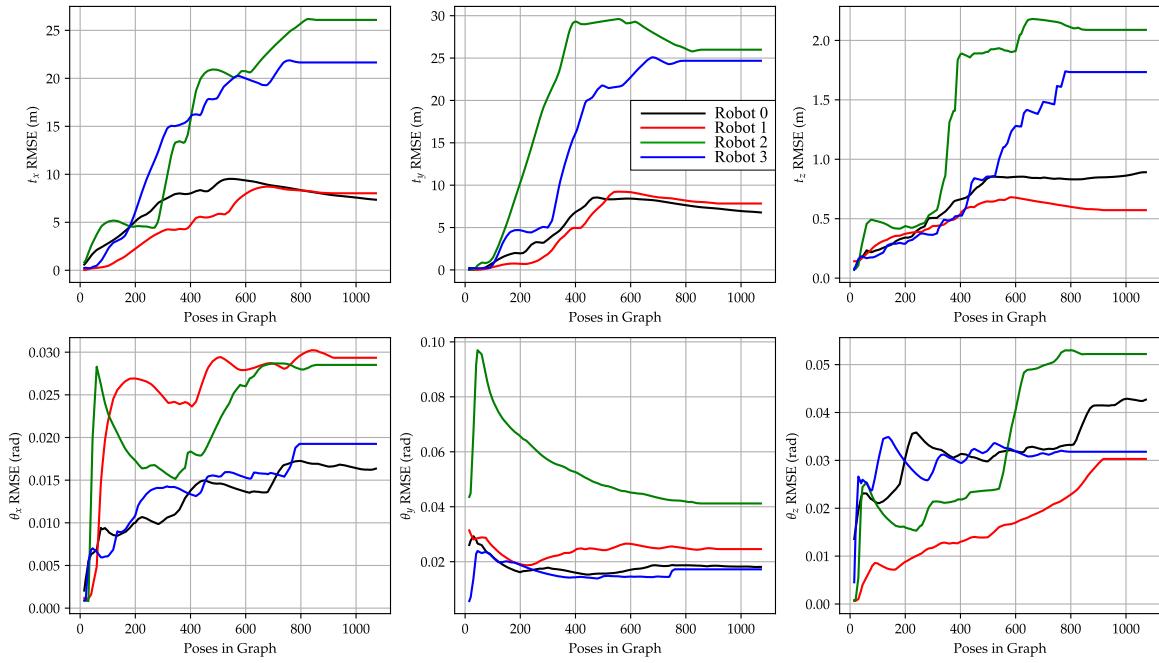


Figure 5-23: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from a feature-starved environment dataset generated from AirSim. The trajectory estimates come from VIO alone.

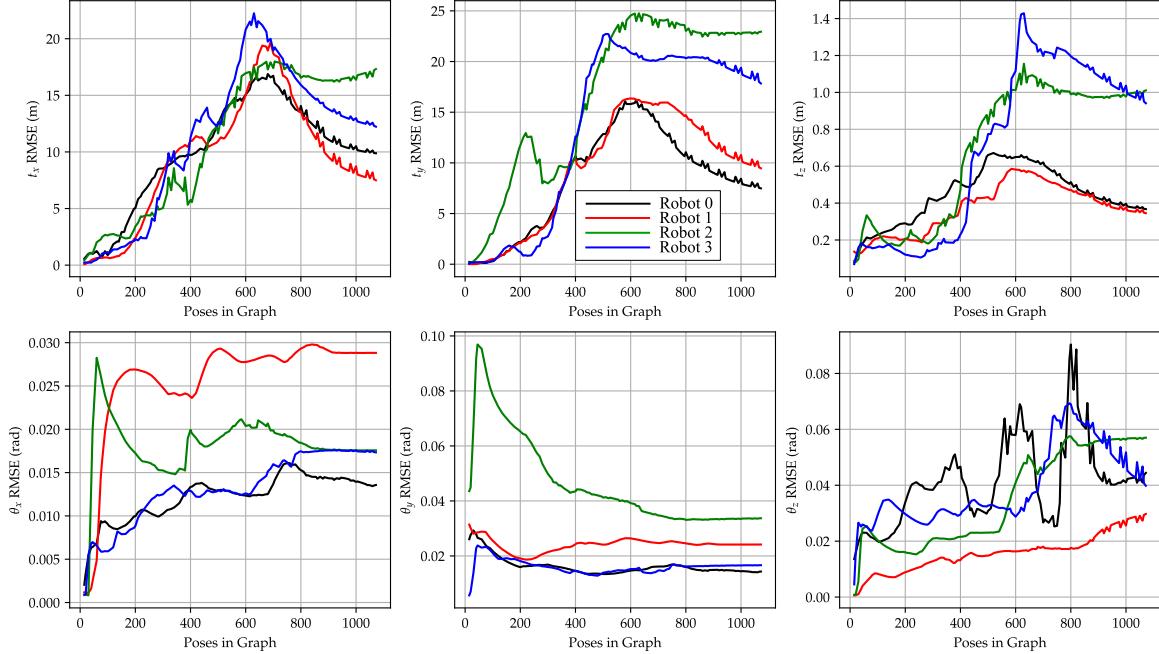


Figure 5-24: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from a feature-starved environment dataset generated from AirSim. The trajectory estimates come from fusing altimeter and inter-agent range data with PGO.

