Doctoral theses at NTNU, 2022:339

Martin Lysvand Sollie

Navigation and Automatic Recovery of Fixed-wing Unmanned Aerial Vehicles

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Martin Lysvand Sollie

Navigation and Automatic Recovery of Fixed-wing Unmanned Aerial Vehicles

Thesis for the Degree of Philosophiae Doctor

Trondheim, November 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-326-6247-0 (printed ver.)
ISBN 978-82-326-6547-1 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:339

Printed by NTNU Grafisk senter

Summary

Today, unmanned aerial vehicles (UAVs) are seeing widespread use as a tool used by the industry, researchers, hobbyists and others around the world. Fixed-wing UAVs, although not in as widespread use a multirotor UAVs, have benefits with longer flight time and range. Low-cost systems typically base their navigation systems on global navigation satellite systems (GNSSs), and are susceptible to natural or artificial interference of GNSS signals. Additionally, while automatic launch of fixed-wing UAVs using a catapult is common, automatic recovery systems are not, which is of special relevance for operating UAVs from ships in the maritime industry. A skilled pilot is typically used for UAV recovery, requiring additional cost and limiting the operation to conditions with sufficient visibility. This thesis considers two main topics: First, UAV navigation both with increased accuracy using GNSS, and local navigation without GNSS for increased resilience. Second, systems for automatic recovery of fixed-wing UAVs.

The first contribution explores the use of a low-cost navigation setup using two independent GNSS receivers, aiding an inertial navigation system by using tighly coupled integration of pseudorange, Doppler frequency and carrier phase measurements from two longitudinally separated GNSS antennas on a fixed-wing UAV. The goal is increased navigation accuracy for high-precision applications, such as georeferencing and photogrammetry. The multiplicative extended Kalman filter (MEKF) is used as the estimation algorithm. Measurement models are derived for the raw GNSS measurements based on the MEKF error state, taking into account antenna lever arms and explicitly including the difference in measurement time between the receivers in the measurement model for double differenced carrier phase. The proposed method is verified using data collected from a UAV flight, and different methods for receiver synchronization is compared.

In a second contribution, the handling of the covariance matrix in connection to the reset of the MEKF error-state after a measurement correction is considered. This is a topic there is disagreement about in the existing literature. Previous work on this topic is reviewed, and it is argued for one specific covariance estimate modification.

The third contribution demonstrates that the recently introduced direction finding feature of Bluetooth, using antenna arrays, is usable for UAV navigation outdoors at a range of up to 700 m. It is shown how using repeated measurements from all array elements, instead of only the initial single-element reference samples as often suggested, can contribute to an improved estimate of the signal's unknown carrier frequency offset, thereby improving the direction estimation performance. To run the direction-of-arrival estimation in real-time with high angular resolution on an embedded computer, we propose a pseudospectrum peak search strategy that combines a coarse search and a local nonlinear optimization for estimate refinement. In an open outdoor environment, using a square antenna array with 12 elements, the azimuthal performance is found to be very consistent with range, with noise standard deviation typically around 1°. While the elevation is significantly affected by multipath at lower elevation angles, with visible disagreement between frequency channels, it is shown to be consistent with simulations of ground reflection multipath.

In a fourth contribution, considerations for choice of Bluetooth array sampling sequence are considered. Bluetooth arrays use a single receiver in combination with software controlled signal switches to receive the incoming signal using all array elements. This implies that measurements are not simultaneous, and that an order of sampling must be chosen. Time-symmetric sampling is considered to reduce the error in estimated direction of arrival due to error in the estimated signal frequency. It is found, using simulation and measurements from an array with 12 antennas in field experiments, that such sampling orders can remove the direct correlation between frequency error and direction error for errors up to approximately 10 kHz. This is at the expense of reduced signal-to-noise ratio as the frequency error increases, as the measurements themselves cancel to remove the effect of the frequency offset.

As a fifth navigation contribution, it is demonstrated how the effect of ground reflection multipath for Bluetooth navigation can be reduced by synchronizing measurements from two independent vertically stacked arrays, allowing hardware modularity using low-cost equipment. Using measurements from a 15 × 15 cm array in field experiments, we demonstrate a significant reduction in elevation error at elevation angles in the 7° to 15° range where the error was largest when using a single array.

In a sixth contribution, a system for automatic recovery of a fixed-wing UAV on a moving platform at sea is developed and demonstrated. The recovery functionality is added as a modular extension based on non-intrusive additions to a commercial off-the-shelf autopilot. Line-of-sight guidance for the UAV is used for line-following along a virtual runway that guides the UAV into the arrest system. The position of the UAV relative to the arrest system, and the orientation of the arrest system, is determined using GNSS receivers supporting Real-Time Kinematic (RTK) processing. The system is validated using two different UAVs in extensive field testing, first with a stationary recovery net, and then a net mounted on a barge towed by a ship.

In the seventh contribution, it is demonstrated that Bluetooth direction finding can be used as a navigation system for automatic recovery of a fixed-wing UAV, guiding the UAV into an arrest system such as a suspended net, independently from GNSSs. The effect of multipath signal interference on the elevation angle estimate is handled by a constant offset calibration. The system is verified in field experiments using a Skywalker X8 fixed-wing UAV and a Bluetooth antenna array.

As an eight and final contribution, a concept for planning approach trajectories for automatic recovery on ships is presented, where optimization is used to generate an approach trajectory that enables aborts as late as safely possible. The concept is demonstrated using a simple three dimensional UAV model, for several different scenarios. This is then compared to less complex approach plans. The implemented example demonstrates that optimization can be useful for planning approach trajectories and allow later aborts than the simple methods.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor (Ph.D.) at the Norwegian University of Science and Technology. The presented work has been carried out at the Department of Engineering Cybernetics with affiliation to the NTNU Center for Autonomous Marine Operations and Systems (AMOS). Professor Tor Arne Johansen has been the main supervisor, and Associate Professor Torleiv Håland Bryne has been the co-supervisor. The work has been supported by the Research Council of Norway and Maritime Robotics AS through the Marlander project, grant number 282427, and AMOS, grant number 223254.

As a part of my Ph.D. work, in addition to the work presented in this thesis, efforts also went towards gravimetric surveying from UAVs using low-cost sensors, which ended without the desired results for several reasons. This was a continuation of the work of postdoc Robert H. Rogne and the results presented in Chapter 2, where a UAV was equipped with four RTK GNSS receivers (antennas on wing tips, nose, and tail) and a Sensonor STIM318 MEMS inertial measurement unit (IMU), for estimating variations in the gravitational field. Gravity anomaly was estimated as a state in the navigation algorithm. This was an ambitious project where it was uncertain whether it would be possible to achieve a useful estimation accuracy with the sensors used. Unfortunately, many time-consuming practical obstacles halted the progress, including significant temperature gradients of the IMU after UAV launch due to airflow, vibrations from motor and propeller causing vibration rectification IMU errors, and flexibility of the airframe and wings making a rigid body assumption inaccurate. The first flight with a Penguin UAV ended with the combustion engine stopping mid-air while flying at Frøya, and the vehicle ending in the sea. A new payload then had to be constructed. A payload was also mounted in a helicopter, courtesy of the Geological Survey of Norway (NGU), where a cable running to a GNSS antenna in the tail boom would degrade over time due to vibrations, in addition to issues with GNSS signal coupling between coaxial cable bulkhead connectors that proved to have insufficient shielding. A long test flight over the sea from Smøla using a different electrically powered UAV was planned, but at a payload shakedown at Brekken the UAV stalled during manual flight and crashed, further delaying progress. While I learned a lot from this work, it is unfortunate that it ended without publishable results.

Acknowledgements

First I would like to thank my supervisor Tor Arne Johansen and co-supervisor Torleiv Håland Bryne, who were also my supervisors for my Master's thesis. In times where I have felt stuck and the way forward has seemed uncertain, a meeting in Tor Arne's office has always brought clarity and opened the way for progress. Torleiv: thanks for the interesting discussions and deep-dives into both theoretical and practical aspects of navigation. I appreciate you sharing your experiences on how taking a Ph.D. *really* works.

Thank you, Kristoffer Gryte, for the lab work, the many weeks spent in the field for experiments, paper writing, software debugging and problem solving we have done together. You are a master of project coordination.

During my work I have been lucky to do a lot of field experiments, moving beyond theory and simulations to make things work in the real world. This has involved many people making it all possible. Thanks to Pål Kvaløy at the department for the assistance and piloting in many of the experiments with unmanned aerial vehicles, and to Terje Haugen and Glenn Angell at the department workshop for making mechanical components used both in UAV payloads and for ground equipment. Erling Jellum provided important support for the use of SentiBoards in experiments. I must also thank the department administration, making it possible to prepare for and conduct field experiments during the restrictive times of the Covid-19 pandemic.

Much of the work has been done with support from or in collaboration with industrial partners: Lars Semb, Morten Einarsve, Torbjørn Houge and Carl Erik Stephansen of Maritime Robotics, Einar Nielsen of Petroleum Geo-Services (PGS), and Carsten Wulff, Paal Kastnes and Bjørn Spockeli of Nordic Semiconductor. Thank you. I would also like to thank Frode Ofstad at the Geological Survey of Norway (NGU) for the opportunity to fly a payload in a helicopter used in one of their survey missions.

Thanks to my office-mates Sverre Rothmund, Pål Mathisen and Sivert Bakken, and my colleagues at the UAV-Lab and department, Frederik Leira, Artur Zolich, Erlend Coates, Dirk Reinhardt, Lars-Christian Ness Tokle, for our many interesting discussions during these years.

Thanks to my parents for their support and patience, listening to all my attempts to explain the topics of my work.

Contents

Chapter 1

Introduction

This thesis considers navigation and automatic recovery of fixed-wing unmanned aerial vehicles (UAVs), in an effort to increase navigation accuracy and robustness, and increase autonomy in the recovery stage of UAV operations. This chapter motivates the topics considered, and presents research objectives and research questions considered in the remaining chapters. Following this, the main contributions and the articles in which they have been published are listed. Finally, the structure of the remainder of the thesis is presented.

1.1 Motivation and research objectives

Norway is a country with a long coastline, a large exclusive economic area, and large fishing zones around Svalbard and Jan Mayen. Petroleum products from offshore petroleum production and seafood contributed about 47% and 13% of Norwegian exports in 2020, respectively [143], and the sea therefore plays an important role in the wealth of the Norwegian people. With plans of offshore wind farms as a future energy source in Norway [40], the importance of the sea is likely to continue. Research on the sea itself and its industrial usage are therefore important fields at Norwegian academic institutions, such as the work on autonomous remote sensing using unmanned aerial vehicles (UAVs), satellites and autonomous surface and underwater vehicles at NTNUs Centre for Autonomous Marine Operations and Systems (AMOS).

Industrial partners of AMOS-affiliated research projects have expressed a desire for use of autonomous UAVs for remote sensing in the maritime environment. Fixed-wing UAVs can cover large areas with the ability to detect and track objects on the sea surface at a lower cost than manned aircraft. Examples of applications are detection and tracking of lost fishing equipment that become obstacles for seismic survey operations, wildlife monitoring, transport of goods between ships, platforms, and the coast, or monitoring of algal blooms near fish farms to improve situational awareness. UAVs can also contribute to surveillance for contingency and emergency preparedness such as in the case of oil spills or search-and-rescue operations at sea, and enforcement of fishing legislation or pollution monitoring. The reduced cost of operating UAVs compared to manned aircraft, and the lower cost of the vehicles themselves, makes UAVs attractive for many uses in the industry, as the threshold for taking such vehicles into use is lower. Unmanned vehicles also have the benefit of reduced risk in the event of adverse weather conditions, where manned aircraft cannot fly due to the safety of the flight crew.

A desired part of detection and tracking of objects at sea using airborne sensors is to accurately georeference the object observations, meaning to be able to determine where the object is, and not just where the UAV is. This requires accurate estimates of the UAV position, and especially its attitude. Small errors in the estimated attitude will result in errors in the georeferenced position that grows as the distance from the object increases. UAVs typically use a single global navigation satellite system (GNSS) receiver combined with low-cost inertial sensors, but flying in patterns consisting of long straight lines with low dynamics, as typical when surveying large areas, results in poor heading observability. Magnetometers can provide aiding for the heading angle but are easily disturbed and work poorly at high latitudes, where the magnetic field lines have a small horizontal component. Commercially available multi-antenna GNSS systems solve this problem, but at a high cost. Improving the UAV navigation estimates while still using low-cost equipment is therefore a research objective where solutions can increase the capabilities of UAVs in widespread use.

While high-accuracy state estimation is desirable in some applications, the ability to operate under degradation of GNSS signals is important for risk reduction in UAV operations. GNSS signals have low power when reaching the Earth's surface, and are therefore easily overpowered by the deliberate transmission of noise, which is known as *jamming*. It is therefore desirable to have resilience to loss of GNSS signals during flight, or inaccurate GNSS position estimates due to degraded signals. Navigation redundancy during operations requiring high accuracy and precision, such as operations close to a ship, can contribute to increased safety and trust in the system, allowing wider limits on the conditions in which the UAV can operate. It is therefore a research objective to find accurate local navigation systems which can be used for UAV navigation outdoors independently of GNSS, and independenly of weather conditions.

Unlike manned fixed-wing aircraft, typical fixed-wing UAVs can easily be launched from ships due to their small size. Automatic recovery is, however, not widely used, and is a limiting factor for use of UAVs in the maritime industry. Recovery using manual flight requires a skilled pilot but is still difficult under adverse conditions with strong or gusty winds and limited visibility. By automating the recovery stage, UAVs can perform completely automated missions from ships, thereby reducing the need for operator training and improving control robustness under windy conditions or with large ship movements due to waves. By operating from ships, UAVs can reach remote areas out of reach of small manned aircraft, such as in the Arctic. To allow the use of existing UAVs in such scenario, it is a research objective to find a solution for automatic recovery that builds on top of existing avionics, and does not require intrusive changes to autopilot software.

1.2 Existing methods and new research questions

Improved estimates of heading can be obtainted by the use of dual-antenna GNSS, and with three or more antennas full attitude can also be determined. Commercial systems using GNSS for heading or attitude, such as the Vectornav VN-300 [154], are however significantly more expensive than commonly used autopilot systems such as the Pixhawk series of flight controllers [106]. This motivates the use of low-cost commercial-off-the-shelf GNSS receivers for aiding in estimation of heading or attitude. The determination of attitude using dedicated multi-antenna receivers, where a single common clock is used for all tracking, has been researched by multiple authors [56, 28, 29, 30, 167, 153]. The use of a common clock, which is not the case when multiple independent receivers are used, simplifies the measurement processing, but requires dedicated hardware. Low-cost receivers from the company u-Blox are commonly used with common open source avionics, and some of their receivers are capable of output of raw measurements including carrier phase, which can enable high accuracy attitude determination using GNSS, tightly integrated with inertial sensors. Combined use of independent GNSS receivers with receiver synchronization was considered by [38, 52]. [38] suggests time extrapolating the carrier phase measurements using the measured Doppler frequency shift but only considered this for a stationary baseline, while [52] proposes a clock correction based on code-only least squares position and clock error estimates. For fusion of GNSS and inertial navigation is it, however, a question whether aiding of the receiver synchronization using smooth INS state estimates, combined with measurement models for the raw GNSS measurements, can improve the results compared to less complex methods.

GNSS receivers are the most common position source used for outdoor navigation, but the very weak signals are susceptible to interference such as jamming and ionospheric scintillation. As a result, having access to supplements or alternatives to GNSS is beneficial to ensure navigation robustness. Several technologies have been demonstrated for this purpose. Visual navigation is a popular technique for relative navigation [70, 77]. What makes this approach tractable is the possibility to construct a self-contained system that does not rely on external communication or measurements, that delivers relative pose measurements at a high rate, like e.g [65, 53, 144, 110]. Drawbacks of visual navigation include high processing requirements, risk of false detection, and sensitivity to visual conditions, such as light/weather conditions and distinctiveness of the tracked features in the images. The latter can to some extent be mitigated by using infrared (IR) cameras, either using natural landmarks [64, 165] or IR lamps in known locations [48]. Radio-based navigation have the benefit of all-weather operability. Ultra-wideband (UWB) ranging with fixed anchors [39, 79, 93, 46, 22] can be used for positioning. To estimate position from these range measurements, the anchors should be positioned to create a sufficiently good measurement geometry, with a low dilution of precision (DOP), the same as is the case for GNSS pseudorange measurements. This is a drawback if the UAV operates from a small ship, where the maximum spread of the anchors can be small, resulting in poor measurement geometry. Phased array radio systems such as [44] can provide accurate positioning with a small equipment footprint, but may be too expensive for many applications and may be limited to licenced frequency bands. Antenna arrays can also be used with UWB hardware [20, 14], and since arrays yield much more precise and accurate relative range measurements than independent anchors, the equipment footprint can be significantly reduced. In 2019, the Bluetooth Special Interest Group (SIG) presented the Bluetooth 5.1 specification, which introduced support for direction finding using antenna arrays. A benefit of Bluetooth direction finding compared to UWB is that Bluetooth uses a single receiver and signal switches to sequentially connect the receiver to each element of the array, while UWB arrays use a separate receiver channels for every element. The cost for a UWB array with many elements will therefore likely be higher than for Bluetooth. Bluetooth also has the benefit of hardware produced in large volumes for consumer electronics, causing component cost to be low. A question of interest then is whether Bluetooth direction finding equipment can be used for UAV navigation outdoors with a useful accuracy, precision, and range.

Recovery of UAVs on a moving platform is often limited by available space. One viable approach to enable recovery on a space-limited, moving platform is to design the operation around a fixed-wing UAV with vertical takeoff and landing (VTOL) capabilities, i.e. a rotary-wing/fixed-wing hybrid [104]. The increased maneuverability and hover capabilities associated with VTOL UAVs make them easier to land, but this comes at a cost of increased drag, mass and complexity, and decreased payload capacity. While fixed-wing VTOL UAVs and rotary-wing only require a flat surface to land on a moving platform, as in [111, 2, 116], conventional fixed-wing UAVs relies on arrest recovery systems to land on a space-limited, moving platform. This enables the design of the UAV to be focused on the main mission, which is usually what adds value for the end user. Relative navigation off-the-shelf options include the laser-based *Object Position and Tracking System* (OPATS) [113], GPS- and radar-based *Dual-Thread Automatic Takeoff and Landing System* (DT-ATLS) [123, 50], as well as the integrated navigation and control solution *UAV Common Automatic Recovery System* (UCARS) [124] for ship landing. Although these systems are well proven, they are also proprietary commercial systems with unknown algorithms and with little flexibility to make customizations. Even though some commercially available autopilots are capable of automatic landing in fixed locations, such as [9], this does not suffice for a moving arrest system. A research question is then if an automatic recovery system for a fixed-wing UAV building non-intrusively on a common open-source autopilot can be developed. Additionally, provided that we find Bluetooth direction finding to be suitable for UAV navigation, can it be used to perform recovery of a fixed-wing UAV in degraded GNSS signal conditions?

With the ability to monitor mission-critical and changing conditions such as wind gusts, and the health of communication links and navigation sensors, autopilots can quickly abort a recovery attempt in case it is considered the safer option. In the existing literature, some work has been done on decision making of UAV aborts, focusing mostly on runway landings where no obstacles need to be considered, or aborting missions. [55] designs a controller for runway landing, and determines a position region from which the landing is considered feasible with the controller used. If the UAV ends up outside this region during the landing, an abort is triggered and the UAV is directed to retry the approach. [72] describes a self-monitoring algorithm that can automatically abort a runway landing based on UAV position and velocity, and status of sensors and communication links, to perform a go-around to a new landing attempt. In [76] a mission abort policy for a vehicle exposed to external impacts is described, where the impacts increase the risk of system failure, and a tradeoff between probabilites for system survivability and mission success is considered. [166] continues this work, attempting to design a mission abort strategy that minimizes the total expected economic loss considering both the cost of a failed mission and vehicle failure, for a UAV operation where early-warning signals such as overheating, vibrations and intermittent loss of communications can indicate problems that increase the risk of system failure. To the best of our knowledge, no work on aborts of UAV recoveries on ships considering collision avoidance is found in existing literature. Therefore, it is an interesting research question how a trajectory for a UAV performing an automatic recovery can be chosen to allow the autopilot to safely abort the approach as late as possible before the recovery plan ends.

1.3 Contributions

Based on the above research questions, the thesis contributes to the research community by

- Deriving measurements models for tightly coupled integration of two independent low-cost GNSS-receivers providing raw measurements, including carrier phase interferometry for attitude aiding, and comparing the results of using different receiver measurement time corrections, using data from a UAV flight. (Chapter 2)
- Providing arguments for one specific transformation of the estimated covariance for the multiplicative extended Kalman filter error-state reset. This has been a topic of disagreement in the literature for many years, where a rigorous and correct solution should improve attitude covariance estimation (Chapter 3)
- Demonstrating in field experiments how Bluetooth direction finding can be used for UAV navigation, as a local system increasing resilience to GNSS dropouts, surpassing 500m range (Chapter 4)
- Extending previous theoretical considerations on sampling sequence considerations for switched antenna arrays, with considerations for a real Bluetooth system and experimental testing. This can contribute to improved accuracy for Bluetooth navigation (Chapter 5)
- Deriving and experimentally demonstrating a method for reducing the effect of ground reflection multipath on Bluetooth elevation estimates by combined pro-

cessing of measurements from two separate low-cost arrays, yielding increased elevation angle accuracy (Chapter 6)

- Developing a system for automatic recovery of a fixed-wing UAV in a moving arrest system, building non-intrusively on an commercial-off-the-shelf autopilot, and experimentally verifying the system in experiments at sea (Chapter 7)
- Developing and experimentally demonstrating that Bluetooth direction finding can be used as an alternative to GNSS for navigation during the final stage of automatic fixed-wing UAV recovery (Chapter 8)
- Proposing a method for approach path planning for automatic recovery of fixed-wing UAVs on ships, which allows the approach to be aborted as late as possible for increased safety (Chapter 9)

1.4 Publications

This thesis is based on the following articles published in, or submitted to, peer-reviewed international journals and conferences:

- [129] M. L. Sollie, T. H. Bryne, and T. A. Johansen, "Pose Estimation of UAVs Based on INS Aided by Two Independent Low-Cost GNSS Receivers," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 1425–1435. DOI: 10.1109/ICUAS.2019.8797746
  Note that this paper is partly based on my master thesis [136]
- [130] M. L. Sollie, T. H. Bryne, and T. A. Johansen, "On the Error-Reset Covariance Transform for the Multiplicative Extended Kalman Filter," *Submitted, in revision*, 2022
- [132] M. L. Sollie, K. Gryte, T. H. Bryne, and T. A. Johansen, "Outdoor Navigation Using Bluetooth Angle-of-Arrival Measurements," *IEEE Access*, vol. 10, pp. 88 012–88 033, 2022. DOI: 10.1109/ACCESS.2022.3199772
- [135] M. L. Sollie, T. A. Johansen, K. Gryte, and T. H. Bryne, "Sampling Sequence Considerations for Bluetooth Angle-of-Arrival Estimation," *Submitted, in revision*, 2022
- [134] M. L. Sollie, T. A. Johansen, K. Gryte, and T. H. Bryne, "Reducing Ground Reflection Multipath Errors for Bluetooth Angle-of-Arrival Estimation by Combining Independent Antenna Arrays," *Submitted, in revision*, 2022
- [47] K. Gryte, M. L. Sollie, and T. A. Johansen, "Control System Architecture for Automatic Recovery of Fixed-Wing Unmanned Aerial Vehicles in a Moving Arrest System," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 103, no. 4, 2021. DOI: 10.1007/s10846-021-01521-z
- [131] M. L. Sollie, K. Gryte, T. H. Bryne, and T. A. Johansen, "Automatic Recovery of Fixed-wing Unmanned Aerial Vehicle Using Bluetooth Angle-of-Arrival Navigation," *Submitted, in revision*, 2022

- [133] M. L. Sollie and T. A. Johansen, “Planning approach trajectories to enable late aborts for fixed-wing UAV recovery on ships,” in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 311–320. DOI: 10.1109/ICUAS51884.2021.9476706

The following article has been published during the Ph.D. work, but is not part of the thesis:

- [67] J. Kim, J. Guivant, M. L. Sollie, T. H. Bryne, and T. A. Johansen, “Compressed pseudo-SLAM: Pseudorange-integrated compressed simultaneous localisation and mapping for unmanned aerial vehicle navigation,” *Journal of Navigation*, vol. 74, no. 5, pp. 1091–1103, 2021. DOI: 10.1017/S037346332100031X

1.5 Outline

The thesis consist of two main parts: navigation in Part I, and automatic UAV recovery in Part II.

In Part I, Chapter 2 presents aided inertial navigation using raw measurements from two GNSS receivers. The handling of the covariance matrix when an observed attitude error is injected into the MEKF nominal state, which was not taken into account in Chapter 2, is considered in Chapter 3. UAV navigation using Bluetooth navigation is presented in Chapter 4. The two next chapters extend the Bluetooth work. Chapter 5 considers the choice of sampling sequence used by the array, and Chapter 6 demonstrates how systematic errors in the elevation angle estimate can be reduced by using two independent arrays acting as one large array by synchronizing their measurements.

In Part II, Chapter 7 presents a system for automatic recovery of fixed-wing UAVs in a moving arrest system, using GNSS navigation. Chapter 8 demonstrates how Bluetooth navigation can be used as an alternative to GNSS for an automatic recovery system. Chapter 9 presents a method for planning recovery approach trajectories that allow the approach to be aborted as late as possible for arrest system recovery on a ship.

Finally, Chapter 10 summarizes the work and suggests some directions for future work.

Part I

Navigation

Chapter 2

Pose Estimation of UAVs Based on INS Aided by Two Independent Low-Cost GNSS Receivers

This chapter is based on the publication

- [129] M. L. Sollie, T. H. Bryne, and T. A. Johansen, “Pose Estimation of UAVs Based on INS Aided by Two Independent Low-Cost GNSS Receivers,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 1425–1435. DOI: 10.1109/ICUAS.2019.8797746

2.1 Introduction

The small unmanned aerial vehicles (UAVs) used by hobbyists and professionals today are commonly equipped with an autopilot system estimating its position, velocity and attitude using one or more inertial measurement units (IMUs) containing gyros and accelerometers, a single global navigation satellite system (GNSS) receiver, a barometer and a magnetometer (magnetic compass). Examples of such systems are the Pixhawk series of flight controllers [106]. GNSSs provides position and velocity measurements which are free of long-term drift. However, there are many challenges with GNSS as a source of position and velocity information. The measurements are normally available at a rate too low for feedback control in highly dynamic systems, such as UAVs, and the measurements can be noisy. Because the signals received from the satellites have low power, the receivers can be disturbed by interference, jamming or spoofing. Obstructions between the receiver and satellites can block the signals, making it less suitable in valleys or dense urban environments and basically unsuitable for indoor navigation.

The use of an inertial navigation system (INS), consisting of an IMU and the processing required to estimate position, velocity and attitude, offers several advantages. As it is completely self contained and does not rely on any external signals, it is not exposed to external interference. The measurements also typically have low noise and are available at a high rate, giving smooth position and attitude outputs. The drawback of inertial navigation is that all IMUs experience slowly varying errors that cause position and attitude estimates, based on the mechanization of angular rate and specific force, to drift over time.

Due to the complementary nature of INS and GNSS, combining the measurements using an estimation algorithm such as the Kalman filter [59] can give the best of both worlds. The long-term drift is eliminated by estimating and compensating the INS errors using GNSS measurements, while the INS is used to smooth the output and provide position, velocity and attitude (PVA) estimates even when not receiving GNSS receiver measurement. The use of tightly coupled integration, where raw GNSS measurements are used directly, instead of intermediate calculation of GNSS-only state estimates, allows correction of the INS estimates even when only a few satellites are visible such that a complete GNSS position solution is not available.

UAVs using today's low cost autopilot systems are mostly used for waypoint flying and other tasks where small attitude errors pose no large problem as a deviation from the desired flight path, which is visible in the observed position and velocity errors, is handled using feedback control. For these uses having a correct course (direction of movement) is more important than a correct heading. Other applications however, such as georeferencing, photogrammetry and mapping, require more accurate state estimates. It is especially the accurate estimation of the vehicle heading which is a challenge [37]. For UAVs flying in a steady state, or hovering, with low acceleration and angular rate, the errors in attitude and IMU biases are not observable with only a single GNSS antenna, and the estimate will rely on the magnetic compass. Magnetometers are susceptible to disturbances from irregularities in the Earth's magnetic field, or ferrous materials or electrical currents close to the sensor [37]. They are also not very useful when navigating near the magnetic poles, and the local magnetic declination values for the areas of operation must be known. Improved estimates of heading can be obtainted by the use of dual-antenna GNSS, and with three or more antennas full attitude can also be determined. Commercial systems using GNSS for heading or attitude, such as the Vectornav VN-300 [154], are however significantly more expensive than the autopilot systems discussed.

This motivates the use of low-cost commercial-off-the-shelf receivers for aiding in estimation of heading or attitude. The determination of attitude using dedicated multi-antenna receivers, where a single common clock is used for all tracking, has been researched by multiple authors [56, 28, 29, 30, 167, 153]. The use of a common clock, which is not the case when multiple independent receivers are used, simplifies the measurement processing, but requires dedicated hardware. The use of non-dedicated receivers has two main properties that must be adressed:

1. Different receivers have different clock errors.

2. Measurements from the different receivers are not necessarily sampled at the same instant.

The second point is mainly a result of the first, as the receivers schedule raw measurement outputs according to their local clocks, but also occurs if the receivers schedule their measurements at different times according to their own clock, i.e. if one receiver measures at the clock's whole seconds while the other does not. The first point is easily adressed by double differencing the measurements, cancelling the receiver clock errors, but this does not handle the difference in measurement time. This is of less or no importance if higher grade receivers are used, that either steer their clock to the GNSS system time, or correct the clock often (i.e. every time a navigation solution is calculated) such that the clock error remains very small. For receivers used in today's consumer products, such as those made by U-Blox AG [149], the clocks are however free to drift within set thresholds on the millisecond level without being corrected. Only if the threshold is violated will a millisecond step correction be applied to the clock. This can lead to measurements being taken a few milliseconds apart. The handling of this is a necessary part of real-time kinematic positioning such as RTKLIB [140], where measurements from the base receiver can be delayed or sampled at a different rate than the rover measurements. [38] suggests time extrapolating the carrier phase measurements using the measured Doppler frequency shift, while [52] proposes a clock correction based on code-only least squares position and clock error estimates. [140] uses the estimated receiver velocity to correct the base station position before the baseline processing.

2.1.1 Main contribution

The main contribution of this paper is tightly coupled integration of dual receiver GPS L1 C/A measurements with inertial navigation using a multiplicative extended Kalman filter (MEKF) [81] (with the attitude error state based on the Gibbs vector), estimating the attitude as a unit quaternion in addition to position, velocity, IMU biases, carrier phase ambiguities and receiver clock offsets and drift rates. The difference in measurement time is explicitly included in the carrier phase measurement model as a function of the estimated vehicle state, also including this in the linearized model used for covariance correction. Thus the corrected smooth INS output is used for measurement time correction, unlike pure GNSS-based estimates or measurements in [38] and [52]. The lever arm of the receivers and the rotation-induced velocity is included in the measurement models, such that the estimated position and velocity is valid for the position of the IMU. The estimation algorithm is tested using data collected during a fixed-wing UAV flight.

The paper proceeds as follows: We first describe the measurement models used for the raw GNSS measurements and the interferometric use of carrier phase for measuring attitude. The compensation of differences in measurement time for carrier phase interferometry is described, before briefly introducing the multiplicative extended Kalman filter (MEKF) and the system dynamics model used. The measurement models used for the MEKF correction step is then presented. The carrier phase measurement model is formulated to take the difference in measurement times into account. Results from verification of the proposed method using the collected UAV flight data is then presented and discussed. Finally, the concluding marks are presented with suggestions for further work.

2.2 GNSS observables

2.2.1 Pseudorange

The pseudorange is the difference between the time of signal reception according to the receiver and the time of signal transmission according to the satellite, scaled to distance using the speed of light c. Receiver and satellite clock errors, ionospheric and tropospheric delays, multipath and other sources all cause errors in ranging. The true geometric range between the antenna of a satellite s and receiver α travelled by the signal in absence of atmospheric effects, using Earth Centered Earth Fixed (ECEF) coordinates, is

$$\begin{aligned} \rho_{\alpha,s}(t_{rx}) &= \|\mathbf{R}^{e,rx}_{e,tx}\boldsymbol{p}^{e,tx}_{es}(t_{tx}) - \boldsymbol{p}^{e,rx}_{e\alpha}(t_{rx})\|_2, && (2.1)\\ &= c(t_{rx} - t_{tx}), && (2.2) \end{aligned}$$

where $\boldsymbol{p}^{e,rx}_{e\alpha}(t_{rx})$ is the position of the receiver antenna at the time of signal reception t_{rx}, given in the ECEF coordinate frame of the same time, $\{e, rx\}$, and $\boldsymbol{p}^{e,tx}_{es}(t_{tx})$ is the position of the satellite antenna at the time of signal transmission t_{tx} in the ECEF frame $\{e, tx\}$. The rotation matrix $\mathbf{R}^{e,rx}_{e,tx} = \mathbf{R}_z(\omega_{ie}(t_{rx} - t_{tx}))$ around the ECEF z-axis accounts for Earth rotation rate ω_{ie} during the signal propagation time, such that the positions used in the expression for range are in the same frame [42]. With the effects of ionospheric and tropospheric delays, and satellite and receiver clock errors included, the psedorange for a single satellite can be modelled as [91]

$$P_{\alpha,s}(t_{rx,\alpha}) = \rho_{\alpha,s}(t_{rx,\alpha}) + c(\delta t_\alpha(t_{rx,\alpha}) - \delta t_s(t_{tx,s})) + I_{\alpha,s} + T_{\alpha,s} + \epsilon_P, \tag{2.3}$$

where $P_{\alpha,s}$ is the measured pseudorange, δt_α and δt_s are the receiver and satellite clock errors, respectively, $I_{\alpha,s}$ and $T_{\alpha,s}$ are the ionospheric and tropospheric delays and ϵ_P is noise and unmodelled errors such as multipath. The times $t_{rx,\alpha} = t_{rx} + \delta t_\alpha$ and $t_{tx,s} = t_{tx} + \delta t_s$ are the time of the receiver and satellite clocks at the true times of reception and transmission. The time of transmission of the satellite clock can be found using the relation $t_{tx,s} = t_{rx,\alpha} - \frac{P}{c}$ and the true time can then be found by calculating the correction δt_s using parameters in the received navigation data.

2.2.2 Doppler frequency shift

The receiver tracks the carrier frequency of the received signal, often as a part of carrier phase tracking. The frequency received depends on the relative movement of the satellite and the receiver due to the Doppler shift. This raw measurement relates to the pseudorange P and carrier phase ϕ measurements as [42]

$$\dot{P} = \lambda\dot{\phi} = -\lambda\Delta f_{\mathrm{m}}, \tag{2.4}$$

where λ is the wavelength, and Δf_{m} is the difference between the measured carrier frequency and the nominal carrier frequency, $\Delta f_{\mathrm{m}} = f_{\mathrm{measured}} - f_{L1}$. Because the rate of change of the ionospheric and tropospheric errors are small, we ommit these terms and use the model

$$\lambda\dot{\phi}_{\alpha,s}(t_{rx,\alpha}) = \dot{\rho}_{\alpha,s}(t_{rx,\alpha}) + c(\dot{\delta t}_{\alpha}(t_{rx,\alpha}) - \dot{\delta t}_{s}(t_{tx,s})) + \epsilon_{\dot{\phi}}. \tag{2.5}$$

Earth rotation also needs to be taken into consideration when using range rate to estimate the antenna velocity. The velocity contribution of the rotating ECEF frame relative to inertial space, with the lever arms from the center of the Earth, should be included in the model [42],

$$\begin{aligned}\dot{\rho}_{\alpha,s}(t_{rx}) = (\boldsymbol{l}^{e,rx})^{\top}\Big(\mathbf{R}^{e,rx}_{e,tx}(\boldsymbol{v}^{e,tx}_{es}(t_{tx}) + \mathbf{S}(\boldsymbol{\omega}^{e}_{ie})\boldsymbol{p}^{e,tx}_{es}(t_{tx})) - \\ (\boldsymbol{v}^{e,rx}_{ea}(t_{rx}) + \mathbf{S}(\boldsymbol{\omega}^{e}_{ie})\boldsymbol{p}^{e,rx}_{ea}(t_{rx}))\Big),\end{aligned} \tag{2.6}$$

where $\boldsymbol{l}^{e,rx}$ is the unit length line of sight (LOS) vector from $\boldsymbol{p}^{e,rx}_{ea}(t_{rx})$ pointing towards $\boldsymbol{p}^{e,rx}_{es}(t_{tx})$.

2.2.3 Carrier phase

The carrier phase of the signal is tracked by phase lock loops (PLLs) in the receiver. The carrier phase measurement is based on the accumulation of Doppler frequency shift [98] in addition to fractional phase measurements [60], and the observable is therefore also known as accumulated Doppler range (ADR). Because the receiver only starts counting cycles from the time at which it locks onto the signal from each satellite, the carrier phase does not provide absolute range measurements, since the number of carrier cycles from satellite to receiver is unknown. The carrier phase measurement can be modelled as [42]

$$\lambda\phi_{\alpha,s} = \rho_{\alpha,s} + \lambda N_{\alpha,s} - I_{\alpha,s} + T_{\alpha,s} + c(\delta t_{\alpha} - \delta t_{s}) + b_{\alpha} - b_{s} + \phi_{p} + \phi_{M} + \epsilon_{\phi}, \tag{2.7}$$

written here without explicit times, where $N_{\alpha,s}$ is an integer number of carrier cycles, b_{α} and b_{s} are receiver and satellite LOS-independent phase biases with respect to the code, occuring in hardware, software and antennas, including phase-windup due to in-plane rotation of the antennas. ϕ_{p} is a LOS-dependent windup error, ϕ_{M} is an error due to signal multipath and signals received only via reflected paths, and ϵ_{ϕ} is the carrier phase tracking error. This model assumes that half cycle errors, caused by the data modulated onto the carrier, are resolved by the receiver after demodulating the data.

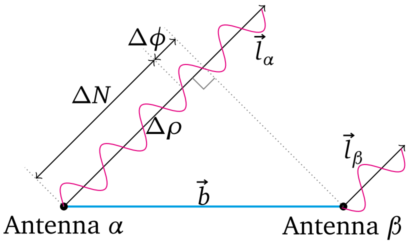


Figure 2.1: Parallel LOS vectors. The LOS for receivers α and β, $\vec{l}_\alpha$ and $\vec{l}_\beta$, can be reasonably approximated as equal, $\vec{l}_\alpha \approx \vec{l}_\beta$. $\Delta\rho$ is the difference in range from each receiver to the satellite, which can be split into an integer wavelength part ΔN and a fractional part $\Delta\phi$.

2.3 Carrier phase interferometry

The use of multiple antennas for estimation of heading or attitude, for short baselines, can be done using an interferometric method with the measured carrier phase. Due to the large distance from a user to the medium Earth orbit (MEO) satellites, antennas fixed at different locations on a vehicle can be reasonably approximated as having parallel LOS vectors. This is illustrated in Fig. 2.1. This approximation means that the signal from a satellite can be seen as a plane wave. Because receivers can start tracking a satellite at different times, because cycle slips can occur causing the integer ambiguity to change, and because some receivers add an integer to the measurement in an attempt to align the measurement with the range measured using code, each of the components $\Delta\phi$ and ΔN in Fig. 2.1 will not necessarily be smaller in magnitude than the length of the baseline in cycles.

By differencing the carrier phase (2.7) between two receivers, called single differencing, we are able to cancel the satellite clock error and phase bias. Because the atmospheric errors are highly spatially correlated, the errors for two receivers mounted on the same vehicle is close to identical, such that these also can be considered to cancel each other. If the antennas are mounted with their planes in parallel, the LOS-dependent phase windup error also cancels for short baselines,

$$\begin{aligned}\lambda\Delta\phi_{\alpha\beta,s} &= \lambda(\phi_{\alpha,s} - \phi_{\beta,s}) \\ &= \Delta\rho_{\alpha\beta,s} + \lambda\Delta N_{\alpha\beta,s} + c(\delta t_\alpha - \delta t_\beta) \\ &\quad + b_\alpha - b_\beta + \Delta\phi_M + \Delta\epsilon_\phi.\end{aligned} \tag{2.8}$$

Furthermore, differencing (2.8) between two satellites s_1 and s_2, called double differencing, cancels errors common to the receivers, including the receiver clock errors and phase biases,

$$\begin{aligned}\lambda\nabla\Delta\phi_{\alpha\beta,s_1s_2} &= \lambda(\Delta\phi_{\alpha\beta,s_1} - \Delta\phi_{\alpha\beta,s_2}) \\ &= \nabla\Delta\rho_{\alpha\beta,s_1s_2} + \lambda\nabla\Delta N_{\alpha\beta,s_1s_2} \\ &\quad + \nabla\Delta\phi_M + \nabla\Delta\epsilon_\phi.\end{aligned} \tag{2.9}$$

The multipath and tracking errors do not cancel, and will be considered as disturbances. Because we normally track many satellites simultaneously, the differencing between satellites can be done in multiple ways. In general we can define a differencing matrix $\mathbf{A} \in \mathbb{R}^{(k-1)\times k}$ for k single differenced measurements, with rows that sum to zero,

$$\lambda \mathbf{A} \Delta \boldsymbol{\phi}_{\alpha\beta} = \mathbf{A}(\Delta \boldsymbol{\rho}_{\alpha\beta} - \lambda \Delta \boldsymbol{N}_{\alpha\beta} + \Delta \boldsymbol{\epsilon}_{\phi}). \tag{2.10}$$

In the following, $\mathbf{A}$ will be chosen to use the highest elevation satellite as the reference to difference all other satellites against.