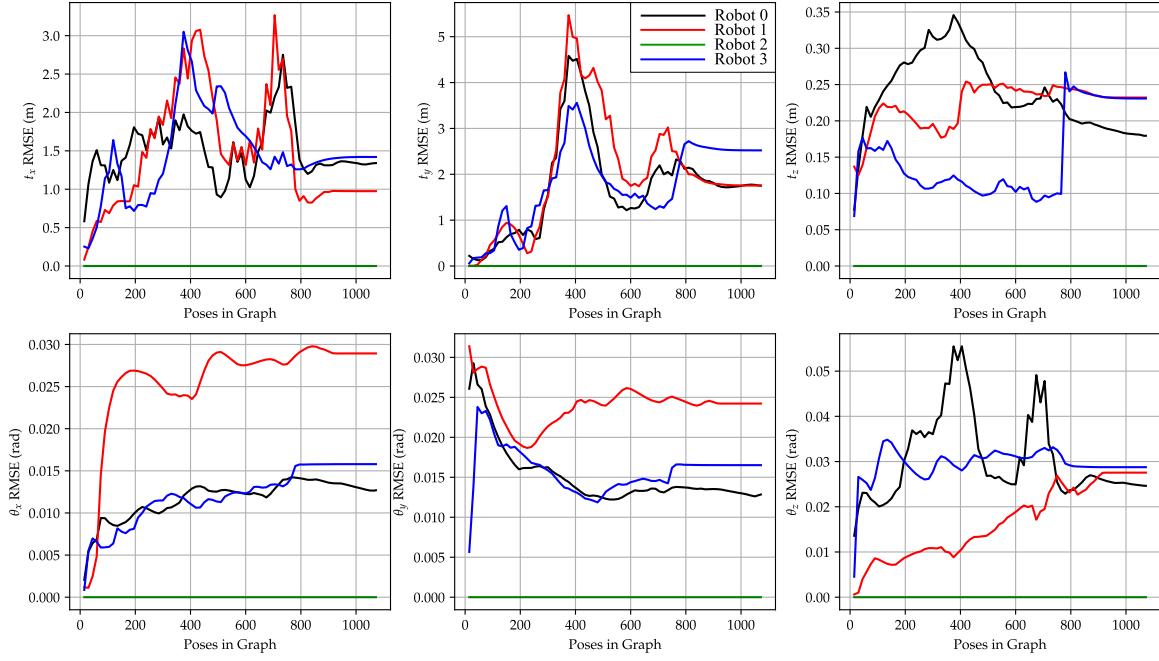


Figure 5-25: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from a feature-starved environment dataset generated from AirSim. The trajectory estimates come from fusing altimeter and inter-agent range data with DRPGO. The distinction from Fig. 5-24 is that in this case, Robot 2’s trajectory estimate is automatically set to its true global estimate, thus anchoring the global consistency of the rest of the trajectory estimates.

entire order of magnitude as would be seen with loop closure measurement incorporation. These results corroborate the notion that for the global accuracy of a swarm’s estimate to improve with inter-agent range measurements, at least one agent in the communication graph must incorporate a separate source of information that improves its own global accuracy directly.

5.2.2 DRPGO Simulation Tests

The presented results from the DRPGO-CSLAM datasets demonstrate with realistic CSLAM sensor data that DRPGO requires a strategy for anchoring the global accuracy of the swarm’s collective trajectory estimate, which can be held together in a relative sense with inter-agent range measurements. The experiments in this section expand on those results, simulating evolving communication sub-graph topologies and anchoring strategies. The data are collected from the DRPGO simulation environ-

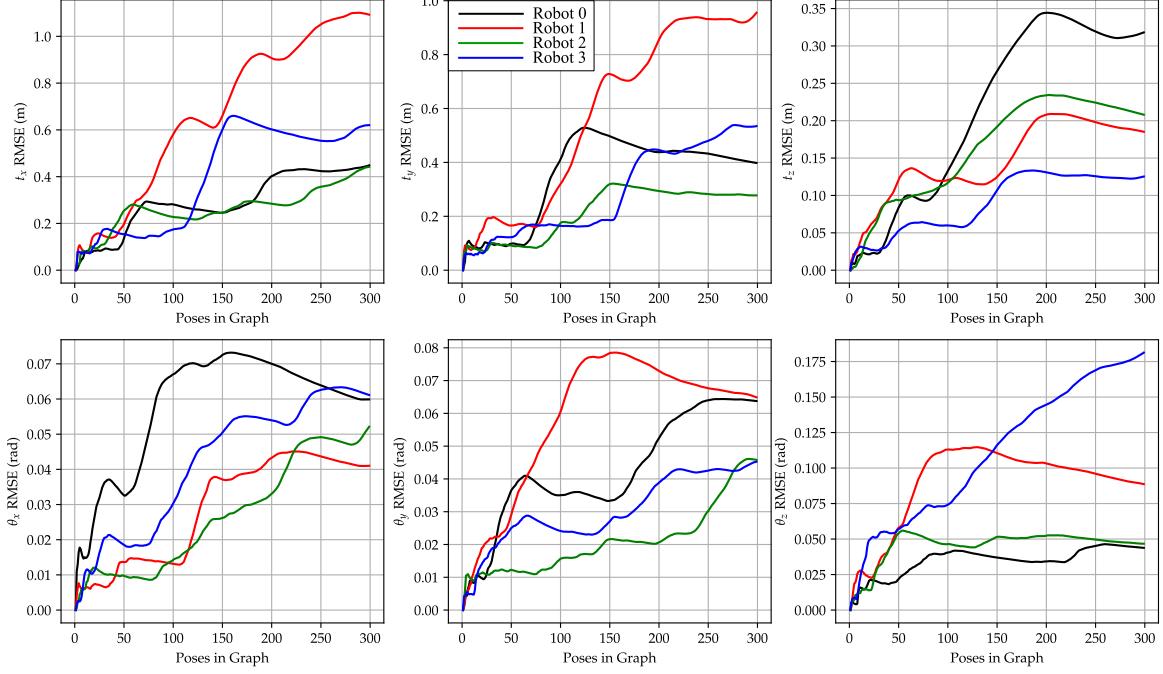


Figure 5-26: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from the DRPGO simulation experiments. In this trial, the agents share no pose or range data with each other through the course of the flight (see Fig. 5-27).

ment described in Section 4.5.

Figure 5-26 presents the error plots for DRPGO in its full form (Algorithm 2), but hampered by imposing no inter-agent communication at all over the course of the flight. Figure 5-27 illustrates the corresponding communication sub-graphs. It does this by coloring the error plots from Fig. 5-26 with a different color for each sub-graph. Thus, the unique colors for each curve indicate that the swarm is totally disconnected. In terms of the raw RMSE, it can be seen that with no inter-agent range measurements, the overall error trend is increasing over time for all degrees of freedom.

Figures 5-28 and 5-29 give the raw RMSE and communication sub-graph data, respectively, for a scenario of DRPGO with medium connectivity. With medium connectivity, non-trivial communication sub-graphs exist (Fig. 5-29), but are allowed to evolve as agents move into and out of each others' communication ranges. Each new sub-graph results in a new anchor agent assignment. In Fig. 5-29, anchor agent assignments are denoted by solid, thicker lines. In terms of raw RMSE, the increasing

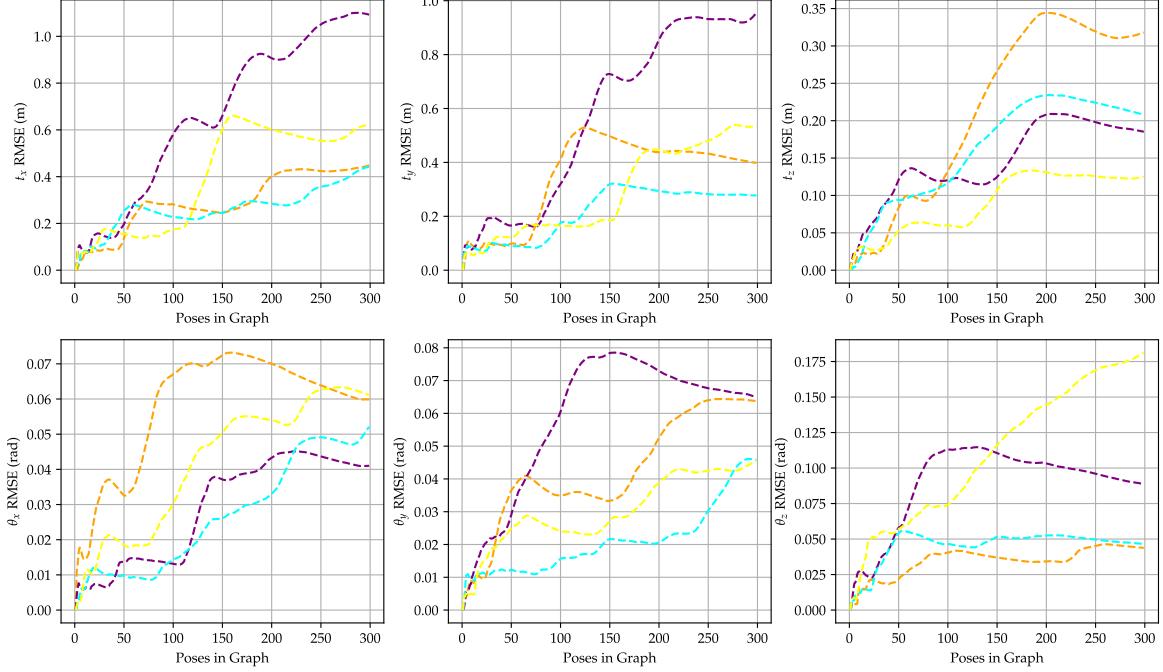


Figure 5-27: Cumulative global RMSE plot from Fig. 5-26, visualizing the communication graph connectivity. Distinct line colors correspond to distinct communication graphs, and dotted/solid lines correspond to unanchored/anchored trajectory segments.

error trend of Fig. 5-26 is absent with full DRPGO, and overall error levels are universally lower by an order of magnitude than in the communication-less case.

A close examination of Fig. 5-29 shows the importance of sub-graph anchoring; as the anchor agent's RMSE decreases with the addition of loop closures, the global error of the rest of the sub-graph also decreases in response, despite the fact that only range measurements are being shared. This denotes a strategy for global accuracy without having to communicate the level of information needed for distributed data association.

Results from experiments with the full DRPGO scheme—this time with full connectivity—are also presented in the raw RMSE and communication sub-graph plots in Fig. 5-30 and 5-31. With full connectivity, the same communication sub-graph persists throughout the entire flight, which means that there is a single assigned anchor agent over the course of the experiment. Once again, the global translation error is observed to be bounded thanks to the inter-agent range measurements, and although the overall

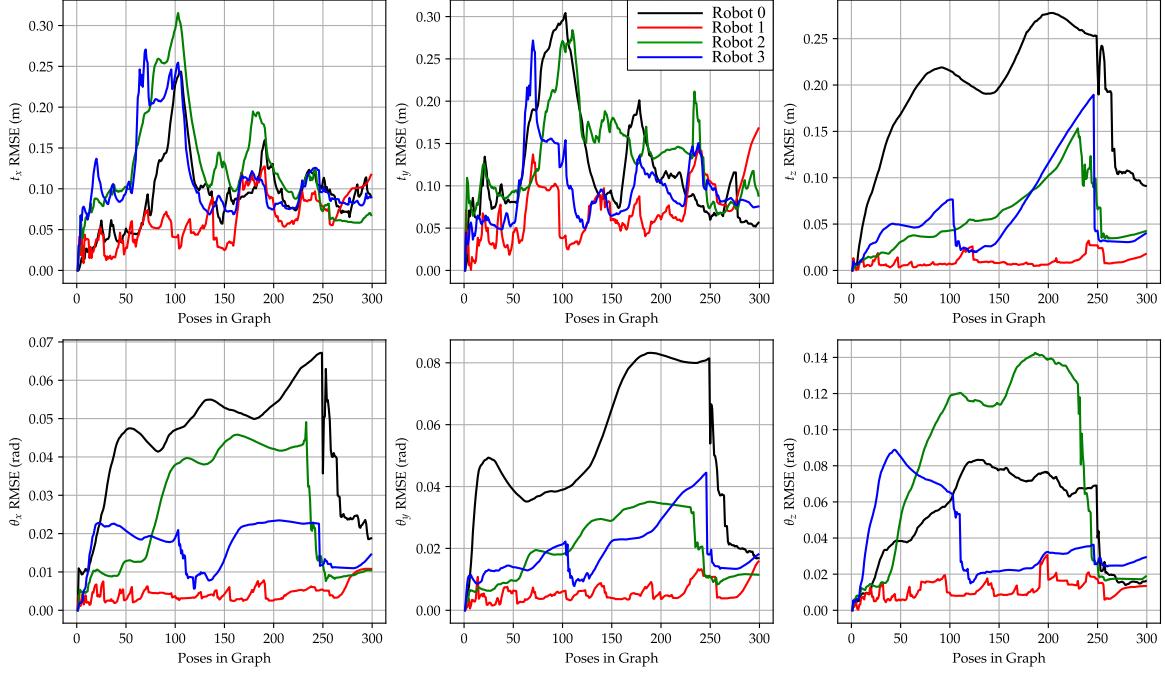


Figure 5-28: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from the DRPGO simulation experiments. In this trial, the agents share pose or range data only within their connected communication sub-graphs, which evolve over the course of the flight (see Fig. 5-29).