The range difference to a satellite between two receivers is the projection of the baseline vector $\vec{b}$ onto the LOS vector,

$$\Delta \rho_{\alpha\beta,s} = \vec{l}_s \cdot \vec{b} = \|\vec{b}\|_2 \cos \theta_{\vec{l}_s,\vec{b}}, \tag{2.11}$$

which in vector form can be written as

$$\Delta \boldsymbol{\rho}_{\alpha\beta} = \lambda(\Delta \boldsymbol{\phi}_{\alpha\beta} - \Delta \boldsymbol{N}_{\alpha\beta} - \Delta \boldsymbol{\epsilon}_{\phi}) = (\mathbf{L}^e)^\top \mathbf{R}^e_b \boldsymbol{b}^b, \tag{2.12}$$

with

$$\Delta \boldsymbol{\phi} = \left[\Delta\phi^1 \cdots \Delta\phi^k\right]^\top, \qquad \Delta \boldsymbol{N} = \left[\Delta N^1 \cdots \Delta N^k\right]^\top, \tag{2.13}$$

$$\mathbf{L}^e = \left[\boldsymbol{l}^e_1 \ldots \boldsymbol{l}^e_k\right] \in \mathbb{R}^{3\times k}. \tag{2.14}$$

The equation being the basis for the MEKF measurement model for double differenced carrier phase (DDCP) is then

$$\mathbf{A}\lambda(\Delta \boldsymbol{\phi}_{\alpha\beta} - \Delta \boldsymbol{N}_{\alpha\beta} - \Delta \boldsymbol{\epsilon}_{\phi}) = \mathbf{A}(\mathbf{L}^e)^\top \mathbf{R}^e_b \boldsymbol{b}^b. \tag{2.15}$$

2.4 Receiver measurement time compensation

The carrier phase differencing in section 2.3 assumes that the measurements are taken at exactly the same time by both receivers. If measurement times are different, the satellites will move, satellite clocks will drift, and, for attitude determination cases like here, both receiver antennas can move in the interval between measurements. Note that the clock drift of the receivers in the period between measurements will cancel in double differencing and does not have to be considered. Using independent receivers, the best performance would be achieved if the measurement times could be steered to a common time (i.e. aligned with Global Positioning System Time (GPST)), but low-cost receivers typically do not do this [60]. Instead the error caused by different measurement times must be compensated afterwards using some form of correction term. The time of validity of the resulting state correction must be chosen, for example as the measurement time of one of the receivers or as a different time, i.e. aligned with GPST. Aligning with GPST has the advantage of making the validity time of the correction independent of the receiver clock errors, but requires two correction terms instead of one.

One possibility is to use the measured Doppler frequency shifts to extrapolate the carrier phase measurements to the chosen common time, using the approximation

$$\phi_\beta(t_\alpha) \approx \phi_\beta(t_\beta) + (t_\alpha - t_\beta)\dot{\phi}_\beta(t_\beta). \tag{2.16}$$

This has the advantage of simplicity, as the measurement model used in a Kalman filter does not have to consider the correction at all, but the Doppler measurement noise will then propagate to the extrapolated carrier phase. An example of this method on estimated double differenced ambiguity is shown in Fig. 2.2. Equivalently, an estimate of the carrier phase rate based on the estimated vehicle state from the INS and the satellite states calculated using the ephemeris parameters, using (2.5), can be used for extrapolation, which can then also be included in the linearized measurement model used for gain calculation and covariance correction, as will be done in section 2.6.3. Simply including the extrapolation of the measurement(s) using modelled carrier phase rate will however also extrapolate the position and clock errors of the satellites, which is unneccessary as these values can be calculated at any time within the validity period of the ephemeris. Thus one could calculate the satellite positions and clock errors for the times needed, and only extrapolate the base antenna position. Consequently, for the case of real-time kinematic positioning where a fixed base antenna is used, no extrapolation is necessary as only the effect of satellite movement and clock drift must be considered, which is well predicted from the broadcast orbital and clock parameters and significant differences in measurement time can thus be handled. For the short time difference between receiver measurements however, the error caused by satellite position extrapolation is on the sub-micrometer level and thus considered negligible. Because the measurement output of a GNSS receiver are commonly more delayed than the output of an IMU,

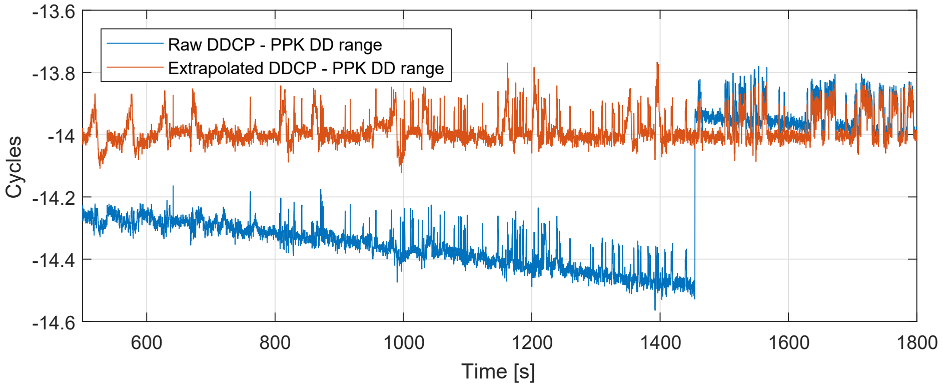


Figure 2.2: Effect of differences in measurement time on DDCP: The differences between the measured DDCP, with and without correction for different measurement times (as shown in Fig. 2.5) for the two receivers, and the predicted double differenced (DD) range from post-processed kinematic (PPK) positioning are plotted. The blue uncorrected value is clearly affected by the clock drift of the two receivers. The red corrected value is very close to the integer value of -14 wavelength cycles.

it is also possible to interpolate the INS-predicted position instead of extrapolating it, if the output rate of the IMU is high enough.

While low-cost receivers do not steer their internal clock to align with GPST, they can correct the clock in millisecond increments in order to keep it *approximately* aligned with GPST [150]. It must also be taken into consideration that the receivers do not necessarily schedule each measurement at the exact same time according to their own clock. The difference in the reported measurement time must be used in addition to the estimated clock errors. The combined difference for two receivers is

$$(t_\alpha - t_\beta) = \underbrace{t_{m\alpha} - t_{m\beta}}_{\text{Timestamps}} + \underbrace{\delta t_\beta - \delta t_\alpha}_{\text{Clock errors}}, \tag{2.17}$$

where $t_{m\alpha}$ and $t_{m\beta}$ are the receiver measurement timestamps.

2.5 MEKF

The multiplicative extendend Kalman filter [81] is an error state Kalman filter used for attitude estimation where the nominal part of the attitude is parametrized using the unit quaternion, while the error component is chosen as a three dimensional parametrization. After the initial transient the error component will be small, staying away from any singularities of the error parametrization, while having a covariance matrix with full rank. The estimator is therefore in practice globally nonsingular. The multiplicative part of the filter is the injection of the estimated error into the nominal estimate, which is done using the quaternion product [128]. Note that the error injection form can be chosen individually for the different states, such that the multiplicative error injection for attitude can be combined with the standard additive injection for other states.

2.5.1 IMU model

The measurement from the angular rate sensor is modelled as

$$\boldsymbol{\omega}^m_{\text{IMU}} = \mathbf{S}_g^{-1} \boldsymbol{\omega}^b_{ib} + \boldsymbol{b}^m_g + \boldsymbol{w}^m_g, \tag{2.18}$$

where $\boldsymbol{\omega}^b_{ib}$ is the angular rate of the body frame $\{b\}$ relative inertial space, approximated as the Earth Centered Inertial (ECI) frame $\{i\}$, $\mathbf{S}_g$ is the gyro calibration matrix which includes the orientation of the measurement frame $\{m\}$ relative $\{b\}$ as well as scale factor and non-orthogonality errors, $\boldsymbol{b}^m_g$ is a measurement bias, modelled as a Wiener process,

$$\dot{\boldsymbol{b}}^m_g = \boldsymbol{w}^m_{b_g}, \tag{2.19}$$

where $\boldsymbol{w}^m_g$ and $\boldsymbol{w}^m_{b_g}$ are Gaussian noise processes with the distributions

$$\boldsymbol{w}^m_g \sim \mathcal{N}(0, \sigma^2_g) \qquad \boldsymbol{w}^m_{b_g} \sim \mathcal{N}(0, \sigma^2_{b_g}), \tag{2.20}$$

which means that the noises are assumed isotropic, and the noise properties are independent of the decomposing axes. The accelerometer measurement is modelled as

$$\begin{aligned} \boldsymbol{f}^m_{\mathrm{IMU}} &= \mathbf{S}_f^{-1}\boldsymbol{f}^b_{ib} + \boldsymbol{b}^m_a + \boldsymbol{w}^m_a & (2.21)\\ &= \mathbf{S}_f^{-1}(\boldsymbol{a}^b_{ib} - \boldsymbol{g}^b) + \boldsymbol{b}^m_a + \boldsymbol{w}^m_a & (2.22) \end{aligned}$$

where $\boldsymbol{f}^b_{ib}$ is the specific force on the sensor, given relative ECI, $\boldsymbol{a}^b_{ib}$ is the coordinate acceleration and $\boldsymbol{g}^m$ is the gravity vector. $\mathbf{S}_f$ is the accelerometer calibration matrix. Bias and noises are modelled as for the angular rate sensor,

$$\dot{\boldsymbol{b}}^m_a = \boldsymbol{w}^m_{b_a}, \tag{2.23}$$

$$\boldsymbol{w}^m_a \sim \mathcal{N}(0, \sigma^2_a) \qquad\qquad \boldsymbol{w}^m_{b_a} \sim \mathcal{N}(0, \sigma^2_{b_a}). \tag{2.24}$$

2.5.2 System kinematic model

As the ECEF frame is used by the Global Positioning System (GPS), it is a reasonable choice of reference frame also for the MEKF. The state vector representing the true values is chosen as

$$\boldsymbol{x} = [\boldsymbol{q}^{e\top}_b \; \boldsymbol{b}^{m\top}_g \; \boldsymbol{p}^{e\;\top}_{eb} \; \boldsymbol{v}^{e\;\top}_{eb} \; \boldsymbol{b}^{m\top}_a \; t_{\varepsilon f} \; t_{\varepsilon\beta} \; t_{df} \; t_{d\beta} \; \boldsymbol{N}^\top_{dd}]^\top, \tag{2.25}$$

where $\boldsymbol{q}^e_b \in \mathbb{R}^4$ is the unit quaternion describing the rotation between ECEF and the body frame, $\boldsymbol{b}^m_g$ and $\boldsymbol{b}^m_a$ are the biases in (2.18) and (2.22), $\boldsymbol{p}^e_{eb}$ and $\boldsymbol{v}^e_{eb}$ are the ECEF position and velocity vectors of the body frame origin (chosen as the location of the IMU), $t_{\varepsilon f}$ and $t_{\varepsilon\beta}$ are the clock offsets of the front and back receiver in unit of meters and t_{df} and $t_{d\beta}$ are the clock drift rates of the front and back receivers in unit of meters per second. $\boldsymbol{N}_{dd}$ are real-valued double differenced carrier phase ambiguities, which will vary in dimension as the number of tracked satellites change over time. The ambiguities are estimated as real-valued, before the LAMBDA algorithm [155] is used to find the best fitting integer vector. For the MEKF the state vector is split into a nominal component

$$\hat{\boldsymbol{x}} = [\hat{\boldsymbol{q}}^{e\top}_b \; \hat{\boldsymbol{b}}^{m\top}_g \; \hat{\boldsymbol{p}}^{e\;\top}_{eb} \; \hat{\boldsymbol{v}}^{e\;\top}_{eb} \; \hat{\boldsymbol{b}}^{m\top}_a \; \hat{t}_{\varepsilon f} \; \hat{t}_{\varepsilon\beta} \; \hat{t}_{df} \; \hat{t}_{d\beta} \; \hat{\boldsymbol{N}}^\top_{dd}]^\top, \tag{2.26}$$

where the ^ operator indicates the expected value, and an error component

$$\delta\boldsymbol{x} = [\delta\boldsymbol{a}^\top \; \delta\boldsymbol{b}^{m\top}_g \; \delta\boldsymbol{p}^{e\;\top}_{eb} \; \delta\boldsymbol{v}^{e\;\top}_{eb} \; \delta\boldsymbol{b}^{m\top}_a \; \delta t_{\varepsilon f} \; \delta t_{\varepsilon\beta} \; \delta t_{df} \; \delta t_{d\beta} \; \delta\boldsymbol{N}^\top_{dd}]^\top, \tag{2.27}$$

where $\delta\boldsymbol{a}$ is the attitude error parametrized by twice the Gibbs vector [81],

$$\delta\boldsymbol{a} := 2\frac{\delta\boldsymbol{\epsilon}}{\delta\eta}, \tag{2.28}$$

with $\delta\boldsymbol{\epsilon}$ being the imaginary vector part of the error quaternion $\delta\boldsymbol{q} = (\hat{\boldsymbol{q}}^e_b)^{-1} \otimes \boldsymbol{q}^e_b$, with $\otimes$ indicating the quaternion product [128], and $\delta\eta$ is the real scalar part. The nominal and error components relate to the true state using

$$\begin{aligned}
\boldsymbol{q}^e_b &= \hat{\boldsymbol{q}}^e_b \otimes \delta\boldsymbol{q}(\delta\boldsymbol{a}) & t_{\varepsilon f} &= \hat{t}_{\varepsilon f} + \delta t_{\varepsilon f} \\
\boldsymbol{b}^m_g &= \hat{\boldsymbol{b}}^m_g + \delta\boldsymbol{b}^m_g & t_{\varepsilon\beta} &= \hat{t}_{\varepsilon\beta} + \delta t_{\varepsilon\beta} \\
\boldsymbol{p}^e_{eb} &= \hat{\boldsymbol{p}}^e_{eb} + \delta\boldsymbol{p}^e_{eb} & t_{df} &= \hat{t}_{df} + \delta t_{df} \\
\boldsymbol{v}^e_{eb} &= \hat{\boldsymbol{v}}^e_{eb} + \delta\boldsymbol{v}^e_{eb} & t_{d\beta} &= \hat{t}_{d\beta} + \delta t_{d\beta} \\
\boldsymbol{b}^m_a &= \hat{\boldsymbol{b}}^m_a + \delta\boldsymbol{b}^m_a & \boldsymbol{N}_{dd} &= \hat{\boldsymbol{N}}_{dd} + \delta\boldsymbol{N}_{dd},
\end{aligned} \tag{2.29}$$

where the error quaternion can be calculated from $\delta\boldsymbol{a}$ using

$$\delta\boldsymbol{q}(\delta\boldsymbol{a}) = \frac{1}{\sqrt{4 + \|\delta\boldsymbol{a}\|_2^2}} \begin{bmatrix} 2 \\ \delta\boldsymbol{a} \end{bmatrix}. \tag{2.30}$$

2.5.2.1 True system dynamics

The attitude and translational kinematics using the true states are

$$\begin{aligned}
\dot{\boldsymbol{q}}^e_b &= \frac{1}{2}\boldsymbol{q}^e_b \otimes \boldsymbol{\omega}^b_{eb}, & (2.31)\\
&= \frac{1}{2}\boldsymbol{q}^e_b \otimes (\mathbf{S}_g(\boldsymbol{\omega}^m_{\text{IMU}} - \boldsymbol{b}^m_g - \boldsymbol{w}^m_g) - \mathbf{R}(\boldsymbol{q}^e_b)^\top \boldsymbol{\omega}^e_{ie}), & (2.32)
\end{aligned}$$

$$\begin{aligned}
\dot{\boldsymbol{p}}^e_{eb} &= \boldsymbol{v}^e_{eb}, & (2.33)\\
\dot{\boldsymbol{v}}^e_{eb} &= \mathbf{R}(\boldsymbol{q}^e_b)\mathbf{S}_f(\boldsymbol{f}^m_{\text{IMU}} - \boldsymbol{b}^m_a - \boldsymbol{w}^m_a) + \boldsymbol{g}^e - 2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\boldsymbol{v}^e_{eb}. & (2.34)
\end{aligned}$$

Note that when the quaternion product notation $\otimes$ here involves a vector, the vector takes the place of the imaginary part of a quaternion, with the real part set to zero. The clock offset and drift rates are assumed to behave as the models

$$\begin{aligned}
\dot{t}_{\varepsilon f} &= t_{df} + w_{t_{\varepsilon f}}, & w_{t_{\varepsilon,f}} &\sim \mathcal{N}(0, \sigma^2_{t_{\varepsilon f}}) & (2.35)\\
\dot{t}_{\varepsilon\beta} &= t_{d\beta} + w_{t_{\varepsilon\beta}}, & w_{t_{\varepsilon,\beta}} &\sim \mathcal{N}(0, \sigma^2_{t_{\varepsilon\beta}}) & (2.36)\\
\dot{t}_{df} &= w_{t_{df}}, & w_{t_{d,f}} &\sim \mathcal{N}(0, \sigma^2_{t_{df}}) & (2.37)\\
\dot{t}_{d\beta} &= w_{t_{d\beta}}, & w_{t_{d,b}} &\sim \mathcal{N}(0, \sigma^2_{t_{d\beta}}) & (2.38)
\end{aligned}$$

and the IMU biases, chosen to be decomposed in the measurement frame, have the models (2.19), (2.23). The ambiguity state is modelled as a slowly varying Wiener process,

$$\dot{\boldsymbol{N}}_{dd} = \boldsymbol{w}_N, \quad \boldsymbol{w}_N \sim \mathcal{N}(0, \sigma^2_N). \tag{2.39}$$

2.5.2.2 Nominal dynamics

The dynamics of the nominal component of the state vector is simply the expected value of the true dynamics, which means that the states are replaced with their expected value and all noises are set to zero. The IMU used was configured to output velocity and angle increments, $\Delta \boldsymbol{v}^m_{\text{IMU}}$ and $\Delta \boldsymbol{\theta}^m_{\text{IMU}}$ at a rate of 250 Hz. The use of increments means that the internal high rate samples are integrated to provide lower rate outputs maintaining all information. The mechanization equations are implemented in discrete form as

$$\hat{\boldsymbol{a}}^e_{eb}\Delta t = \mathbf{R}(\hat{\boldsymbol{q}}^e_b)\mathbf{S}_f(\Delta \boldsymbol{v}^m_{\text{IMU}} - \hat{\boldsymbol{b}}^m_a \Delta t) + \boldsymbol{g}^e \Delta t - 2\mathbf{S}(\boldsymbol{\omega}^e_{ie})\hat{\boldsymbol{v}}^e_{eb}\Delta t \tag{2.40}$$

$$\hat{\boldsymbol{p}}^e_{eb} \leftarrow \hat{\boldsymbol{p}}^e_{eb} + \hat{\boldsymbol{v}}^e_{eb}\Delta t + \hat{\boldsymbol{a}}^e_{eb}\frac{\Delta t^2}{2} \tag{2.41}$$

$$\hat{\boldsymbol{v}}^e_{eb} \leftarrow \hat{\boldsymbol{v}}^e_{eb} + \hat{\boldsymbol{a}}^e_{eb}\Delta t \tag{2.42}$$

$$\Delta\hat{\boldsymbol{\theta}}^b := \mathbf{S}_g(\Delta \boldsymbol{\theta}^m_{\text{IMU}} - \hat{\boldsymbol{b}}^m_g \Delta t) - \mathbf{R}(\hat{\boldsymbol{q}}^e_b)^\top \boldsymbol{\omega}^e_{ie} \tag{2.43}$$

$$\boldsymbol{q}(\Delta\hat{\boldsymbol{\theta}}^b) = \begin{bmatrix} \cos\left(\frac{\|\Delta\hat{\boldsymbol{\theta}}^b\|}{2}\right) \\ \frac{\Delta\hat{\boldsymbol{\theta}}^b}{\|\Delta\hat{\boldsymbol{\theta}}^b\|}\sin\left(\frac{\|\Delta\hat{\boldsymbol{\theta}}^b\|}{2}\right) \end{bmatrix} \tag{2.44}$$

$$\hat{\boldsymbol{q}}^e_b \leftarrow \hat{\boldsymbol{q}}^e_b \otimes \boldsymbol{q}(\Delta\hat{\boldsymbol{\theta}}) \tag{2.45}$$

$$\hat{t}_{\varepsilon f} \leftarrow \hat{t}_{\varepsilon f} + \hat{t}_{df}\Delta t \tag{2.46}$$

$$\hat{t}_{\varepsilon\beta} \leftarrow \hat{t}_{\varepsilon\beta} + \hat{t}_{d\beta}\Delta t \tag{2.47}$$

2.5.2.3 Error state dynamics

Since the expected error before a measurement is always zero, and the observed error is injected into the nominal state after a measurement, the expected error is zero at every timstep and the state prediction step of the Kalman filter does therefore not have to be implemented. The linearized error dynamics are however still needed for covariance propagation. From [81] we have the attitude error dynamics

$$\delta\dot{\boldsymbol{a}} = \Delta\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}^b_{eb} \times \delta\boldsymbol{a} - \frac{1}{2}\Delta\boldsymbol{\omega} \times \delta\boldsymbol{a} + \frac{1}{4}(\Delta\boldsymbol{\omega} \cdot \delta\boldsymbol{a})\delta\boldsymbol{a} \tag{2.48}$$

with $\Delta\boldsymbol{\omega}$ extended with the effect of the Earth rotation,

$$\Delta\boldsymbol{\omega} = -\mathbf{S}_g\delta\boldsymbol{b}^m_g - \mathbf{S}_g\boldsymbol{w}_g + \mathbf{R}(\hat{\boldsymbol{q}}^e_b)^\top\boldsymbol{\omega}^e_{ie} - \mathbf{R}(\boldsymbol{q}^e_b)^\top\boldsymbol{\omega}^e_{ie}. \tag{2.49}$$

The translational error dynamics are

$$\delta\dot{\boldsymbol{p}}^e_{eb} = \delta\boldsymbol{v}^e_{eb}. \tag{2.50}$$

$$\begin{aligned}\delta\dot{\boldsymbol{v}}^{e}_{eb} = -2\mathbf{S}(\boldsymbol{\omega}^{e}_{ie})\delta\boldsymbol{v}^{e}_{eb} + (\mathbf{R}(\boldsymbol{q}^{e}_{b}) - \mathbf{R}(\hat{\boldsymbol{q}}^{e}_{b}))\mathbf{S}_{f}(\boldsymbol{f}^{m}_{\text{IMU}} - \hat{\boldsymbol{b}}^{m}_{a})& \\ - \mathbf{R}(\boldsymbol{q}^{e}_{b})\mathbf{S}_{f}(\delta\boldsymbol{b}^{m}_{a} + \boldsymbol{w}^{m}_{a}).& \quad (2.51)\end{aligned}$$

The error dynamics for the remaining biases, clock errors and ambiguity states are simply given by their process noises. These equations are linearized in order to find the linear system matrix **F**. For attitude it is used that $\mathbf{R}(\boldsymbol{q}^{e}_{b}) = \mathbf{R}(\hat{\boldsymbol{q}}^{e}_{b})\mathbf{R}(\delta\boldsymbol{a})$ and the first order approximation $\mathbf{R}(\delta\boldsymbol{a}) \approx \mathbf{I}_{3\times3} + \mathbf{S}(\delta\boldsymbol{a})$. The resulting linearized, continous time system matrix $\bar{\mathbf{F}}$ associated with the fixed part of the state vector, without the variable size ambiguity vector, is given in (2.52),

$$\bar{\mathbf{F}} = \begin{bmatrix} -\mathbf{S}(\hat{\boldsymbol{\omega}}^{b}_{ib}) & -\mathbf{S}_{g} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ -\mathbf{R}(\hat{\boldsymbol{q}}^{e}_{b})\mathbf{S}(\hat{\boldsymbol{f}}^{b}_{ib}) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -2\mathbf{S}(\boldsymbol{\omega}^{e}_{ie}) & -\mathbf{R}(\hat{\boldsymbol{q}}^{e}_{b})\mathbf{S}_{f} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 1 & 0 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & 1 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.52)$$

Combining this with the zero dynamics of the double differenced ambiguity errors, we get the complete matrix

$$\mathbf{F} = \begin{bmatrix} \bar{\mathbf{F}} & \mathbf{0}_{19\times(k-1)} \\ \mathbf{0}_{(k-1)\times19} & \mathbf{0}_{(k-1)\times(k-1)} \end{bmatrix} \in \mathbb{R}^{(19+k-1)\times(19+k-1)}. \quad (2.53)$$

The process noise input matrix for the fixed part of the error state is

$$\bar{\mathbf{G}} = \begin{bmatrix} -\mathbf{S}_{g} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{S}_{f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & 0 & 0 & 0 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 1 & 0 & 0 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 1 & 0 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.54)$$

and the complete matrix including the ambiguities is

$$\mathbf{G} = \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{0}_{19\times(k-1)} \\ \mathbf{0}_{(k-1)\times16} & \mathbf{I}_{(k-1)\times(k-1)} \end{bmatrix} \in \mathbb{R}^{(19+k-1)\times(16+k-1)}. \quad (2.55)$$

The linear process model is then

$$\delta\dot{\boldsymbol{x}} = \mathbf{F}\delta\boldsymbol{x} + \mathbf{G}\boldsymbol{w}, \quad (2.56)$$

where the noise vector is

$$\boldsymbol{w} = [\boldsymbol{w}^{\top}_{g}\ \boldsymbol{w}^{\top}_{bg}\ \boldsymbol{w}^{\top}_{a}\ \boldsymbol{w}^{\top}_{ba}\ w_{t_{\varepsilon f}}\ w_{t_{\varepsilon\beta}}\ w_{t_{df}}\ w_{t_{d\beta}}\ \boldsymbol{w}^{\top}_{N}]^{\top}. \quad (2.57)$$

The discrete time equivalents of **F** and **G** are found using Van Loan's method [151].

2.6 Measurement models

For the measurements from each receiver, we take into account that the antenna positions are offset from the body frame coordinate origin, defined at the position of the IMU, by the vectors $\boldsymbol{r}_f^b$ and $\boldsymbol{r}_\beta^b$,

$$\boldsymbol{p}_{eb,f}^e = \boldsymbol{p}_{eb}^e + \mathrm{R}(\boldsymbol{q}_b^e)\boldsymbol{r}_f^b, \tag{2.58}$$

$$\boldsymbol{p}_{eb,\beta}^e = \boldsymbol{p}_{eb}^e + \mathrm{R}(\boldsymbol{q}_b^e)\boldsymbol{r}_\beta^b. \tag{2.59}$$

The velocity of each receiver is then

$$\boldsymbol{v}_{eb,f}^e = \boldsymbol{v}_{eb}^e + \mathrm{R}(\boldsymbol{q}_b^e)\mathrm{S}(\boldsymbol{\omega}_{eb}^b)\boldsymbol{r}_f^b, \tag{2.60}$$

$$\boldsymbol{v}_{eb,\beta}^e = \boldsymbol{v}_{eb}^e + \mathrm{R}(\boldsymbol{q}_b^e)\mathrm{S}(\boldsymbol{\omega}_{eb}^b)\boldsymbol{r}_\beta^b. \tag{2.61}$$

In the following the frame $\{e\}$ denotes the ECEF frame of the time of reception, $\{e, rx\}$.

2.6.1 Pseudorange

As a function of the vehicle state, using the lever arm, the pseudorange model (2.3) of the front receiver is written as

$$P_f = ||\hat{\mathrm{R}}_{e,tx}^e \boldsymbol{p}_{es}^{e,tx} - (\boldsymbol{p}_{eb}^e + \mathrm{R}(\boldsymbol{q}_b^e)\boldsymbol{r}_f^b)||_2 + t_{\varepsilon f} + \epsilon_P + I + T - c\delta t_s. \tag{2.62}$$

where δt_s is the clock error computed from the correction parameters in the satellite navigation message [41] along with the satellite position $p_{es}^{e,tx}$ at the corrected time of signal transmission. I and T are values for the ionospheric delay from the Klobuchar [69] ionospheric model and the NATO STANAG tropospheric model [42]. Linearizing the model at the expected zero error, for a vector of k measurements, for each receiver, the linear measurement matrices become

$$\mathbf{H}_{P,f} = \begin{bmatrix} \hat{\boldsymbol{l}}_{f,1}^\top \mathrm{R}(\hat{\boldsymbol{q}}_b^e)\mathrm{S}(r_f^b) & \mathbf{0}_{1\times3} & -\hat{\boldsymbol{l}}_{f,1}^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & 0 & 0 & 0 & \mathbf{0}_{1\times(k-1)} \\ \hat{\boldsymbol{l}}_{f,2}^\top \mathrm{R}(\hat{\boldsymbol{q}}_b^e)\mathrm{S}(r_f^b) & \mathbf{0}_{1\times3} & -\hat{\boldsymbol{l}}_{f,2}^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & 0 & 0 & 0 & \mathbf{0}_{1\times(k-1)} \\ & & & & \vdots & & & & & \\ \hat{\boldsymbol{l}}_{f,k}^\top \mathrm{R}(\hat{\boldsymbol{q}}_b^e)\mathrm{S}(r_f^b) & \mathbf{0}_{1\times3} & -\hat{\boldsymbol{l}}_{f,k}^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & 0 & 0 & 0 & \mathbf{0}_{1\times(k-1)} \end{bmatrix}, \tag{2.63}$$

$$\mathbf{H}_{P,\beta} = \begin{bmatrix} \hat{\boldsymbol{l}}_{\beta,1}^\top \mathrm{R}(\hat{\boldsymbol{q}}_b^e)\mathrm{S}(r_\beta^b) & \mathbf{0}_{1\times3} & -\hat{\boldsymbol{l}}_{\beta,1}^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 1 & 0 & 0 & \mathbf{0}_{1\times(k-1)} \\ \hat{\boldsymbol{l}}_{\beta,2}^\top \mathrm{R}(\hat{\boldsymbol{q}}_b^e)\mathrm{S}(r_\beta^b) & \mathbf{0}_{1\times3} & -\hat{\boldsymbol{l}}_{\beta,2}^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 1 & 0 & 0 & \mathbf{0}_{1\times(k-1)} \\ & & & & \vdots & & & & & \\ \hat{\boldsymbol{l}}_{\beta,k}^\top \mathrm{R}(\hat{\boldsymbol{q}}_b^e)\mathrm{S}(r_\beta^b) & \mathbf{0}_{1\times3} & -\hat{\boldsymbol{l}}_{\beta,k}^\top & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 1 & 0 & 0 & \mathbf{0}_{1\times(k-1)} \end{bmatrix}, \tag{2.64}$$

each $\in \mathbb{R}^{k\times(19+k-1)}$.

2.6.2 Doppler

Using (2.4) and (2.5) and the antenna velocity (2.60), the Doppler measurement can be written as

$$\begin{aligned}\lambda\Delta f_f = \boldsymbol{l}_f^{e\top}\Big(&\boldsymbol{v}_{eb}^e + \mathbf{R}(\boldsymbol{q}_b^e)\mathbf{S}(\boldsymbol{\omega}_{eb}^b)\boldsymbol{r}_f^b + \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\boldsymbol{p}_{eb}^e \\ &- \hat{\mathbf{R}}_{e,tx}^e\big(\boldsymbol{v}_{es}^{e,tx} - \mathbf{S}(\boldsymbol{\omega}_{ie}^e)\boldsymbol{p}_{es}^{e,tx}\big)\Big) - t_{df} + c\dot{\delta t_s} + \epsilon_{\Delta f}.\end{aligned} \tag{2.65}$$

The dependency of the model on position errors through the Earth rotation rate and the dependency on position and attitude errors throught the LOS-vector are ignored as these terms are negligible due to the slow Earth-rotation rate and low sensitivity of the LOS vector to small position errors. The linearization of the model at the expected zero error gives the measurement matrices (2.66), (2.67).

$$\mathbf{H}_{\Delta f,f} = \begin{bmatrix} -\hat{\boldsymbol{l}}_{f,1}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(\mathbf{S}(\hat{\boldsymbol{\omega}}_{eb}^b)r_f^b) & \hat{\boldsymbol{l}}_{f,1}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(r_f^b)\mathbf{S}_g & \mathbf{0}_{1\times3} & \hat{\boldsymbol{l}}_{f,1}^{e\top} & \mathbf{0}_{1\times3} & 0 & 0 & -1 & 0 & \mathbf{0}_{1\times(k-1)} \\ -\hat{\boldsymbol{l}}_{f,2}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(\mathbf{S}(\hat{\boldsymbol{\omega}}_{eb}^b)r_f^b) & \hat{\boldsymbol{l}}_{f,2}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(r_f^b)\mathbf{S}_g & \mathbf{0}_{1\times3} & \hat{\boldsymbol{l}}_{f,2}^{e\top} & \mathbf{0}_{1\times3} & 0 & 0 & -1 & 0 & \mathbf{0}_{1\times(k-1)} \\ & & & & \vdots & & & & & \\ -\hat{\boldsymbol{l}}_{f,k}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(\mathbf{S}(\hat{\boldsymbol{\omega}}_{eb}^b)r_f^b) & \hat{\boldsymbol{l}}_{f,k}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(r_f^b)\mathbf{S}_g & \mathbf{0}_{1\times3} & \hat{\boldsymbol{l}}_{f,k}^{e\top} & \mathbf{0}_{1\times3} & 0 & 0 & -1 & 0 & \mathbf{0}_{1\times(k-1)} \end{bmatrix} \in \mathbb{R}^{k\times(19+k-1)}, \tag{2.66}$$

$$\mathbf{H}_{\Delta f,\beta} = \begin{bmatrix} -\hat{\boldsymbol{l}}_{\beta,1}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(\mathbf{S}(\hat{\boldsymbol{\omega}}_{eb}^b)r_\beta^b) & \hat{\boldsymbol{l}}_{\beta,1}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(r_\beta^b)\mathbf{S}_g & \mathbf{0}_{1\times3} & \hat{\boldsymbol{l}}_{\beta,1}^{e\top} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & -1 & \mathbf{0}_{1\times(k-1)} \\ -\hat{\boldsymbol{l}}_{\beta,2}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(\mathbf{S}(\hat{\boldsymbol{\omega}}_{eb}^b)r_\beta^b) & \hat{\boldsymbol{l}}_{\beta,2}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(r_\beta^b)\mathbf{S}_g & \mathbf{0}_{1\times3} & \hat{\boldsymbol{l}}_{\beta,2}^{e\top} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & -1 & \mathbf{0}_{1\times(k-1)} \\ & & & & \vdots & & & & & \\ -\hat{\boldsymbol{l}}_{\beta,k}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(\mathbf{S}(\hat{\boldsymbol{\omega}}_{eb}^b)r_\beta^b) & \hat{\boldsymbol{l}}_{\beta,k}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}_b^e)\mathbf{S}(r_\beta^b)\mathbf{S}_g & \mathbf{0}_{1\times3} & \hat{\boldsymbol{l}}_{\beta,k}^{e\top} & \mathbf{0}_{1\times3} & 0 & 0 & 0 & -1 & \mathbf{0}_{1\times(k-1)} \end{bmatrix} \in \mathbb{R}^{k\times(19+k-1)}. \tag{2.67}$$

2.6.3 Double differenced carrier phase

The carrier phase measurement matrix will be constructed first for the fixed part of the state vector, $\bar{\boldsymbol{x}}$, for single differenced measurements, with the corresponding double differenced measurement matrix then simply calculated as

$$\mathbf{H}_{dd,\bar{x}} = \mathbf{A}\mathbf{H}_{sd,\bar{x}}. \tag{2.68}$$

The complete double differenced measurement matrix including the double differenced ambiguities is then

$$\mathbf{H}_{dd} = \begin{bmatrix}\mathbf{H}_{dd,\bar{x}} & \mathbf{I}_{(k-1)\times(k-1)}\end{bmatrix} \in \mathbb{R}^{(k-1)\times(19+k-1)}. \tag{2.69}$$

Using the linear time extrapolation with the measurement time of the front receiver chosen as the time of differencing, the single differenced measurement model becomes

$$\phi_\beta + (t_f - t_\beta)\dot{\phi}_\beta - \phi_f = \frac{1}{\lambda}\boldsymbol{l}^{e\top}\mathbf{R}(\boldsymbol{q}_b^e)\boldsymbol{b}^b + N_{sd}. \tag{2.70}$$

Substituting in the expression for the carrier phase derivative using (2.65) and (2.4) gives

$$\begin{aligned}\phi_\beta - \phi_f = \frac{1}{\lambda}\boldsymbol{l}^{e\top}\mathbf{R}(\boldsymbol{q}^e_b)\boldsymbol{b}^b + N_{sd} + (t_f - t_\beta)\frac{1}{\lambda}\Big(\boldsymbol{l}^{e\top}(\boldsymbol{v}^e_{eb} + \mathbf{R}(\boldsymbol{q}^e_b)\mathbf{S}(\boldsymbol{\omega}^b_{eb})\boldsymbol{r}^b_\beta + \mathbf{S}(\boldsymbol{\omega}^e_{ie})\boldsymbol{p}^e_{eb} \\ - \hat{\mathbf{R}}^e_{e,tx}\big(\boldsymbol{v}^{e,tx}_{es} - \mathbf{S}(\boldsymbol{\omega}^e_{ie})\boldsymbol{p}^{e,tx}_{es}\big)) - t_{d\beta} + c\dot{\delta t}_s + \epsilon_{\Delta f}\Big) + \epsilon_{\phi_{sd}}.\end{aligned} \tag{2.71}$$

Note that because we will difference the model between satellites later, terms that are the same for all satellites will cancel, in this case this only applies to the drift rate $t_{d\beta}$, which will therefore be ignored in the following derivatives. The derivative with respect to the attitude error is

$$\frac{\partial\Delta\phi_{\beta f}}{\partial\delta\boldsymbol{a}} = -\frac{1}{\lambda}\boldsymbol{l}^{e\top}\mathbf{R}(\hat{\boldsymbol{q}}^e_b)\Big(\mathbf{S}(\boldsymbol{b}^b) - (t_f - t_\beta)\mathbf{S}(\mathbf{S}(\boldsymbol{\omega}^b_{eb})\boldsymbol{r}_\beta)\Big). \tag{2.72}$$

The angular rate $\boldsymbol{\omega}^b_{eb}$ makes the model dependent on the gyro bias. Differentiating with respect the gyro bias error gives

$$\frac{\partial\Delta\phi_{\beta f}}{\partial\delta\boldsymbol{b}^m_g} = \frac{t_f - t_\beta}{\lambda}\boldsymbol{l}^{e\top}\mathbf{R}(\boldsymbol{q}^e_b)\mathbf{S}(r_\beta)\mathbf{S}_g. \tag{2.73}$$

Differentiating with respect to the clock biases, using (2.17), gives

$$\begin{aligned}\frac{\partial\Delta\phi_{\beta f}}{\partial\delta\boldsymbol{t}_{\varepsilon f}} = -\frac{1}{c\lambda}\Big(&\boldsymbol{l}^{e\top}(\boldsymbol{v}^e_{eb} + \mathbf{R}(\boldsymbol{q}^e_b)\mathbf{S}(\boldsymbol{\omega}^b_{eb})\boldsymbol{r}^b_\beta + \mathbf{S}(\boldsymbol{\omega}^e_{ie})\boldsymbol{p}^e_{eb} \\ &- \hat{\mathbf{R}}^e_{e,tx}\big(\boldsymbol{v}^{e,tx}_{es} - \mathbf{S}(\boldsymbol{\omega}^e_{ie})\boldsymbol{p}^{e,tx}_{es}\big)) + c\dot{\delta t}_s + \epsilon_{\Delta f}\Big),\end{aligned} \tag{2.74}$$

$$\frac{\partial\Delta\phi_{\beta f}}{\partial\delta\boldsymbol{t}_{\varepsilon\beta}} = -\frac{\partial\Delta\phi_{\beta f}}{\partial\delta\boldsymbol{t}_{\varepsilon f}}. \tag{2.75}$$

With respect to the velocity we have

$$\frac{\partial\Delta\phi_{\beta f}}{\partial\delta\boldsymbol{v}^e_{eb}} = (t_f - t_\beta)\frac{1}{\lambda}\boldsymbol{l}^{e\top}. \tag{2.76}$$

Writing $\Delta\phi_{\beta f}$ for satellite i as $\Delta\phi_i$ and using expected values, these can be combined to the matrix

$$\mathbf{H}_{sd,\bar{\boldsymbol{x}}} = \begin{bmatrix} \widehat{\frac{\partial\Delta\phi_1}{\partial\boldsymbol{a}}} & \widehat{\frac{\partial\Delta\phi_1}{\partial\delta\boldsymbol{b}_g}} & \mathbf{0}_{1\times3} & \widehat{\frac{\partial\Delta\phi_1}{\partial\delta\boldsymbol{v}^e_{eb}}} & \mathbf{0}_{1\times3} & \widehat{\frac{\partial\Delta\phi_1}{\partial\delta\boldsymbol{t}_{\varepsilon f}}} & \widehat{\frac{\partial\Delta\phi_1}{\partial\delta\boldsymbol{t}_{\varepsilon\beta}}} & 0 & 0 \\ \widehat{\frac{\partial\Delta\phi_2}{\partial\boldsymbol{a}}} & \widehat{\frac{\partial\Delta\phi_2}{\partial\delta\boldsymbol{b}_g}} & \mathbf{0}_{1\times3} & \widehat{\frac{\partial\Delta\phi_2}{\partial\delta\boldsymbol{v}^e_{eb}}} & \mathbf{0}_{1\times3} & \widehat{\frac{\partial\Delta\phi_2}{\partial\delta\boldsymbol{t}_{\varepsilon f}}} & \widehat{\frac{\partial\Delta\phi_2}{\partial\delta\boldsymbol{t}_{\varepsilon\beta}}} & 0 & 0 \\ & & & \vdots & & & & & \\ \widehat{\frac{\partial\Delta\phi_k}{\partial\boldsymbol{a}}} & \widehat{\frac{\partial\Delta\phi_k}{\partial\delta\boldsymbol{b}_g}} & \mathbf{0}_{1\times3} & \widehat{\frac{\partial\Delta\phi_k}{\partial\delta\boldsymbol{v}^e_{eb}}} & \mathbf{0}_{1\times3} & \widehat{\frac{\partial\Delta\phi_k}{\partial\delta\boldsymbol{t}_{\varepsilon f}}} & \widehat{\frac{\partial\Delta\phi_k}{\partial\delta\boldsymbol{t}_{\varepsilon\beta}}} & 0 & 0 \end{bmatrix} \tag{2.77}$$

$\in \mathbb{R}^{k\times19}$, and the complete measurement matrix, including the ambiguity states, is constructed using (2.68) and (2.69).

2.6.4 Fixed integer ambiguity correction

After all measurement corrections have been performed, integer ambiguity resolution is attempted using the LAMBDA algorithm [141]. If an integer vector $\hat{\boldsymbol{N}}_{\text{fixed}}$ sufficiently better than the next best alternative is found, a "pseudo-measurement" correction is performed where $\boldsymbol{N}_{dd}$ is assumed measured with no uncertainty, such that $\mathbf{R}_{\hat{N}_{\text{fixed}}} = \mathbf{0}$. With the simple measurement matrix $\mathbf{H} = [\mathbf{0}_{(k-1)\times 19} \quad \mathbf{I}_{(k-1)\times(k-1)}]$, the fixed state vector is

$$\hat{\boldsymbol{x}}_{\text{fixed}} = \hat{\boldsymbol{x}} + \mathbf{K}(\hat{\boldsymbol{N}}_{\text{fixed}} - \mathbf{H}\hat{\boldsymbol{x}}). \tag{2.78}$$

Note that because it is not desired to feed the fixed integers back in the MEKF, due to the potentially destabilizing effect it can have, the fixed estimates $\hat{\boldsymbol{x}}_{\text{fixed}}$ are propagated using the expressions in section 2.5.2.2 in parallel with $\hat{\boldsymbol{x}}$ until the next time measurements corrections (and integer ambiguity resolution) are performed, in order to get a high rate "fixed" output.