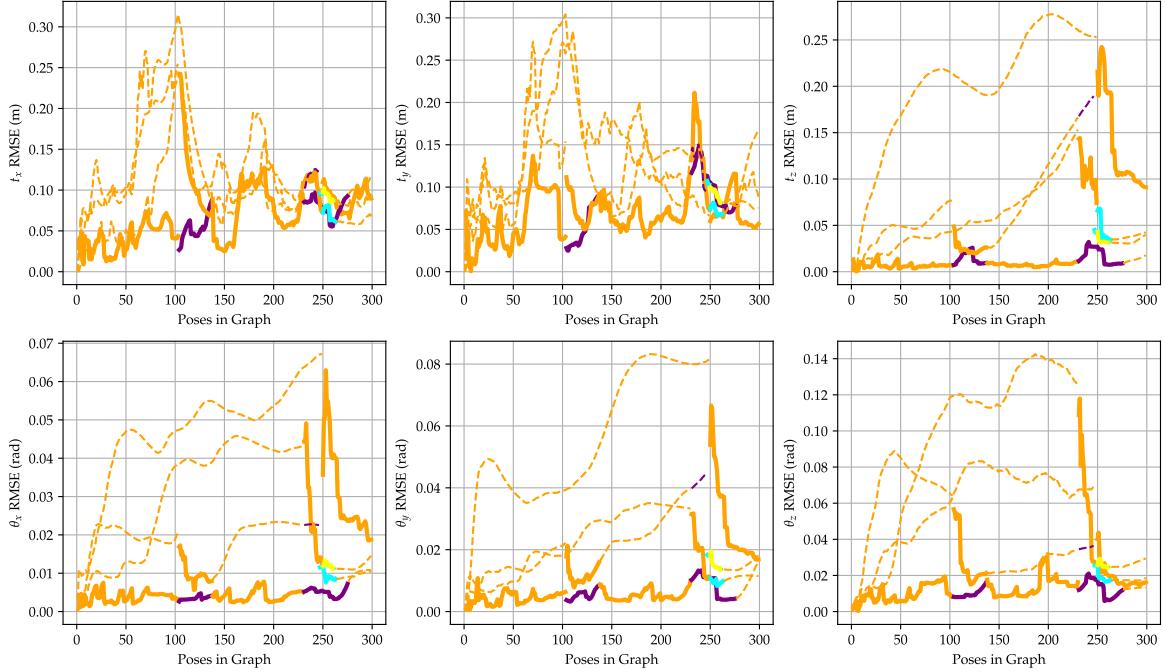


Figure 5-29: Cumulative global RMSE plot from Fig. 5-28, visualizing the communication graph connectivity. Distinct line colors correspond to distinct communication graphs, and dotted/solid lines correspond to unanchored/anchored trajectory segments.

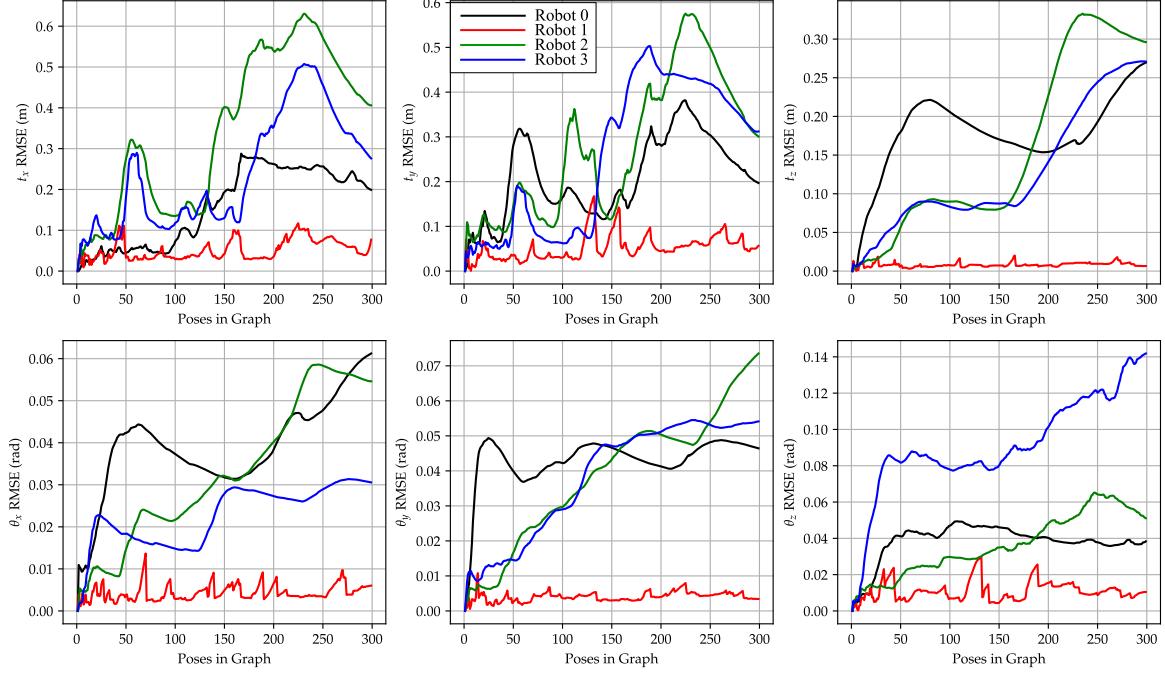


Figure 5-30: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from the DRPGO simulation experiments. In this trial, all agents share necessary pose or range data with each other throughout the flight (see Fig. 5-31).

RMSE is higher than that of the medium-connectivity experiment, it is still considerably lower than that of the communication-less experiment. The fact that the overall RMSE is higher than with the medium-connectivity experiment is explained by the smaller number of sub-graphs, which translates to less loop closures being added to the pose graphs overall.

As a final illustrative experiment, the full-connectivity experiment is repeated, but this time with no anchor agent assignment. The results are presented in Fig. 5-32 (raw RMSE) and 5-33 (communication sub-graph). The resulting global errors look similar to the case with no inter-agent communication at all, with unbounded error growth in translation and greater error in the z-component of rotation. Even with full connectivity, anchoring loop closure measurements are needed to prevent the aggregate trajectory for all agents from drifting globally.

One immediate question that remains to be answered pertains to determining exact accuracy expectations given reasonable operating conditions. In other words, Monte Carlo testing over many different communication graph topology evolutions

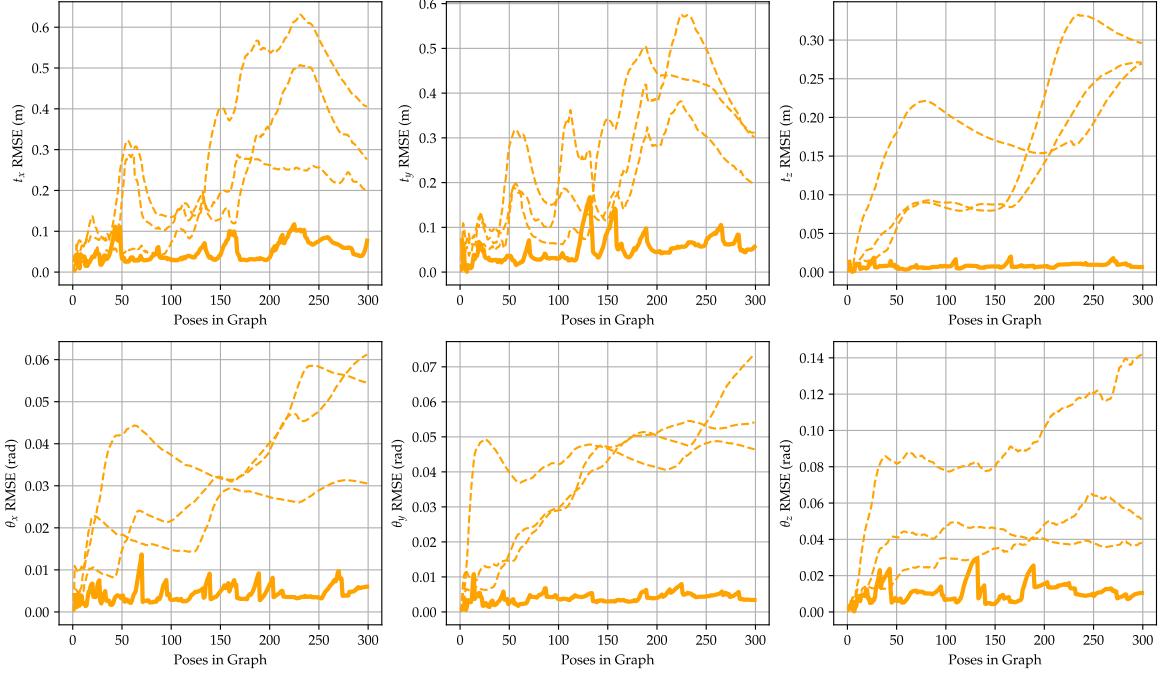


Figure 5-31: Cumulative global RMSE plot from Fig. 5-30, visualizing the communication graph connectivity. Distinct line colors correspond to distinct communication graphs, and dotted/solid lines correspond to unanchored/anchored trajectory segments.

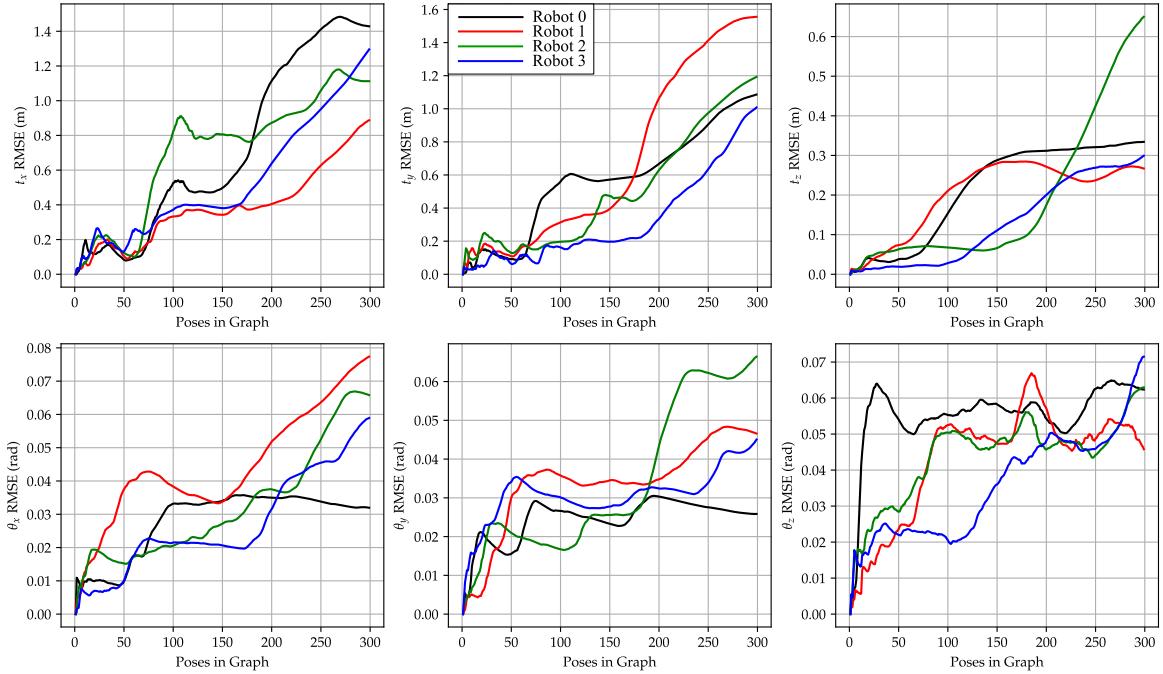


Figure 5-32: Cumulative global RMSE of the 6-DOF estimated trajectories of the four agents from the DRPGO simulation experiments. In this trial, all agents share necessary pose or range data with each other, but no agent ever is assigned to act as an anchor (see Fig. 5-33).