2.6.5 Initialization and handling changes in usable satellites

Position, velocity, clock offsets and clock drift rates are initialized using single-epoch least-squares GNSS estimates. Starting on the ground before launch, roll and pitch are initialized using accelerometer leveling [42],

$$\hat{\phi} = \operatorname{atan2}(-\boldsymbol{f}^b_{\text{IMU},y}, -\boldsymbol{f}^b_{\text{IMU},z}), \tag{2.79}$$

$$\hat{\theta} = \operatorname{atan2}(\boldsymbol{f}^b_{\text{IMU},x}, \sqrt{(\boldsymbol{f}^b_{\text{IMU},y})^2 + (\boldsymbol{f}^b_{\text{IMU},z})^2}), \tag{2.80}$$

and heading using a magnetometer. Following this the double differenced ambiguity estimates are initialized using the attitude estimate,

$$\widehat{\nabla\Delta\rho}_{\beta f,s_1s_2} = (\boldsymbol{l}^e_{s_1} - \boldsymbol{l}^e_{s_2})^\top \hat{\mathbf{R}}^e_b \boldsymbol{b}^b, \tag{2.81}$$

$$\widehat{\nabla\Delta N}_{\beta f,s_1s_2} = \frac{1}{\lambda}\widehat{\nabla\Delta\rho}_{\beta f,s_1s_2} - \nabla\Delta\phi_{\beta f,s_1s_2}. \tag{2.82}$$

This method is also used to initialize ambiguities whenever a satellite, which was not usable the previous epoch, is to be used, both the first time it is observed and after any signal tracking issues such as cycle slips. Double differenced pseudoranges can also be used for initialization, but for very short baselines the noise present in the pseudoranges can make this undesirable. Gyro biases are initialized by utilizing the known $\boldsymbol{\omega}^b_{eb} = \mathbf{0}$ on the ground before launch, and accelerometer biases are initialized by using the initial attitude, calculated gravity vector and known $\boldsymbol{a}^b_{eb} = \mathbf{0}$.

2.7 Experimental testing

The UAV used for data collection was the ET-Air Cruiser Mini, equipped with a Sensonor STIM300 IMU (engineering sample), running at 250Hz output rate, and and two longitudinally spaced antennas connected to U-Blox NEO-M8T receivers. The flight path of the UAV is shown in Fig. 2.3. All measurements were received by the *SenTiBoard* (sensor timing board) [4], which also receives synchronization signals from each sensor. Using the known delay characteristics of the IMU measurement output [120], the time-of-validity (TOV) output signal and a pulse-per-second (PPS) output of one of the GPS receivers, which signals at the top of each GPST second, the fractional time of validity of the IMU measurements in GPST can be found accurately. This is illustrated in Fig. 2.4. Combining this with the GNSS measurement timestamps and the estimated clock errors, the absolute time of validity of the IMU measurements can be found. This was used to apply the GNSS measurement corrections with a timing error less than one IMU sample interval (4ms). The difference in measurement time of the GNSS receivers was taken into account in the DDCP measurement, but was not considered for the time of validity of the other measurements, as this difference is smaller than the time between IMU samples.

The MEKF was implemented in Matlab, running on a desktop computer for offline processing. The Matlab implementation of the LAMBDA algorithm from [155] was used for integer ambiguity resolution, using the ratio test with $\mu = 0.5$ to accept integer results. The detection of half cycle errors and cycle slips done by the receivers, and output as flags together with the raw measurements, was used to exclude unusable satellites for the DDCP correction.

A stationary GNSS receiver was placed near the launch site, logging raw GPS observables. The raw data from the base station and each antenna on the UAV was used for PPK positioning of each antenna using RTKLIB [140]. In addition, RTKLIB was run in "moving base" mode using only the onboard receivers. The estimates using the base station were less noisy, but did not always have the carrier phase integer ambiguities resolved. The moving base results were stiched into these parts of the output. These results were only used for comparison with the results from the

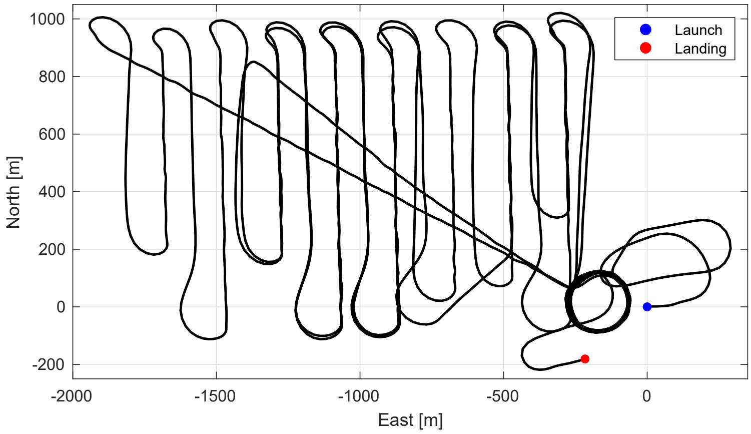


Figure 2.3: North-East plot of the flight path.

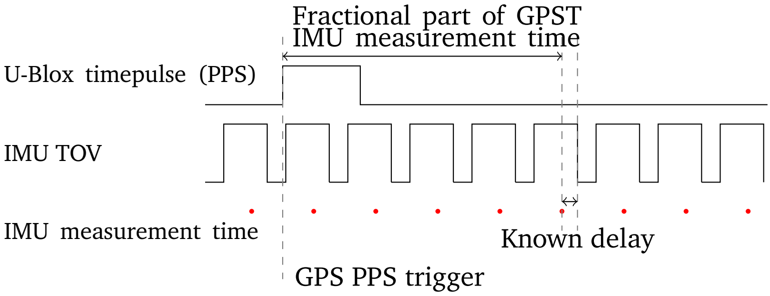


Figure 2.4: Time syncronization using the SenTiBoard.

MEKF. While the PPK positions are very precise, the accuracy depends on how well the location of the base station is known. Here the position of the base station was assumed to be the average of the "single" position calculated using pseudoranges. Unlike with the implemented algorithm, both GPS, GLONASS and European Geostationary Navigation Overlay System (EGNOS) corrections were used in RTKLIB to calculate the reference signals for comparison.

2.7.1 Body frame definition and IMU calibration

When working with vehicle attitude estimation a choice which must be made is how the body frame $\{b\}$ should be defined. One possibility is to use the IMU measurement frame $\{m\}$ as the definition. The calibration matrices $\mathbf{S}_g$ and $\mathbf{S}_f$ would then not include rotation, only scale factor and axis non-orthogonality corrections. The GNSS receiver antenna positions would then need to be known in this frame, which can be difficult if the exact orientation of the IMU in the UAV fuselage is uncertain. Online estimation of the antenna lever arms would then be a possibility. Another option is to define the body frame by the antenna positions themselves, and treat $\{m\}$ and $\{b\}$ as different frames with the same origin.

In this case the vehicle body frame was defined as having its origin at the location of the IMU, x-axis along the baseline between antennas (from back antenna to front), y-axis towards the starboard wing and z-axis downwards giving a right-handed coordinate frame. This allows the simple computation of pitch and heading from antenna positions by using the four-quadrant inverse tangent function. This does however leave the roll-component of the body frame uncertain, in the absence of a third antenna. A separate "alignment" mode was implemented in the MEKF, where the 18 values of the IMU calibration matrices $\mathbf{S}_g$ and $\mathbf{S}_g$ used in (2.18) and (2.22) were estimated online. In order to make these observable, the roll estimates from the autopilot, a Pixhawk 2, were used as measurements in the MEKF. For the testing of the MEKF the roll measurement and calibration values in the state vector were removed. The roll measurement used to enable observability of the matrices is of course an error source in itself, which would not be necessary if a third GNSS antenna had been used.

2.8 Results

Five different measurement configurations were tested:

1. Front antenna only, pseudorange and Doppler.
2. Both antennas, pseudorange, Doppler and DDCP:
    a) No time correction.
    b) Time correction handling satellite movement only, omitting UAV velocities in the correction term of (2.71).
    c) Time correction using measured Doppler shift, and a simpler measurement model replacing the estimated Doppler shift in (2.71) with the measurement.
    d) Time correction using the complete model (2.71).

The estimated clock errors and receiver measurement time difference are plotted in Fig. 2.5. The estimated mean and RMS errors when compared to PPK (except the roll error, which is compared to the flight controller estimate) are shown in Tab. 2.1. The

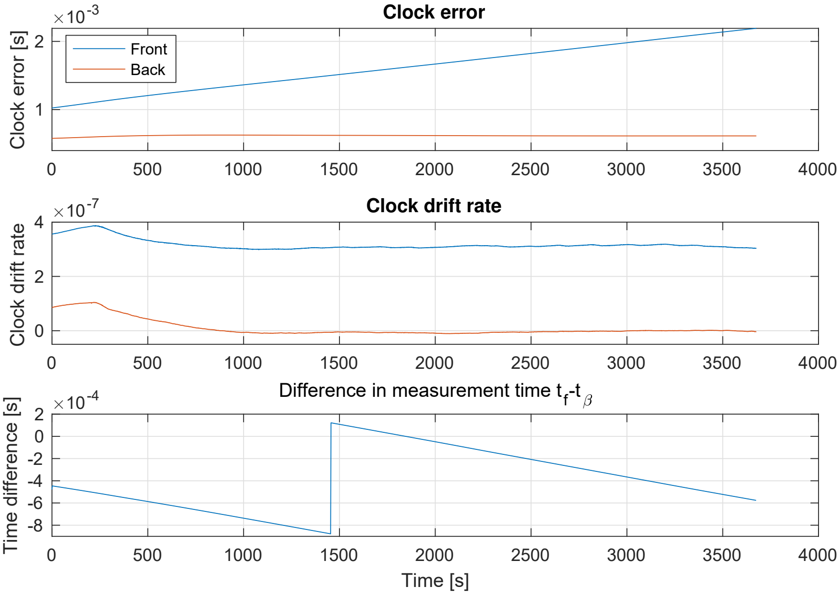


Figure 2.5: Estimates of the clock errors and the difference in measurement time between receivers, $t_f - t_\beta$. The jump at around 1450 seconds is caused by the front receiver changing its local measurement time by 1ms.

Table 2.1: MEKF estimation errors.

Mean	1	2a	2b	2c	2d
$\boldsymbol{p}^n_x$	0.150m	0.187m	0.188m	0.188m	0.188m
$\boldsymbol{p}^n_y$	1.530m	1.478m	1.482m	1.482m	1.482m
$\boldsymbol{p}^n_z$	0.856m	1.067m	1.068m	1.065m	1.065m
$\boldsymbol{v}^n_x$	$1.104\frac{\text{mm}}{\text{s}}$	$0.217\frac{\text{mm}}{\text{s}}$	$-1.315\frac{\text{mm}}{\text{s}}$	$-1.104\frac{\text{mm}}{\text{s}}$	$-1.124\frac{\text{mm}}{\text{s}}$
$\boldsymbol{v}^n_y$	$-0.597\frac{\text{mm}}{\text{s}}$	$-0.627\frac{\text{mm}}{\text{s}}$	$-0.967\frac{\text{mm}}{\text{s}}$	$-0.579\frac{\text{mm}}{\text{s}}$	$-0.570\frac{\text{mm}}{\text{s}}$
$\boldsymbol{v}^n_z$	$0.155\frac{\text{mm}}{\text{s}}$	$1.114\frac{\text{mm}}{\text{s}}$	$1.228\frac{\text{mm}}{\text{s}}$	$0.673\frac{\text{mm}}{\text{s}}$	$0.656\frac{\text{mm}}{\text{s}}$
Roll	0.129°	0.156°	0.159°	0.160°	0.160°
Pitch	0.524°	0.542°	0.529°	0.519°	0.519°
Heading	0.563°	0.757°	0.102°	0.099°	0.099°
RMS					
$\boldsymbol{p}^n_x$	0.356m	0.336m	0.337m	0.337m	0.337m
$\boldsymbol{p}^n_y$	1.548m	1.494m	1.497m	1.498m	1.498m
$\boldsymbol{p}^n_z$	1.843m	1.834m	1.837m	1.838m	1.838m
$\boldsymbol{v}^n_x$	$9.936\frac{\text{cm}}{\text{s}}$	$9.998\frac{\text{cm}}{\text{s}}$	$9.850\frac{\text{cm}}{\text{s}}$	$9.838\frac{\text{cm}}{\text{s}}$	$9.838\frac{\text{cm}}{\text{s}}$
$\boldsymbol{v}^n_y$	$4.095\frac{\text{cm}}{\text{s}}$	$4.087\frac{\text{cm}}{\text{s}}$	$3.980\frac{\text{cm}}{\text{s}}$	$3.946\frac{\text{cm}}{\text{s}}$	$3.946\frac{\text{cm}}{\text{s}}$
$\boldsymbol{v}^n_z$	$8.354\frac{\text{cm}}{\text{s}}$	$8.137\frac{\text{cm}}{\text{s}}$	$8.138\frac{\text{cm}}{\text{s}}$	$8.144\frac{\text{cm}}{\text{s}}$	$8.143\frac{\text{cm}}{\text{s}}$
Roll	0.898°	0.903°	0.902°	0.902°	0.902°
Pitch	1.318°	1.322°	1.310°	1.308°	1.308°
Heading	0.731°	2.150°	0.372°	0.381°	0.381°

attitude errors are also plotted in Fig. 2.6. The differences in position, velocity, roll and pitch results are only minor. Case 2a, with no measurement time correction, is able to resolve integer ambiguities only 18.9% of the time, including many false fixes resulting in a large heading error. This is to be expected from Fig. 2.2, noting that the estimated float ambiguities do not correspond well to integers if the difference in measurement time is ignored. For case 2b, integer ambiguity resolution failed with a rate of 4.908×10^{-4} and for 2c and 2d only 1.088×10^{-6}. Cases 2c and 2d give nearly identical results, with mean and RMS differences of 1.80×10^{-5}°and 1.64×10^{-4}°respectively, suggesting that the more detailed measurement model may not be worthwhile.

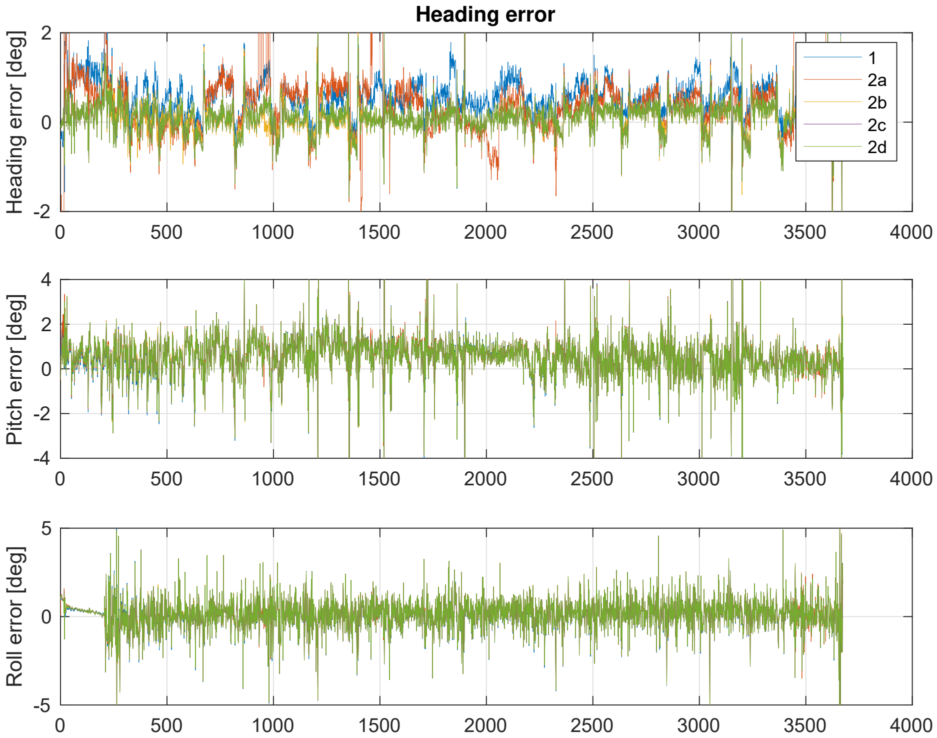


Figure 2.6: Attitude error plotted as Euler angles

2.9 Conclusion

The results show that use of carrier phase interferometry using low cost receivers can improve the heading estimate compared to the use of a single receiver, and that handling the difference in measurement time is necessary to reliably resolve the integer ambiguities. However, the proposed measurement model yields estimates which are almost identical to those resulting from handling the measurement times using measured Doppler shift, despite increased complexity. Estimates of roll, pitch, position and velocity are almost identical is all the tested cases. An extension left for the future is to add a third receiver, making attitude completely observable, allowing IMU errors to be estimated more accurately. The use of a higher IMU sample rate to reduce integration errors and improve IMU-GNSS timing should also improve the estimates.

Chapter 3

On the Error-Reset Covariance Transform for the Multiplicative Extended Kalman Filter

This chapter is based on the publication

- [130] M. L. Sollie, T. H. Bryne, and T. A. Johansen, “On the Error-Reset Covariance Transform for the Multiplicative Extended Kalman Filter,” *Submitted, in revision*, 2022

and considers a small implementation detail of the MEKF used in Chapter 2, which was not included in that chapter, since it is a topic of disagreement in the literature.

3.1 Nomenclature

- $\{b\}$ body frame
- $\{\bar{b}\}$ nominal body frame estimate
- $\boldsymbol{b}_g^b$ Angular rate sensor bias in body frame
- $\boldsymbol{F}_k$ continuous time system matrix
- $\boldsymbol{f}(\cdot)$ discrete time state prediction function
- $\boldsymbol{f}_{\text{err}}(\cdot)$ continuous time error state dynamics function
- $\boldsymbol{H}$ linearized measurement matrix
- $\boldsymbol{h}(\cdot)$ measurement model function
- $\boldsymbol{I}$ identity matrix
- $\boldsymbol{K}$ Kalman gain matrix

- k timestep
- $\boldsymbol{l}$ lever-arm vector for position measurement
- $\{n\}$ navigation frame
- $\boldsymbol{P}$ covariance matrix. The covariance of a specific variable $\boldsymbol{x}$ can be written $\boldsymbol{P}_k = \mathrm{Cov}(\boldsymbol{x})$ for $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_k)$ at timestep k
- $\boldsymbol{R}_k = \mathrm{Cov}(\boldsymbol{v})$ for $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$ is the measurement covariance matrix at time step k
- $\boldsymbol{Q}_k = \mathrm{Cov}(\boldsymbol{w})$ for $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ is the process noise matrix at timestep k
- $\boldsymbol{q}$ Unit quaternion $\begin{bmatrix} q_w & \boldsymbol{q}_v^\top \end{bmatrix}^\top$
- $\boldsymbol{q}(\delta\boldsymbol{a})$ quaternion corresponding to the three-dimensional rotation parameters $\delta\boldsymbol{a}$
- $\boldsymbol{R}_b^n$ rotation matrix transforming vectors from frame $\{b\}$ to $\{n\}$, $\boldsymbol{v}^n = \boldsymbol{R}_b^n \boldsymbol{v}^b$
- $\boldsymbol{S}(\boldsymbol{v}) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$ skew-symmetrix matrix of the vector $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^\top$
- $\boldsymbol{T}$ small-error attitude equivalence transform matrix, e.g. $\boldsymbol{T}_{\boldsymbol{\epsilon}_1}^{\boldsymbol{\epsilon}_2}$ such that $\boldsymbol{R}(\boldsymbol{\epsilon}_1 + \delta\boldsymbol{\epsilon}_1)\boldsymbol{R}(\boldsymbol{\epsilon}_2) = \boldsymbol{R}(\boldsymbol{\epsilon}_1)\boldsymbol{R}(\boldsymbol{\epsilon}_2 + \boldsymbol{T}_{\boldsymbol{\epsilon}_1}^{\boldsymbol{\epsilon}_2}\delta\boldsymbol{\epsilon}_1)$
- Δt prediction interval
- $\delta\boldsymbol{a}$ three-dimensional rotation error parameters
- $\delta\boldsymbol{a}(\boldsymbol{q})$ three-dimensional rotation parameters corresponding to the quaternion $\boldsymbol{q}$
- $\delta\boldsymbol{b}_g$ Angular rate sensor bias error
- $\boldsymbol{\epsilon}$ infinitesimal attitude error
- $\boldsymbol{\theta}$ rotation vector
- $\boldsymbol{\Phi}_k$ state transition matrix
- $\boldsymbol{\omega}_{ab}^c$ angular rate of frame $\{b\}$ relative to frame $\{a\}$, decomposed in frame $\{c\}$
- $\otimes$ Hamiltonian quaternion product [128]
- $\oplus$ generalized composition (quaternion product, additive sum, attitude parameter composition)
- $(\cdot)^+$ posterior updated value
- $(\cdot)^-$ prior predicted value
- $(\cdot)^\top$ vector or matrix transpose
- $||\boldsymbol{v}||$ Euclidean norm of $\boldsymbol{v}$
- $\hat{\boldsymbol{v}}$ expected value of the stochastic vector $\boldsymbol{v}$

3.2 Introduction

The Multiplicative Extended Kalman Filter (MEKF) [81] is a state estimation algorithm used for navigation, where a unit quaternion is used for a nominal attitude estimate, propagated outside the filter, while a three-dimensional attitude parameterization representing small errors from the nominal orientation is estimated by an error-state EKF. Additive states such as position, velocity and sensor biases can also be included. Consider for example a state consisting of a quaternion and biases of an angular rate sensor (gyro)

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}_b^n & \boldsymbol{b}_g^b \end{bmatrix}^\top, \tag{3.1}$$

which can be split in a nominal estimate and an error component,

$$\hat{\boldsymbol{x}} = \begin{bmatrix} \hat{\boldsymbol{q}}_b^n & \hat{\boldsymbol{b}}_g^b \end{bmatrix}^\top, \tag{3.2}$$

$$\delta\boldsymbol{x} = \begin{bmatrix} \delta\boldsymbol{a} & \delta\boldsymbol{b}_g \end{bmatrix}^\top, \tag{3.3}$$

related as

$$\boldsymbol{q}_b^n = \hat{\boldsymbol{q}}_b^n \otimes \delta\boldsymbol{a} \tag{3.4}$$

$$\boldsymbol{b}_g^b = \hat{\boldsymbol{b}}_g^b + \delta\boldsymbol{b}_g. \tag{3.5}$$

This can be written in state-space form as

$$\boldsymbol{x} = \hat{\boldsymbol{x}} \oplus \delta\boldsymbol{x}. \tag{3.6}$$

Note that the rotation matrix convention $\boldsymbol{R}_b^n$ is used, opposite of [102, 103, 81]. That is, $\boldsymbol{R}_b^n = \boldsymbol{A}^\top$ where $\boldsymbol{A}$ is that attitude matrix used by the cited works. There are several common parameter choices for $\delta\boldsymbol{a}$, such as two times the Gibbs vector, four times the modified Rodrigues parameters, or the rotation vector [81].

The EKF estimates the error $\delta\boldsymbol{x}$. When a measurement is processed, the estimate of the error is updated to a non-zero value, which is then injected into the nominal estimate, before resetting the error estimate to zero. This means that there is no need to propagate the error estimate over time since it is always trivially zero. This parameter combination provides a globally non-singular attitude estimate as long as the error estimate is kept small, away from singularities of the three-dimensional attitude error parameterization, and at the same time avoids the use of a singular 4x4 covariance matrix for the three degrees of freedom of orientation.

The steps of the MEKF are propagation of the nominal state

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_k), \tag{3.7}$$

propagation of the covariance estimate for the error state

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{f}_{\text{err}}(\delta\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{3.8}$$

$$\boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{f}_{\text{err}}}{\partial \delta\boldsymbol{x}}\right|_{\delta\boldsymbol{x}=0} \tag{3.9}$$

$$\boldsymbol{\Phi}_k = e^{\boldsymbol{F}_k \Delta t} \tag{3.10}$$

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_k \boldsymbol{P}_{k-1}^+ \boldsymbol{\Phi}_k^\top + \boldsymbol{Q}_k \tag{3.11}$$

measurement update

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}} \tag{3.12}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^\top (\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^\top + \boldsymbol{R}_k)^{-1} \tag{3.13}$$

$$\delta\hat{\boldsymbol{x}} = \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)) \tag{3.14}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{3.15}$$

and error injection and reset

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- \oplus \delta\hat{\boldsymbol{x}}, \quad \delta\hat{\boldsymbol{x}} = 0 \tag{3.16}$$

There is some disagreement in the literature on whether moving the estimated attitude correction from the three-dimensional error value into the nominal quaternion,

$$\hat{\boldsymbol{q}}_b^n \leftarrow \hat{\boldsymbol{q}}_b^n \otimes \boldsymbol{q}(\delta\hat{\boldsymbol{a}}), \quad \delta\hat{\boldsymbol{a}} \leftarrow \boldsymbol{0}, \tag{3.17}$$

also requires a modification to the covariance matrix,

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}_k^-, \tag{3.18}$$

$$\boldsymbol{P}_{k,\text{reset}}^+ = \boldsymbol{T}\boldsymbol{P}_k^+\boldsymbol{T}^\top. \tag{3.19}$$

This paper attempts to clarify this specific aspect of the MEKF.

3.3 Previous work

In [81, 80] it is argued against the covariance modification in (3.19), respectively stating that

"*The reset does not modify the covariance, since it neither increases nor decreases the total information content of the estimate; it merely moves this information from one part of the attitude representation to another.*"[81]

"*This reset rotates the reference frame for the attitude covariance, so that we might expect the covariance to be rotated, even though no new information is added. However,* *the covariance depends on the assumed statistics of the measurements, not on the actual measurements. Therefore, because the update is zero mean, the mean rotation caused by the reset is actually zero, and so the covariance is in fact not affected by the reset.*"[80]

In [80] it is, however, acknowledged in a footnote that there are authors who disagree with this statement. Despite not advocating its use, and asserting that ignoring it has not led to problems in applications, [82] shows that if a transform of (3.19) should be used, it would have the form

$$\boldsymbol{T} = \frac{1}{1+\frac{1}{4}||\delta\hat{\boldsymbol{a}}||^2}\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}})\right) \tag{3.20}$$

$$\boldsymbol{P}^+_{k,\text{reset}} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{P}^+_k \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^\top \tag{3.21}$$

when using twice the Gibbs vector as the attitude error parameters, and additional additive states with the appropriately sized identity matrix $\boldsymbol{I}$. It is important to note that this transform is not a rotation, but a parameter transform that maps from uncertainty in the parameter value $\delta\boldsymbol{a}$, representing the transform between the estimated body frame and the prior predicted estimate, to small-angle rotation uncertainty about the body frame axes.

[128] states "*After error injection into the nominal state, the error state mean $\delta\hat{\boldsymbol{x}}$ gets reset. This is especially relevant for the orientation part, as the new orientation error will be expressed locally with respect to the orientation frame of the new nominal state. To make the ESKF update complete, the covariance of the error needs to be updated according to this modification.*"[128]

and derives the first-order transform equivalent to (3.20),

$$\boldsymbol{T} = \boldsymbol{I} - \frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}}), \tag{3.22}$$

valid for both the rotation vector, two times the Gibbs vector and four times the modified Rodrigues parameters, but not valid for Euler angles, as choice of attitude error parameterization.

[108] uses the Gibbs vector as attitude error parameterization (that is, not two times the Gibbs vector) and finds the transform matrix equivalent to that found for the covariance projection approach, using twice the Gibbs vector, in [82] and states:

"*Each time the quaternion estimate is propagated or updated, the covariance estimate must be mapped from one tangent space to the next.*"[108]

"*(...) it is clear that we need to transform $\boldsymbol{P}^+_k$ to the tangent space of the updated quaternion $T_{\hat{\boldsymbol{q}}_k}S^3$ to maintain a consistent local representation of the covariance estimate*"[108]

Transforming or mapping the parameter covariance estimate from one tangent frame to another is a rotation if the expected attitude offset in both frames are zero, that is

$\delta\hat{\boldsymbol{a}} = \mathbf{0}$. Suppose we have an attitude offset in a tangent frame, and the covariance represents errors in the offset parameter values. In that case, a transform to the origin of another tangent frame is not a rotation.

A brief discussion of the covariance reset based on the results of [108, 82] can be found in [21, p.91].

[103] uses a different covariance modification,

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R}(\delta\hat{\boldsymbol{a}}) & \boldsymbol{Z}(\delta\hat{\boldsymbol{a}}) \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \tag{3.23}$$

with

$$\boldsymbol{R}(\boldsymbol{\theta}) = (\cos\|\boldsymbol{\theta}\|)\mathbf{I} - \frac{\sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|}\boldsymbol{S}(\boldsymbol{\theta}) + \frac{1-\cos\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^2}\boldsymbol{\theta}\boldsymbol{\theta}^T, \tag{3.24}$$

$$\boldsymbol{Z}(\boldsymbol{\theta}) = \tau\left[\frac{\sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|}\mathbf{I} - \frac{1-\cos\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^2}\boldsymbol{S}(\boldsymbol{\theta}) + \frac{\|\boldsymbol{\theta}\| - \sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^3}\boldsymbol{\theta}\boldsymbol{\theta}^T\right], \tag{3.25}$$

where (3.24) is the Rodrigues rotation matrix formula for rotation vectors and (3.25) is the time interval τ multiplied by the average rotation matrix during the interval (see [42, p.188]). (3.23) is, for the attitude correction $\delta\hat{\boldsymbol{a}}$, equivalent to the linearized state transition matrix with the inertial measurement unit (IMU) angular increment $\boldsymbol{\phi}_{k-1}(\tau)$,

$$\boldsymbol{\Phi}(t_{k-1}+\tau, t_{k-1}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\phi}_{k-1}(\tau)) & \boldsymbol{Z}(\boldsymbol{\phi}_{k-1}(\tau)) \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \tag{3.26}$$

used for covariance propagation in (3.11). This is explained as: "*The updated covariance $\boldsymbol{P}_{k|k}$ is also referenced to the predicted attitude frame. The reference frame for $\boldsymbol{P}_{k|k}$ should be updated from $\boldsymbol{A}_{k|k-1}$ to $\boldsymbol{A}_{k|k}$ by applying the rotation correction $\delta\hat{\boldsymbol{\phi}}_{k|k}$.*"[103]

The inclusion of $\boldsymbol{Z}(\delta\hat{\boldsymbol{a}})$ results in the correction $\delta\hat{\boldsymbol{a}}$ being treated as a gyro measurement over a duration τ (sampling time of the gyro/IMU) even though no time has passed, and propagates gyro bias uncertainty for this period into the attitude. It is also stated that "*For efficiency, this frame update can be combined with the attitude prediction on the next filter cycle by computing R and S from the combined rotation $\hat{\boldsymbol{\phi}}_k^{k+1} \circ \delta\hat{\boldsymbol{\phi}}_{k|k}$*". This would, however, not result in the same result, as

$$\begin{bmatrix} \boldsymbol{R}(\boldsymbol{\phi}_{k-1}(\tau)) & \boldsymbol{Z}(\boldsymbol{\phi}_{k-1}(\tau)) \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \boldsymbol{R}(\delta\hat{\boldsymbol{\phi}}_{k|k}) & \boldsymbol{Z}(\delta\hat{\boldsymbol{\phi}}_{k|k}) \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \tag{3.27}$$

$$\neq \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\phi}_{k-1}(\tau) \circ \delta\hat{\boldsymbol{\phi}}_{k|k}) & \boldsymbol{Z}(\boldsymbol{\phi}_{k-1}(\tau) \circ \delta\hat{\boldsymbol{\phi}}_{k|k}) \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \tag{3.28}$$

$$\approx \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\phi}_{k-1}(\tau)) & \boldsymbol{Z}(\boldsymbol{\phi}_{k-1}(\tau)) \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \boldsymbol{R}(\delta\hat{\boldsymbol{\phi}}_{k|k}) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \tag{3.29}$$

since (3.27) will propagate gyro bias twice while (3.28) only does it once. We note that the injection is not equivalent to the passage of time, and therefore the covariance should not be modified using the state transition matrix unless $\tau = 0$ is used.

In a different paper from the same author, it is stated: *"In the reduced-order filter, this corresponds to updating $\boldsymbol{P}_3^+$ at time $k-1$ with $\boldsymbol{T}(\boldsymbol{r}_{k-1})\boldsymbol{P}_3^+\boldsymbol{T}(\boldsymbol{r}_{k-1})^\top$, where $\boldsymbol{T}(\boldsymbol{r}_{k-1})$ is the transformation matrix corresponding to $\boldsymbol{r}_{k-1}$. To the author's knowledge this covariance propagation is not mentioned in the literature and is never done in practice, though it should be done in order to maintain the correct frame of reference for the covariance matrix and to prevent "leakage" in the covariance of the full quaternion filter."*[102]. The matrix $\boldsymbol{T}(\boldsymbol{r}_{k-1})$ is the same as (3.24), meaning that in this case the state transition matrix is not used as in (3.23).

3.4 Parameter covariance transform

For the rotation from body frame to navigation frame, the uncertainty in an estimate can be represented in multiple ways. Consider the sequence of rotations

$$\boldsymbol{R}_b^n = \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{R}(\boldsymbol{\epsilon}^{\bar{b}})\boldsymbol{R}(\delta\boldsymbol{a}_b^{\bar{b}})\boldsymbol{R}(\boldsymbol{\epsilon}^b) \tag{3.30}$$

where $\boldsymbol{q}_{\bar{b}}^n$ is a nominal quaternion, $\boldsymbol{\epsilon}^{\bar{b}}$ is an error in the nominal frame with expected value $\boldsymbol{0}$, $\delta\boldsymbol{a}_b^{\bar{b}}$ is an offset from the nominal frame, which can be nonzero, and $\boldsymbol{\epsilon}^b$ is an error in the body frame with expected value $\boldsymbol{0}$. If the MEKF is reset, we have $\delta\hat{\boldsymbol{a}}_b^{\bar{b}} = \boldsymbol{0}$, and uncertainty of $\boldsymbol{\epsilon}^{\bar{b}}$, $\delta\boldsymbol{a}_b^{\bar{b}}$, $\boldsymbol{\epsilon}^b$ all have the same effect. This can then be reduced to

$$\begin{aligned} \boldsymbol{R}_b^n &= \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{R}(\delta\boldsymbol{a}_b^{\bar{b}}) \\ \delta\boldsymbol{a}_b^{\bar{b}} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_{\delta\boldsymbol{a}_b^{\bar{b}}}) \end{aligned} \tag{3.31}$$

After a measurement update, before a reset, we will in general have $\delta\hat{\boldsymbol{a}}_b^{\bar{b}} \neq \boldsymbol{0}$, and therefore, uncertainty in the different errors have different effects. We could represent the uncertainty in the estimated body frame

$$\boldsymbol{\epsilon}^{\bar{b}} = \boldsymbol{0}, \quad \delta\boldsymbol{a}_b^{\bar{b}} = \delta\hat{\boldsymbol{a}}_b^{\bar{b}}, \quad \boldsymbol{\epsilon}^b \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_{\boldsymbol{\epsilon}^b}), \tag{3.32}$$

in the prior nominal body frame,

$$\boldsymbol{\epsilon}^{\bar{b}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}_{\boldsymbol{\epsilon}^{\bar{b}}}),\ \delta\boldsymbol{a}_b^{\bar{b}} = \delta\hat{\boldsymbol{a}}_b^{\bar{b}},\ \boldsymbol{\epsilon}^b = \boldsymbol{0}, \tag{3.33}$$

or instead represent it as uncertainty in the parameters of the offset from the nominal frame,

$$\boldsymbol{\epsilon}^{\bar{b}} = \boldsymbol{0},\ \delta\boldsymbol{a}_b^{\bar{b}} \sim \mathcal{N}(\delta\hat{\boldsymbol{a}}_b^{\bar{b}}, \boldsymbol{P}_{\delta\boldsymbol{a}_b^{\bar{b}}}),\ \boldsymbol{\epsilon}^b = \boldsymbol{0} \tag{3.34}$$

Transforming between these covariance matrices requires the transformation matrix between small perurbations of $\boldsymbol{\epsilon}^{\bar{b}}$, $\boldsymbol{\epsilon}^b$, and $\delta\boldsymbol{a}_b^{\bar{b}}$. The transform from small-angle errors about the body frame axes to parameter error can be expressed as

$$T^{\delta \boldsymbol{a}^{\bar{b}}_{b}}_{\boldsymbol{\epsilon}^{b}} = \frac{\partial}{\partial \boldsymbol{\epsilon}^{b}} \delta \boldsymbol{a}(\boldsymbol{q}(\hat{\boldsymbol{a}}^{\bar{b}}_{b}) \otimes \boldsymbol{q}(\boldsymbol{\epsilon}^{b}))\bigg|_{\boldsymbol{\epsilon}^{b}=\mathbf{0}} \tag{3.35}$$

$$\boldsymbol{P}_{\boldsymbol{a}^{\bar{b}}_{b}} = \boldsymbol{T}^{\delta \boldsymbol{a}^{\bar{b}}_{b}}_{\boldsymbol{\epsilon}^{b}} \boldsymbol{P}_{\boldsymbol{\epsilon}^{b}} (\boldsymbol{T}^{\delta \boldsymbol{a}^{\bar{b}}_{b}}_{\boldsymbol{\epsilon}^{b}})^{\top}, \tag{3.36}$$

from small-angle error about the prior predicted body frame axes to parameter errors as

$$T^{\delta \boldsymbol{a}^{\bar{b}}_{b}}_{\boldsymbol{\epsilon}^{\bar{b}}} = \frac{\partial}{\partial \boldsymbol{\epsilon}^{\bar{b}}} \delta \boldsymbol{a}(\boldsymbol{q}(\boldsymbol{\epsilon}^{\bar{b}}) \otimes \boldsymbol{q}(\delta \hat{\boldsymbol{a}}^{\bar{b}}_{b}))\bigg|_{\boldsymbol{\epsilon}^{\bar{b}}=\mathbf{0}} \tag{3.37}$$

$$\boldsymbol{P}_{\delta \boldsymbol{a}^{\bar{b}}_{b}} = \boldsymbol{T}^{\delta \boldsymbol{a}^{\bar{b}}_{b}}_{\boldsymbol{\epsilon}^{\bar{b}}} \boldsymbol{P}_{\boldsymbol{\epsilon}^{\bar{b}}} (\boldsymbol{T}^{\delta \boldsymbol{a}^{\bar{b}}_{b}}_{\boldsymbol{\epsilon}^{\bar{b}}})^{\top} \tag{3.38}$$

and between small angle-errors in the two frames as

$$\boldsymbol{T}^{\boldsymbol{\epsilon}^{\bar{b}}}_{\boldsymbol{\epsilon}^{b}} = \boldsymbol{R}(\delta \hat{\boldsymbol{a}}^{\bar{b}}_{b}) \tag{3.39}$$

$$\boldsymbol{P}_{\boldsymbol{\epsilon}^{\bar{b}}} = \boldsymbol{T}^{\boldsymbol{\epsilon}^{\bar{b}}}_{\boldsymbol{\epsilon}^{b}} \boldsymbol{P}_{\boldsymbol{\epsilon}^{b}} (\boldsymbol{T}^{\boldsymbol{\epsilon}^{\bar{b}}}_{\boldsymbol{\epsilon}^{b}})^{\top}. \tag{3.40}$$

The intuitive difference between these is that $\boldsymbol{P}_{\boldsymbol{\epsilon}^{b}}$ and $\boldsymbol{P}_{\boldsymbol{\epsilon}^{\bar{b}}}$ represent angular perturbations about the axes of the nominal body and estimated body frames, while $\boldsymbol{P}_{\delta \boldsymbol{a}^{\bar{b}}_{b}}$ represent perturbations in the nonzero parameter values of $\delta \boldsymbol{a}^{\bar{b}}_{b}$. Expressions for these transforms are easy to find, because small changes in values transform the same as parameter rates, using the differential equation for the parameter. For example, for the rotation vector, we have the rate transform [80, p.73]:

$$\dot{\boldsymbol{\theta}}^{\bar{b}}_{b} = \left(\boldsymbol{I} + \frac{1}{2} \boldsymbol{S}(\boldsymbol{\theta}^{\bar{b}}_{b}) + \frac{1}{\|\boldsymbol{\theta}^{\bar{b}}_{b}\|^{2}} \left(1 - \frac{\|\boldsymbol{\theta}^{\bar{b}}_{b}\|}{2} \cot\left(\frac{\|\boldsymbol{\theta}^{\bar{b}}_{b}\|}{2} \right) \boldsymbol{S}(\boldsymbol{\theta}^{\bar{b}}_{b})^{2} \right) \right) \boldsymbol{\omega}^{b}_{\bar{b}b}, \tag{3.41}$$

and therefore also have for small-angle error relation

$$\delta \boldsymbol{\theta}^{\bar{b}}_{b} = \left(\boldsymbol{I} + \frac{1}{2} \boldsymbol{S}(\boldsymbol{\theta}^{\bar{b}}_{b}) + \frac{1}{\|\boldsymbol{\theta}^{\bar{b}}_{b}\|^{2}} \left(1 - \frac{\|\boldsymbol{\theta}^{\bar{b}}_{b}\|}{2} \cot\left(\frac{\|\boldsymbol{\theta}^{\bar{b}}_{b}\|}{2} \right) \boldsymbol{S}(\boldsymbol{\theta}^{\bar{b}}_{b})^{2} \right) \right) \boldsymbol{\epsilon}^{b}. \tag{3.42}$$

This transformation matrix can be used to transform the covariance matrix from body frame attitude errors to a covariance matrix for rotation vector parameters, rotating from the body frame to another frame. The inverse relation is

$$\boldsymbol{\omega}^{b}_{\bar{b}b} = \left(\boldsymbol{I} - \frac{1 - \cos\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^{2}} \boldsymbol{S}(\boldsymbol{\theta}) + \frac{\|\boldsymbol{\theta}\| - \sin\|\boldsymbol{\theta}\|}{\|\boldsymbol{\theta}\|^{3}} \boldsymbol{S}(\boldsymbol{\theta})^{2} \right) \dot{\boldsymbol{\theta}}^{\bar{b}}_{b}. \tag{3.43}$$

In [128] this is referred to as the *right Jacobian* of the rotation vector. Of other popular attitude parametrizations we have for two times the Gibbs vector

$$\delta \dot{\boldsymbol{a}}^{\bar{b}}_{b} = \left(\boldsymbol{I} + \frac{1}{2} \boldsymbol{S}(\delta \boldsymbol{a}^{\bar{b}}_{b}) + \frac{1}{4} \delta \boldsymbol{a}^{\bar{b}}_{b} (\delta \boldsymbol{a}^{\bar{b}}_{b})^{\top} \right) \boldsymbol{\omega}^{b}_{\bar{b}b}, \tag{3.44}$$

$$\boldsymbol{\omega}^{b}_{\bar{b}b} = \frac{4}{4 + \|\delta \boldsymbol{a}^{\bar{b}}_{b}\|^2} \left(\boldsymbol{I} - \frac{1}{2} \boldsymbol{S}\left(\delta \boldsymbol{a}^{\bar{b}}_{b}\right) \right) \dot{\delta \boldsymbol{a}}^{\bar{b}}_{b}, \tag{3.45}$$

and four times the modified Rodrigues parameters

$$\dot{\delta \boldsymbol{a}}^{\bar{b}}_{b} = \left(\left(1 - \frac{1}{16} \|\delta \boldsymbol{a}^{\bar{b}}_{b}\|^2 \right) \boldsymbol{I} + \frac{1}{2} \boldsymbol{S}(\delta \boldsymbol{a}^{\bar{b}}_{b}) + \frac{1}{8} \delta \boldsymbol{a}^{\bar{b}}_{b} (\delta \boldsymbol{a}^{\bar{b}}_{b})^{\top} \right) \boldsymbol{\omega}^{b}_{\bar{b}b}, \tag{3.46}$$

$$\boldsymbol{\omega}^{b}_{\bar{b}b} = \frac{16}{16 + \|\delta \boldsymbol{a}^{\bar{b}}_{b}\|^2} \left(\boldsymbol{I} + \frac{2\boldsymbol{S}(\delta \boldsymbol{a}^{\bar{b}}_{b})^2 - 8\boldsymbol{S}(\delta \boldsymbol{a}^{\bar{b}}_{b})}{16 + \|\delta \boldsymbol{a}^{\bar{b}}_{b}\|^2} \right) \dot{\delta \boldsymbol{a}}^{\bar{b}}_{b}. \tag{3.47}$$

A first-order approximation for these three parameter choices is

$$\dot{\delta \boldsymbol{a}}^{\bar{b}}_{b} = \left(\boldsymbol{I} + \frac{1}{2} \boldsymbol{S}(\delta \boldsymbol{a}^{\bar{b}}_{b}) \right) \boldsymbol{\omega}^{b}_{\bar{b}b}, \tag{3.48}$$

$$\boldsymbol{\omega}^{b}_{\bar{b}b} = \left(\boldsymbol{I} - \frac{1}{2} \boldsymbol{S}\left(\delta \boldsymbol{a}^{\bar{b}}_{b}\right) \right) \dot{\delta \boldsymbol{a}}^{\bar{b}}_{b}. \tag{3.49}$$

Covariance can also be transformed to a 4x4 quaternion covariance matrix the same way, as stated by [122]:

"*The errors in the Euler-Rodrigues symmetric parameters can be written in terms of the attitude error vector in the same way that the rate of change of the quaternion was written in terms of the angular velocity.*"[122]

$$\begin{aligned} \dot{\boldsymbol{q}}^{n}_{b} &= \frac{1}{2} \boldsymbol{q}^{n}_{b} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}^{b}_{nb} \end{bmatrix} \\ &= \frac{1}{2} \Xi(\boldsymbol{q}^{n}_{b}) \boldsymbol{\omega}^{b}_{nb} \end{aligned} \tag{3.50}$$

$$\delta \boldsymbol{q}^{n}_{b} = \frac{1}{2} \Xi(\boldsymbol{q}^{n}_{b}) \boldsymbol{\epsilon}^{b} \tag{3.51}$$

$$\Xi(\boldsymbol{q}) = \begin{bmatrix} -\boldsymbol{q}_{v}^{\top} \\ q_{w} \boldsymbol{I} + \boldsymbol{S}(\boldsymbol{q}_{v}) \end{bmatrix} \tag{3.52}$$

The covariance can then be transformed as

$$\boldsymbol{P}_{\boldsymbol{q}^{n}_{b}} = \frac{1}{4} \Xi(\boldsymbol{q}^{n}_{b}) \boldsymbol{P}_{\boldsymbol{\epsilon}^{b}} \Xi(\boldsymbol{q}^{n}_{b})^{\top}. \tag{3.53}$$

This type of transform is used as an intermediate step in [108], for derivation of the reset covariance transform.