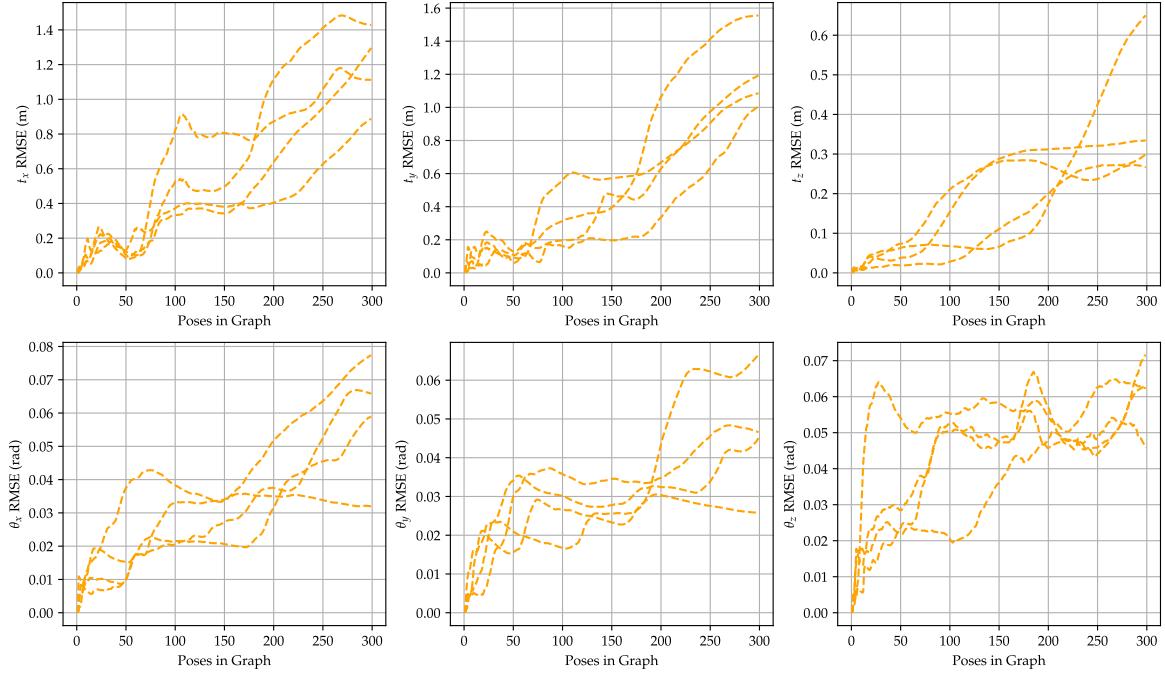


Figure 5-33: Cumulative global RMSE plot from Fig. 5-32, visualizing the communication graph connectivity. Distinct line colors correspond to distinct communication graphs, and dotted/solid lines correspond to unanchored/anchored trajectory segments.

can give a more quantitative measure to the expected upper and lower global accuracy bounds, as well as how those bounds scale with the number of agents and sizes of the communication sub-graphs. Nevertheless, the presented DRPGO results act as a proof-of-concept to indicate that inter-agent range measurements, when coupled with a strategy for intelligently and efficiently incorporating loop closures on a single-agent basis, can reliably facilitate more globally accurate swarm trajectory estimates with minimal inter-agent communication requirements. These characteristics are essential for applications like radiological mapping, where the ability to explore larger spaces in a time-sensitive manner while not simultaneously straining the inter-agent communication network is essential.

Chapter 6

Conclusion

6.1 Summary

This thesis presents two different UAS designed to accomplish accurate and reliable sensing challenging real-world settings. For the air wake measurement system, this is accomplished by tailoring classical techniques in control and estimation to the specific idiosyncrasies of both tethered flight and the maritime environment. The use of a tether in the air wake measurement system affords a measure of passive flight stabilization which, when coupled with the presented control strategy, allows the UAS to attain sufficient coverage to map both the predominantly turbulent and free stream regions of the air wake profile. The relative state estimation algorithm, which fuses measurements from a differential GPS as well as a vision-based pose estimation scheme using active IR beacons, provides a centimeter-level-accuracy estimate of the UAS state relative to the ship, even at large distances and in the midst of temporary sensor dropouts.

The presented DRPGO formulation for swarms performing radiological mapping is tailored to the well-documented constraints on inter-agent communication and flight time that are inherent to multi-agent coverage problems. Simulation demonstrations show that, given a strategy for allocating tasks pertaining to exploration versus exploitation, a UWB-augmented, distributed pose graph optimization system can achieve similar global localization accuracies to full-fledged distributed CSLAM meth-

ods at a fraction of the communication cost. These proof-of-concept results open the door for further utilization of UWB-based ranging for increasing the robustness of mapping with swarms and inexpensive sensor suites.

6.2 Future Work

The air wake measurement system demonstrations presented in this thesis validate the approaches taken in estimation and control, particularly when subjected to real-world disturbances and sensing conditions. With validated autonomy performance, the system is ready to be deployed for in situ air wake measurements on a maritime vessel to accomplish the ultimate goal of validating CFD-based air wake models for improved maritime landing operations. The demonstrations related to DRPGO, by comparison, require more thorough testing and demonstration on systems that are not simply artificially constrained—particularly through moving beyond simulation and into extensive hardware testing in actual UAS swarms. Varied and convincing demonstrations of DRPGO’s ability to improve the global consistency and accuracy of a swarm’s collective localization under real-world communication and computational constraints is still needed. Once these demonstrations are accomplished in a laboratory environment, DRPGO can confidently be integrated into a distributed CSLAM system for facilitating faster radiological mapping in time-sensitive and otherwise constrained and challenging environments.

Bibliography

- [1] Dinuka M. W. Abeywardena, Sarath Kodagoda, Gamini Dissanayake, and S. Rohan Munasinghe. Improved state estimation in quadrotor mavs: A novel drift-free velocity estimator. *CoRR*, abs/1509.03388, 2015.
- [2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. <http://ceres-solver.org>.
- [3] Kairos Autonomi. Gps correction comparisons – RTK vs DGPS. Online, January 2000. <http://www.kairosautonomi.com/uploads/files/129/Bulletin---RTK-vs-DGPS-010400.pdf>. Accessed August 2018.
- [4] Ross Barnowski, Andrew Haefner, Lucian Mihailescu, and Kai Vetter. Scene data fusion: Real-time standoff volumetric gamma-ray imaging. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 800:65–69, 2015.
- [5] Z. Berman. Outliers rejection in kalman filtering; some new observations. *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, May 2014.
- [6] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.
- [7] Titus Cieslewski, Siddharth Choudhary, and Davide Scaramuzza. Data-Efficient Decentralized Visual SLAM. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2466–2473, Brisbane, QLD, May 2018. IEEE.
- [8] Cooper Cone, Samantha Sullivan, Malik Hancock, Kevin Carey, and Andrew Kopeikin. *Distributed and Adaptive Multi-UAS Behaviors to Support Nuclear Response*.
- [9] Charles Champagne Cossette, Mohammed Shalaby, David Saussié, James Richard Forbes, and Jérôme Le Ny. Relative position estimation between two uwb devices with imus. *IEEE Robotics and Automation Letters*, 6(3):4313–4320, 2021.

- [10] M. Cutler, B. Michini, and J. P. How. Lightweight infrared sensing for relative navigation of quadrotors. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1156–1164, 2013.
- [11] Gregory M. Dooley, Pablo Carrica, Juan Martin, Austin Krebill, and James Buchholz. Effects of waves, motions and atmospheric turbulence on ship air-wakes. *AIAA Scitech 2019 Forum*, Jan 2019.
- [12] Brandon Draper. Long-range outdoor monocular localization with active features for ship air wake measurement. Master’s thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2019.
- [13] Rodolphe Dubois, Alexandre Eudes, and Vincent Frémont. On Data Sharing Strategy for Decentralized Collaborative Visual-Inertial Simultaneous Localization And Mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019)*, Macau, China, November 2019.
- [14] Gary Ellingson, Kevin Brink, and Tim McLain. Cooperative relative navigation of multiple aircraft in global positioning system-denied/degraded environments. *Journal of Aerospace Information Systems*, 17(8):470–480, 2020.
- [15] Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. A monocular pose estimation system based on infrared leds. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.
- [16] Kanishke Gamagedara, Kalpesh Patil, Taeyoung Lee, and Murray R. Snyder. Vision-based relative localization for airborne measurements of ship air wake. *2018 AIAA Atmospheric Flight Mechanics Conference*, Jan 2018.
- [17] Matthew Giamou, Kasra Khosoussi, and Jonathan P. How. Talk Resource-Efficiently to Me: Optimal Communication Planning for Distributed Loop Closure Detection. *arXiv:1709.06675 [cs]*, March 2018. arXiv: 1709.06675.
- [18] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [19] Kexin Guo, Xiuxian Li, and Lihua Xie. Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control. *IEEE Transactions on Cybernetics*, 50(6):2590–2603, 2020.
- [20] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. *AIAA Guidance, Navigation and Control Conference and Exhibit*, Aug 2007.
- [21] Haomiao Huang, Gabriel M. Hoffmann, Steven L. Waslander, and Claire J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *2009 IEEE International Conference on Robotics and Automation*, pages 3277–3282, 2009.

- [22] Michael Hutchinson, Cunjia Liu, and Wen-Hua Chen. Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments. *IEEE Transactions on Control Systems Technology*, 27(6):2388–2402, 2019.
- [23] Michael Hutchinson, Cunjia Liu, and Wen-Hua Chen. Source term estimation of a hazardous airborne release using an unmanned aerial vehicle. *Journal of Field Robotics*, 36(4):797–817, 2019.
- [24] J. Jackson, G. Ellingson, and T. McLain. Rosflight: A lightweight, inexpensive mav research and development tool. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 758–762, 2016.
- [25] Ivan Kalinov, Evgenii Safronov, Ruslan Agishev, Mikhail Kurenkov, and Dzmitry Tssetserukou. High-precision uav localization system for landing on a mobile collaborative robot based on an ir marker pattern recognition. *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Apr 2019.
- [26] Vsevolod Khithov, Alexander Petrov, Igor Tishchenko, and Konstantin Yakovlev. Toward autonomous uav landing based on infrared beacons and particle filtering. *Robot Intelligence Technology and Applications 4*, page 529–537, Jul 2016.
- [27] Michael Klinker. Tethered UAV Flight using a Spherical Position Controller. Master’s thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2016.
- [28] Anil Kumar and Pinhas Ben-Tzvi. Novel wireless sensing platform for experimental mapping and validation of ship air wake. *Mechatronics*, 52:58–69, 2018.
- [29] Ming Li, Zhuang Chang, Zhen Zhong, and Yan Gao. Relative localization in multi-robot systems based on dead reckoning and uwb ranging. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–7, 2020.
- [30] H. W. LIEPMANN. On the application of statistical concepts to the buffeting problem. *Journal of the Aeronautical Sciences*, 19(12):793–800, 1952.
- [31] Iker Lluvia, Elena Lazkano, and Ander Ansuategi. Active mapping and robot exploration: A survey. *Sensors*, 21(7), 2021.
- [32] Christopher J. Mallon, Benjamin J. Muthig, Kanishke Gamagedara, Kalpesh Patil, Chen Friedman, Taeyoung Lee, and Murray R. Snyder. Measurements of ship air wake using airborne anemometers. *55th AIAA Aerospace Sciences Meeting*, Jan 2017.
- [33] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A study of vicon system positioning performance. *Sensors*, 17(7):1591, 2017.