3.5 The covariance of the updated error estimate

With an expected zero error, the covariance is updated with a measurement matrix linearized about zero. Before the measurement update, we consider that the body and navigation frames relate as

$$\boldsymbol{R}_b^n = \boldsymbol{R}(\hat{\boldsymbol{q}}_{\bar{b}}^n)\boldsymbol{R}(\boldsymbol{\epsilon}), \tag{3.54}$$

$$\boldsymbol{P}^- = \mathrm{Cov}(\boldsymbol{\epsilon}). \tag{3.55}$$

Since the covariance is updated under the expectation of zero error, the posterior attitude and covariance are given as

$$\boldsymbol{R}_b^n = \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{R}(\boldsymbol{\epsilon}^{\bar{b}})\boldsymbol{R}(\delta\hat{\boldsymbol{a}}_b^{\bar{b}}), \tag{3.56}$$

$$\boldsymbol{P}^+ = \mathrm{Cov}(\boldsymbol{\epsilon}^{\bar{b}}), \tag{3.57}$$

meaning that the covariance is updated in the predicted frame regardless of the attitude correction found from the actual measurement. This means that we have a frame mismatch between the updated error estimate and its covariance due to the measurement model linearization. Since $\delta\hat{\boldsymbol{a}}_b^{\bar{b}}$ is updated to first order, it is not affected by the assumed interpretation of the error as any of the mentioned parameter choices, even though our interpretation does affect the rotation this results in, due to higher-order terms when converting to a rotation matrix or quaternion. We would like the posterior uncertainty to be given in the new body frame as

$$\boldsymbol{R}_b^n = \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{R}(\delta\boldsymbol{a}_b^{\bar{b}})\boldsymbol{R}(\boldsymbol{\epsilon}^b), \tag{3.58}$$

$$\boldsymbol{P}^+ = \mathrm{Cov}(\boldsymbol{\epsilon}^b), \tag{3.59}$$

such that the error estimate and the covariance are in the same frame. This is necessary as the state transition matrix $\boldsymbol{\Phi}$ in (3.11) uses angular rate measurements $\boldsymbol{\omega}_{nb}^b$ assumed to be in the current estimated body frame, not the previous predicted nominal estimate, and thereby assumes that both the error estimate and the covariance are given in the updated frame. To achieve this, the covariance must be rotated using $\boldsymbol{R}(\delta\boldsymbol{a}_b^{\bar{b}})^\top$,

$$\mathrm{Cov}(\boldsymbol{\epsilon}^b) = \boldsymbol{R}(\delta\boldsymbol{a}_b^{\bar{b}})^\top \mathrm{Cov}(\boldsymbol{\epsilon}^{\bar{b}})\boldsymbol{R}(\delta\boldsymbol{a}_b^{\bar{b}}), \tag{3.60}$$

which is consistent with the rotation performed on the covariance by the state transition matrix in (3.11) due to angular rate measurements, and the same covariance modification done by [102] and [103] if we consider $\tau = 0$, such that gyro bias error is not propagated into the attitude uncertainty.

The transforms (3.20) and (3.22) assume that the posterior pre-update covariance matrix is $\mathrm{Cov}(\boldsymbol{a}_b^{\bar{b}})$, and that we want to transform this to $\mathrm{Cov}(\boldsymbol{\epsilon}^b)$. If this assumption was true, the proposed transforms of [82, 108, 128] would be correct. For two times the Gibbs vector as attitude error parameter this is

$$\boldsymbol{T}^{\boldsymbol{\epsilon}^b}_{\delta\boldsymbol{a}_b^{\bar{b}}} = \frac{\partial}{\partial\delta\boldsymbol{a}_b^{\bar{b}}}\delta\boldsymbol{a}(\boldsymbol{q}(-\delta\hat{\boldsymbol{a}}_b^{\bar{b}})\otimes\boldsymbol{q}(\delta\boldsymbol{a}_b^{\bar{b}}))\bigg|_{\delta\boldsymbol{a}_b^{\bar{b}}=\delta\hat{\boldsymbol{a}}_b^{\bar{b}}} = \frac{1}{1+\frac{1}{4}\|\delta\hat{\boldsymbol{a}}_b^{\bar{b}}\|^2}\left(\boldsymbol{I}-\frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}}_b^{\bar{b}})\right) \tag{3.61}$$

However, this requires the MEKF to know the updated error-state when performing the covariance update, which it does not. Unless an iterated measurement correction is performed, the measurement matrix $\boldsymbol{H}$ can depend only on previous measurements, not on the current. The covariance measurement update is based on *assumptions* about the measurement, not the measurement itself. This, however, does not mean that the updated covariance matrix is given in the form we would like, and therefore we may have to modify it, as in this case using (3.60). Depending on the chosen attitude error parameters, the attitude error rotation matrix is calculated as

$$\boldsymbol{R}(\delta\boldsymbol{a}) = \boldsymbol{I} + \frac{4}{4 + \|\delta\boldsymbol{a}\|^2}\boldsymbol{S}(\delta\boldsymbol{a}) + \frac{2}{4 + \|\delta\boldsymbol{a}\|^2}\boldsymbol{S}^2(\delta\boldsymbol{a}) \tag{3.62}$$

for two times the Gibbs vector,

$$\boldsymbol{R}(\delta\boldsymbol{a}) = \boldsymbol{I} + \frac{(1 - \frac{1}{16}\|\delta\boldsymbol{a}\|^2)\boldsymbol{S}(\delta\boldsymbol{a}) + \frac{1}{2}\boldsymbol{S}^2(\delta\boldsymbol{a})}{(1 - \frac{1}{16}\|\delta\boldsymbol{a}\|^2)^2} \tag{3.63}$$

for four times the modified Rodrigues parameters, or

$$\boldsymbol{R}(\delta\boldsymbol{a}) = \boldsymbol{I} + \frac{\sin\|\delta\boldsymbol{a}\|}{\|\delta\boldsymbol{a}\|}\boldsymbol{S}(\delta\boldsymbol{a}) + \frac{1 - \cos\|\delta\boldsymbol{a}\|}{\|\delta\boldsymbol{a}\|^2}\boldsymbol{S}^2(\delta\boldsymbol{a}) \tag{3.64}$$

for the rotation vector [80]. A first-order approximation is

$$\boldsymbol{R}(\delta\boldsymbol{a}) = \boldsymbol{I} + \boldsymbol{S}(\delta\boldsymbol{a}) \tag{3.65}$$

Note that the rotation transform also can be written as a combination of transforms via rotation parameter errors,

$$\boldsymbol{T}^{\boldsymbol{\epsilon}^b}_{\boldsymbol{\epsilon}^{\bar{b}}} = \boldsymbol{R}(\delta\boldsymbol{a}^{\bar{b}}_b)^\top = \boldsymbol{T}^{\boldsymbol{\epsilon}^b}_{\delta\boldsymbol{a}^{\bar{b}}_b}\boldsymbol{T}^{\delta\boldsymbol{a}^{\bar{b}}_b}_{\boldsymbol{\epsilon}^{\bar{b}}} \tag{3.66}$$

where $\boldsymbol{T}^{\delta\boldsymbol{a}^{\bar{b}}_b}_{\boldsymbol{\epsilon}^{\bar{b}}}$ for twice the Gibbs vector is given as

$$\boldsymbol{T}^{\delta\boldsymbol{a}^{\bar{b}}_b}_{\boldsymbol{\epsilon}^{\bar{b}}} = \frac{\partial}{\partial\boldsymbol{\epsilon}^b}\delta\boldsymbol{a}(\boldsymbol{q}(\boldsymbol{\epsilon}^b)\otimes\boldsymbol{q}(\delta\hat{\boldsymbol{a}}^{\bar{b}}_b))\bigg|_{\boldsymbol{\epsilon}^b=\boldsymbol{0}} = \left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}}^{\bar{b}}_b) + \frac{1}{4}\delta\hat{\boldsymbol{a}}^{\bar{b}}_b(\delta\hat{\boldsymbol{a}}^{\bar{b}}_b)^\top\right). \tag{3.67}$$

and $\boldsymbol{T}^{\boldsymbol{\epsilon}^b}_{\delta\boldsymbol{a}^{\bar{b}}_b}$ is given by (3.61). The second step $\boldsymbol{T}^{\boldsymbol{\epsilon}^b}_{\delta\boldsymbol{a}^{\bar{b}}_b}$ in this transform is the reset transform proposed in [82].

3.6 Example: Using position measurements with lever arm

The following example demonstrates how linearization at a non-zero error can take the covariance transform into account directly in the measurement matrix. Consider that we estimate position and attitude using an inertial navigation system (INS) and GNSS position measurements, with a (simplified) nominal state vector (compared to a complete INS)

$$\boldsymbol{x}_{\text{nom}} = \begin{bmatrix} \boldsymbol{q}^n_{\bar{b}} \\ \boldsymbol{p}^n_{\text{nom}} \end{bmatrix}. \tag{3.68}$$

The error in attitude and position estimated by an EKF uses the error-state

$$\delta \boldsymbol{x} = \begin{bmatrix} \delta \boldsymbol{a} \\ \delta \boldsymbol{p}^n \end{bmatrix}, \tag{3.69}$$

such that

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}_b^n \otimes \boldsymbol{q}(\delta \boldsymbol{a}) \\ \boldsymbol{p}_{\text{nom}}^n + \delta \boldsymbol{p}^n \end{bmatrix}. \tag{3.70}$$

Consider that we receive a position measurement $\boldsymbol{p}_m^n$ where the body frame lever arm from the IMU location to the GNSS antenna is $\boldsymbol{l}^b$. This is a typical case for a GNSS-aided INS. The full-state measurement model is then

$$\boldsymbol{p}_m^n = \boldsymbol{p}^n + \boldsymbol{R}(\boldsymbol{q}_b^n)\boldsymbol{l}^b + \boldsymbol{v} \tag{3.71}$$

with the measurement noise $\boldsymbol{v}$ and the error state model is

$$\boldsymbol{p}_m^n - \hat{\boldsymbol{p}}_m^n = \boldsymbol{p}^n - \hat{\boldsymbol{p}}^n + (\boldsymbol{R}(\boldsymbol{q}_b^n) - \boldsymbol{R}(\hat{\boldsymbol{q}}_b^n))\boldsymbol{l}^b + \boldsymbol{v}. \tag{3.72}$$

For an MEKF where the error-state is reset such that the predicted error is zero, $\delta \hat{\boldsymbol{x}} = \boldsymbol{0}$, which is the way the MEKF is typically is used, we linearize about $\delta \boldsymbol{x} = \boldsymbol{0}$ to find the measurement matrix. Using the first-order relation $\boldsymbol{R}(\delta \boldsymbol{a}) = \boldsymbol{I} + \boldsymbol{S}(\delta \boldsymbol{a})$ and $\boldsymbol{R}(\boldsymbol{q}_b^n) = \boldsymbol{R}(\hat{\boldsymbol{q}}_b^n)\boldsymbol{R}(\delta \boldsymbol{a})$ this can be written as

$$\boldsymbol{p}_m^n - \hat{\boldsymbol{p}}_m^n = \delta \boldsymbol{p}^n - \hat{\boldsymbol{R}}_b^n \boldsymbol{S}(\boldsymbol{l}^b)\delta \boldsymbol{a} + \boldsymbol{v} \tag{3.73}$$

yielding the measurement matrix

$$\boldsymbol{H}_0 = \begin{bmatrix} -\hat{\boldsymbol{R}}_b^n \boldsymbol{S}(\boldsymbol{l}^b) & \boldsymbol{I} \end{bmatrix}. \tag{3.74}$$

After updating the attitude error estimate using

$$\delta \hat{\boldsymbol{x}} = \boldsymbol{K}\left(\boldsymbol{p}_m^n - (\boldsymbol{p}_{\text{nom}}^n + \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{l}^b)\right) \tag{3.75}$$

and updating the covariance estimate, we can inject the error into the nominal estimate and reset the error, or if we have more measurements taken simultaneously, we can continue to use the remaining measurements before the reset.

"*Resets can be performed after each measurement update, in which case the term $H_a \hat{\boldsymbol{a}}(-)$ in Eq. (55) is identically zero; but the reset is usually delayed until all the updates for a set of simultaneous measurements have been performed, for computational efficiency.*"[81]

It is implicit in this statement that the linearization point remains at zero as before, as a re-linearization would remove the need for the term $\boldsymbol{H}_0 \delta \hat{\boldsymbol{x}}$ in the measurement prediction $\boldsymbol{h}(\boldsymbol{x}_{\text{nom}}) + \boldsymbol{H}_0 \delta \hat{\boldsymbol{x}}$. This is a reasonable approximation if the measurements only make very small corrections to the attitude estimate. Additional measurements would then update the error estimate as

$$\delta \hat{\boldsymbol{x}} \leftarrow \delta \hat{\boldsymbol{x}} + \boldsymbol{K}\left((\boldsymbol{p}_m^n - (\boldsymbol{p}_{\text{nom}}^n + \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{l}^b) - \boldsymbol{H}_0 \delta \hat{\boldsymbol{x}})\right). \tag{3.76}$$

If measurement corrections are large, the most reasonable it to perform error injection and reset, but alternatively, we could re-linearize the system about a nonzero error. Since we have $\boldsymbol{P} = \mathrm{Cov}(\boldsymbol{\epsilon}^{\bar{b}})$ and $\delta \boldsymbol{x} \neq \mathbf{0}$ after the first measurement correction, we will use the second measurement to find an additional correction to the error estimate. Since the position error state is additive, nothing changes with that state, and we focus on the attitude component. The additional error correction is combined with the previous estimated error using

$$\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}} = \frac{\boldsymbol{\epsilon}^{\bar{b}} + \delta\hat{\boldsymbol{a}} - \frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}})\boldsymbol{\epsilon}^{\bar{b}}}{1 - \frac{1}{4}(\boldsymbol{\epsilon}^{\bar{b}})^{\top}\delta\hat{\boldsymbol{a}}}, \tag{3.77}$$

where both parameters are considered to be parametrized as two times the Gibbs vector. This combination formula can be found by using the quaternion product,

$$\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}} = \delta\boldsymbol{a}(\boldsymbol{q}(\boldsymbol{\epsilon}^{\bar{b}}) \otimes \boldsymbol{q}(\delta\hat{\boldsymbol{a}})), \tag{3.78}$$

using

$$\delta\boldsymbol{a}(\boldsymbol{q}) = 2\frac{\boldsymbol{q}_v}{q_w} \tag{3.79}$$

and

$$\boldsymbol{q}(\delta\boldsymbol{a}) = \frac{1}{4 + \delta\boldsymbol{a}^{\top}\delta\boldsymbol{a}}\begin{bmatrix} 2 \\ \delta\boldsymbol{a} \end{bmatrix}. \tag{3.80}$$

We can then expand (3.72) to

$$\boldsymbol{p}^n_m - \hat{\boldsymbol{p}}^n_m = \boldsymbol{p}^n - \hat{\boldsymbol{p}}^n + (\boldsymbol{R}(\boldsymbol{q}^n_{\bar{b}})\boldsymbol{R}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}}) - \boldsymbol{R}(\boldsymbol{q}^n_{\bar{b}})\boldsymbol{R}(\delta\hat{\boldsymbol{a}}))\boldsymbol{l}^b. \tag{3.81}$$

Using the rotation matrix (3.62) we can write this as

$$\begin{aligned}\boldsymbol{p}^n_m - \hat{\boldsymbol{p}}^n_m &= \boldsymbol{p}^n - \hat{\boldsymbol{p}}^n \\ &+ \boldsymbol{R}(\boldsymbol{q}^n_{\bar{b}})\left(\frac{\frac{1}{2}\boldsymbol{S}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})^2 + \boldsymbol{S}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})}{1 + \frac{1}{4}\|\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}}\|^2} - \frac{\frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}})^2 + \boldsymbol{S}(\delta\hat{\boldsymbol{a}})}{1 + \frac{1}{4}\|\delta\hat{\boldsymbol{a}}\|^2}\right)\boldsymbol{l}^b.\end{aligned} \tag{3.82}$$

This results in the measurement matrix attitude component

$$\begin{aligned}\boldsymbol{H}_{\delta\hat{\boldsymbol{a}}} &= \frac{\partial}{\partial\boldsymbol{\epsilon}^{\bar{b}}}(\boldsymbol{p}^n_m - \hat{\boldsymbol{p}}^n_m)\bigg|_{\boldsymbol{\epsilon}^{\bar{b}}=\mathbf{0}} = \\ &\frac{\partial}{\partial\boldsymbol{\epsilon}^{\bar{b}}}\boldsymbol{R}(\boldsymbol{q}^n_{\bar{b}})\left(\frac{\frac{1}{2}\boldsymbol{S}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})^2 + \boldsymbol{S}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})}{1 + \frac{1}{4}\|\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}}\|^2}\right)\boldsymbol{l}^b \\ &= \boldsymbol{R}(\boldsymbol{q}^n_{\bar{b}})\left[\frac{\partial}{\partial(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})}\left(\frac{\frac{1}{2}\boldsymbol{S}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})^2 + \boldsymbol{S}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})}{1 + \frac{1}{4}\|\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}}\|^2}\right)\boldsymbol{l}^b \frac{\partial}{\partial\delta\boldsymbol{a}}(\boldsymbol{\epsilon}^{\bar{b}} \oplus \delta\hat{\boldsymbol{a}})\right]_{\delta\boldsymbol{a}=\mathbf{0}} \\ &= \boldsymbol{R}(\boldsymbol{q}^n_{\bar{b}})\Bigg(\frac{-\boldsymbol{S}(\boldsymbol{l}^b) + \frac{1}{2}\boldsymbol{S}(\boldsymbol{l}^b)\boldsymbol{S}(\delta\hat{\boldsymbol{a}}) - \boldsymbol{S}(\delta\hat{\boldsymbol{a}})\boldsymbol{S}(\boldsymbol{l}^b)}{1 + \frac{1}{4}\delta\hat{\boldsymbol{a}}^{\top}\delta\hat{\boldsymbol{a}}} \\ &- \frac{\boldsymbol{S}(\delta\hat{\boldsymbol{a}})^2 - 2\boldsymbol{S}(\boldsymbol{l}^b)}{4(1 + \frac{1}{4}\delta\hat{\boldsymbol{a}}^{\top}\delta\hat{\boldsymbol{a}})^2}\boldsymbol{l}^b\delta\hat{\boldsymbol{a}}^{\top}\Bigg)\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}}) + \frac{1}{4}\delta\hat{\boldsymbol{a}}\delta\hat{\boldsymbol{a}}^{\top}\right).\end{aligned} \tag{3.83}$$

The factor $\boldsymbol{I} - \frac{1}{2}\boldsymbol{S}(\delta\hat{\boldsymbol{a}}) + \frac{1}{4}\delta\hat{\boldsymbol{a}}\delta\hat{\boldsymbol{a}}^\top$ is the transform $\boldsymbol{T}_{\boldsymbol{\epsilon}^{\bar{b}}}^{\delta\boldsymbol{a}_b^{\bar{b}}}$, transforming small offsets in rotation about the nominal body frame axes to small offsets in the attitude offset parameters around $\delta\hat{\boldsymbol{a}}$, and

$$\boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\left(\frac{-\boldsymbol{S}(\boldsymbol{l}^b) + \frac{1}{2}\boldsymbol{S}(\boldsymbol{l}^b)\boldsymbol{S}(\delta\hat{\boldsymbol{a}}) - \boldsymbol{S}(\delta\hat{\boldsymbol{a}})\boldsymbol{S}(\boldsymbol{l}^b)}{1 + \frac{1}{4}\hat{\boldsymbol{a}}^\top\delta\hat{\boldsymbol{a}}} - \frac{\boldsymbol{S}(\delta\hat{\boldsymbol{a}})^2 - 2\boldsymbol{S}(\boldsymbol{l}^b)}{4(1 + \frac{1}{4}\delta\hat{\boldsymbol{a}}^\top\delta\hat{\boldsymbol{a}})^2}\boldsymbol{l}^b\delta\hat{\boldsymbol{a}}^\top\right)$$

transforms small parameter offsets about $\delta\hat{\boldsymbol{a}}$ to small offsets in antenna position. The error estimate is then updated using

$$\delta\hat{\boldsymbol{x}} \leftarrow \boldsymbol{K}\Big(\boldsymbol{p}_m^n - \Big(\boldsymbol{p}_{\text{nom}}^n + \delta\hat{\boldsymbol{p}}^n + \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{R}(\delta\hat{\boldsymbol{a}})\boldsymbol{l}^b\Big)\Big) \oplus \delta\hat{\boldsymbol{x}} \tag{3.84}$$

or equivalent to first order,

$$\begin{aligned} \delta\hat{\boldsymbol{x}}^+ &\approx \delta\hat{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{T}_{\boldsymbol{\epsilon}^{\bar{b}}}^{\delta\boldsymbol{a}_b^{\bar{b}}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{K}\Big(\boldsymbol{p}_m^n - \hat{\boldsymbol{p}}_m^n\Big), \\ \hat{\boldsymbol{p}}_m^n &= \boldsymbol{p}_{\text{nom}}^n + \delta\hat{\boldsymbol{p}}^n + \boldsymbol{R}(\boldsymbol{q}_{\bar{b}}^n)\boldsymbol{R}(\delta\hat{\boldsymbol{a}})\boldsymbol{l}^b. \end{aligned} \tag{3.85}$$

Since this approach for the non-zero-error linearization operates using small errors and covariance in the nominal body frame, the covariance can be updated and is maintained in this frame. The reset and covariance transform to the new body frame is then performed at the end after processing all measurements. It should, however, still be clear that an error reset between measurements is simpler than this procedure and that if the linearization point does change significantly between measurements, an iterated measurement update using all measurements simultaneously can be a better option.

To visualize the covariance transform, we consider a numerical example of a single-step Kalman measurement update. With prior position and attitude estimates $\boldsymbol{p}^{n-} = \boldsymbol{0}$ and $\boldsymbol{q}_b^{n-} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^\top$ (correspoding to $\boldsymbol{R}_b^n = \boldsymbol{I}$), and covariance matrix

$$\boldsymbol{P}^- = \text{diag}([(0.2\text{rad})^2\ (0.07\text{rad})^2\ (0.4\text{rad})^2\ (0.01\text{m})^2\ (0.01\text{m})^2\ (0.01\text{m})^2]), \tag{3.86}$$

where diag($\cdot$) is a matrix with the given entries on the diagonal, we receive a position measurement with a known lever-arm $\boldsymbol{l}^b = \begin{bmatrix}0.8 & 0 & 0\end{bmatrix}^\top$ m, and covariance matrix

$$\boldsymbol{R}_0 = (0.05\text{m})^2\boldsymbol{I}_{3\times3}. \tag{3.87}$$

The measurement is $\boldsymbol{p}_m = \begin{bmatrix}0.725 & 0.338 & 0\end{bmatrix}^\top$ m (corresponding to a 25° rotation about the z-axis) and our prediction of the measurement is $\hat{\boldsymbol{p}}_m = \boldsymbol{l}^b = \begin{bmatrix}0.8 & 0 & 0\end{bmatrix}^\top$ m. Using (3.74) and (3.13), the calculated correction is

$$\begin{aligned} \delta\hat{\boldsymbol{x}} &= \boldsymbol{K}(\boldsymbol{p}_m - \hat{\boldsymbol{p}}_m) && (3.88) \\ &= \begin{bmatrix}0 & 0 & 0.4122\text{rad} & -0.0029\text{m} & 0.0003\text{m} & 0\end{bmatrix}^\top, && (3.89) \end{aligned}$$

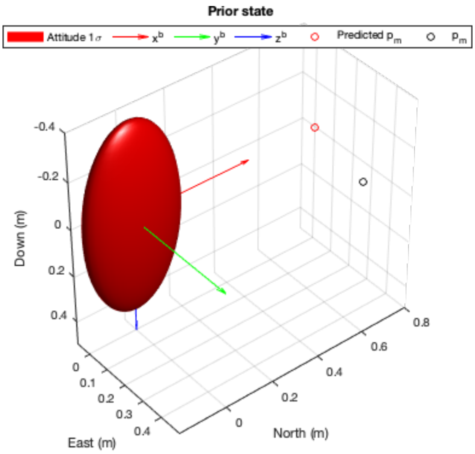


Figure 3.1: A priori attitude estimate and attitude covariance.

with the resulting updated quaternion

$$\boldsymbol{q}_b^{n+} = \begin{bmatrix} 0.97884 & 0 & 0 & 0.2046 \end{bmatrix}^\mathsf{T}. \tag{3.90}$$

Updating the covariance using (3.15) results in

$$\boldsymbol{P}^+ = \begin{bmatrix} 0.04 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0022 & 0 & 0 & 0 & 0.000068 \\ 0 & 0 & 0.00396 & 0 & -0.00012 & 0 \\ 0 & 0 & 0 & 0.000096 & 0 & 0 \\ 0 & 0 & -0.00012 & 0 & 0.0001 & 0 \\ 0 & 0.00007 & 0 & 0 & 0 & 0.0001 \end{bmatrix}. \tag{3.91}$$

The prior state and the 3 × 3 upper left attitude covariance submatrix from (3.86), along with the predicted and actual measurement are illustrated in Fig. 3.1. The attitude covariance matrix is visualized as a one standard deviation covariance ellipsoid, where the large dimension in the body frame z-axis indicates a large angular uncertainty about this axis, while the small y-axis dimension indicates a low pitch angle uncertainty. If the updated covariance matrix is interpreted as being given in the updated body frame, the result is as visualized in Fig. 3.2. Since the prediction of the measurement lies on the body x-axis, the measurement does not contribute to a reduced angular uncertainty about this axis, and the body x-axis dimension of the covariance ellipsoid therefore remains the same as in Fig. 3.1.

If a rotation transform (3.60) is used, the result is as illustrated in Fig. 3.3. In this case, the updated covariance matrix is interpreted as being given in the prior body frame, which in this case is equal to the navigation frame. This is the same result we would get if we had $\delta\hat{\boldsymbol{x}} = \boldsymbol{0}$ followed by a 25° rotation about the z-axis from error-free IMU measurements which is what one would expect from (3.89)–(3.91). On the other hand, Fig. 3.4 shows the result of using the transform (3.22), which is equivalent to rotation of half the attitude correction to the first order. The orientation uncertainty shown in Fig. 3.4 therefore appears to have an orientation in between that shown in Fig. 3.2 and Fig. 3.3.

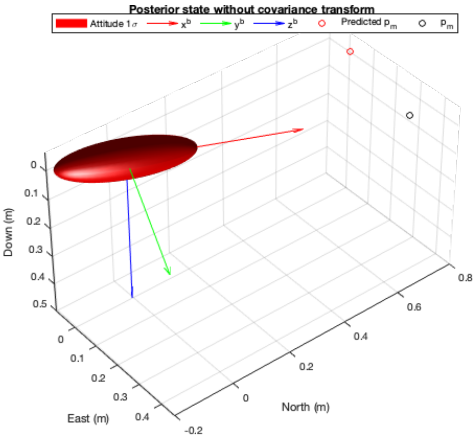


Figure 3.2: Posterior attitude and attitude covariance without covariance transform

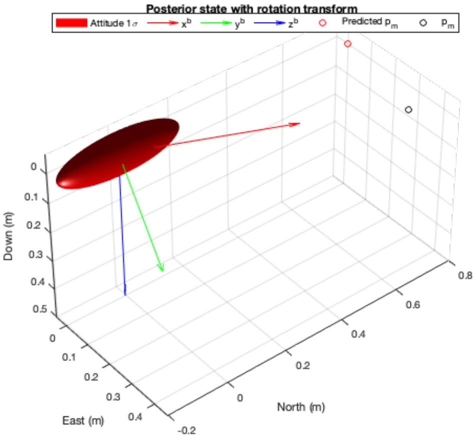


Figure 3.3: Posterior attitude estimate and attitude covariance using (3.60) to transform the posterior covariance.

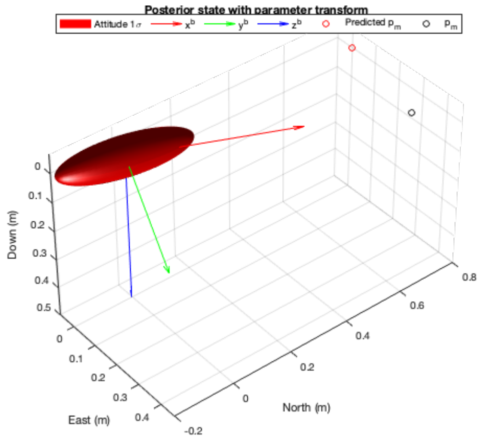


Figure 3.4: Estimated attitude and attitude covariance using (3.22) to transform the posterior covariance.

3.7 Conclusion

Based on the presented material, it is the authors' opinion that the covariance should be modified when resetting the error state, by performing

$$\boldsymbol{P}_{k,\text{reset}}^{+} = \boldsymbol{R}^{\top}(\delta\hat{\boldsymbol{a}}_{b}^{\bar{b}})\boldsymbol{P}_{k}^{+}\boldsymbol{R}(\delta\hat{\boldsymbol{a}}_{b}^{\bar{b}}) \tag{3.92}$$

to rotate the covariance from the prior predicted body frame to the updated posterior body frame. The rotation matrix is computed using 3.62 to 3.64, depending on the error parametrization assumed, or the first-order rotation matrix (3.65). For measurements performing only tiny corrections to the attitude estimate, this is, however, a minor modification unlikely to cause a significant difference in the estimation result.

Chapter 4

Outdoor Navigation Using Bluetooth Angle-of-Arrival Measurements

This chapter is based on the publication

- [132] M. L. Sollie, K. Gryte, T. H. Bryne, and T. A. Johansen, “Outdoor Navigation Using Bluetooth Angle-of-Arrival Measurements,” *IEEE Access*, vol. 10, pp. 88 012–88 033, 2022. doi: 10.1109/ACCESS.2022.3199772

4.1 Introduction

In 2019, the Bluetooth Special Interest Group (SIG) presented the Bluetooth 5.1 specification, which introduced support for direction finding using antenna arrays. This can provide estimates of the direction from the array to a moving tag in the antenna coordinate frame, but does not at the present time provide ranging. While the suggested applications typically focus on indoor tracking of devices, it can also be used outdoors where signal conditions are typically better, with fewer reflections off walls and other objects. Global Navigation Satellite System (GNSS) receivers are the most common position source used for outdoor navigation, but the very weak signals are susceptible to interference such as jamming and ionospheric scintillation. As a result, having access to supplements or alternatives to GNSS is beneficial to ensure navigation robustness.

In 2019, [24] implemented the Bluetooth 5.1 direction finding using Software Defined Radios (SDRs) with a two-antenna array, since commercial off-the-shelf (COTS) equipment was not yet available, testing both outdoors and indoors. It was found that if the signal propagation direction is close to the linear array direction, phase delays appeared almost random. Different frequency channels performed differently, and averaging between them was found beneficial. We note that each packet is sent on a single channel, and that to average over several channels many packets must be used, which is a disadvantage since measurements on different channels will not be simultaneous.

In [115], COTS direction finding equipment from Texas Instruments was tested indoors and outdoors at short range, finding good performance in general, but significantly better outdoors. Two separate receiver arrays were used for 2D position estimation. Indoor experiments have also been conducted by others [121, 170, 145].

It should be noted that all previous work using Bluetooth 5.1 direction finding we have found focused on short-range indoor positioning applications. Outdoor testing was only performed for comparison with an environment offering cleaner signal propagation conditions, mainly with less multipath impact. This is very different from the outdoor applications we consider, demonstrating new capabilities for this low cost-equipment. Most previous experimental work has also focused on the estimation of a single azimuth angle for positioning in the horizontal plane using linear arrays or pairs of linear arrays such as the Texas Instruments BOOSTXL-AOA [10, 20, 170, 115, 109]. An exception is [121] where both a uniform rectangular array and an array of the same type as used in this paper were tested.

Outdoor navigation using phased array radio has been demonstrated in [44] using a system from Radionor Communications. This is a more expensive system primarily designed for long-range communication, with a range of tens of kilometers or more. The Radionor system estimates direction in order to perform active beamforming between the units, while Bluetooth direction finding only uses a single transceiver and hardware radio frequency (RF) signal switches on the array. Bluetooth is aimed at mass-market use, while the Radionor system uses licensed frequency bands. Similar to the Radionor system, Bluetooth can be used as a combined communication and navigation system. The Radionor system supports ranging, while this is not yet a part of the Bluetooth specification (see [16] for information about a planned ranging feature), although custom ranging solutions compatible with the standard exist, see e.g. [173].

Indoor short-range comparison of Ultra-wideband (UWB) and Bluetooth arrays for direction finding was performed in [20], finding that UWB in general had better performance, especially for complex environments with multipath and signal obstruction. The test did however use very small arrays with 2 or 3 elements, finding that 3 elements performed significantly better than 2 elements, suggesting that larger arrays could improve performance further. UWB uses a high sampling rate and one receiver per antenna, all with a common clock, and is therefore more expensive. The cost of additional array elements with Bluetooth is low, but as the number of elements increases the maximum number of measurements from each element is reduced due to the maximum length of the received signal. For UWB it is also common to position using trilateration with range measurements to multiple separate fixed anchors [22, 93], but this has the disadvantage of a requiring a more elaborate equipment setup than one or two arrays.

Bluetooth is typically considered a short-range system for use under 100 m distance, but a range of hundreds of meters is possible outdoors, and even kilometer level within the Bluetooth specification limits of transmit power, depending on the directionality of the antennas used. Bluetooth supports a mode offering increased range at the expense of reduced data rate, known as "LE Coded PHY", but this is not compatible with direction finding [15, p. 292]. The Bluetooth standard does not specify the direction estimation method, and many algorithms can be used for measurement processing. Examples of outdoor applications where Bluetooth direction finding could be used, if range and direction estimation performance is sufficient, are automatic docking of boats and ferries, robotic lawn mowers, and precision landing of unmanned aerial vehicles (UAVs).

The main contribution of this paper is the demonstration of long range use of Bluetooth direction finding outdoors in field experiments. Methods for

- high precision carrier frequency offset estimation utilizing all available measurements
- efficient high resolution spatial pseudo-spectrum peak search using optimization

are proposed, resulting in improved estimation accuracy and precision of the direction of the received signal, and reduced computational load and latency. While the optimization-based peak-search has not been used in the field experiments, it has been used in offline estimation for simulated multipath measurements, where the direction has been estimated for a large number of measurement sets with very high resolution, showing its usefulness.

The paper proceeds as follows: Section 4.2 explains the mathematical notation used throughout the paper. Section 4.3 introduces the Bluetooth direction finding feature, explains the measurements sampled by the receiver and how these can be used for Angle-of-Arrival estimation. Methods for high precision frequency deviation estimation and efficient high resolution pseudo-spectrum peak search are proposed. In Section 4.4, field experiments using a multirotor UAV and a fixed-wing UAV are conducted, demonstrating direction estimation out to 700 m range. Section 4.5 investigates the effect of ground reflection multipath interference on the direction estimate for vehicle navigation, and how different parameters affect the estimation error. Section 4.6 concludes the paper with suggestions for future work.

4.2 Preliminaries

4.2.1 Notation

A vector decomposed in coordinate frame $\{a\}$ uses lowercase bold letters, e.g. $\boldsymbol{v}^a$, while matrices use uppercase bold, $\boldsymbol{V}$. A matrix of vectors $\boldsymbol{v}_k^a$ is written as $\boldsymbol{V}^a =$

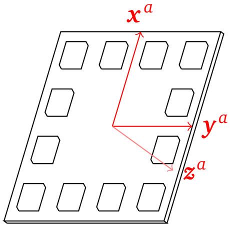


Figure 4.1: Array coordinate frame $\{a\}$, with array seen from the front. $-\boldsymbol{z}^a$ is the boresight direction, the direction where the antenna elements have the maximum gain.

$\left[\boldsymbol{v}_1^a, \ldots, \boldsymbol{v}_m^a\right]$ for m elements. A rotation matrix $\boldsymbol{R}_a^b \in SO(3)$ transforms vectors between frames $\{a\}$ and $\{b\}$, $\boldsymbol{v}^b = \boldsymbol{R}_a^b \boldsymbol{v}^a$. The Euclidean norm is denoted $\|\cdot\|_2$ with $\cdot$ being the variable placeholder. The magnitude of a complex number is denoted $|\cdot|$, the matrix transpose $(\cdot)^\top$, and the complex conjugate transpose $(\cdot)^H$. The imaginary unit is denoted j and the set of real and complex numbers $\mathbb{R}$ and $\mathbb{C}$, respectively. $\boldsymbol{v} = \boldsymbol{v}_1 \odot \boldsymbol{v}_2$ is the element-wise product of the vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

4.2.2 Coordinates

The antenna array coordinate frame $\{a\}$ is illustrated in Fig. 4.1. The navigation coordinate frame $\{n\}$ has its origin coincident with $\{a\}$, but with axes pointing towards North-East-Down (NED). If the array is placed flat on the ground with $\boldsymbol{x}^a$ towards North, the frames are identical. Directions in the antenna frame are parameterized using the polar angle α and the azimuthal angle Ψ. α is the angle of incidence, which is 0 in the boresight direction and $\frac{\pi}{2}$ for a direction in the array xy-plane. Ψ is measured in the antenna xy-plane about $\boldsymbol{z}^a$ using the right-hand rule, with $\Psi = 0$ for the direction $\boldsymbol{x}^a$. In $\{n\}$ we have the azimuth angle Ψ_n measured relative to North and elevation angle α_n measured from the horizontal tangent plane.

4.3 Bluetooth direction finding

Bluetooth is a technical standard for short-range wireless communication using the 2.4-2.5 GHz industrial, scientific and medical (ISM) band. Bluetooth Low Energy (BLE), which is a subset of the Bluetooth standard intended for low power consumption, uses a total of 40 channels, from 2402 MHz to 2480 MHz, with 2 MHz channel spacing. The three channels at 2402 MHz, 2426 MHz, and 2480 MHz are used for advertising, while the rest are used for data transmission. Gaussian frequency-shift keying (GFSK) is used for modulation [16], which means that the frequency deviates with a positive offset for a binary 1, and a negative offset for a binary 0, but that a Gaussian filter is used on the baseband data before modulation to smooth the transitions and reduce sideband levels. For BLE using 1 Mb/s the nominal deviation is 250 kHz, and the average deviation should be between 225kHz and 275kHz, with 99.9% being above 185kHz [17].

The Bluetooth 5.1 Core specification introduced direction finding capability using antenna arrays. This is done by appending a Constant Tone Extension (CTE) to the Bluetooth packet transmitted, which is essentially a stream of binary ones, resulting in a sine wave at a fixed frequency at the end of the message. Receivers can measure the phase of this signal and perform carrier phase interferometry calculations. While antenna arrays are used either on the receive or transmit side, active beamforming is not used, and only a single antenna of the array is used at once. This allows low-cost hardware, using single-channel transceivers combined with electronically controlled RF switches. The sequential use of each antenna in the array does however mean that processing software is more complex compared to systems where all measurements are simultaneous.