- [34] Jason Metzger, Murray R. Snyder, John Blacksher Burks, and Hyung Suk Kang. Measurement of ship air wake impact on a remotely piloted aerial vehicle. 2012.
- [35] R. Pavlovsky, J. W. Cates, W. J. Vanderlip, T. H. Y. Joshi, A. Haefner, E. Suzuki, R. Barnowski, V. Negut, A. Moran, K. Vetter, and B. J. Quiter. 3d gamma-ray and neutron mapping in real-time with the localization and mapping platform from unmanned aerial systems and man-portable configurations, 2019.
- [36] Ryan Pavlovsky, Andrew Haefner, Tenzing Joshi, Victor Negut, Keith McManus, Erika Suzuki, Ross Barnowski, and Kai Vetter. 3-d radiation mapping in real-time with the localization and mapping platform lamp from unmanned aerial systems and man-portable configurations, 2018.
- [37] David Phelps, Kanishke Gamagedara, Jeremey Waldron, Kalpesh Patil, and Murray Snyder. Ship air wake detection using small fixed wing unmanned aerial vehicle. *2018 AIAA Aerospace Sciences Meeting*, Jan 2018.
- [38] Susan A. Polksky, Colin Wilkinson, Jay Nichols, Douglas Ayers, Johana Mercado-Perez, and T. Scott Davis. Development and application of the safedi tool for virtual dynamic interface ship airwake analysis. *54th AIAA Aerospace Sciences Meeting*, Jan 2016.
- [39] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *CoRR*, abs/1708.03852, 2017.
- [40] Jorge Peña Queralta, Li Qingqing, Fabrizio Schiano, and Tomi Westerlund. Vio-uwb-based collaborative localization and dense scene reconstruction within heterogeneous multi-robot systems. *CoRR*, abs/2011.00830, 2020.
- [41] J. Rankin. An error model for sensor simulation gps and differential gps. In *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS'94*, pages 260–266, 1994.
- [42] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.
- [43] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, Portland, OR, USA, June 2013. IEEE.
- [44] Patrik Schmuck and Margarita Chli. Multi-uav collaborative monocular slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3863–3870, 2017.

- [45] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.
- [46] Yongjie SHI, Xiang HE, Yi XU, and Guohua XU. Numerical study on flow control of ship airwake and rotor airload during helicopter shipboard landing. *Chinese Journal of Aeronautics*, 32(2):324 – 336, 2019.
- [47] S. Shukla, S.S. Sinha, and S.N. Singh. Ship-helo coupled airwake aerodynamics: A comprehensive review. *Progress in Aerospace Sciences*, 106:71 – 107, 2019.
- [48] Joan Solà. Quaternion kinematics for the error-state kalman filter. *CoRR*, abs/1711.02508, 2017.
- [49] Hannes Sommer, Igor Gilitschenski, Michael Bloesch, Stephan Weiss, Roland Siegwart, and Juan I. Nieto. Why and how to avoid the flipped quaternion multiplication. *CoRR*, abs/1801.07478, 2018.
- [50] Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. Flightmare: A flexible quadrotor simulator. 2020.
- [51] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.
- [52] Hauke Strasdat and Others. Sophus. <https://github.com/strasdat/Sophus>.
- [53] A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, and J. Nieto. Collaborative transportation using mavs via passive force control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5766–5773, 2017.
- [54] Yulun Tian, Kasra Khosoussi, and Jonathan P. How. Resource-Aware Algorithms for Distributed Loop Closure Detection with Provable Performance Guarantees. *arXiv:1901.05925 [cs]*, January 2019. arXiv: 1901.05925.
- [55] Yulun Tian, Kasra Khosoussi, David M. Rosen, and Jonathan P. How. Distributed certifiably correct pose-graph optimization, 2021.
- [56] Charles E. Tinney, Jeremy Shipman, and Praveen Panickar. Reduced-order models for characterizing ship airwake interactions. *2018 Fluid Dynamics Conference*, Jun 2018.
- [57] Andrew J. Torgesen, Jonathan P. How, and Benjamin Cameron. *Airborne Sensing for Ship Air Wake Surveys with a Tethered Autonomous UAV*.
- [58] Dominik Van Opdenbosch, Tamay Aykut, Nicolas Alt, and Eckehard Steinbach. Efficient Map Compression for Collaborative Visual SLAM. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 992–1000, March 2018.
- [59] Velodyne. Vlp-16 (puck lite). Online, August 2018. <https://velodynelidar.com/vlp-16-lite.html>. Accessed August 2018.

- [60] VICON. Motion capture systems. Online, August 2018. <http://www.vicon.com/>. Accessed August 2018.
- [61] X. Q. Wang, D. Sarhaddi, Z. Wang, M. P. Mignolet, and P. C. Chen. Modeling-based hyper reduction of multidimensional cfd data: Application to ship airwake data. In *AIAA Scitech 2019 Forum*, AIAA Scitech 2019 Forum. American Institute of Aeronautics and Astronautics Inc, AIAA, January 2019. AIAA Scitech Forum, 2019 ; Conference date: 07-01-2019 Through 11-01-2019.
- [62] Weisong Wen, Xiwei Bai, Guohao Zhang, Shengdong Chen, Feng Yuan, and Li-Ta Hsu. Multi-agent collaborative gnss/camera/ins integration aided by inter-ranging for vehicular navigation in urban areas. *IEEE Access*, 8:124323–124338, 2020.
- [63] Hao Xu, Luqi Wang, Yichen Zhang, Kejie Qiu, and Shaojie Shen. Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm. *CoRR*, abs/2003.05138, 2020.
- [64] Shichao Yang and Sebastian Scherer. CubeSLAM: Monocular 3D Object SLAM. *IEEE Transactions on Robotics*, 35(4):925–938, August 2019. arXiv: 1806.00557.
- [65] Weixing Yuan, Alanna Wall, and Richard Lee. Combined numerical and experimental simulations of unsteady ship airwakes. *Computers & Fluids*, 172:29 – 53, 2018.
- [66] Jianan Zhu and Solmaz S. Kia. An imm-based decentralized cooperative localization with los and nlos uwb inter-agent ranging, 2020.
- [67] Danping Zou, Ping Tan, and Wenxian Yu. Collaborative visual SLAM for multiple agents:A brief survey. *Virtual Reality & Intelligent Hardware*, 1(5):461–482, October 2019.