Direction finding can use either Angle-of-Arrival, AoA, where the moving tag transmits the CTE and the array receiver measures phase, or Angle-of-Departure, AoD, where CTE transmission is switched between all array antennas, and the moving tag measures phase [162]. These two methods each have benefits and drawbacks. A disadvantage of AoD is that antenna switching occurs at the transmitting side, where significant transmit power must be switched. If multiple moving tags should be located, AoA requires all tracked devices to transmit data to the receiving array, and they should not transmit at the same time. Randomization of the transmit time can make the probability of traffic collision less likely, but with more devices, it can still occur. AoD only requires that the array transmits and all devices receive, allowing an unlimited amount of devices to estimate their direction from the array. If multiple arrays are to be used to track a single moving tag, however, AoA only requires the tag to transmit, while all arrays sample simultaneously. For AoD all arrays would need to transmit separately, meaning that direction estimates for each array would not be simultaneous. In this paper, only AoA is used, as it is the only feature supported by the equipment setup we are using for experiments. In the following, the terms *moving tag* and *transmitter* are therefore interchangeable.

4.3.1 CTE transmission and measurement sampling

The Bluetooth specification sets the maximum length of the CTE at 160 µs. The first 4 µs is the *guard* period, where no sampling is performed. This is followed by a 8 µs *reference* period, where one of the array elements is sampled repeatedly 8 times with 1 µs spacing. After the reference period, the remainder of the CTE is used to sample the array elements by using the signal switches to sequentially connect each element to the receiver. This period is divided into alternating switching slots and sampling slots of 1 µs or 2 µs duration [97]. Since a transient occurs when the receiver switches between antennas, the switching slots are intended to allow the

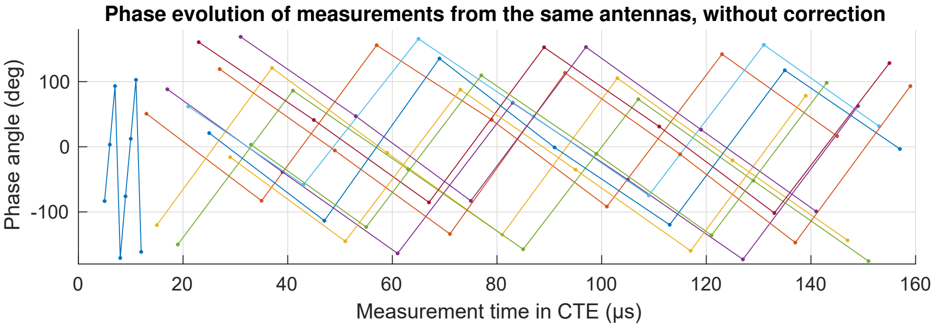


Figure 4.2: Phase angle of measurements from each antenna for a CTE, as output by the receiver. The 8 reference measurements are visible in the beginning. The connected dots of different colors are the measurements for each of the 11 remaining antennas in the array, which are sampled in cycles until the CTE ends.

signal to settle before sampling begins.

The transmitter broadcasts the CTE with a frequency $f_t = f_c + \Delta f_m + \Delta f_t$, where f_c is the channel center frequency. The frequency deviation Δf_m used for data modulation, nominally 250 kHz, is added since the CTE consists solely of digital ones. Δf_t is the error in the channel center frequency due to the inaccuracy of the transmitter oscillator. The Bluetooth modulation frequency is not required to be very accurate, or constant over time, but can reasonably be assumed stable for the short duration of a single CTE.

The receiver demodulates the signal using a local reference frequency set to the channel center frequency according to its own clock, $f_r = f_c + \Delta f_r$, where Δf_r is the receiver frequency error. For advertising channel 39 with a nominal frequency of 2480 MHz, where the CTE will have an intended frequency of 2480.25 MHz, the in-phase and quadrature (IQ) samples are generated using a local reference with a nominal frequency of 2480 MHz. This means that the IQ samples are not at baseband, but at an intermediate frequency offset from the baseband by approximately 250 kHz. This is known as the carrier frequency offset (CFO). The frequency deviation leads to a change in the phase relative to the receiver reference of approximately 90 degrees for each microsecond between samples. This is visible for the initial reference samples in Fig. 4.2.

Unlike receivers with a separate channel for each antenna in the array, allowing simultaneous sampling of all elements, BLE direction finding uses a single receiver connected to the array antennas using electronically controlled RF switches. The receiver samples all antennas sequentially by switching between them with known order and timing. Since measurements are not simultaneous and not at baseband, the frequency deviation must be estimated and taken into account in the direction estimation. Fig. 4.3 shows an example of samples from a CTE where the nominal 250 kHz has been used for compensation. There is still significant phase drift remaining,

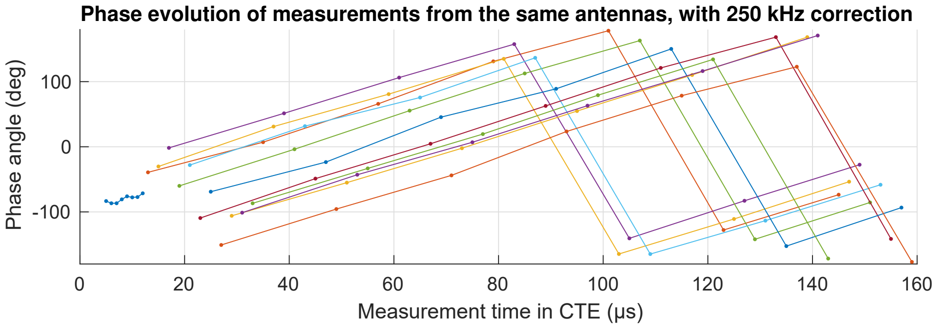


Figure 4.3: Phase angle of measurements, corrected for the nominal 250 kHz deviation. Some drift in phase is still present.

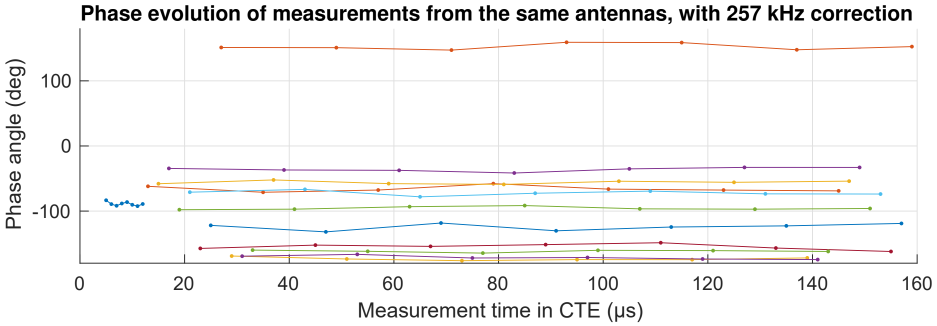


Figure 4.4: Phase angle of measurements, corrected for an estimated 257 kHz deviation. The systematic drift appears to be removed, and the corrected measurements can be used as if they were sampled simultaneously.

indicating that the actual deviation frequency is not 250 kHz. Compensating with an estimated 257 kHz gives the result in Fig. 4.4, where phases are close to constant. The carrier wavelength used can also be calculated from the estimated signal wavelength, but the error in this caused by a few tens of kilohertz error is very small. Because of the measurement noise present in the IQ samples, increasing the number of samples from a CTE used for frequency estimation can reduce the noise in the resulting frequency estimates, as will be explained in Section 4.3.2.3.

4.3.2 Direction estimation from CTE IQ samples

Using the Bluetooth IQ measurements, the direction to the transmitter is estimated using the following steps:

1. Estimate the CFO of the signal using the classical periodogram [105] generalized to non-uniform sampling, and in our case, multiple antennas. This creates

a one-dimensional frequency spectrum. A frequency spectrum peak search is then used to determine the signal frequency.

2. Use the CFO estimate to correct the IQ measurements such that they can be treated as being simultaneous.
3. Create a spatial pseudo-spectrum using a conventional beamformer method [71, 152] with a steering vector based on the far-field assumption.
4. Find the direction estimate using a peak search of the spatial pseudo-spectrum.

4.3.2.1 Steering vector - far field measurement model

The problem of estimating the direction to a transmitter, based on measuring the phase shift between antennas in a receiver array, is typically based on the measurement model [71]

$$\boldsymbol{x}(t) = \boldsymbol{A}(\boldsymbol{\Psi}, \boldsymbol{\alpha})\boldsymbol{s}(t) + \boldsymbol{n}(t) \tag{4.1}$$

for a signal downconverted to baseband before sampling, where $\boldsymbol{x}(t) = \boldsymbol{i}(t) + j\boldsymbol{q}(t) \in \mathbb{C}^m$ is the measurement vector containing IQ samples $\boldsymbol{i}(t), \boldsymbol{q}(t) \in \mathbb{R}^m$. $\boldsymbol{s}(t) \in \mathbb{C}^l$ is the vector of l transmitted baseband signal waveforms, $\boldsymbol{n}(t) \in \mathbb{C}^m$ is additive noise and

$$\boldsymbol{A}(\boldsymbol{\Psi}, \boldsymbol{\alpha}) = \left[\boldsymbol{a}(\Psi_1, \alpha_1), \ldots, \boldsymbol{a}(\Psi_l, \alpha_l)\right] \in \mathbb{C}^{m \times l} \tag{4.2}$$

is the matrix of steering vectors for each signal, which uses the far-field assumption such that it depends only on the source direction parameterized by Ψ, α. This is a typical simplification that assumes that all array elements receive the incoming signal from the same direction as a plane wave with all elements sampled simultaneously. Defining the line-of-sight direction unit vector

$$\boldsymbol{l}^a(\Psi, \alpha) = \begin{bmatrix} \cos\Psi \sin\alpha \\ \sin\Psi \sin\alpha \\ -\cos\alpha \end{bmatrix}, \tag{4.3}$$

the steering vector, which is the predicted phase shifts for a given signal direction, takes the form

$$\boldsymbol{a}(\Psi, \alpha) = e^{\frac{2\pi j}{\lambda}\boldsymbol{P}^{a\top}\boldsymbol{l}^a(\Psi,\alpha)} \in \mathbb{C}^m, \tag{4.4}$$

where λ is the wavelength of the signal, and $\boldsymbol{P}_a$ is the matrix of array antenna positions

$$\boldsymbol{P}^a = \left[\boldsymbol{p}_1^a, \cdots, \boldsymbol{p}_m^a\right] \in \mathbb{R}^{3 \times m}. \tag{4.5}$$

In our case, only a single transmitter is involved, and the model reduces to

$$\boldsymbol{x}(t) = \boldsymbol{a}(\Psi, \alpha)s(t) + \boldsymbol{n}(t), \tag{4.6}$$

$s(t) \in \mathbb{C}$. This model is not directly applicable for Bluetooth direction finding, because the signal sampled is not downconverted to baseband and the measurements are not simultaneous. This can be accounted for either by performing a measurement time correction on the data, as will be explained in Section 4.3.2.3, or by including an equivalent but opposite correction factor in the steering vector.

For an antenna array where elements are evenly spaced on a grid, calculating (4.4) with the full position matrix $\boldsymbol{P}^a$ is not necessarily the most efficient, as the sine/-cosine used in the calculation of the exponential may be slow depending on the computational platform architecture, and this is calculated for many direction pairs Ψ, α. An alternative method we propose is to calculate the phase step along each of the grid unit vectors, and then use complex multiplication to calculate the steering vector values. For the array in Fig. 4.1, we can calculate the phase difference between adjacent antennas with position differences $\boldsymbol{p}_x^a$ and $\boldsymbol{p}_y^a$ along the directions $\boldsymbol{x}^a$ and $\boldsymbol{y}^a$, respectively, as

$$a_x(\Psi,\alpha) = e^{\frac{2\pi j}{\lambda}(\boldsymbol{p}_x^a)^\top \boldsymbol{l}^a(\Psi,\alpha)}, \tag{4.7}$$

$$a_y(\Psi,\alpha) = e^{\frac{2\pi j}{\lambda}(\boldsymbol{p}_y^a)^\top \boldsymbol{l}^a(\Psi,\alpha)}. \tag{4.8}$$

Complex products can then be used to calculate the value for all elements

$$a(\Psi,\alpha)_{(n_x,n_y)} = a_x(\Psi,\alpha)^{n_x} a_y(\Psi,\alpha)^{n_y}, \tag{4.9}$$

where the index pair (n_x, n_y) refers to the grid position of the antenna element along the unit vectors. To limit the number of operations, avoiding repeated multiplications, this can be performed stepwise as

$$a(\Psi,\alpha)_{(0,0)} = 1 \tag{4.10}$$

$$a(\Psi,\alpha)_{(1,0)} = a_x(\Psi,\alpha) \tag{4.11}$$

$$a(\Psi,\alpha)_{(2,0)} = a(\Psi,\alpha)_{(1,0)} a_x(\Psi,\alpha) \tag{4.12}$$

and so on for each direction, and combined as e.g.

$$a(\Psi,\alpha)_{(2,2)} = a(\Psi,\alpha)_{(2,0)} a(\Psi,\alpha)_{(0,2)} \tag{4.13}$$

4.3.2.2 Beamformer pseudo-spectrum

The conventional beamformer [71, 152], also called the Bartlett beamformer, is a simple method for estimation of the direction from which a signal is received by an antenna array. This is essentially a spatial equivalent to the discrete Fourier transform, applicable to arbitrary array geometries. It is a spectral method where a spatial spectrum is to be searched to find the direction where the steering vector correlates best with the measurements. While algorithms such as Multiple Signal Classification (MUSIC) [117] and Estimation of Signal Parameters via Rotational Invariance Technique (ESPIRIT) [112] offer benefits such as increased resolution and better handling of multiple signals, beamformers have low computational complexity, without the need to compute the received signal correlation matrix or performing an eigenvalue decomposition. In our case, we only have a single transmitter and use a single-board embedded computer for real-time processing, and therefore use the beamformer approach.

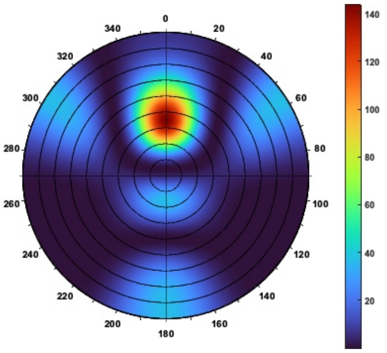


Figure 4.5: Beamformer spatial pseudo-spectrum. α is 0 at the center, and 90° at the edges, meaning that this covers the half-sphere in front of the array.

For the measurement vector $\boldsymbol{x}$ and steering vector (4.4), the beamformer pseudo-spectrum is

$$P(\Psi,\alpha) = |\boldsymbol{a}(\Psi,\alpha)^H \boldsymbol{x}|^2. \tag{4.14}$$

Fig. 4.5 shows an example of such spectrum, with dark red indicating the peak value. For Bluetooth CTE samples the measurement vector $\boldsymbol{x}$ typically contains multiple measurements from each array element. The steering vector must match the measurements, and if $\boldsymbol{x}$ contains more than one measurement from an element, the steering vector entry for that antenna must also be repeated. If the measurements have been corrected for CFO, such that they can be treated as being simultaneous, a general way to reduce the size of the steering and measurement vectors is to sum all measurements from the same antenna element, which correspond to the same complex steering vector value. The n_k measurements $x_{k,1},\ldots,x_{k,n_k}$ for antenna k is summed,

$$x_k = \sum_{i=1}^{n_k} x_{k,i}, \tag{4.15}$$

and $\boldsymbol{x} = [x_1,\ldots,x_m]^\top \in \mathrm{C}^m$ is used in (4.14). This yields the same estimation result as without the summation, as this only rearranges terms in the dot product of (4.14),

$$\begin{bmatrix} a_1(\Psi,\alpha) \\ a_2(\Psi,\alpha) \\ \vdots \\ a_m(\Psi,\alpha) \\ \vdots \\ a_1(\Psi,\alpha) \\ a_2(\Psi,\alpha) \\ \vdots \\ a_m(\Psi,\alpha) \end{bmatrix}^\top \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{m,1} \\ \vdots \\ x_{1,n_1} \\ x_{2,n_2} \\ \vdots \\ x_{m,n_m} \end{bmatrix} = \begin{bmatrix} a_1(\Psi,\alpha) \\ a_2(\Psi,\alpha) \\ \vdots \\ a_m(\Psi,\alpha) \end{bmatrix}^\top \begin{bmatrix} \sum_{i=1}^{n_1} x_{1,i} \\ \sum_{i=1}^{n_2} x_{2,i} \\ \vdots \\ \sum_{i=1}^{n_m} x_{m,i} \end{bmatrix}. \tag{4.16}$$

Antennas with a higher number of measurements, and with higher measurement magnitude, will have a greater impact on the pseudo-spectrum. After summation $\boldsymbol{a}(\Psi, \alpha)$ and $\boldsymbol{x}$ have only m elements without values for the same antennas repeated in $\boldsymbol{a}(\Psi, \alpha)$, making the time spent on the spectrum peak search independent of the number of measurements for each antenna.

To find the estimated direction in a common navigation frame independent of the orientation of the array, we want to find the direction parameters Ψ_n, α_n. After finding the antenna frame direction pair Ψ, α maximizing the pseudo-spectrum, the line-of-sight vector $\boldsymbol{l}^a(\Psi, \alpha)$ can be transformed to the local direction Ψ_n, α_n using

$$\boldsymbol{l}^n = \boldsymbol{R}_a^n \boldsymbol{l}^a(\Psi, \alpha), \tag{4.17}$$

$$\Psi_n = \text{atan2}(l_y^n, l_x^n), \tag{4.18}$$

$$\alpha_n = \tan^{-1}\left(\frac{-l_z^n}{\sqrt{(l_x^n)^2 + (l_y^n)^2}}\right), \tag{4.19}$$

where $\boldsymbol{R}_a^n$ depends on the array orientation. $\text{atan2}(y, x) = \text{Arg}(x + jy)$ is the two-argument four-quadrant arctangent function standard in many programming languages, which is equivalent to $\tan^{-1}(\frac{y}{x})$ for $x > 0$.

4.3.2.3 CFO estimation and measurement correction

To use (4.14) with the steering vector (4.4), we need to compensate for the CFO of the measurements. Since this can change over time it must be estimated, as shown in Section 4.3.1. The unknown CFO needed to convert the measurements to baseband, allowing them to be treated as simultaneous, can be estimated using a similar approach as for direction, using the classical periodogram [105] adapted to nonuniform sampling and multiple antennas. A one-dimensional spectrum search is used to find the CFO estimate, if it is assumed constant throughout a CTE. If it is desirable to not assume a constant CFO, a linear drift term can additionally be estimated, making the spectrum two-dimensional, but this is not considered in the following. The Bluetooth specification [15] allows a maximum channel center frequency deviation of ±150kHz, corresponding to a clock error range of approximately ±60ppm, which comes in addition to the modulation deviation range mentioned in Section 4.3. In the approach suggested here, each receiver antenna contributes individually to the frequency spectrum, and knowledge of the antenna geometry is not needed for the CFO estimation step. Any antenna element where multiple measurements are sampled during the CTE can be included, which helps to refine the frequency deviation estimate compared to only using the reference measurements. See Appendix A.3 for a method to predict the effect on the direction estimate caused by small errors in the CFO used for correction.

Using only the reference period is the most straightforward and often suggested method for estimating the CFO [168, 162]. The low number of samples available from the reference period motivates the additional use of samples from the sample slots to improve the estimate. In [168] it is proposed to return to the same antenna every other sample, and include these repeated measurements in the frequency estimation. If the array has more than two elements, allocating half of the possible samples to one element obviously has the drawback of reducing the number of measurements available for direction estimation from the remaining elements. An analysis of several other approaches is found in [23], including both estimation of a global CFO for a whole CTE and local estimation where the frequency is considered to vary throughout the CTE. All methods considered there have frequency estimation ranges smaller than what the data allows with 1 µs spacing between reference samples, and smaller than the maximum possible CFO.

The frequency "steering vector" depends on the frequency and the time at which the included measurements were taken relative to each other,

$$\boldsymbol{a}_k(f) = e^{2\pi j f \boldsymbol{t}_k}, \tag{4.20}$$

where $\boldsymbol{t}_k$ is a column vector of the relative measurement times for antenna k, e.g. relative to the first measurement. A spectrum that can be used to estimate the CFO is then

$$P_{\text{freq}}(f) = \sum_{k=1}^{m} |\boldsymbol{a}_k(f)^H \boldsymbol{x}_k|^2, \tag{4.21}$$

where $\boldsymbol{x}_k$ is the subset of the measurement vector $\boldsymbol{x}$ with measurements from antenna k. An example spectrum using all 82 samples (8 reference samples and 74 from sample slots for 160 µs CTE with 1 µs slots) compared to the same method used with reference samples alone is shown in Fig. 4.6. Doppler shift due to movement of the transmitter or receiver along the direction between the two during the 160 µs CTE is absorbed into the frequency estimate. After finding the frequency with the maximum value of $P_{\text{freq}}(f)$, this is used to correct either the IQ measurements themselves or the direction steering vector. The measurement correction is

$$\boldsymbol{x} = \boldsymbol{x}_{\text{raw}} \odot e^{-2\pi j \hat{f} \boldsymbol{t}}, \tag{4.22}$$

where $\hat{f}$ is the estimated frequency deviation. After the correction, $\boldsymbol{x} = \left[x_1, \ldots, x_m\right]$ can be used as if all measurements have the same time of validity. Example frequency estimates from a data set collected in the field, from extracting the peak frequencies in the spectrum for both cases shown in Fig. 4.6, is plotted in Fig. 4.7. The estimate using all measurements is also shown enlarged, and with filtering, in Fig. 4.8. The noise level is reduced considerably by the increased amount of measurements included.

The CFO estimate will be ambiguous with the frequency change required to change the phase of successive measurements by multiples of a full cycle. For a mix of different measurement intervals, sidelobe peaks will appear in the spectrum in addition to the ambiguity peaks, corresponding to each measurement interval. For reference measurements taken at the same antenna at 1µs intervals, this frequency ambiguity is $(1\mu s)^{-1} = 1$MHz. The ambiguities are not a problem for the CFO measurement correction as any of the equally strong peaks will produce the same corrected measurements.

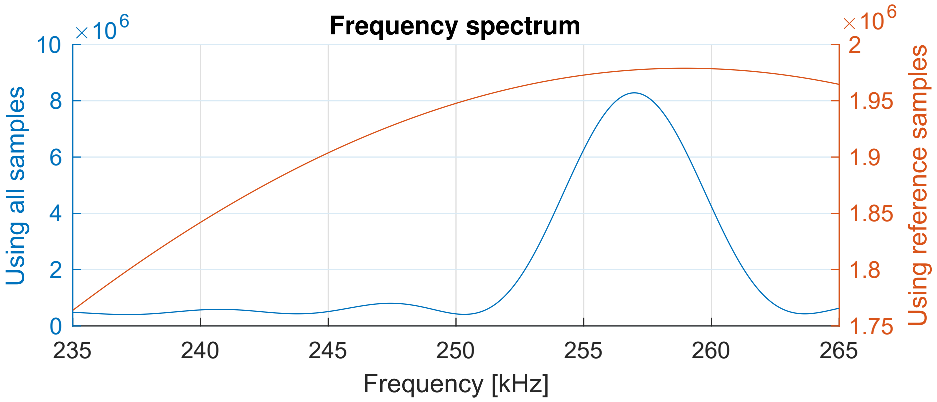


Figure 4.6: Example periodogram frequency spectrum for a single CTE, from field data. The blue spectrum, with its scale on the left, uses all available measurements. The red, with its scale on the right, only uses the reference samples. The measurements correlate best with the steering vector for the frequency with the maximum spectrum value. Note the very different axis values; the estimate using only reference samples is much flatter than the spectrum using all available measurements.

4.3.2.4 Pseudospectrum peak search strategy

Once (4.14) is found from the measurements, a typical approach for finding the best fitting direction is to calculate the value $P(\Psi, \alpha)$ on a grid of α from 0 to 90 degrees and Ψ from 0 to 360 degrees with a chosen resolution. The direction estimate is then found as the direction with the highest value. The downside of this approach, which is well-known [1], is that the grid resolution must be high to get a small discretization error in the direction estimate, while at the same time keeping the resolution low enough to meet limits on computational resources, especially for real-time applications such as vehicle navigation. Multi-step searches using e.g. a coarse search followed by a fine search in a limited area around the coarse peak can make this more efficient, but it is still limited by the resolution of the final search.

Under the assumption that we are only receiving a signal from a single transmitter, and that any multipath either results in a combined peak, or a separate peak that is lower than the direct signal peak, we only need to find the highest value of the spectrum. This can be done efficiently using the following two-step process:

1. Run a coarse search. Knowing the geometry of the array, the expected power of any multipath peak compared to the direct peak, and ideally the radiation pattern of the array elements, we can use the main lobe beamwidth to determine an appropriate search resolution that is as coarse as possible, while still ensuring that the highest spectrum value found is somewhere on the largest peak. The resolution does not have to be uniform, as the main lobe size can vary for different search directions due to array geometry projection and the

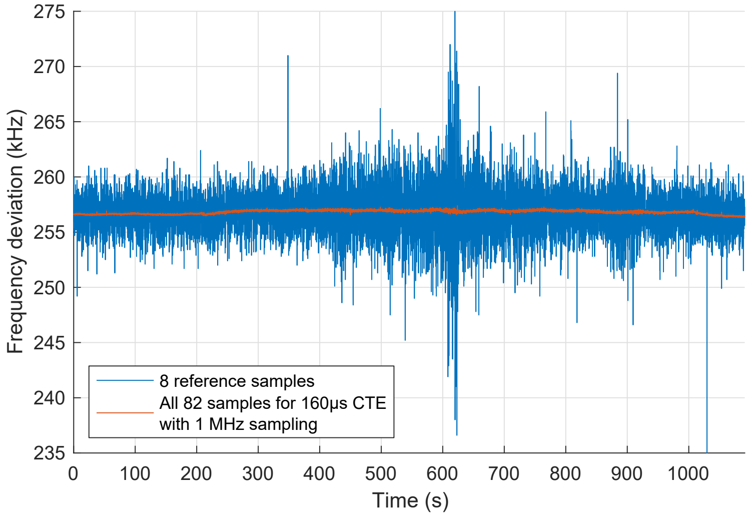


Figure 4.7: Comparison of frequency estimates using the 8 reference measurements and using all available measurements from a 160 µs CTE with 1 MHz sampling. The few remaining small spikes in the estimates can be removed by outlier rejection, and noise can be lowered by low pass filtering, which is done in Fig. 4.8.

radiation pattern. Any directions that are considered impossible can be excluded from the search. As an example, the array power pattern of a square array, as shown in Fig. 4.1, but using omnidirectional elements, for a signal source in the boresight direction, was calculated using Matlab and is shown in Fig. 4.9. The essential point here is that a smaller array, which has a lower angular resolution, can use a coarser search than a larger array.

2. With the coarse direction found as the initial value, use a nonlinear programming (NLP) [96] solver to maximize $P(\Psi, \alpha)$. The spectrum is locally very smooth, especially for small arrays with a low angular resolution, and convergence to the maximum does not need many iterations. Nelder-Mead [96] gradient-free search can also be used, but for beamforming methods or MUSIC the spectrum functions are differentiable, meaning that the gradient can easily be computed, which should be taken advantage of. See Appendix A.2 for the Matlab code used to implement this with CasADi [6].

As the direction towards the transmitter is unlikely to change significantly between measurements (at e.g. 10 Hz measurement rate), it is not always necessary to run the coarse search for every new measurement set. Once a new measurement set is available the NLP solver can be used immediately, using the previous direction estimate or a prediction also based on other sensors, such as an inertial measurement unit (IMU) and an inertial navigation system, as the initial guess. Such sensors and systems are standard in many robotic applications. See Appendix A.1 for a method for synchronization of measurements from Bluetooth and UAV-borne inertial sensors.

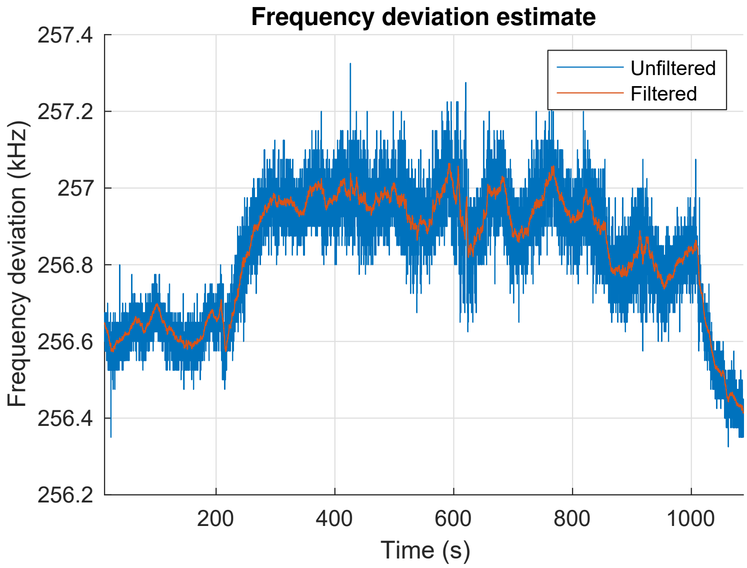


Figure 4.8: Example of frequency estimate from a multirotor flight. The motors started spinning on the ground at about 200s, which might have affected the temperature of the transmitter board hanging exposed on the UAV underside. The oscillations in the latter part of the plot is likely the result of a combination of Doppler shift and temperature variations due to airspeed, as the vehicle alternates between flying towards and away from the array.

If λ in the steering vector (4.4) is calculated based on the nominal carrier frequency, instead of the total frequency from the estimated CFO, the steering vector for all coarse search directions can be precomputed once and simply read from memory for each coarse search, which can speed up processing. The frequency deviation estimate is most important for measurement correction as explained in Section 4.3.2.3, while its use for wavelength calculation is of little importance as it only changes λ by single-digit parts per million.

If Ψ, α are used as the antenna-frame direction parameters, a uniform grid of these values will not be uniformly spread in direction, as illustrated in Fig. 4.10a. Near the array boresight direction, neighboring azimuth values will be closer together. The result is that the search is inefficient, as some points are unnecessarily close. The azimuthal resolution close to boresight can be reduced as shown in Figs. 4.10b and 4.10c, or a different parameterization can be used as in Fig. 4.10d, to reduce processing time used for the coarse search. These are however simple examples that do not take into account knowledge of a specific antenna array.

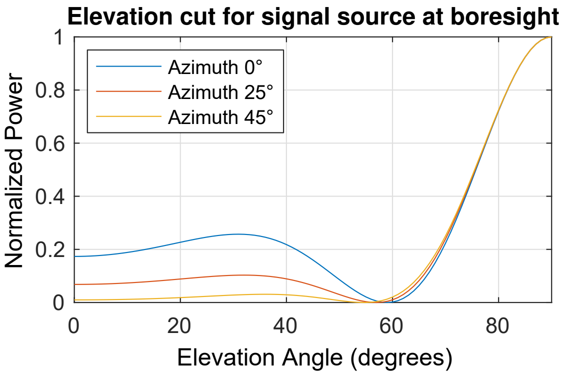
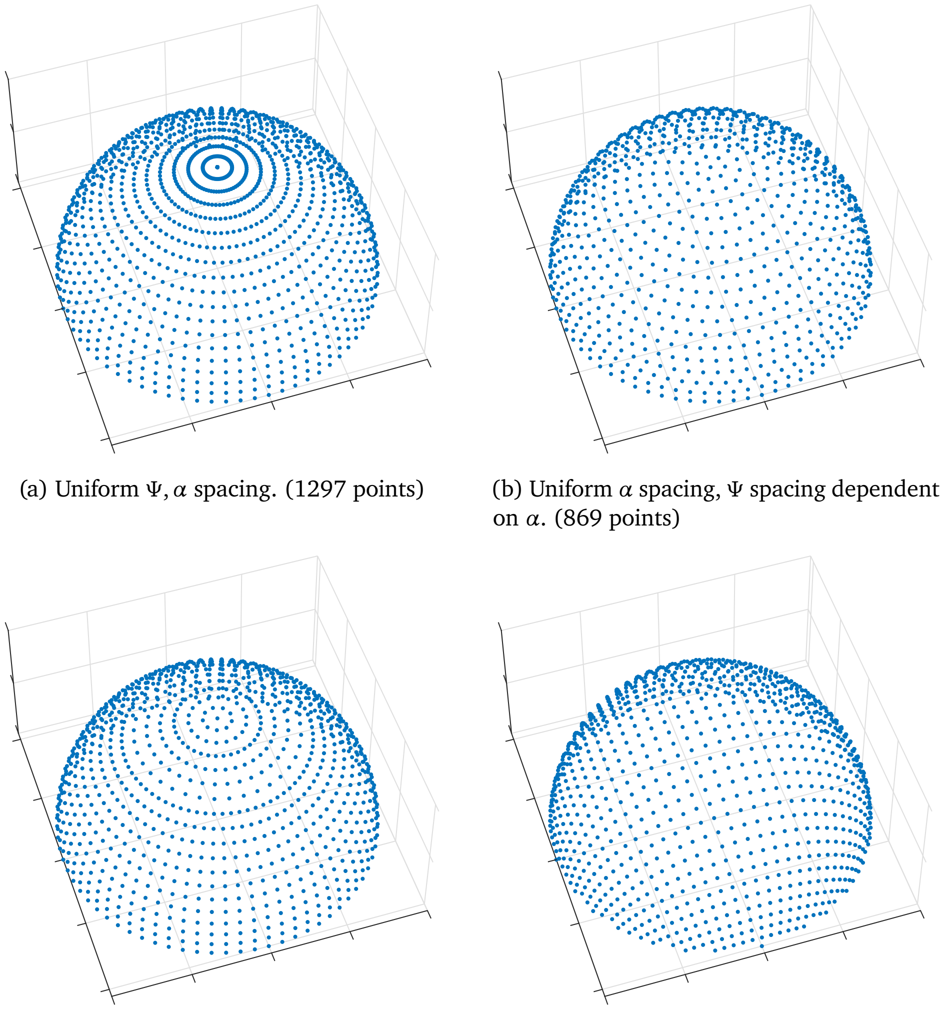


Figure 4.9: Matlab normalized power pattern for the array illustrated in Fig. 4.1, using omnidirectional elements for different azimuth angles with a signal source in the boresight direction. The boresight direction is at 90°.



(a) Uniform Ψ, α spacing. (1297 points)

(b) Uniform α spacing, Ψ spacing dependent on α. (869 points)

(c) Uniform α spacing, Ψ spacing dependent on α, but only by doubling the azimuthal step size, to maintain a clear grid pattern for constant values of Ψ. (1036 points)

(d) Cartesian grid on half sphere. (1009 points)

Figure 4.10: Search direction distribution for different point selection strategies. For the three first cases: 5° steps in elevation angle, and along in the array plane for azimuth. Azimuth angle steps can be increased closer to the boresight direction to maintain a more even direction coverage.

4.4 Field experiments

4.4.1 Equipment

4.4.1.1 Ground antenna equipment

Fig. 4.11 illustrates the hardware components connected to the antenna array. The array used is a Nordic Semiconductor experimental reference design, using 12 truncated corner right-handed circular polarization (RHCP) patch antennas in a square 15x15cm pattern, with 5cm antenna spacing. An nRF52833 board is directly attached to the back of the antenna using two header rows and a separate coaxial cable, where the header rows control the hardware signal switches in the array printed circuit board (PCB). The receiver board scans for advertising packets and outputs measurements only if the media access control address (MAC address) of the transmitter matches a pre-programmed value. Thus the system operates connectionless on the advertising channels with frequencies 2402 MHz, 2426 MHz and 2480 MHz. Due to the connectionless setup, Bluetooth was only used for direction estimation and not as a vehicle telemetry and command link, although this is a potential future extension. The board is configured to sample the CTE at a rate of 1 MHz, meaning once in every sample slot, for a total of 82 measurements.

The nRF52833 board is connected to a SentiBoard [4] universal asynchronous receiver-transmitter (UART) port. The SentiBoard forwards the data using Universal Serial Bus (USB) communication to a Beaglebone Black single-board computer, where the data is parsed in DUNE[73]. A custom binary protocol is used for the output of the measurements from the nRF52833. This includes phase measurements as in-phase and quadrature components, the frequency of the channel used, and metadata about measurement timing and sampling order. The Beaglebone Black is either connected using an ethernet cable to the vehicle ground station, where a Ubiquiti Rocket M5 radio allows communication with the vehicle, or directly to a radio that is wirelessly connected to the ground station for remote array placement. The receiver data are transmitted to the payload computer of the vehicle where the direction estimation runs. A two-step coarse-fine spectrum search is used in these experiments, with coarse steps of 2.5° and fine steps of 0.1°. The CFO is estimated using 25 Hz search steps. A uBlox ZED-F9P GNSS receiver with a helix antenna is used as a Real-Time-Kinematic (RTK) GNSS base, transmitting measurements using the RTCM3 protocol to the UAV over the network, in order to evaluate the positioning performance of the Bluetooth system. The components are mounted to a custom bracket, pictured in Fig. 4.12, with slots at different angles for the array PCB relative to the base plate. This allows changing the pitch angle of the array while the bubble level on the bracket is used to level the assembly.

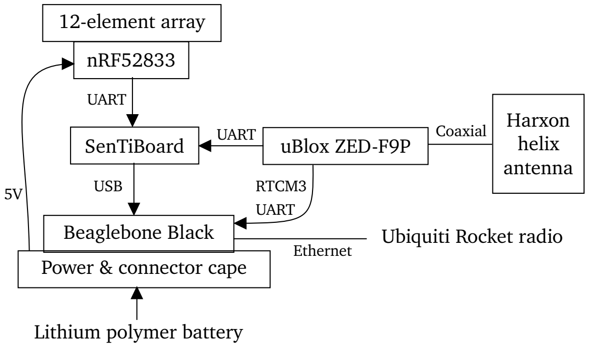


Figure 4.11: Ground hardware schematic

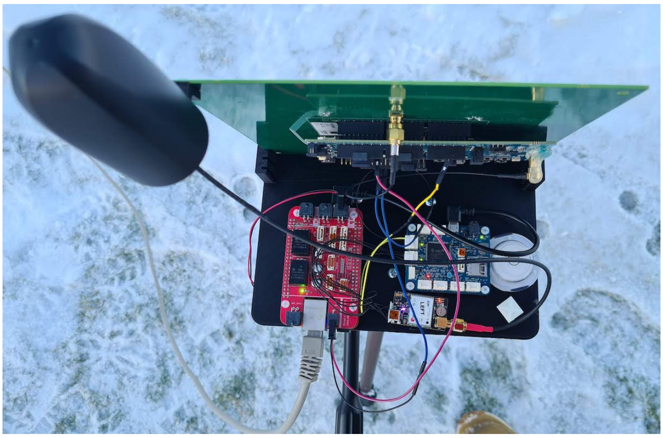


Figure 4.12: Ground hardware

4.4.1.2 Vehicle payload

Fig. 4.13 shows a schematic with the relevant hardware components onboard the vehicle. A directional TrueRC Canada X-AIR 2.4GHz RHCP antenna is used. Fig. 4.14 shows the antenna mounted in the nose of a Skywalker X8 UAV pointing forwards. The nose of the UAV, in front of the antenna, is made of expanded polystyrene with a layer of canvas tape for strength. Lab tests show no significant reduction in signal strength from this placement. The antenna is specified to have a gain of 8 dB, a −3 dB beamwidth of 75° and performance equal to an omnidirectional antenna in a 120° beam [146]. The antenna is connected to the nRF52833 board running transmitter firmware using a coaxial cable. The antenna connector on the board contains a signal switch that connects the transmitter to a linear polarization PCB trace antenna when an external antenna is not connected. A uBlox ZED-F9P GNSS receiver with a helix antenna is used on the UAV, receiving measurements from the antenna mounted on the array. The RTK GNSS setup yields very accurate and precise estimates of the UAV's position relative to the GNSS antenna attached to the

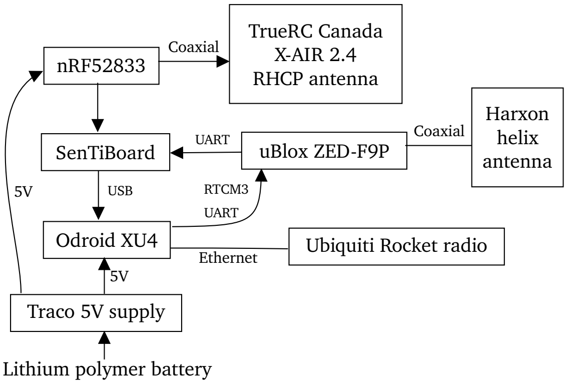


Figure 4.13: Payload hardware schematic

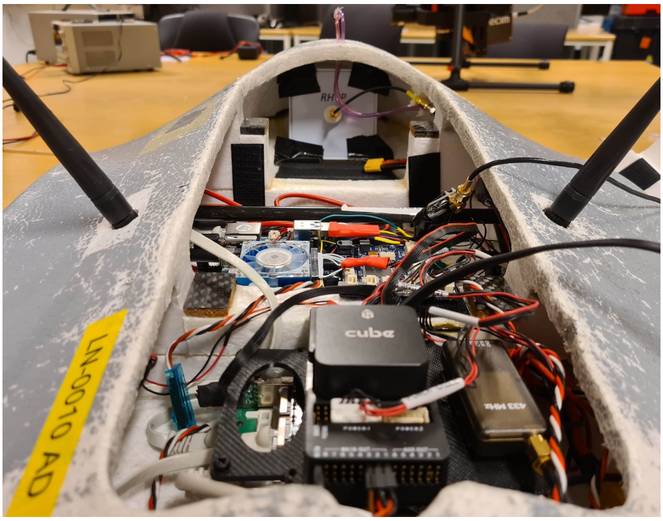


Figure 4.14: Vehicle payload: the RHCP antenna is visible in the front of the fuselage. The payload hardware is in the center and the Cube Black autopilot at the bottom.

array, with position errors on the centimeter level.

The transmitter broadcasts advertising packets with a 160 µs CTE at an average rate of 10 Hz (with a small random variation for conflict reduction with other Bluetooth devices). The SentiBoard outputs measurements from all connected sensors to the Odroid XU4 computer where they are both logged for later analysis and parsed for real-time use. The Odroid and SentiBoard are shown in Fig. 4.15, where they are mounted on top of the nRF52833.

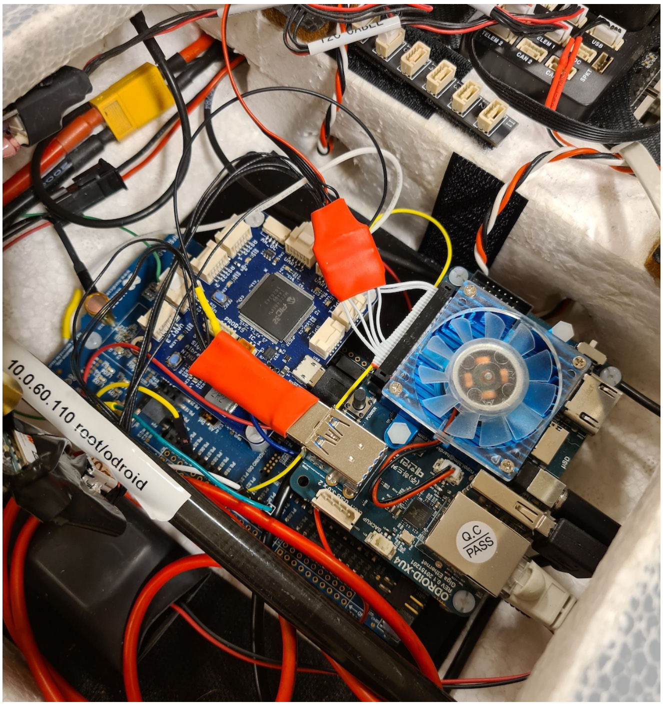


Figure 4.15: Vehicle payload: Odroid XU4 (right, with fan) and SentiBoard (left, blue) on top of nRF52833.

4.4.2 Practical sampling aspects: sample rate, switching intervals and CTE duration

The number of IQ measurements from a CTE is configurable in multiple ways. The CTE length can be configured, up to 160 µs, the duration of the switch and sample slots can be set to either 1 µs or 2 µs, and the sampling frequency can be set higher than 1 measurement in each slot. For the Nordic Semiconductor equipment used in the experiments in this paper, the maximum sample rate is 8 MHz, yielding 8 samples for each 1 µs sample slot, for a total maximum of 600 samples (or 1192 if sampling is also performed in the switch slots). Depending on the processing power available, the rate of CTE transmissions, and the algorithm used, this can be too much to process in real-time. Sampling can be done in both the switching and sampling periods and it can be configured in the external processing software whether any measurements in the last part of the switch slots are to be used for estimation if the switching transients settle early in the switch slots.

A limitation of the communication interface between the nRF52833 board and the computer used for processing is the maximum UART baudrate of 1 Mb/s. With a start bit and stop bit, the data capacity is 800 kbit/s or an average of 80 kbit for each sample set at 10 Hz. With 32 bits being used for an IQ sample pair, 1 MHz sampling results in 82 IQ pairs for a total of 2624 bits, and 8 MHz sampling equivalently yields 19200 bits, or 38144 bits if switch slot samples are output as well, which is close to half the link capacity. If additional metadata is included, such as timestamps within the CTE or the number identifying the array element used for each measurement, or if the measurement rate is increased beyond 10 Hz, the required data rate could reach the link limit. Since the sample order and sample timestamp sequence within the CTE is always the same, they can be hard-coded in the processing software instead of output on the link, to reduce link usage and latency, although any changes in the receiver configuration would require changes in the external processing software.

4.4.3 Experimental results

4.4.3.1 Multirotor

Initial testing was performed using a multirotor UAV, using the built-in linear polarization PCB trace antenna on the transmitter board, and the default 0 dBm transmit power. The array was placed flat on the ground pointing upwards, with axes aligned as best as possible to NED. The UAV was flown above the array at different heights up to 100 m, and at different horizontal distances. This is pictured in Fig. 4.16. Using height from RTK GNSS, Bluetooth direction measurements were transformed to North-East horizontal position for comparison with the RTK GNSS position estimates. Directly above the array, there was no significant loss of signal at 100 m height using the transmitter PCB trace antenna. The signal strength reduced with increasing horizontal distance, when the array received the signal at a more acute angle. Since the patch antenna elements in the array are somewhat directional, this is expected. The most interesting observation was that the direction estimation performance appeared significantly better for Bluetooth in the North direction as opposed to East, as clearly visible in Fig. 4.17 (the abbreviation BT is used for Bluetooth). This difference was independent of the UAV yaw angle, and when the array was yawed on the ground, the effect stayed the same in the antenna frame. Changing the array sampling order did not have any effect. Rotating the payload on the UAV by 90 degrees yielded the result in Fig. 4.18, which was much improved. Returning the payload to the original orientation again gave a different performance in the two directions. A possible explanation for this effect is that the transmitter antenna is linearly polarized and that the RC receiver onboard the UAV, which also uses 2.4GHz and sends telemetry data to the RC controller, in some way interferes with the Bluetooth system. For this reason, it was decided to try an external circularly polarized antenna, depicted in Fig. 4.19, which should be a better match for the circular polarization of the antenna elements on the array. This antenna is directional with a quite wide beam, and has a PCB ground plane that should shield the antenna from equipment above it on the UAV. Using the new transmitter antenna, Fig. 4.20 shows the result from a 20m diameter circle flown at 50m height, compared to RTK GNSS. The results for the three frequencies used are plotted separately in Fig. 4.21. Both directions appear similar in performance. It is visible that estimates from the three frequencies

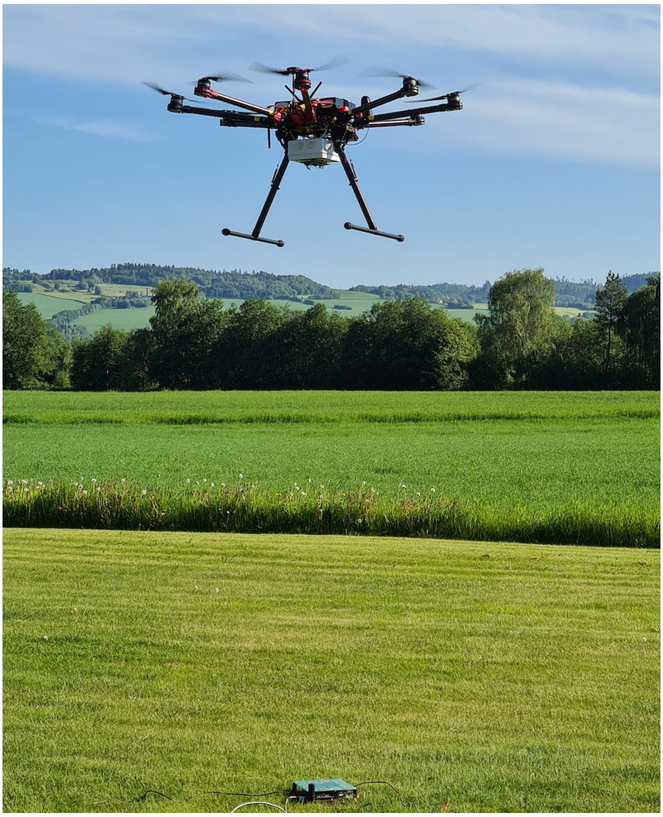


Figure 4.16: Multirotor UAV flying above the array, which is placed flat on the ground facing upwards. The PCB trace antenna on the transmitter board was used.

are slightly different, which could be related to a lack of frequency-dependent IQ origin calibration for the specific device. If the frequencies could be calibrated to remove the relative offsets, the performance looks very promising for multirotor precision landing without GNSS. Results for low elevation angles, $\alpha_n < 40°$, were much worse than for high elevation angles, and the direction search in the spatial pseudo-spectrum was therefore limited to angles within 50° of the array boresight, which was also carried over to the next fixed-wing test.

4.4.3.2 Fixed-wing

With the payload and RHCP antenna mounted in a Skywalker X8 fixed-wing UAV, tests were performed with the array boresight direction close to horizontal, with a 10° upwards pitch. Transmit power was increased to the maximum supported, 8 dBm, to provide the maximum range possible. Fig. 4.22 shows the array setup, which was leveled using a two-dimensional bubble level on the array mounting plate.

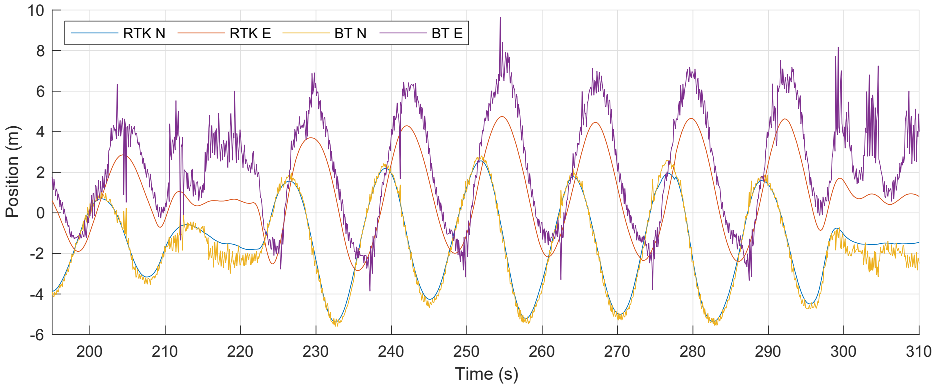


Figure 4.17: North-East position plots for RTK GNSS and Bluetooth (BT) for the multirotor at 15m height using the PCB trace antenna in the transmitter board. Heigth from RTK GNSS is used to transform from direction to horizontal position.

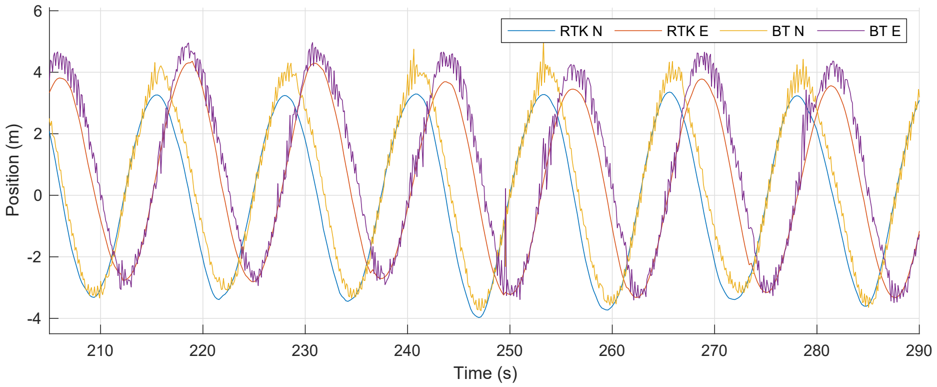


Figure 4.18: North-East position plots for the multirotor at 15m height using PCB trace antenna, after rotating the payload 90 degrees on the UAV. Height from RTK GNSS is used to transform from direction to horizontal position.

A flight was conducted where several maneuvers were tested, including straight lines towards the array at a constant height from multiple directions, crossing the boresight vector, changing both height and direction while approaching the array, and loiters at several distances, including one passing over the array. Figs. 4.23 and 4.24 show the RTK GNSS position of the UAV relative to the array, with color used to visualize the Bluetooth Received Signal Strength Indicator (RSSI) for all the Bluetooth measurements during this flight. The array was pointing South-West along the long straight line in Fig. 4.23, and Fig. 4.24 has a view from the side, perpendicular to the long straight line. The sensitivity of the receiver is −93 dBm according to the specification, and weaker signals will not be received. The maximum

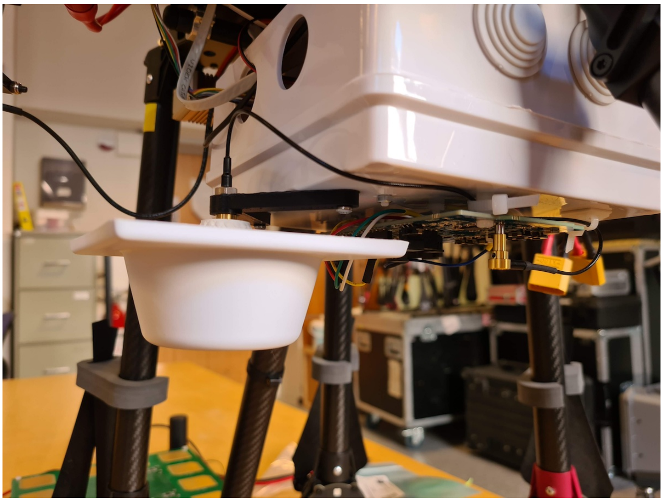


Figure 4.19: RHCP antenna mounted on the multirotor UAV, poining downwards.

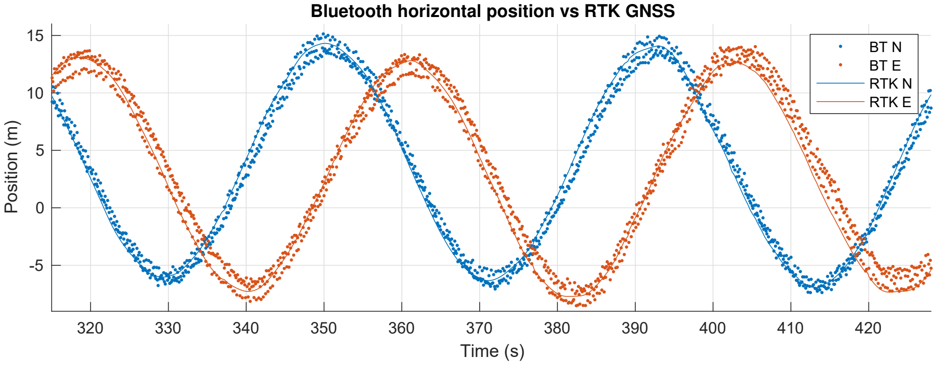


Figure 4.20: North-East position plots for the multirotor at 50m height, compared to RTK GNSS. Height from RTK GNSS is used to transform from direction to horizontal position.

range in this test was approximately 700 m, which should be more than enough for the system to be usable in a UAV recovery scenario. An interesting observation is that at close range several spatial bands appear (see Fig. 4.25).

Fig. 4.26 shows the azimuth and elevation angle estimates compared to RTK GNSS after calibrating the array heading by matching the azimuth angles from both systems. The azimuth estimate from Bluetooth is very consistent, with no large systematic errors, although the array orientation assumed when transforming GNSS positions into the array frame can be slightly misaligned. The azimuth plots are shown enlarged in Fig. 4.27. The azimuthal standard deviation for the near-constant azimuth intervals 393-424s, 500-527s and 602-634s are 1.08°, 1.00° and 0.92° respectively.

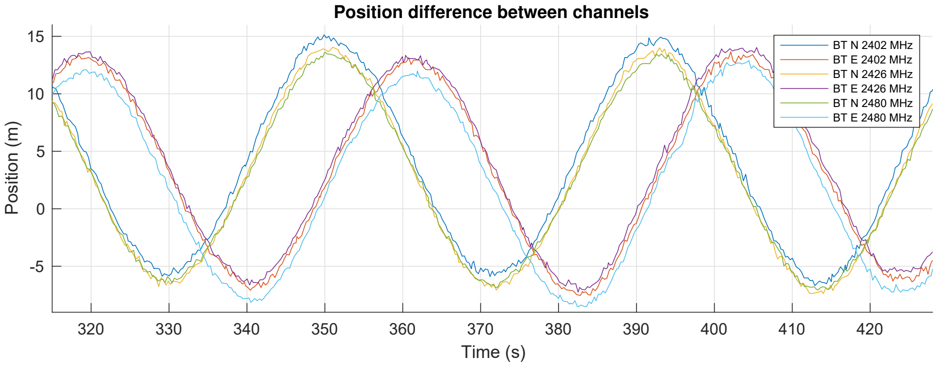


Figure 4.21: North-East position plots for the multirotor at 50m height showing each channel separately.

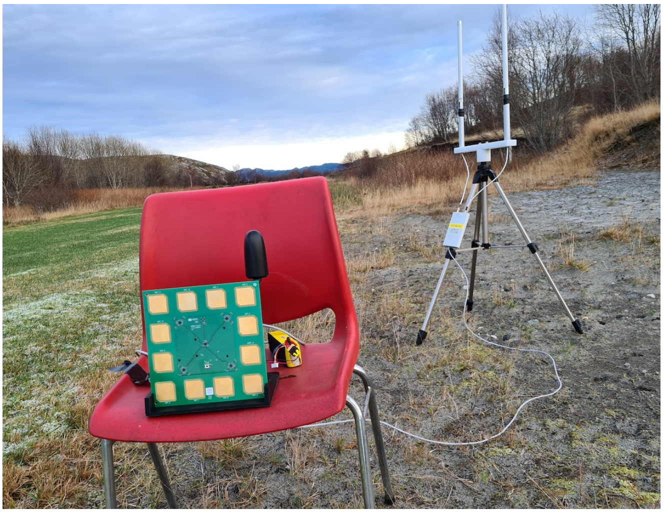


Figure 4.22: Array setup with Ubiquiti Rocket radio for remote equipment placement.

The elevation on the other hand has very clear systematic and repeatable errors at elevation angles up to approximately 25°, with larger errors for lower elevation angles. This was a trend for all maneuvers in this flight, with good azimuth performance and elevation angles struggling with systematic errors. The elevation angles are plotted separately for each channel in Fig. 4.28, showing systematic differences. To transform the Bluetooth angle estimates α, Ψ to horizontal and vertical antenna frame xy-positions, denoted $\boldsymbol{p}^a_{xy,\mathrm{BT}}$, the range along the array boresight vector from GNSS, $-p^a_{z,\mathrm{GNSS}}$, was extracted from the NED GNSS position transformed to the array frame $\boldsymbol{p}^a_{\mathrm{GNSS}} = \boldsymbol{R}^a_n \boldsymbol{p}^n_{\mathrm{GNSS}}$. Using

$$\boldsymbol{p}^a_{xy,\mathrm{BT}} = -p^a_{z,\mathrm{GNSS}} \tan(\alpha) \begin{bmatrix} \cos(\Psi) \\ \sin(\Psi) \end{bmatrix}, \tag{4.23}$$

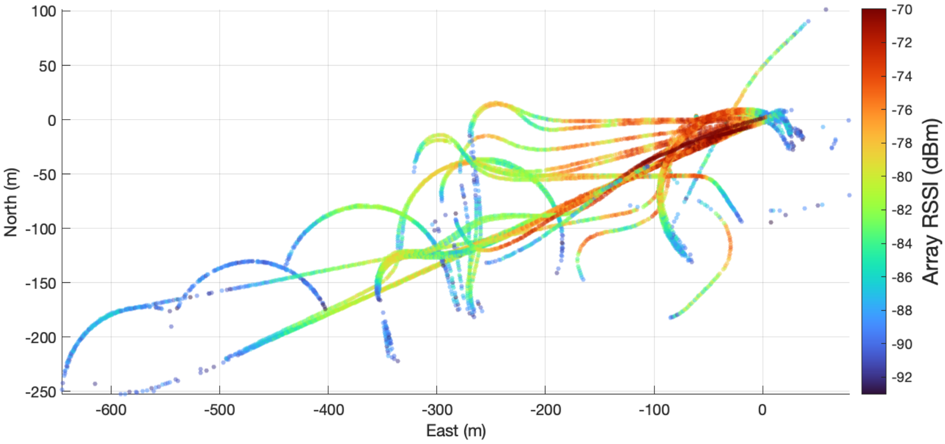


Figure 4.23: A top view of the X8 flight, with RSSI shown using color dots with position from GNSS.

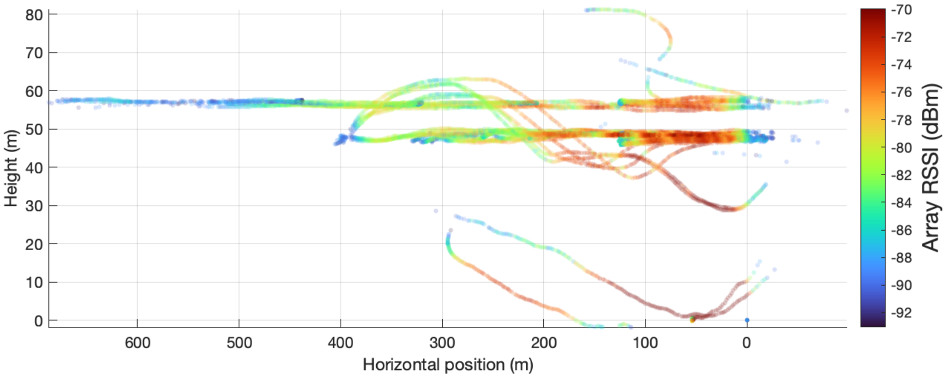


Figure 4.24: A view of the X8 flight from the side, perpendicular to the long straight line in Fig. 4.23. Note that the vertical and horizontal axes have different scales.

we can see the position error the angles corresponds to in the middle part of Fig. 4.26. Fig. 4.29 shows corresponding plots for the four loiters with increasing distance shown on the left side of Fig. 4.23.

An angular noise level will give rise to an increasing error in position as the range increases, but the angular noise level is decent at all distances where a signal was received. We expected direction estimation performance to become poor before losing the signal, so this exceeded our expectations.

The most likely source of the systematic errors in elevation is signal multipath. The signal received by the array is not only the signal propagating directly from the transmitter to the array but also a signal reflected from the flat ground surface, which causes signal interference and a signal that appears to originate from a different

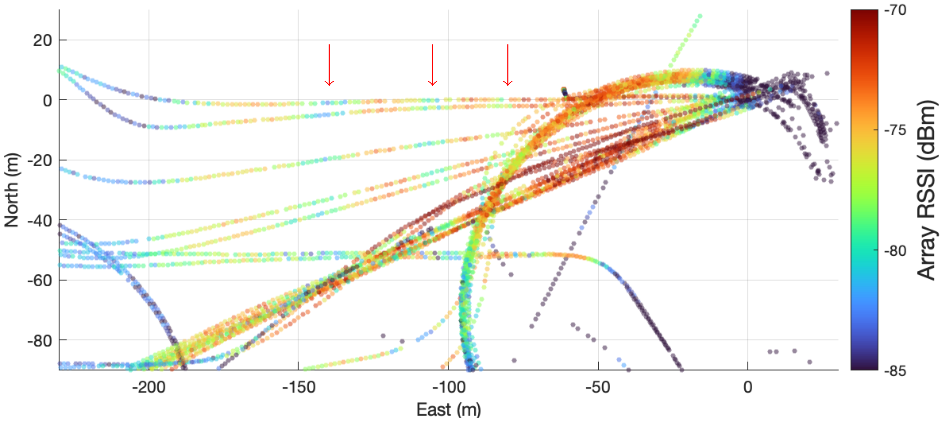


Figure 4.25: Visible spatial bands in the RSSI.

elevation angle.

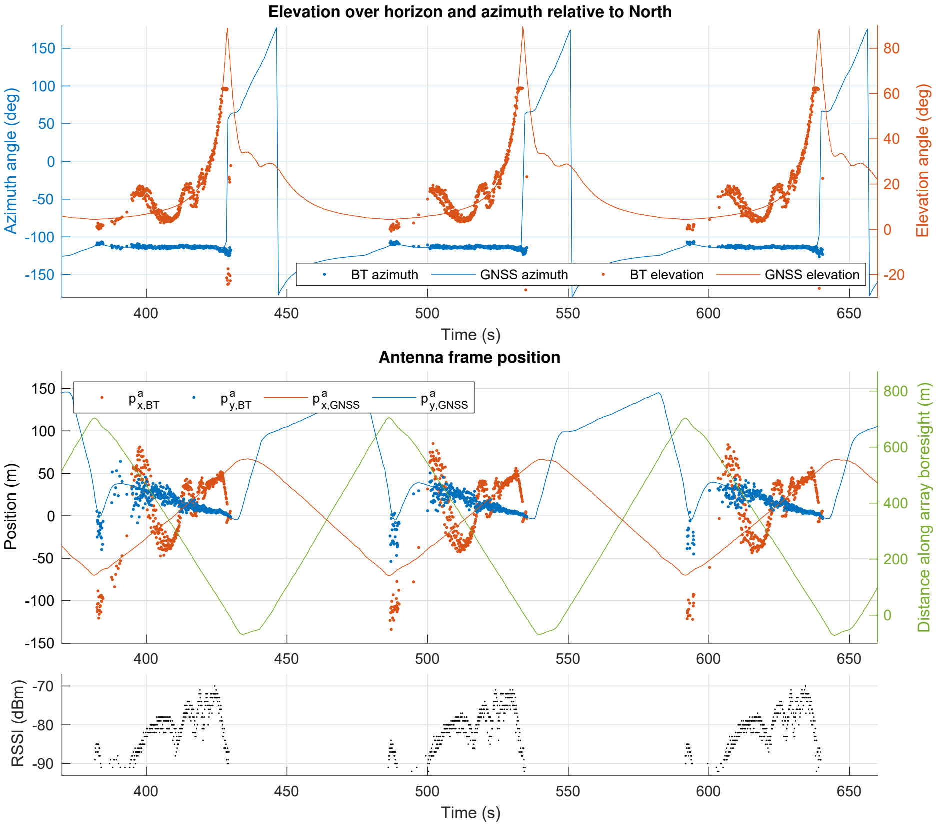


Figure 4.26: Estimation results for three straight flight lines towards and over the array. Note that due to a limitation $\alpha < 50°$ used in the direction search, elevation angles were limited to maximum 60° and directions are erroneous outside this range. RSSI values were output by the receiver as integer values.

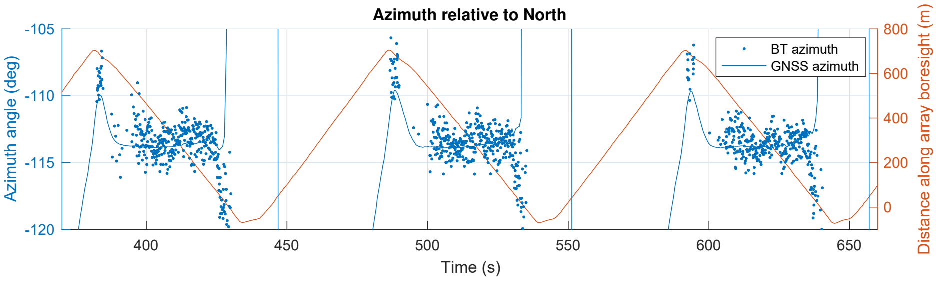


Figure 4.27: Enlarged plot of azimuth angle from Fig. 4.26.

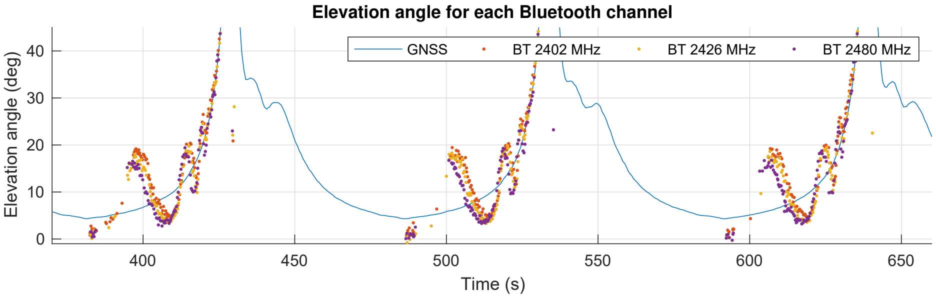


Figure 4.28: Difference in elevation angle from Fig. 4.26 for the three channels.

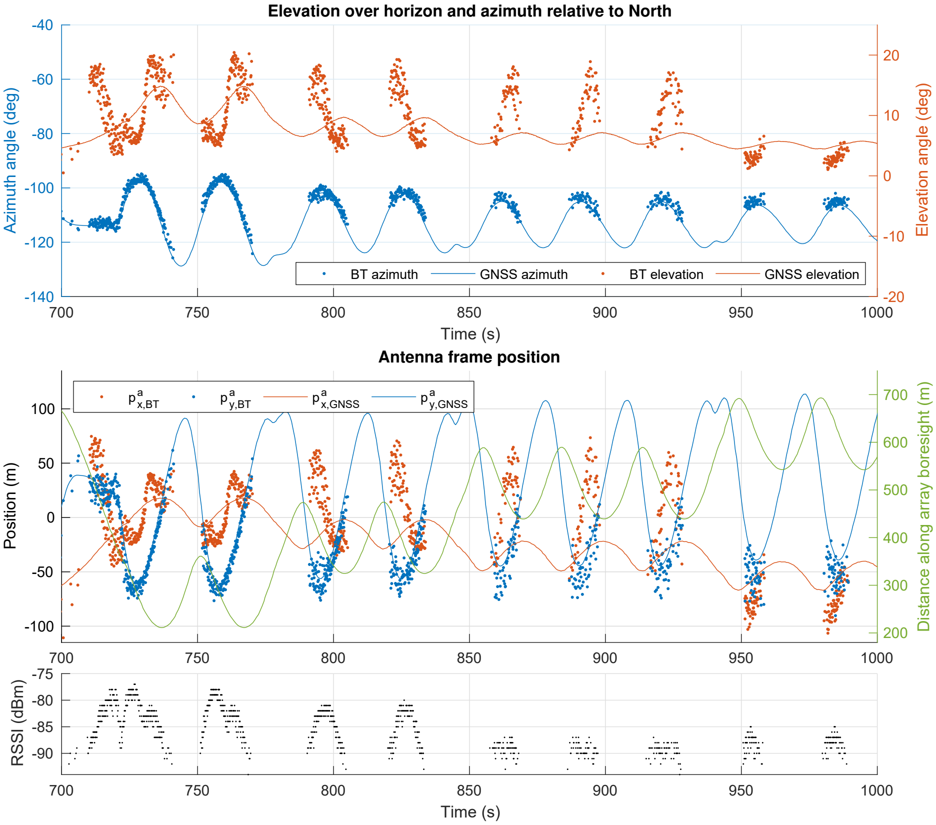


Figure 4.29: Estimation results for loiters at four different distances from the array, with increasing distance, as visible on the left side in Fig. 4.23.

4.5 Multipath

If the signal broadcasted by the moving transmitter not only propagates directly to the receiver array but also reflects off surfaces and continues to reach the array, we have a multipath situation. For a long-range navigation scenario, the antenna array boresight may point close to horizontally, with the direction to the transmitter having a shallow elevation angle over the horizon. The signal transmitted can then be expected to reflect off the ground or water surface below, and the signal reaching the array will be the sum of a direct and a reflected signal, as illustrated in Fig. 4.30.

If the array is large enough, the angular resolution will allow the separation of the direct and reflected signals if the angle between them is sufficient. For a transmitter far away, the angle separating the direct and ground reflected signals is approximately two times the transmitter elevation measured from the antenna location, and separability thus requires a larger array for lower elevation angles. If the array is too small to satisfy the *Rayleigh criterion* for the transmitter direction and its reflection, a single combined peak appears in the spectrum. The Rayleigh criterion is satisfied if the angular separation between the peaks is at minimum as large as the separation of each peak and the first null of its response, known as the *Rayleigh resolution limit* [152]. For the array used here, Fig. 4.9 show this to be approximately 30° if the first signal is in the boresight direction (90°), with the null at approximately 60° along $\Psi = 0$. It is important to note that since the projection of the array element positions is smaller in other directions, the angular resolution of the array is reduced for directions away from the boresight. The interference of the two signals will result in an error in the direction of the pseudo-spectrum peak compared to the pseudo-spectrum of only a direct signal.

Signal reflection behaves differently for different frequencies, surface materials, and reflection angles. For a circularly polarized signal, the reflected signal is linearly polarized if the angle of incidence equals Brewster's angle,

$$\theta_B = \operatorname{atan}\left(\frac{n_{\text{surface}}}{n_{\text{air}}}\right), \tag{4.24}$$

where θ_B is measured from the surface normal. n_{surface} and n_{air} are the refractive

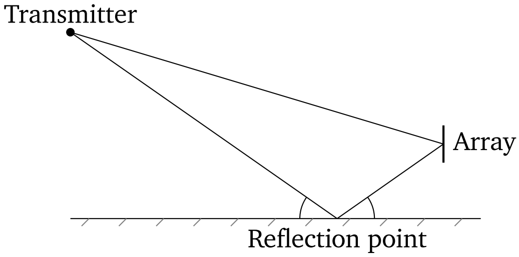


Figure 4.30: Multipath geometry

indices of the surface and air, respectively. A RHCP signal reflected off a surface at normal (i.e. 0°) incidence will have left-handed circular polarization (LHCP). For incidence at the Brewster angle θ_B, the reflected signal is linearly polarized in the horizontal direction [138]. For angles between the Brewster angle and the normal, the polarization is left-handed elliptical. For signals incident at large angles of incidence, or small grazing angles, the reflection will remain right-handed polarized, but also elliptical. Since an RHCP receiver antenna rejects LHCP signals, incidence at near-normal angles results in less multipath issues than small grazing angles, which is the case for the experimental results in Fig. 4.26.

For radio frequencies, n_{air} is just slightly greater than 1. As a few examples, [114] lists the refractive index of dry snow (0.05% liquid water content) at 0°C and 2-3 GHz as 1.016, yielding $\theta_B \approx 45.5°$, while moist snow (1% liquid water) has a value of 1.123, $\theta_B \approx 48.3°$. Comparably, ice was reported to have a value of 1.78 [27, 114], $\theta_B \approx 60.7°$, while pure water (i.e. rain water) has a value of approximately 9 [27, 75, 114], resulting in $\theta_B \approx 84°$.

Since the refractive index of the surface material is higher than that of air, the phase of the reflected signal changes by a half cycle. For a circularly polarized signal, this occurs for all incidence angles [138].

4.5.1 Multipath simulation

A simulation study was conducted to determine how a ground reflection would manifest in direction estimates. Assumptions are a completely flat ground, and a half cycle phase shift in the reflection due to refractive index. The distance traveled by the direct signal path between array element i and the transmitter antenna is

$$d_{\text{direct},i} = \|\boldsymbol{p}^n_{\text{a},i} - \boldsymbol{p}^n_{\text{t}}\|_2, \tag{4.25}$$

where $\boldsymbol{p}^n_{\text{a},i}$ is the North-East-Down (NED) position of the i-th receiver array antenna element, and $\boldsymbol{p}^n_{\text{t}}$ is the position of the transmitter antenna. The complex phase measurement of receiver antenna i is calculated from the distance as

$$x_{\text{direct},i} = e^{\frac{2\pi j}{\lambda} d_{\text{direct},i}}. \tag{4.26}$$

Note that the measurements for each antenna, unlike the real measurements, are simulated as simultaneous. For a completely flat ground surface, assuming that the angles of incidence and reflection are equal as illustrated in Fig. 4.30, the coordinate of the reflection point $\boldsymbol{p}_{\text{r},i}$ for the i-th array element is

$$\boldsymbol{p}_{\text{r},i} = \begin{bmatrix} \boldsymbol{p}^n_{\text{a, NE},i} + \left(\frac{-p^n_{\text{a, D},i}}{-p^n_{\text{a, D},i} - p^n_{\text{t, D}}}\right) \frac{\boldsymbol{p}^n_{\text{t, NE}} - \boldsymbol{p}^n_{\text{a, NE},i}}{\|\boldsymbol{p}^n_{\text{a, NE},i} - \boldsymbol{p}^n_{\text{t, NE}}\|_2} \\ 0 \end{bmatrix}, \tag{4.27}$$

where subscript NE denotes the two-dimensional North-East component of the vector and subscript D denotes the scalar Down (vertical) component. The path distance traveled by the reflected signal is

$$d_{\text{reflected},i} = \|\boldsymbol{p}^n_{\text{a},i} - \boldsymbol{p}^n_{\text{r},i}\|_2 + \|\boldsymbol{p}^n_{\text{t}} - \boldsymbol{p}^n_{\text{r},i}\|_2, \tag{4.28}$$

and the phase measurement for the reflected signal alone is

$$x_{\text{reflected},i} = c_r e^{\frac{2\pi j}{\lambda} d_{\text{reflected}} + \pi j}. \tag{4.29}$$

A coefficient c_r, typically between 0 and 1, is used for attenuation of the reflected signal. It should be noted that the phase is inverted by the addition of πj in the exponent. The phase measured at the receiver antenna is considered to be the sum of the direct and reflected signals,

$$x_{\text{measured},i} = x_{\text{direct},i} + x_{\text{reflected},i}. \tag{4.30}$$

These simulated measurements can then be used in the direction estimation algorithm in the same way as real measurements but without any need for frequency estimation or correction for the difference in measurement time. To visualize the error in estimated elevation angles for different positions in front of the array, we estimate direction for positions on a grid of heights and horizontal distances from the array. Simulated measurements are created for a transmitter at each position, and the pseudo-spectrum is searched using an NLP solver initialized using the known true direction. Simulation results are shown in Figs. 4.32b to 4.35b compared with experimental data explained in Section 4.5.2.

4.5.2 Experimental multipath testing

To validate that the effect of multipath on the elevation angle estimate behaves as described in Section 4.5.1, a field experiment was performed using a DJI S1000 multirotor UAV carrying the Bluetooth transmitter board and antenna. The test was performed on a grass runway covered with snow, shown in Fig. 4.31. The antenna array was set up on three different heights, with the array center at 1.35 m, 0.7 m and 0.15 m above ground, and upwards pitch angle of 10°. The mounting bracket had a 10° angle built-in and was leveled using a two-dimensional bubble level. Finally, a test was conducted at roughly 0.17 m height with a pitch angle of approximately 45° without leveling (later estimated as 48.5° degrees from the data). The runway was quite flat, see Fig. 4.31, but the slope was not perfectly known. RTK GNSS was used for accurate and precise relative positioning, for comparison with Bluetooth estimates, as in the previous experiments.

The UAV was flow on a flight path along the runway in front of the array, with approximately a constant azimuth angle Ψ_n, distances out to 180m and a height up to 100m. The flight path consisted of straight line segments, as shown using a side-on view in Fig. 4.32a where the array is located at the origin in the bottom right corner. The error in the Bluetooth elevation angle estimates were computed by subtracting the elevation calculated from RTK GNSS position measurements, which were used

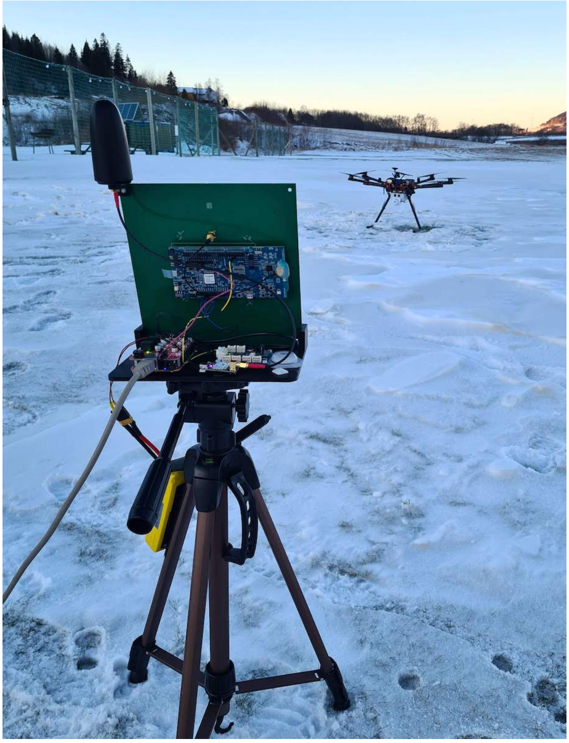
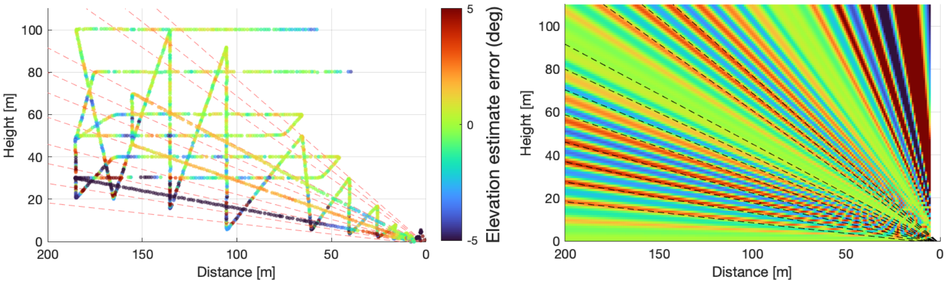


Figure 4.31: The antenna array on a tripod, with array center height 0.7 m over the ground. The DJI S1000 UAV in visible in the background.

as ground truth values. Figs. 4.32a to 4.35a shows the computed Bluetooth elevation angle errors from the field experiments, using only the 2480 MHz channel, using color along the flight trajectory. Red color indicates that the Bluetooth elevation estimate is higher than the RTK GNSS elevation, and the opposite for blue color. Figs. 4.32b to 4.35b shows the predicted result from simulated measurements for the same array placement scenarios as in the experiments. For a position grid of distances and heights, elevation angles were estimated using simulated measurements with both $c_r = 0.25$ and $c_r = 0$ (multipath-free). By subtracting the multipath-free estimates from the estimates with multipath, the simulated error was found. The dotted lines are drawn with the same angle in both figures, centered in positive error rays from the experimental data, for easier comparison of the patterns.

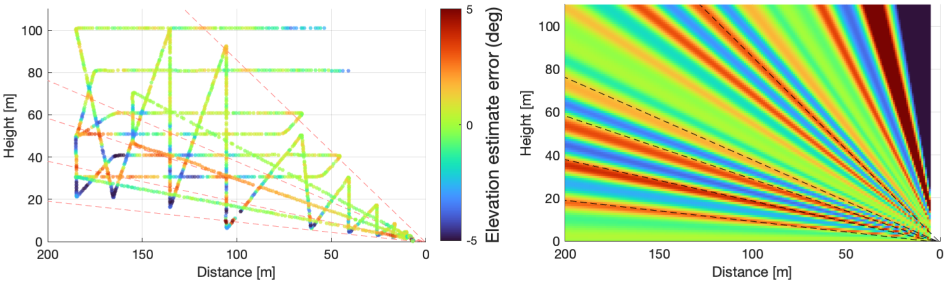
The simulation does not take polarization of the reflected signal, or antenna properties such as radiation pattern, into account. The polarization mostly causes disagreement at higher elevation values, where the error is reduced since the reflection is then left-hand polarized, which is not what the array is designed to receive. The effect of radiation pattern has the most effect in the scenario in Fig. 4.35 with the array pointing upwards in the direction of the UAV when flying at high elevation angles. The reflection is then received by the array from a direction almost in the



(a) Experimental Bluetooth elevation estimate error.

(b) Elevation estimate error for simulated measurements with multipath.

Figure 4.32: Multirotor validation test using RTK GNSS, array center 1.35 m over ground and 10° upwards array pitch.



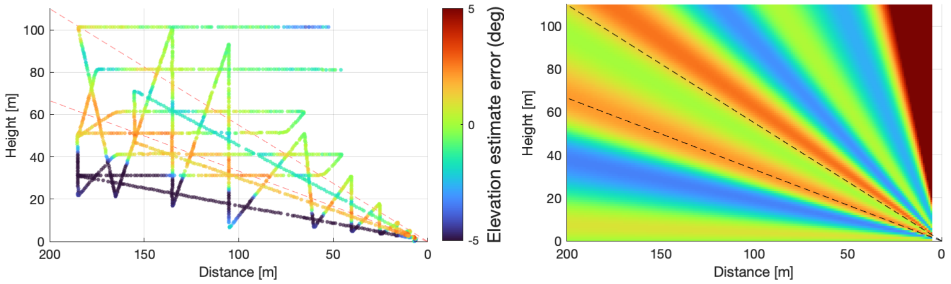
(a) Experimental Bluetooth elevation estimate error.

(b) Elevation estimate error for simulated measurements with multipath.

Figure 4.33: Multirotor validation test using RTK GNSS, array center 0.7 m over ground and 10° upwards array pitch.

array PCB plane, where the gain is much lower for the directional patch antennas.

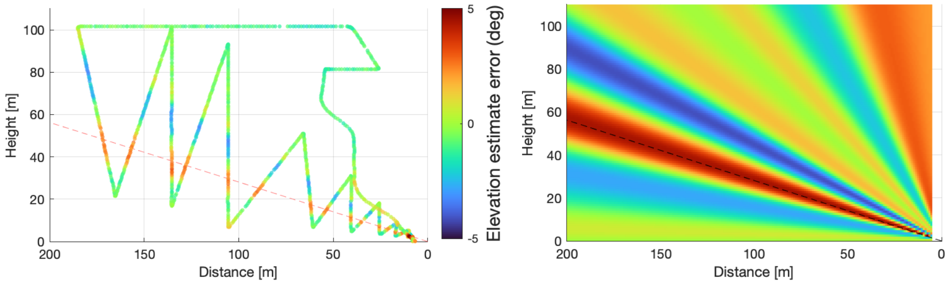
Comparing the experimental and simulated results, where the experimental results are considered samples of an underlying error pattern covering the area shown in the simulated results, it is clear that the observed pattern is similar to the simulation, especially for low elevation angles. If the offset between the experimental and simulated results was the same for all three heights, a reasonable source of the offset would be runway slope or a bias in the array leveling. Since the array was re-leveled for each of the heights over the ground, differences in the residual leveling error would lead to differences in the resulting angle offsets. The location on the ground in front of the array where signals reflect depends on the array height above ground, being closer to the array for low array heights. The simulation assumes a perfectly flat ground, which is not the case in reality, being a significant error source.



(a) Experimental Bluetooth elevation estimate error.

(b) Elevation estimate error for simulated measurements with multipath.

Figure 4.34: Multirotor validation test using RTK GNSS, array center 0.15 m over ground and 10° upwards array pitch. In this case, the upper dashed line was placed in a positive error peak for the experimental data, but it appears that the minimum-amplitude direction between the dashed lines for simulation occurs at a higher elevation value for the experimental data, resulting in a very narrow positive peak at the edge of the red beam visible in the simulation result. The increased negative error between the dashed lines in the experimental data also supports this.



(a) Experimental Bluetooth elevation estimate error.

(b) Elevation estimate error for simulated measurements with multipath.

Figure 4.35: Multirotor validation test using RTK GNSS, array center approximately 0.17 m over ground and 48.5° upwards array pitch. Compared to 10° tilt, the 25° zero amplitude lobe is moved up to approximately 40°, and the oscillation spatial frequency is increased a little for low angles due to the effectively smaller array from this direction.

4.5.3 Factors affecting elevation error

Based on simulating different scenarios, the effect of a number of parameters was observed, which are shown in Fig. 4.38 compared to the reference case Fig. 4.38a:

4.5.3.1 Reflection magnitude

(Fig. 4.38b) A stronger reflection increases the error amplitude. This depends on ground material including water content, e.g. wet grass reflects more than dry grass. Seawater is highly reflective for radio frequencies.

4.5.3.2 Height above surface

(Fig. 4.38c) Increasing the height of the array over the surface increases the spatial frequency of the elevation error oscillation with respect to the true elevation angle. To explain why these oscillations occur, we combine (4.14) with the multipath signal sum (4.30), resulting in

$$P(\Psi,\alpha) = |\boldsymbol{a}(\Psi,\alpha)^{\top}(\boldsymbol{x}_{\text{direct}} + \boldsymbol{x}_{\text{reflected}})|^2. \tag{4.31}$$

For elevation angles where the complex value $\boldsymbol{a}(\Psi,\alpha)^{\top}\boldsymbol{x}_{\text{reflected}}$ aligns with $\boldsymbol{a}(\Psi,\alpha)^{\top}\boldsymbol{x}_{\text{direct}}$ in the same or opposite direction, the addition of the reflected signal does not influence the estimated elevation angle, and the error is zero. When the components do not align, the elevation angle maximizing the spectrum changes.

Note that the wide "bands" of reduced error remain in the same directions, and that these depend on the amplitude of the error, not the phase. Minima in the amplitude occurs when the argument (phase angle) of the values of the complex vector $\boldsymbol{a}(\Psi,\alpha)\odot\boldsymbol{x}_{\text{reflected}}$ spread out, such that their sum $\boldsymbol{a}(\Psi,\alpha)^{\top}\boldsymbol{x}_{\text{reflected}}$ has a small magnitude. On the other hand, when the values align such that $\boldsymbol{a}(\Psi,\alpha)^{\top}\boldsymbol{x}_{\text{reflected}}$ has a maximum magnitude, the error amplitude is maximized.

4.5.3.3 Antenna pitch angle

(Fig. 4.38d) By pitching the array up there is essentially a reduction in the vertical array size seen from the front through projection. In the simulated case this moves zero-amplitude rays to higher angles without changing the phase of the oscillations.

A second effect that is not included in the simulation is that by pitching the array, the radiation pattern maximum, which points in the boresight direction, points upwards. This reduces the gain in downwards directions from which the reflected signals are received. Additionally, the directions blocked by the array ground plane change. Consider a pitch angle where the direct signal has 0 dB gain while the reflection has -3dB, and compare this to a pitch angle where the direct signal has gain -3dB while the reflection has -10dB. The effect of multipath will be reduced, at the cost of reduced range. This can go as far as having the reflected signal arrive at the array from behind.

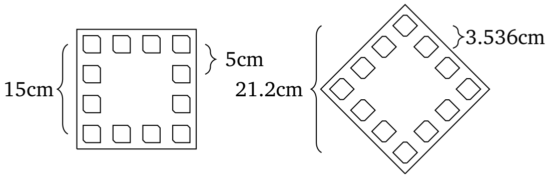


Figure 4.36: Orienting the array with the diagonal vertically changes the effect of multipath.

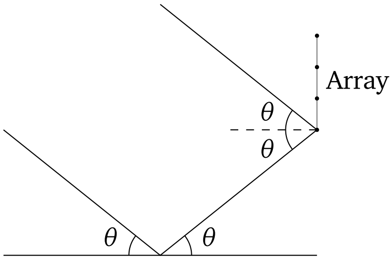


Figure 4.37: For a signal at an infinite distance with a positive upwards elevation angle, the angle between the direct and reflected signals reaching each array element is twice the source elevation angle.

4.5.3.4 Antenna size and in-plane orientation

(Figs. 4.38e and 4.38f) Increasing the size of the array moves the elevation bands of small error to lower angles, and increases their frequency. This reduces multipath error, but also requires an increased number of elements to avoid ambiguity errors or too high sidelobe power. The orientation of the array also matters if the maximum dimension is different in different in-plane directions, e.g. a square antenna can be mounted with a diagonal vertically, giving an increased vertical size, as illustrated in Fig. 4.36.

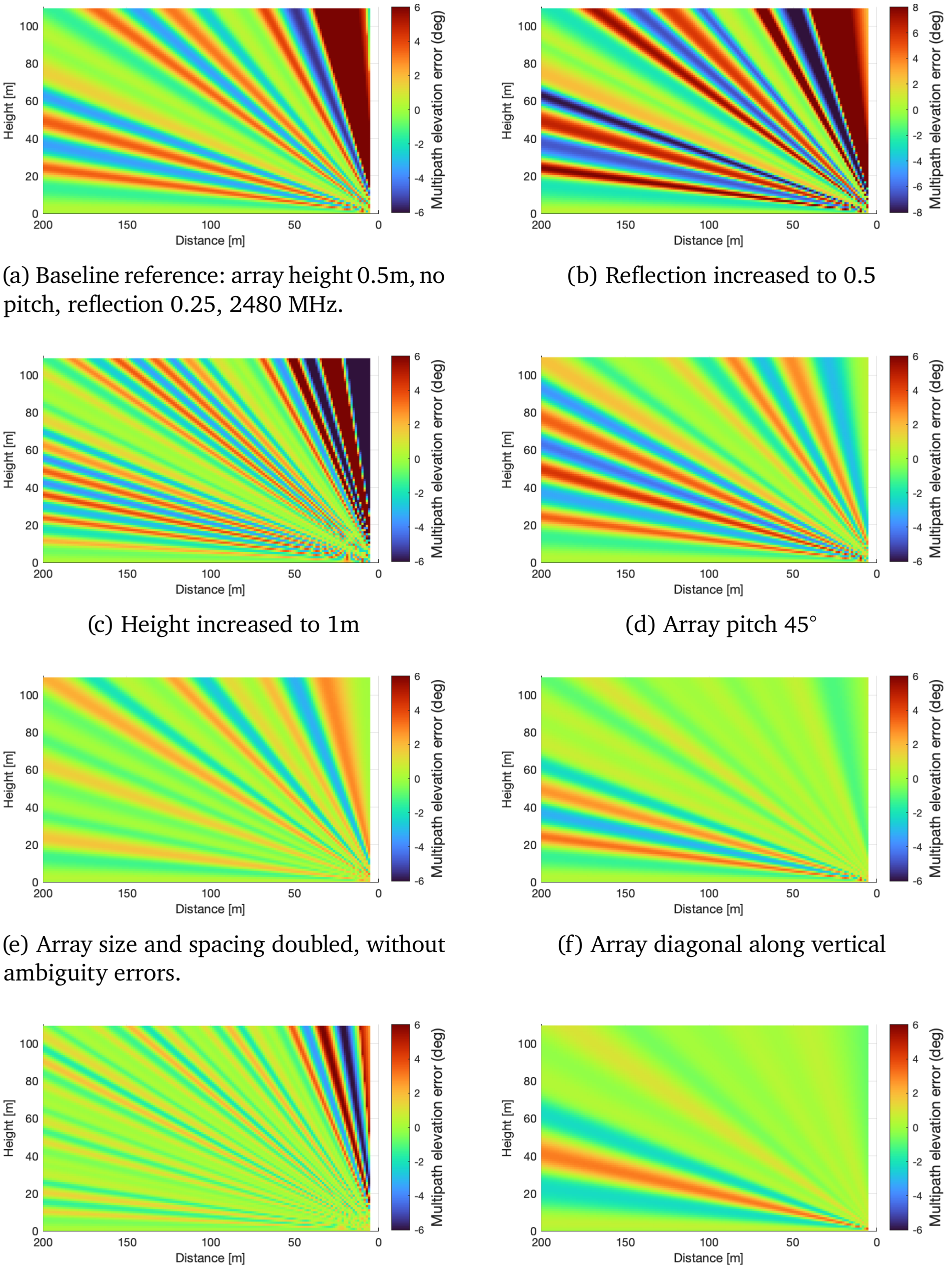
For a signal source at an infinite distance (far field) in front and higher than the array, the reflected signal will reach the array at an angle from below equivalent to the angle from above, illustrated in Fig. 4.37.

4.5.3.5 Signal frequency

(Fig. 4.38g) For Bluetooth, the carrier frequencies of the different channels give a minor effect on the error, which is more pronounced at higher elevation angles. If higher frequencies could be used, however, such as 5.8 GHz, multipath would be reduced since an array of the same physical size would be larger relative to the signal wavelength. This would however require more elements to avoid ambiguity errors.

4.5.4 Calibration

Canceling the multipath elevation error using a calibration for a specific scenario could be beneficial for the practical use of Bluetooth for navigation. A simple elevation angle calibration is however not globally possible above a reflection threshold that depends on the situation, as the true and estimated elevation angles do not always have a one-to-one relationship: More than one geometric elevation angle can result in the same measurement due to the multipath error. Fig. 4.39 shows simulated examples of estimated elevation angle plotted as a function of the true elevation angle for a transmitter at 500 m distance from the array, with the different colors representing different reflection coefficients. This is created using the method described in Section 4.5.1, but with positions at a constant Euclidean distance with a range of elevation angles. One way the calibration problem could be handled is to track the direction over time, but this does not solve the problem if the initial direction is unknown. Fig. 4.40 shows a similar plot for field experiment data.



(a) Baseline reference: array height 0.5m, no pitch, reflection 0.25, 2480 MHz.

(b) Reflection increased to 0.5

(c) Height increased to 1m

(d) Array pitch 45°

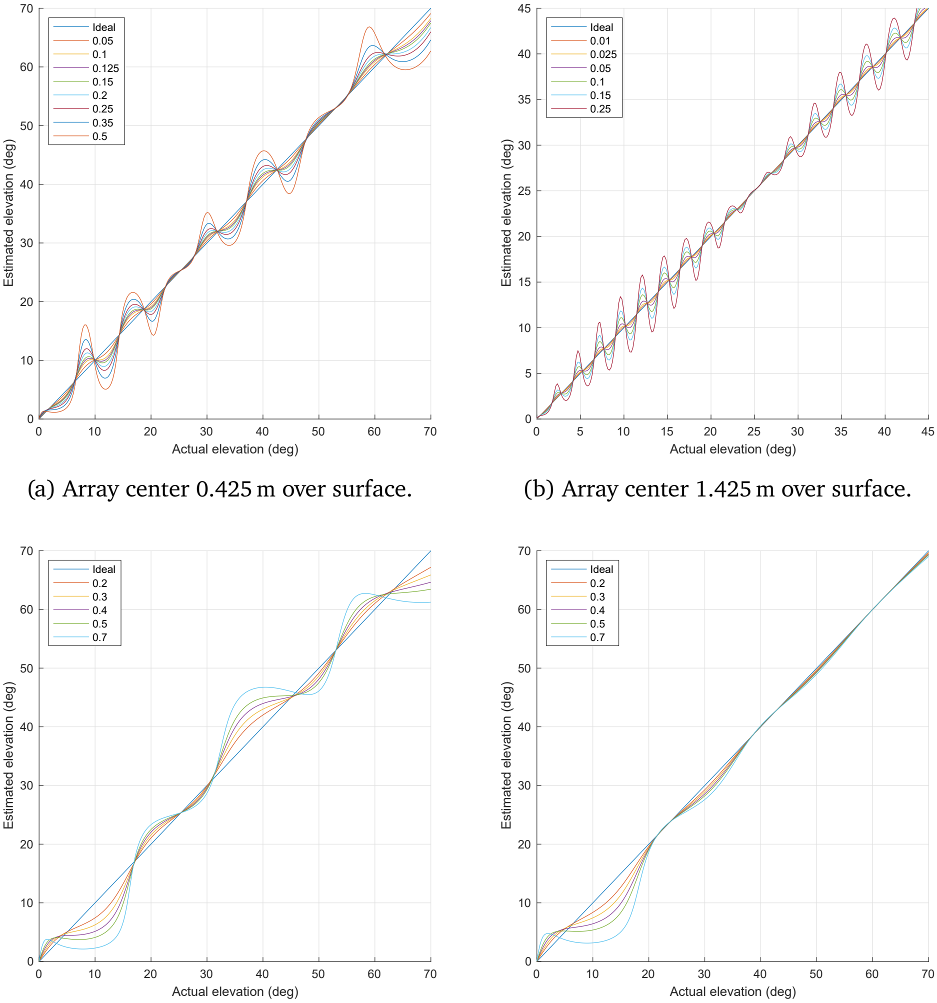
(e) Array size and spacing doubled, without ambiguity errors.

(f) Array diagonal along vertical

(g) Frequency increased to 5800 MHz, without ambiguity errors.

(h) Array pitch 10°, top antenna height 0.4m, diagonal along vertical

Figure 4.38: Comparison of multipath simulation results, illustrating the effect of different parameters.



(a) Array center 0.425 m over surface.

(b) Array center 1.425 m over surface.

(c) Array center 0.15 m over surface.

(d) Array center 0.15 m over surface, diagonal aligned vertically.

Figure 4.39: Simulated elevation angle estimates plotted as a function of the actual elevation angle, for different scenarios and reflection coefficients c_r. In all cases the frequency is 2480 MHz and the array boresight points 10° over the horizontal.

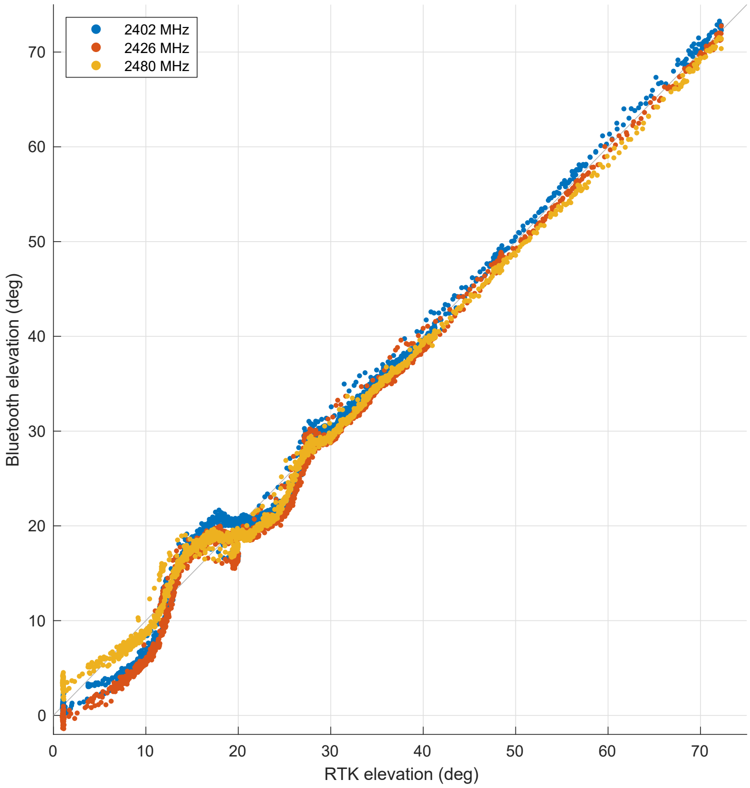


Figure 4.40: Elevation angle estimated from Bluetooth plotted as a function of elevation angle estimate from RTK GNSS, for the flight in Fig. 4.35a, with all three frequency channels included.

4.6 Conclusion and future work

The use of Bluetooth direction finding for outdoor navigation has been shown to be feasible at distances up to 700m, close to the receiver sensitivity of -93dBm, without significant loss of angular precision, which exceeded our expectations. Azimuth angle estimates were accurate, with standard deviation on the 1° level, while the elevation estimate suffered from systematic errors due to multipath. The effect of multipath was investigated using simulation and field experiments, showing quite consistent results. The range and performance of the system look promising for fixed-wing UAV recovery and ship docking in GNSS-denied environments, which will be a focus of future work. The proposed peak-search method should be implemented for online use in a payload, benchmarking its computational requirements compared with other methods. Methods to reduce the effect of multipath should be considered.

Chapter 5

Sampling Sequence Considerations for Bluetooth Angle-of-Arrival Estimation

This chapter is based on the publication

- [135] M. L. Sollie, T. A. Johansen, K. Gryte, and T. H. Bryne, "Sampling Sequence Considerations for Bluetooth Angle-of-Arrival Estimation," *Submitted, in revision*, 2022

and considers the possibility of reducing the effect of frequency estimation error on the direction estimate by using array sampling sequences that are time-symmetric.

5.1 Introduction

Different methods for sampling antenna arrays for estimation of received signal direction have been used through the last decades. Motivated by reducing hardware complexity and cost, it has in some cases been a desire to avoid the typical assumption that each element of the arrays has its separate receiver, which may be accomplished in different ways. The use of analog hardware to sum the signal from each element using complex weighing [33, 157] allowed the correlation of the measurements with signal steering vectors to be done in hardware, before sampling the total response, instead of in software. A different method was proposed in [36], where a signal switch connected a single antenna at a time to analog downconversion (to baseband or an intermediate frequency), before passing the signal to individual low-pass filters for each element. Digital sampling could then be performed simultaneously with separate digital sampling at the output of each filter. Using a single receiver with sequential element sampling was described in [148], with the main focus being on the hardware implementation of different line delays for each array element to compensate for the difference in measurement time. The alternative digital compensation for measurement time differences after sampling was also mentioned and later considered by others [119, 107]. There are also methods using a reduced number of receivers greater than one to switch between different subsets of the array elements [164], or the switching between sub-arrays with analog weighing [163, 118].

Direction finding was introduced as a feature in the Bluetooth 5.1 specification in 2019 to allow the determination of the direction from which a Bluetooth packet is received using antenna arrays [15]. The antenna array can be used either on the receiving side, called angle-of-arrival (AoA), or on the transmitting side, called angle-of-departure (AoD) [162]. A pure carrier is transmitted at the end of a Bluetooth packet for this use, called the Constant Tone Extension (CTE). Bluetooth uses a single transceiver and signal switching to reduce cost, power consumption, and requirements for hardware components. Since only one array element can be used at any time, an order of events for the element switching procedure must be chosen. The use of a single transceiver means that Bluetooth arrays cannot be used for transmission beamforming, but makes it possible to perform carrier phase interferometry for direction determination. With the use of low-cost oscillators, there is significant uncertainty in both the frequency transmitted and the frequency of the reference oscillator used for in-phase and quadrature (IQ) sampling on the receiving side. Combined with the difference in sampling time, this results in a carrier frequency offset (CFO) in the IQ samples which must be taken into account in the direction estimation, and the frequency must therefore be determined.

In [11] it was shown using theoretical considerations and simulations that the sequence of switching and sampling has a significant effect on the lower bounds for errors in the estimation of signal frequency and arrival direction. It was taken into account that after switching between elements the transient at the receiver input must be allowed to settle before sampling, such that frequent switching reduces the time where sampling can be performed. It was concluded that for frequency estimation the interval between the first and last measurement for at least one array element should be maximized. The number of samples from each element should be roughly the same, the number of lost measurements due to switching should be minimized, and the set of measurement instants for each element should share the same centroid. The last point means that the sampling sequence should have a mirrored or time-symmetric structure, as this has the effect of removing the direct effect of frequency error on the direction determination. Bluetooth AoA was used as an example case, although no experimental testing was performed.

The main contributions of this paper is the practical considerations for different choices of sampling orders for Bluetooth AoA, including experiments using a Bluetooth antenna array. [11] considered only one type of mirrored sampling sequence, while we consider several different possible options, and take practical constraints such as the sampling structure from the Bluetooth specification [15] into account.

It is of interest to understand the reasons behind and limitations of the CFO-independence of direction estimation using mirrored sampling orders, and if it is possible to omit the CFO estimation completely and still get an accurate direction estimate, or reduce the required CFO estimation accuracy.

The paper proceeds as follows: In Section 5.2 we consider the effect of using time-symmetric sampling orders in direction determination for different errors in the CFO estimate. In Section 5.3 we discuss different structural options for such sequences. Results from processing of measurements from experiments using a Bluetooth antenna array are presented in Section 5.4. Finally, conclusions are drawn in Section 5.5.

5.2 Time-symmetric sampling sequences

Consider that we have an antenna array of m elements, and sample a complex measurement vector $\boldsymbol{x} \in \mathrm{C}^n$, assumed to be at baseband, of in-phase and quadrature components from a single signal source. Since we have $n > m$ measurements, at least one element has been sampled more than once. We estimate the direction of arrival by finding the antenna frame line-of-sight vector $\boldsymbol{l} \in \mathrm{R}^3$ correlating best with the signal received by maximizing the spatial pseudo-spectrum

$$P(\boldsymbol{l}) = |\boldsymbol{a}(\boldsymbol{l})^H \boldsymbol{x}|^2, \tag{5.1}$$

with the steering vector $\boldsymbol{a}(\boldsymbol{l}) = e^{(\frac{2\pi j}{\lambda} \boldsymbol{P}^\top \boldsymbol{l})}$. This is a conventional beamformer approach [71, 152]. $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_1 & \dots & \boldsymbol{p}_n \end{bmatrix} \in \mathrm{R}^{3\times n}$ is the matrix of array element positions and $()^H$ denotes the conjugate transpose. This means that repeated measurements from the same element i is multiplied with the same value $a_i(\boldsymbol{l})^H = e^{(-\frac{2\pi j}{\lambda} \boldsymbol{l}^\top \boldsymbol{p}_i)}$. Thus, if we have measurements $x_{i,k}$ for element $i = 1, \dots, m$ and sample $k = 1, \dots, n_k$, i.e. we have n_k measurements for element k, we can aggregate the measurements without altering the spectrum value,

$$\begin{bmatrix} a_1(\boldsymbol{l}) \\ a_2(\boldsymbol{l}) \\ \vdots \\ a_m(\boldsymbol{l}) \\ \vdots \\ a_1(\boldsymbol{l}) \\ a_2(\boldsymbol{l}) \\ \vdots \\ a_m(\boldsymbol{l}) \end{bmatrix}^H \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{m,1} \\ \vdots \\ x_{1,n_1} \\ x_{2,n_2} \\ \vdots \\ x_{m,n_m} \end{bmatrix} = \begin{bmatrix} a_1(\boldsymbol{l}) \\ a_2(\boldsymbol{l}) \\ \vdots \\ a_m(\boldsymbol{l}) \end{bmatrix}^H \begin{bmatrix} \sum_{i=1}^{n_1} x_{1,i} \\ \sum_{i=1}^{n_2} x_{2,i} \\ \vdots \\ \sum_{i=1}^{n_m} x_{m,i} \end{bmatrix}. \tag{5.2}$$

We can write this as the aggregation function

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}\left(\begin{bmatrix} x_{1,1} & x_{2,1} & \dots & x_{m,1} & \dots & x_{1,n_1} & x_{2,n_2} & \dots & x_{m,n_m} \end{bmatrix}^\top\right) = \begin{bmatrix} \sum_{i=1}^{n_1} x_{1,i} \\ \sum_{i=1}^{n_2} x_{2,i} \\ \vdots \\ \sum_{i=1}^{n_m} x_{m,i} \end{bmatrix}, \tag{5.3}$$

and if two measurement vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have $\boldsymbol{g}(\boldsymbol{x}_1) = \boldsymbol{g}(\boldsymbol{x}_2)$, then the direction estimation will yield the same result.

For a measurement vector assumed to be at baseband after compensation of the estimated frequency offset, a residual frequency compensation error results in the actual measurement vector $\boldsymbol{x} \odot e^{2\pi f \boldsymbol{t}}$, where $\boldsymbol{x}$ is the baseband measurements, f is the CFO error, $\boldsymbol{t}$ is the vector of measurement times, and $\odot$ denotes element-wise multiplication. Consider the array with elements numbered as shown in Fig. 5.1, and an element sampling sequence as shown in the timeline of Fig. 5.2. This sequence is equivalent to a linescan over the array, starting from the top left to bottom right line by line, and then runs the same in reverse, ending at the same antenna as it started with. That is, the sampling order is symmetric in time. Both measurements from each antenna is sampled with equal time spacing from the central time of symmetry $t = 23$µs. The spacing from the central time for a measurement is denoted Δt. Considering the ideal noise-free case where all IQ measurements are identical if the residual CFO error after correction is 0, then the aggregated measurements from a single antenna, both with baseband value m, can be expressed as

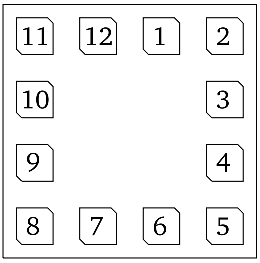


Figure 5.1: Array element numbering, with the array seen from the front.

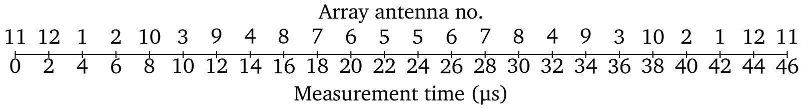


Figure 5.2: Top-bottom-top time-symmetric sampling order. Every element is sampled twice, with a total duration of 46µs.

$$\boldsymbol{g}\left(\begin{bmatrix} me^{2\pi j f \Delta t} & me^{-2\pi j f \Delta t}\end{bmatrix}\right) = me^{2\pi j f \Delta t} + me^{-2\pi j f \Delta t} = 2m\cos(2\pi f \Delta t). \tag{5.4}$$

The result is a scalar multiple of the baseband value. Since the time of symmetry is the same for all measurement pairs due to the time-symmetric measurement order, the phase angles of the aggregated measurements maintain their value relative to each other as long as $\cos(2\pi f \Delta t) > 0$. As $f \Delta t$ approaches $\frac{1}{4}$ the magnitude of the aggregated measurement approaches 0, and for $\frac{1}{4} < f \Delta t < \frac{3}{4}$ the sign changes. The measurements at the ends of the interval will be the first to change sign, which means that the measurements close to the middle of the sampling period suffer the least from a reduction in total magnitude, and therefore have the strongest influence on the spectrum initially for increasing CFO error. For large CFO errors, we might, however, see that the furthest spaced measurements wrap around the complex origin multiple times as the closely spaced measurements approach $\cos(2\pi f \Delta t) = 0$. As an example, consider the 46µs spacing between samples from antenna 11. With a residual CFO error of 2 kHz, the measurements would be separated by 33.12°, which is illustrated in Fig. 5.3. Summing the measurements results in a magnitude of $2\cos(2\pi \times 2000\text{Hz} \times 23 \times 10^{-6}\text{s}) = 1.917$ times the length of each measurement's magnitude, which is less than the factor 2 which would be the result without the CFO error.

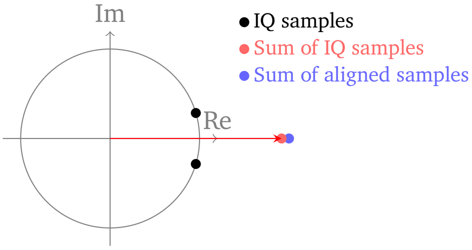


Figure 5.3: Spread of two IQ measurements (black dots) for a CFO error of 2 kHz and a time separation of 46µs. The sum of the measurements is shown in red, and the blue dot illustrates the magnitude of the measurement sum if both measurements had been identical, as would be the case without CFO error.

For the whole sequence of measurement time spacings, $\boldsymbol{\Delta t} = \begin{bmatrix} 1\mu\text{s} & 3\mu\text{s} & \dots & 23\mu\text{s} \end{bmatrix}$, we have

$$\boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix} \odot e^{2\pi j f [-\boldsymbol{\Delta t}_r \;\; \boldsymbol{\Delta t}]}\right) = \boldsymbol{g}(\boldsymbol{x} \odot 2\cos(2\pi f \boldsymbol{\Delta t})), \tag{5.5}$$

where $\boldsymbol{x}_r$ is the time-reversal of the measurement vector $\boldsymbol{x}$, $\boldsymbol{\Delta t}_r = \begin{bmatrix} 23\mu\text{s} & 21\mu\text{s} & \dots & 1\mu\text{s} \end{bmatrix}$ is the reversal of $\boldsymbol{\Delta t}$. Since the time of symmetry is common for all the measurements, a small frequency error is simply removed at the expense of a slight reduction in signal magnitude, with relative phase angles being the same as without a CFO error. This is the case until the measurements from antenna 11 reach a separation of 180°, which occurs for $f = \frac{1}{4 \times 23 \times 10^{-6}\text{s}} = 10870\text{Hz}$. This situation is illustrated in Fig. 5.4. In this case, the measurements from antenna 11 sum to 0, while the reduction in measurement magnitude for the rest of the antennas depends on the distance from the time of symmetry. Beyond this frequency, the phase of the sum of the measurement pairs with the largest time separation

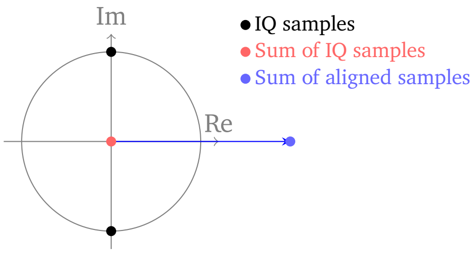


Figure 5.4: Cancellation of measurements with half cycle phase difference.

reverses its direction in the complex plane, as illustrated in Fig. 5.5. This will start to introduce severe errors in the direction spatial spectrum.

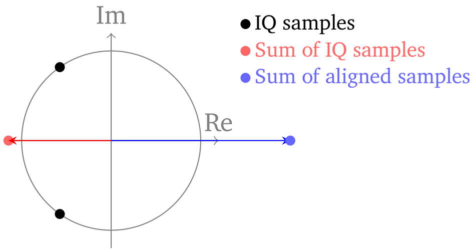


Figure 5.5: Reversal of the direction of aggregated measurements with large phase separation.

Fig. 5.6 illustrates the combined magnitude along the direction of the baseband measurements for the 12 aggregated measurement pairs, as a function of the CFO error f. The measurement pair with the largest Δt is the first to become negative.

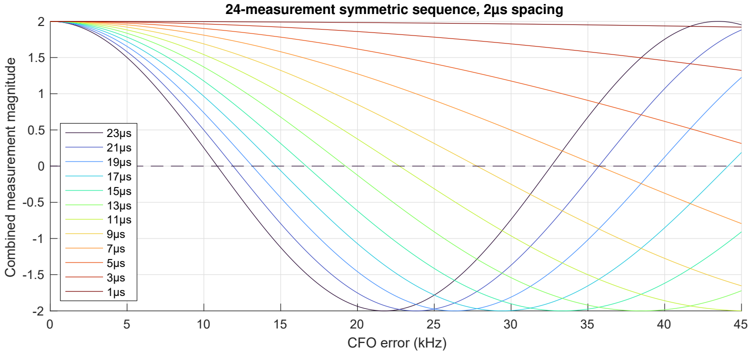


Figure 5.6: Magnitude of aggregated measurements for sampling order structure $\left[\boldsymbol{x} \quad \boldsymbol{x}_r\right]$. The legend indicates the value Δt between the time of symmetry and the time of measurements for each pair.

For the sampling order in Fig. 5.2, the measurements from the top of the array will be the first to have the aggregated measurements reversed. This has an influence on the resulting pattern in the calculated spatial spectrum. For a more randomly chosen antenna sequence, the phase reversal would appear more spread out over the array. It should be noted that if all array elements have the phase reversal occuring simultaneously, the spectrum would still indicate the correct direction, and only a mix causes the error. As the error increases, the measurements spread out more and more, resulting in a cyclical error for small error thresholds initially, until the direction of the measurement sums appear nearly randomly reversed.

5.3 Alternatives for long time-symmetric sequences using Bluetooth

With enough time in a 160µs CTE to freely sample for 148µs after subtracting the 12µs used for the 4µs *guard* and 8µs *reference* sampling periods [162], which is more than the 46µs sequence considered in Section 5.2, we can increase the number of samples by two or three times. This can be done in more than one way while maintaining time symmetry. We can repeat the previous sequence to $\begin{bmatrix}\boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r\end{bmatrix}$ or $\begin{bmatrix}\boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r\end{bmatrix}$, or sample each element multiple times in a row before switching. It should be noted that if we sample the same element using multiple sampling slots in a row, no switching occurs between these, and sampling can also be done in the switching slots in between, as considered in [11]. Thus, if we sample the same element once in each sample slot, and use two successive sampling slots for the same element, we gain an additional measurement from the switching slot in-between, thus increasing the total number of measurements compared to switching every time. To create a time-symmetric sampling order where all measurements are used for estimation, it must be considered that the initial reference period where one element is sampled consecutively for 8µs must be repeated at the end of the sampling sequence. Otherwise, these samples must be excluded from the direction estimation.

Repeating the previous sampling order, $\begin{bmatrix}\boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r\end{bmatrix}$, results in a total of 48 measurements over 94µs. The first and last measurements are now separated by the central time of symmetry by 47µs. For small CFO errors, the measurements all contribute in the direction of the baseband measurements, as illustrated in Fig. 5.7. Due to the increased time spread, the first and last measurements reach a separation of 180° at the lower frequency of 5319 Hz. Unlike in the previous case, once the first measurements pass $f\Delta t = \frac{1}{4}$, the sum is not immediately negative, since we also have measurements with a smaller spacing from the same element. This is illustrated in Fig. 5.8. The aggregated measurements for an array element only changes direction when one pair of measurements contribute more in the negative direction than the other does in the positive direction, as illustrated in Fig. 5.9.

As we have a new longer sequence, which is a combination of two smaller time-

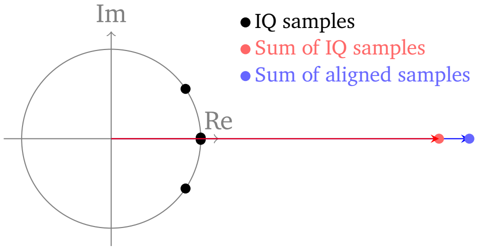


Figure 5.7: Small total phase spread of measurements from the same array antenna, due to CFO error and measurement time intervals. The sum of the spread measurements is close in magnitude to the combined measurement magnitude.

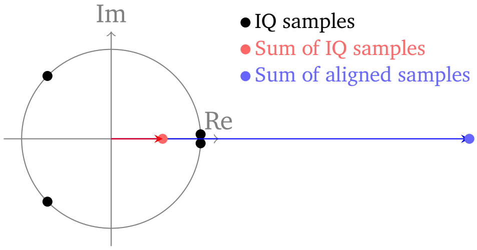


Figure 5.8: A total phase spread where the furthest spread measurements contribute in the opposite direction, but since the measurement sum still points in the correct direction, the estimate will still be the same in the absence of measurement noise. With IQ noise, the result is a reduction in the measurement signal-to-noise ratio.

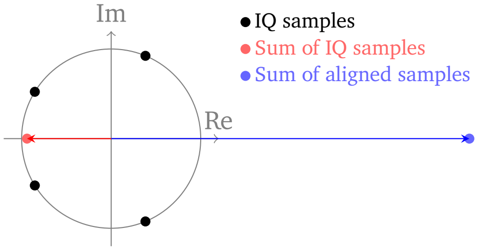


Figure 5.9: In this case the phase spread is such that the direction of the measurement sum is reversed. If this is the case for all array elements, the estimate is unaffected. The effect on the direction estimate occurs when this occurs for only parts of the array elements.

symmetrical sequences, we will use $\Delta \boldsymbol{t}$ to denote the time spacing vector for the complete new measurement vector, in this case $\Delta \boldsymbol{t} = \begin{bmatrix} 1\mu\mathrm{s} & 3\mu\mathrm{s} & \dots & 47\mu\mathrm{s} \end{bmatrix}$, while $\bar{\Delta} \boldsymbol{t}$ denotes the time spacing vector for each of the repeated measurement vector components, $\bar{\Delta} \boldsymbol{t} = \begin{bmatrix} 1\mu\mathrm{s} & 3\mu\mathrm{s} & \dots & 23\mu\mathrm{s} \end{bmatrix}$. For a time-symmetrical pattern repeated twice this yields the relation

$$\begin{bmatrix} -\Delta \boldsymbol{t}_r & \Delta \boldsymbol{t} \end{bmatrix} = \begin{bmatrix} -\bar{\Delta} \boldsymbol{t}_r - \frac{t_p}{2} & \bar{\Delta} \boldsymbol{t} - \frac{t_p}{2} & -\bar{\Delta} \boldsymbol{t}_r + \frac{t_p}{2} & \bar{\Delta} \boldsymbol{t} + \frac{t_p}{2} \end{bmatrix}, \tag{5.6}$$

where t_p is the pattern repetition period, which for 12 elements with 2µs spacing and a time-symmetrical pattern is $t_p = 48\mu s$. Because constant phase offsets does not affect the estimation result, this is also equivalent to using $\begin{bmatrix} -\bar{\Delta \boldsymbol{t}}_r & \bar{\Delta \boldsymbol{t}} & -\bar{\Delta \boldsymbol{t}}_r + t_p & \bar{\Delta \boldsymbol{t}} + t_p \end{bmatrix}$. Similarly, for a pattern repeated three times we would have the equivalence

$$\begin{bmatrix} -\Delta \boldsymbol{t}_r & \Delta \boldsymbol{t} \end{bmatrix} \equiv \begin{bmatrix} -\bar{\Delta \boldsymbol{t}} & \bar{\Delta \boldsymbol{t}} & t_p - \bar{\Delta \boldsymbol{t}} & t_p + \bar{\Delta \boldsymbol{t}} & 2t_p - \bar{\Delta \boldsymbol{t}} & 2t_p + \bar{\Delta \boldsymbol{t}} \end{bmatrix}, \tag{5.7}$$

with $\Delta \boldsymbol{t} = \begin{bmatrix} 1\mu s & 3\mu s & \dots & 71\mu s \end{bmatrix}$. Such a sequence can be aggregated as

$$\begin{aligned} & \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix} \odot e^{2\pi j f [-\Delta \boldsymbol{t}_r \ \Delta \boldsymbol{t}]}\right) \\ & = \boldsymbol{g}\left(\left[{\scriptstyle [\boldsymbol{x}\ \boldsymbol{x}_r] \odot e^{2\pi j f \left[-\bar{\Delta \boldsymbol{t}}_r \ \bar{\Delta \boldsymbol{t}}\right]} \quad [\boldsymbol{x}\ \boldsymbol{x}_r] \odot e^{2\pi j f \left[-\bar{\Delta \boldsymbol{t}}_r \ \bar{\Delta \boldsymbol{t}}\right]} e^{2\pi j f t_p}} \right]\right) && (5.8) \\ & = \boldsymbol{x} \odot 2\cos(2\pi f \bar{\Delta \boldsymbol{t}})(1 + e^{2\pi j f t_p}), && (5.9) \end{aligned}$$

or in general for multiple repetitions,

$$\begin{aligned} & \boldsymbol{g}\left(\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r & \dots \end{bmatrix} \odot e^{2\pi j f [-\Delta \boldsymbol{t} \ \Delta \boldsymbol{t}]}\right) = \\ & \qquad \boldsymbol{x} \odot 2\cos(2\pi f \bar{\Delta \boldsymbol{t}})(1 + e^{2\pi j f t_p} + e^{4\pi j f t_p} + \dots). \end{aligned} \tag{5.10}$$

The resulting total magnitude is increased by an additional measurement sequence for $|(1 + e^{2\pi j f t_p})| > 1$, which for $t_p = 48\mu s$ occurs for $|f| < 6944$Hz, and $|(1 + e^{2\pi j f t_p} + e^{4\pi j f t_p})| > 1$ for $t_p = 48\mu s$ and $|f| < 5208$Hz. It is a problem for direction estimation if $1 + e^{2\pi j f t_p} + e^{4\pi j f t_p} + \dots = 0$, as the measurements then cancel completely if the magnitudes of the repeated sequences are equal, leaving only measurement noise. We have $1 + e^{2\pi j f t_p} = 0$ for $f = 10417$Hz and $1 + e^{2\pi j f t_p} + e^{4\pi j f t_p} = 0$ for $f = 6944$Hz. These zeros due to repeated sampling order segments also occur in non-time-symmetric cases. The magnitude of the aggregated measurements along the baseline measurement direction for the double repetition $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$ is shown in Fig. 5.10. The zero at $f = 10417$Hz is clearly visible, in addition to a second zero at $f = 31250$Hz for when $f \cdot t_p = \frac{3}{2}$ yielding another full rotation of $e^{2\pi j f t_p}$. For CFO errors above the first zero, there is a mix of positive and negative values, which indicates that the direction estimate will be erroneous. Below 10 kHz the values are however all positive, suggesting that estimates will remain correct. The equivalent for the triple repetition $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$ is shown in Fig. 5.11. This shows a different behavior above the first zero at $f = 6944$Hz, where all magnitudes are negative, which will then again yield a correct estimation result. This is the case for most of the measurements until the second zero at 13889 Hz, when $f \cdot t_p = \frac{2}{3}$.

For the case $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r \end{bmatrix}$, which is illustrated in Fig. 5.12, the f where the first antenna element has its aggregated measurements reverse its direction occurs at a lower frequency. For $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r & \boldsymbol{x}_r \end{bmatrix}$ it is even lower, as shown in Fig. 5.13. For both of these there are CFO error ranges where most or all antennas have a aggregated magnitude with the same sign, for example the range 10-13 kHz in Fig. 5.13.

By using the structure $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$, but sampling every element two sampling slots in a row with additional measurements gained from the switching slots where no

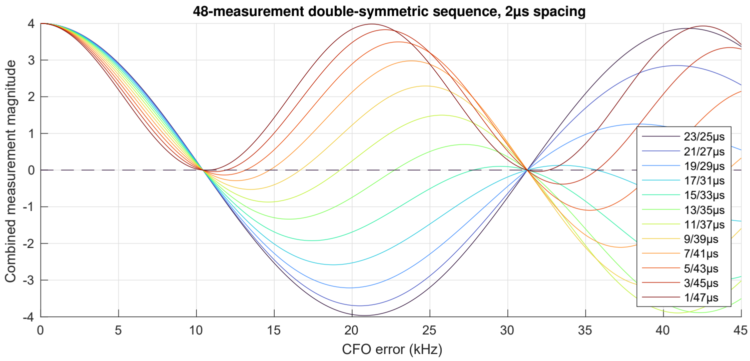


Figure 5.10: Magnitude of aggregated measurements for sampling order structure $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$.

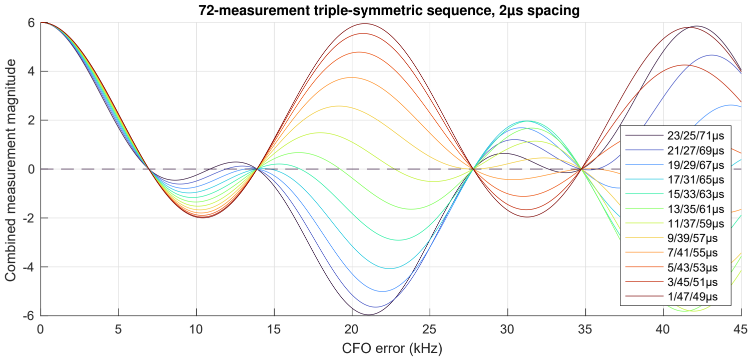


Figure 5.11: Magnitude of aggregated measurements for sampling order structure $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$.

switching occurs, the aggregated magnitudes are as shown in Fig. 5.14. The result is similar to Fig. 5.6, but compressed to lower frequencies. Extending this to using three samples in a row yields similar results at even lower frequencies, shown in Fig. 5.15.

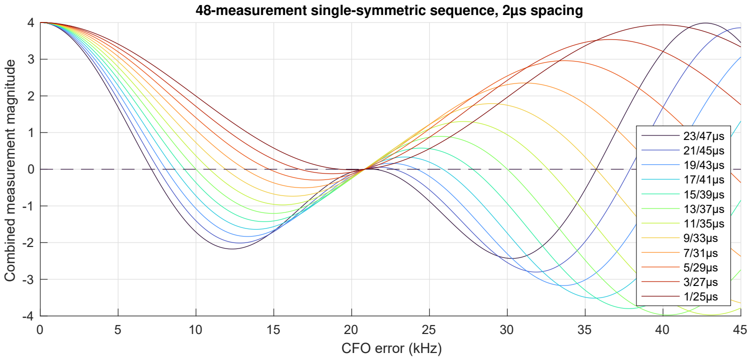


Figure 5.12: Magnitude of aggregated measurements for sampling order structure $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r \end{bmatrix}$.

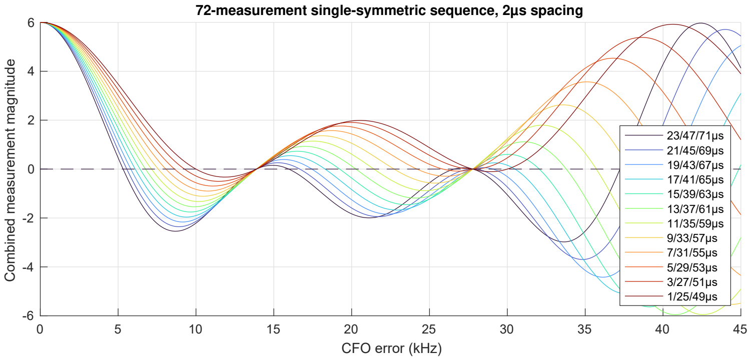


Figure 5.13: Magnitude of aggregated measurements for sampling order structure $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r & \boldsymbol{x}_r \end{bmatrix}$.

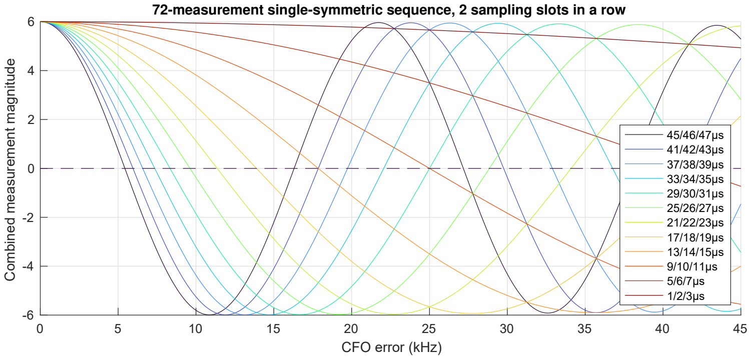


Figure 5.14: Magnitude of aggregated measurements for single-symmetric sampling order where the same element is sampled in 2 consecutive sampling slots, and therefore also in the switching slots where no switching occurs.

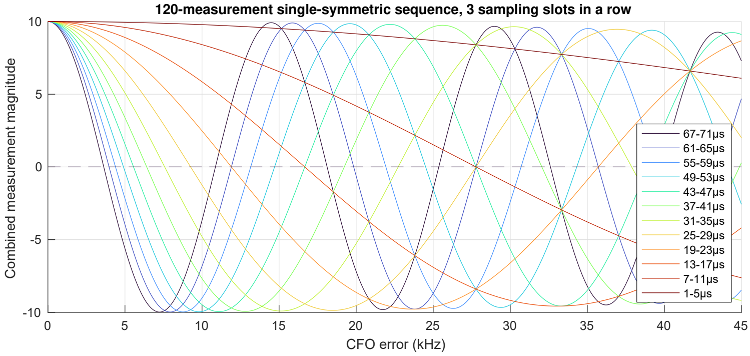


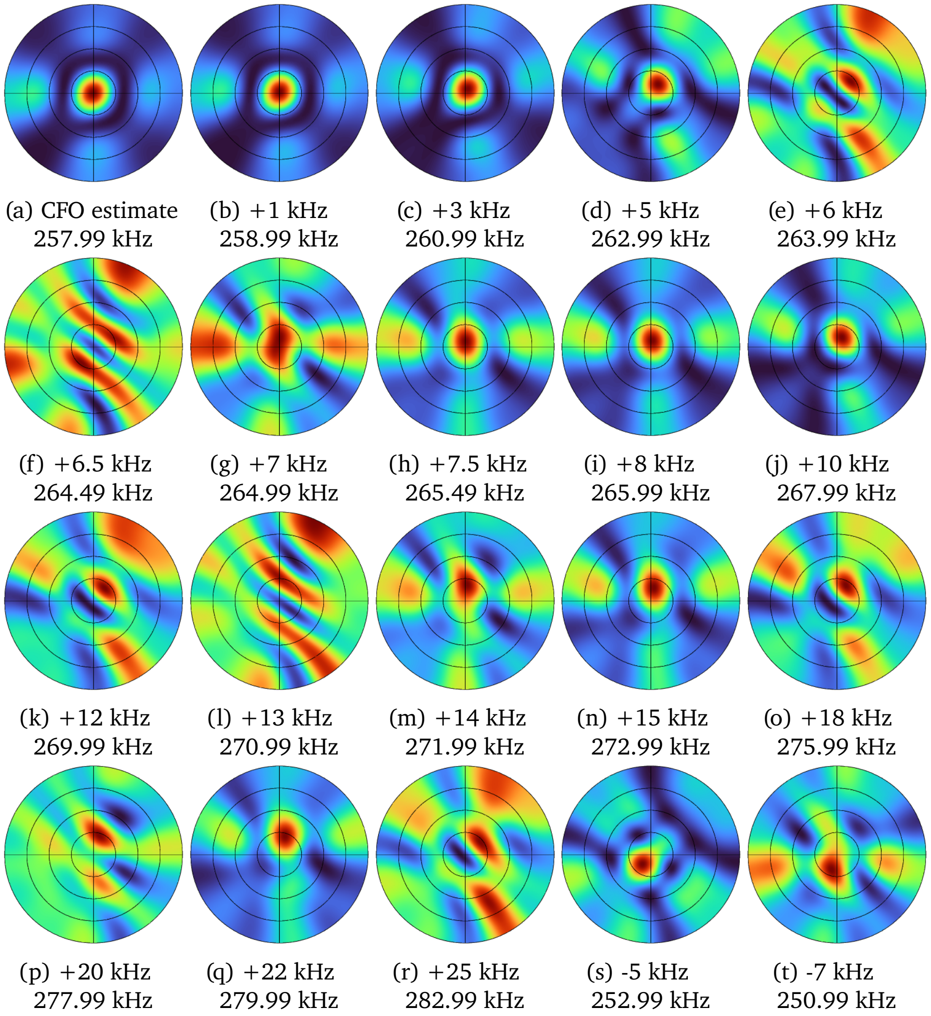
Figure 5.15: Magnitude of aggregated measurements for single-symmetric sampling order where the same element is sampled in 2 consecutive sampling slots, and therefore also in the switching slots where no switching occurs.

5.4 Experimental verification

To verify that the considerations for the ideal conditions assumed in Section 5.2 and Section 5.3 also hold under realistic conditions, data was collected indoors using a Nordic Semiconductor experimental reference design Bluetooth antenna array using different sampling sequences. The array, with 12 elements in a 15x15cm square configuration, was set up oriented as in Fig. 5.1. The array and transmitter were positioned to yield a direction estimate along the array boresight. For a set of measurements from a single CTE for each sample order, the CFO was estimated using all samples. Spatial spectra as in (5.1) were then generated by using the estimated CFO with different frequency offsets added.

As a baseline, Fig. 5.16 shows the results for a non-time-symmetric sampling order, consisting of 8 reference measurements from antenna 11 and then cyclic sampling in the order 12,1,2,10,3,9,4,8,7,6,5 until the end of the CTE, for a total of 82 measurements. The spectra covers a half sphere in front of the array and are shown from the array's perspective. This means that the top of the spectra points towards the zenith, the center is the boresight direction, and the right edge corresponds to a signal direction from the left side in Fig. 5.1 where the array is seen from the front. Even for the small errors of 3 kHz (Fig. 5.16c) and 5 kHz (Fig. 5.16d) it can be seen how the direction estimate is affected by the CFO error, and that error patterns appear which repeat for several CFO errors, such as for 6.5 kHz (Fig. 5.16f) and 13 kHz (Fig. 5.16l). For the remainder of the figures, where time-symmetric orders were used, the reference samples are not included in the direction estimation, and 24, 48 or 72 of the measurements are used depending on the desired sampling sequence.

In Fig. 5.17, 24 measurements are used with the time-symmetric sampling sequence from Fig. 5.2. In this case the peak at the center starts deforming at 12 kHz error (Fig. 5.17k) until splitting into two peaks from 15 kHz error (Fig. 5.17n). From Fig. 5.6 no error would be expected for CFO errors below 10 kHz, which matches the observed behavior. The angular resolution of the array results in a minimum direction separation required for distinct peaks to appear in the spatial spectrum. The low angular resolution of a 15x15cm Bluetooth array increases the CFO error required for the peak to separate compared to that of a larger array. In Fig. 5.18 the same sequence is repeated once more, for a total of 48 samples. The result is almost identical to that in Fig. 5.17, which is reasonable given that Fig. 5.10 shows all positive values until approximately 10 kHz and mixed signs for the higher CFO errors tested, which was also the case in Fig. 5.6. In Fig. 5.19 the sampling sequence is extended again to three repetitions of the time-symmetric sequence, with a total of 72 samples used. In this case the result is very different, with the the CFO errors 7 kHz (Fig. 5.19g) and 14 kHz (Fig. 5.19m) matching close enough with the zeros at 6944 Hz and 13889 Hz in Fig. 5.11 to yield spectra significantly affected by measurement noise. As expected, the result for 7.5 kHz (Fig. 5.19h), 8 kHz (Fig. 5.19i) and 10 kHz (Fig. 5.19j) CFO error all appear unaffected as the aggregated measurement magnitudes for all antennas are negative in Fig. 5.11. Fig. 5.20 uses the same symmetric sequence structure, but using a different order

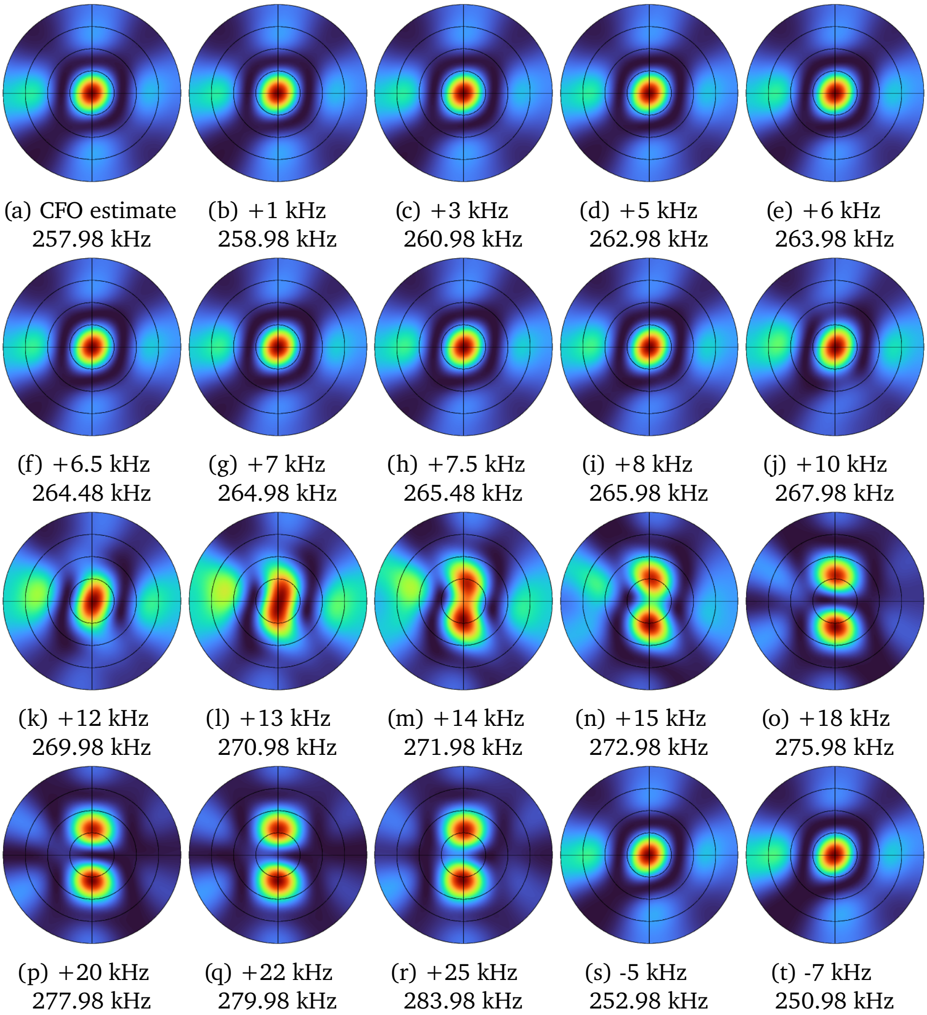


(a) CFO estimate 257.99 kHz
(b) +1 kHz 258.99 kHz
(c) +3 kHz 260.99 kHz
(d) +5 kHz 262.99 kHz
(e) +6 kHz 263.99 kHz
(f) +6.5 kHz 264.49 kHz
(g) +7 kHz 264.99 kHz
(h) +7.5 kHz 265.49 kHz
(i) +8 kHz 265.99 kHz
(j) +10 kHz 267.99 kHz
(k) +12 kHz 269.99 kHz
(l) +13 kHz 270.99 kHz
(m) +14 kHz 271.99 kHz
(n) +15 kHz 272.99 kHz
(o) +18 kHz 275.99 kHz
(p) +20 kHz 277.99 kHz
(q) +22 kHz 279.99 kHz
(r) +25 kHz 282.99 kHz
(s) -5 kHz 252.99 kHz
(t) -7 kHz 250.99 kHz

Figure 5.16: Spatial pseudo-spectrum for different CFO errors with default non-symmetric order.

that does not scan across the array. This creates very different patterns for the 15 kHz (Fig. 5.20n) to 25 kHz (Fig. 5.20r) CFO errors.

Fig. 5.21 shows the result from using the longer symmetric combination $\begin{bmatrix}\boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r\end{bmatrix}$ that is not the combination of shorter symmetric sequences. Deformation of the central peak is visible already for 8 kHz error (Fig. 5.21i), which is reasonable from the point at which signs start to change in Fig. 5.12. Both 20 kHz (Fig. 5.21p) and 22 kHz (Fig. 5.21q) appear to be far enough away from the zero between these frequencies to yield spectra with a clear peak, although the peak for 20 kHz is shifted slightly away from the correct direction. Fig. 5.20 show the result for

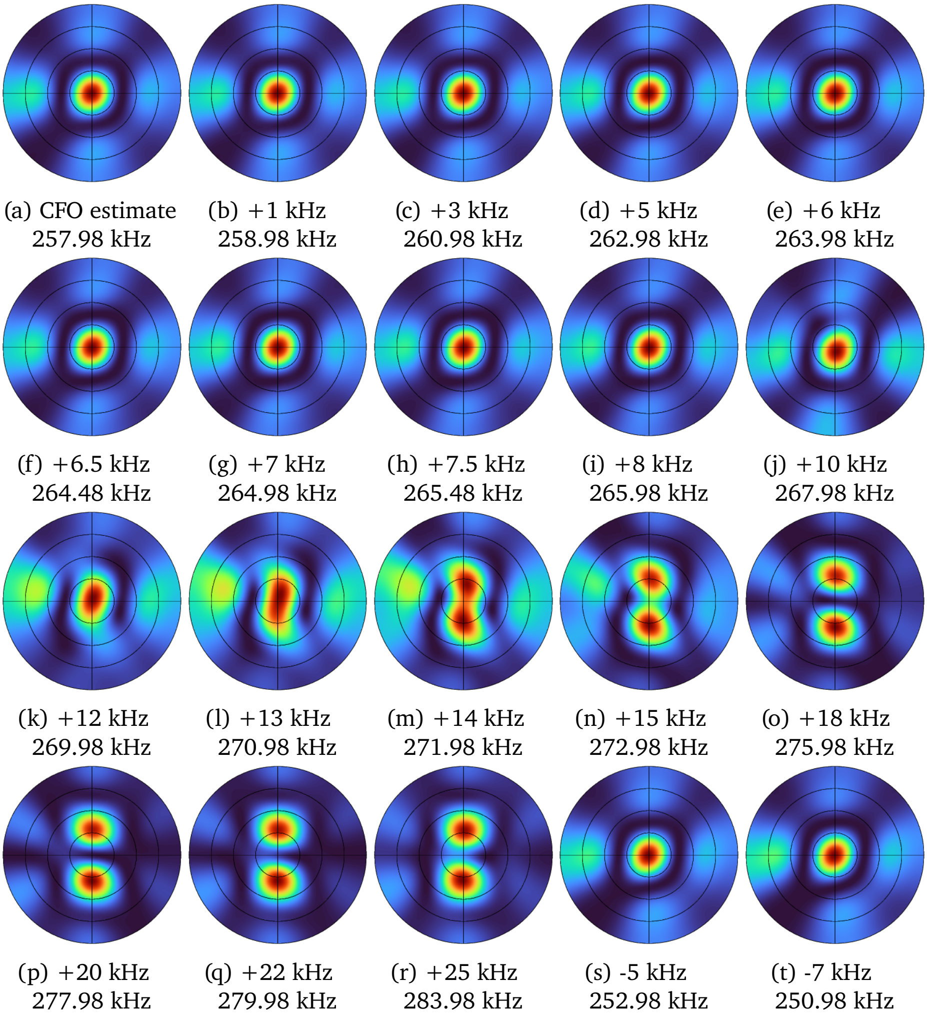


(a) CFO estimate 257.98 kHz
(b) +1 kHz 258.98 kHz
(c) +3 kHz 260.98 kHz
(d) +5 kHz 262.98 kHz
(e) +6 kHz 263.98 kHz
(f) +6.5 kHz 264.48 kHz
(g) +7 kHz 264.98 kHz
(h) +7.5 kHz 265.48 kHz
(i) +8 kHz 265.98 kHz
(j) +10 kHz 267.98 kHz
(k) +12 kHz 269.98 kHz
(l) +13 kHz 270.98 kHz
(m) +14 kHz 271.98 kHz
(n) +15 kHz 272.98 kHz
(o) +18 kHz 275.98 kHz
(p) +20 kHz 277.98 kHz
(q) +22 kHz 279.98 kHz
(r) +25 kHz 283.98 kHz
(s) -5 kHz 252.98 kHz
(t) -7 kHz 250.98 kHz

Figure 5.17: Spatial pseudo-spectrum for different CFO errors with mirrored direction, $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$, $\boldsymbol{x} = [11, 12, 1, 2, 10, 3, 9, 4, 8, 7, 6, 5]$.

the similar but even longer sequence with the combination $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r & \boldsymbol{x}_r \end{bmatrix}$. The spectrum peak is starting to split in two already at 6 kHz error (Fig. 5.22e) in this case, matching well with Fig. 5.13.

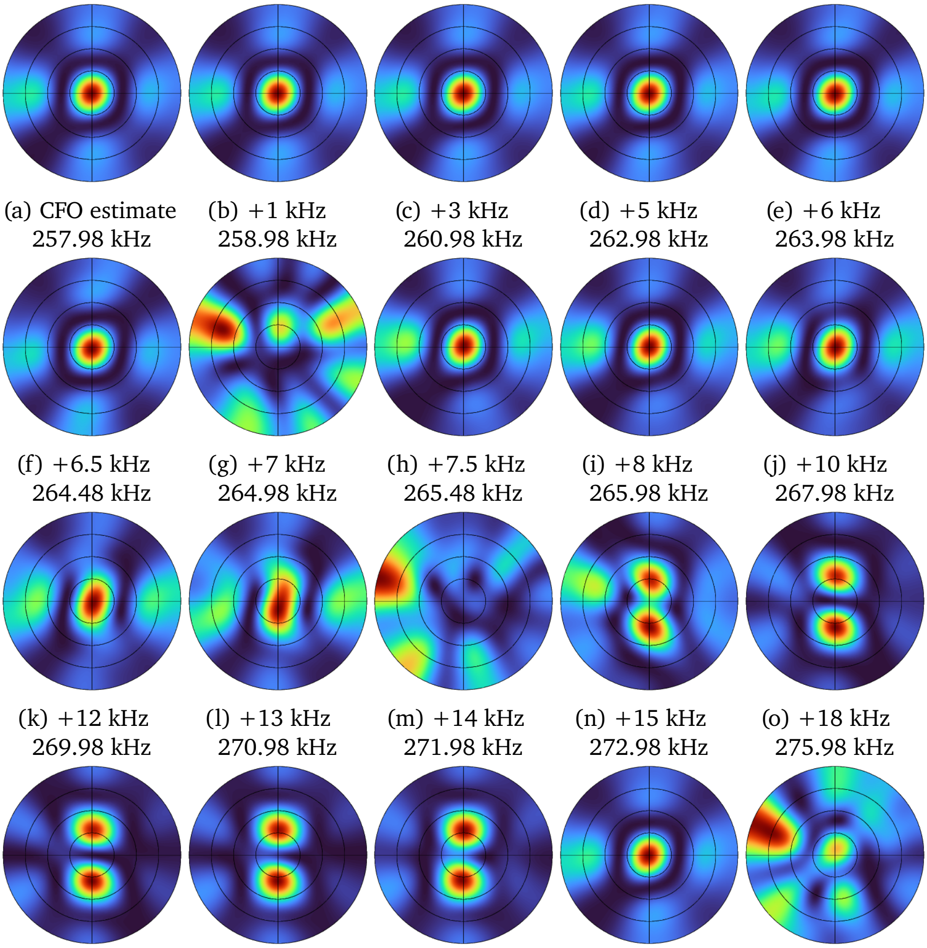
To see the effect of the sampling order in an outdoor direction estimation scenario, measurements using two different sampling orders were collected from an array on the ground and a transmitter attached to a multirotor UAV. The array setup is shown in Fig. 5.23, where the lower array was used in the experiment. Two different CFO estimates were used for measurement correction: one utilizing all 82 measurements available for each CTE, and another using only the 8 reference measurements as



(a) CFO estimate 257.98 kHz
(b) +1 kHz 258.98 kHz
(c) +3 kHz 260.98 kHz
(d) +5 kHz 262.98 kHz
(e) +6 kHz 263.98 kHz
(f) +6.5 kHz 264.48 kHz
(g) +7 kHz 264.98 kHz
(h) +7.5 kHz 265.48 kHz
(i) +8 kHz 265.98 kHz
(j) +10 kHz 267.98 kHz
(k) +12 kHz 269.98 kHz
(l) +13 kHz 270.98 kHz
(m) +14 kHz 271.98 kHz
(n) +15 kHz 272.98 kHz
(o) +18 kHz 275.98 kHz
(p) +20 kHz 277.98 kHz
(q) +22 kHz 279.98 kHz
(r) +25 kHz 283.98 kHz
(s) -5 kHz 252.98 kHz
(t) -7 kHz 250.98 kHz

Figure 5.18: Spatial pseudo-spectrum for different CFO errors with mirrored direction, $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$, $\boldsymbol{x} = [11, 12, 1, 2, 10, 3, 9, 4, 8, 7, 6, 5]$.

often suggested [168, 162]. Fig. 5.24 shows the result for the default non-symmetric sampling order as used in Fig. 5.16. It is clear the CFO estimate using only reference measurements is much noisier than the CFO estimate using all measurements. Since the array is diagonally mounted, the approximately 45° error direction for small CFO errors as visible for the 3-6 kHz errors in Fig. 5.16 now points along the vertical. It can clearly be seen that the elevation estimate for the reference CFO is very noisy. Both the elevation and azimuth angles show spikes, which is expected from the pseudospectrum for CFO errors such as 6.5 kHz and 13 kHz in Fig. 5.16. Fig. 5.25 shows the result from using the sampling order from Fig. 5.19. In this case the noise in the elevation estimate when using the reference CFO is comparable to



(a) CFO estimate 257.98 kHz

(b) +1 kHz 258.98 kHz

(c) +3 kHz 260.98 kHz

(d) +5 kHz 262.98 kHz

(e) +6 kHz 263.98 kHz

(f) +6.5 kHz 264.48 kHz

(g) +7 kHz 264.98 kHz

(h) +7.5 kHz 265.48 kHz

(i) +8 kHz 265.98 kHz

(j) +10 kHz 267.98 kHz

(k) +12 kHz 269.98 kHz

(l) +13 kHz 270.98 kHz

(m) +14 kHz 271.98 kHz

(n) +15 kHz 272.98 kHz

(o) +18 kHz 275.98 kHz

(p) +20 kHz 277.98 kHz

(q) +22 kHz 279.98 kHz

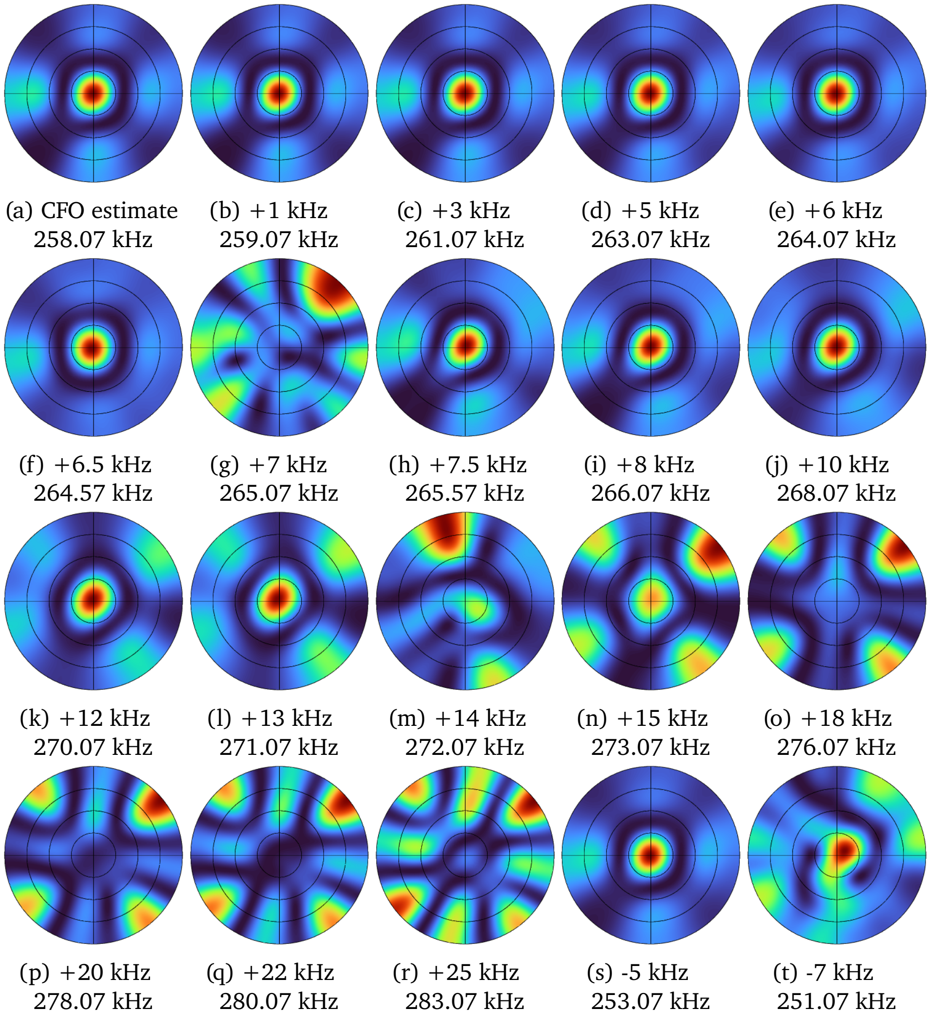
(r) +25 kHz 283.98 kHz

(s) -5 kHz 252.98 kHz

(t) -7 kHz 250.98 kHz

Figure 5.19: Spatial pseudo-spectrum for different CFO errors with mirrored direction, $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r \end{bmatrix}$, $\boldsymbol{x} = [11, 12, 1, 2, 10, 3, 9, 4, 8, 7, 6, 5]$.

that using the better CFO estimate, but spikes still appear. The apparently random spectrum for 7 kHz and 14 kHz errors in Fig. 5.19, close to the expected zeros in the combined measurement magnitude, are expected to appear due to the error levels in the reference CFO estimate.



(a) CFO estimate 258.07 kHz

(b) +1 kHz 259.07 kHz

(c) +3 kHz 261.07 kHz

(d) +5 kHz 263.07 kHz

(e) +6 kHz 264.07 kHz

(f) +6.5 kHz 264.57 kHz

(g) +7 kHz 265.07 kHz

(h) +7.5 kHz 265.57 kHz

(i) +8 kHz 266.07 kHz

(j) +10 kHz 268.07 kHz

(k) +12 kHz 270.07 kHz

(l) +13 kHz 271.07 kHz

(m) +14 kHz 272.07 kHz

(n) +15 kHz 273.07 kHz

(o) +18 kHz 276.07 kHz

(p) +20 kHz 278.07 kHz

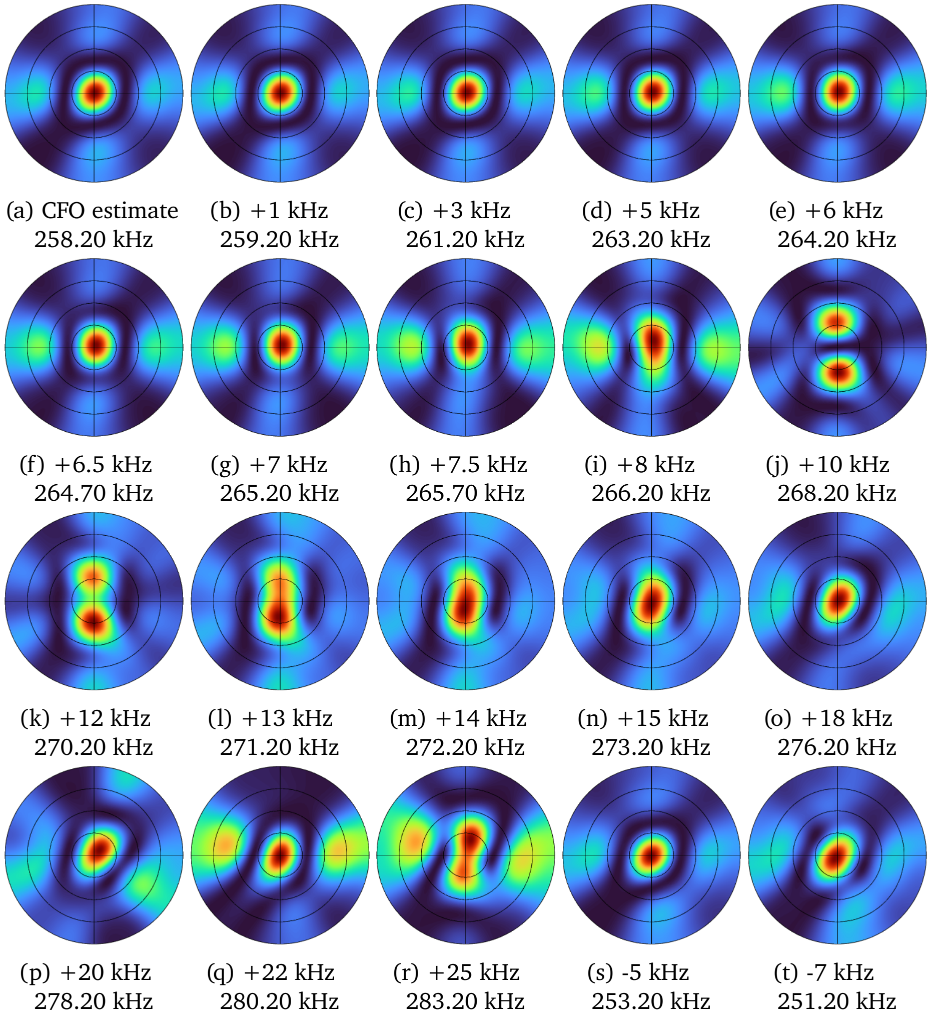
(q) +22 kHz 280.07 kHz

(r) +25 kHz 283.07 kHz

(s) -5 kHz 253.07 kHz

(t) -7 kHz 251.07 kHz

Figure 5.20: Spatial pseudo-spectrum for different CFO errors with mirrored direction $\begin{bmatrix}\boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x} & \boldsymbol{x}_r\end{bmatrix}$, $\boldsymbol{x} = [11, 5, 8, 2, 7, 10, 1, 4, 9, 12, 3, 6]$.



(a) CFO estimate 258.20 kHz

(b) +1 kHz 259.20 kHz

(c) +3 kHz 261.20 kHz

(d) +5 kHz 263.20 kHz

(e) +6 kHz 264.20 kHz

(f) +6.5 kHz 264.70 kHz

(g) +7 kHz 265.20 kHz

(h) +7.5 kHz 265.70 kHz

(i) +8 kHz 266.20 kHz

(j) +10 kHz 268.20 kHz

(k) +12 kHz 270.20 kHz

(l) +13 kHz 271.20 kHz

(m) +14 kHz 272.20 kHz

(n) +15 kHz 273.20 kHz

(o) +18 kHz 276.20 kHz

(p) +20 kHz 278.20 kHz

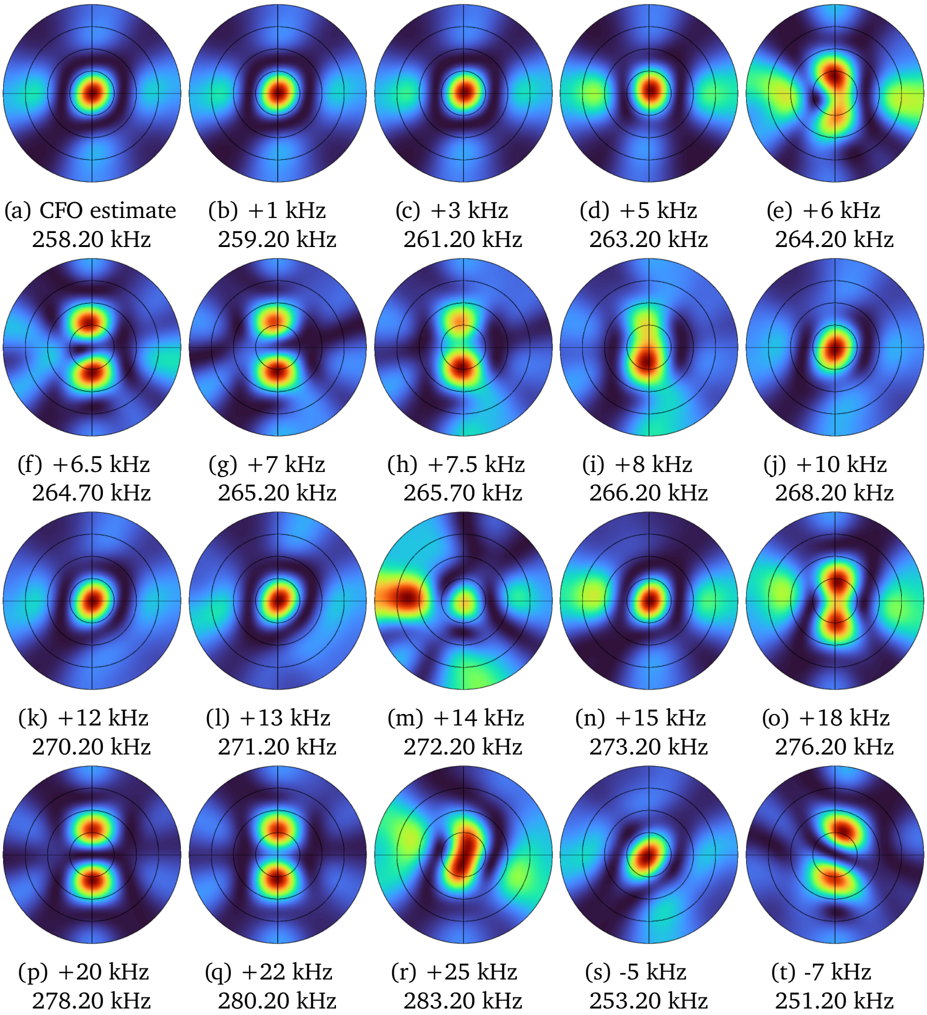
(q) +22 kHz 280.20 kHz

(r) +25 kHz 283.20 kHz

(s) -5 kHz 253.20 kHz

(t) -7 kHz 251.20 kHz

Figure 5.21: Spatial pseudo-spectrum for different CFO errors with mirrored direction, $\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x}_r & \boldsymbol{x}_r \end{bmatrix}$, $\boldsymbol{x} = [11, 12, 1, 2, 10, 3, 9, 4, 8, 7, 6, 5]$.



(a) CFO estimate 258.20 kHz

(b) +1 kHz 259.20 kHz

(c) +3 kHz 261.20 kHz

(d) +5 kHz 263.20 kHz

(e) +6 kHz 264.20 kHz

(f) +6.5 kHz 264.70 kHz

(g) +7 kHz 265.20 kHz

(h) +7.5 kHz 265.70 kHz

(i) +8 kHz 266.20 kHz

(j) +10 kHz 268.20 kHz

(k) +12 kHz 270.20 kHz

(l) +13 kHz 271.20 kHz

(m) +14 kHz 272.20 kHz

(n) +15 kHz 273.20 kHz

(o) +18 kHz 276.20 kHz

(p) +20 kHz 278.20 kHz

(q) +22 kHz 280.20 kHz

(r) +25 kHz 283.20 kHz

(s) -5 kHz 253.20 kHz

(t) -7 kHz 251.20 kHz

Figure 5.22: Spatial pseudo-spectrum for different CFO errors with mirrored direction, $\left[\boldsymbol{x} \quad \boldsymbol{x} \quad \boldsymbol{x} \quad \boldsymbol{x}_r \quad \boldsymbol{x}_r \quad \boldsymbol{x}_r\right]$, $\boldsymbol{x} = [11, 12, 1, 2, 10, 3, 9, 4, 8, 7, 6, 5]$.

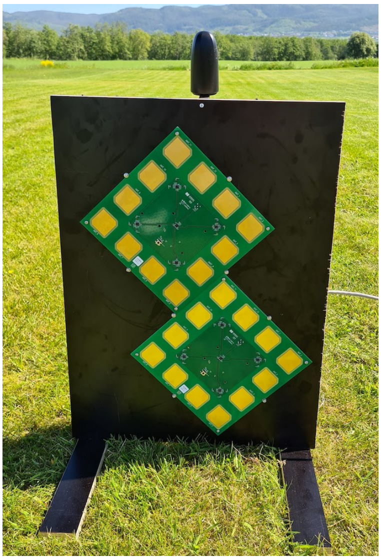


Figure 5.23: Array setup. Data from the lower array was used in the outdoor experiment. Element 11 as per Fig. 5.1 is highest above ground, and element 5 is the lowest.

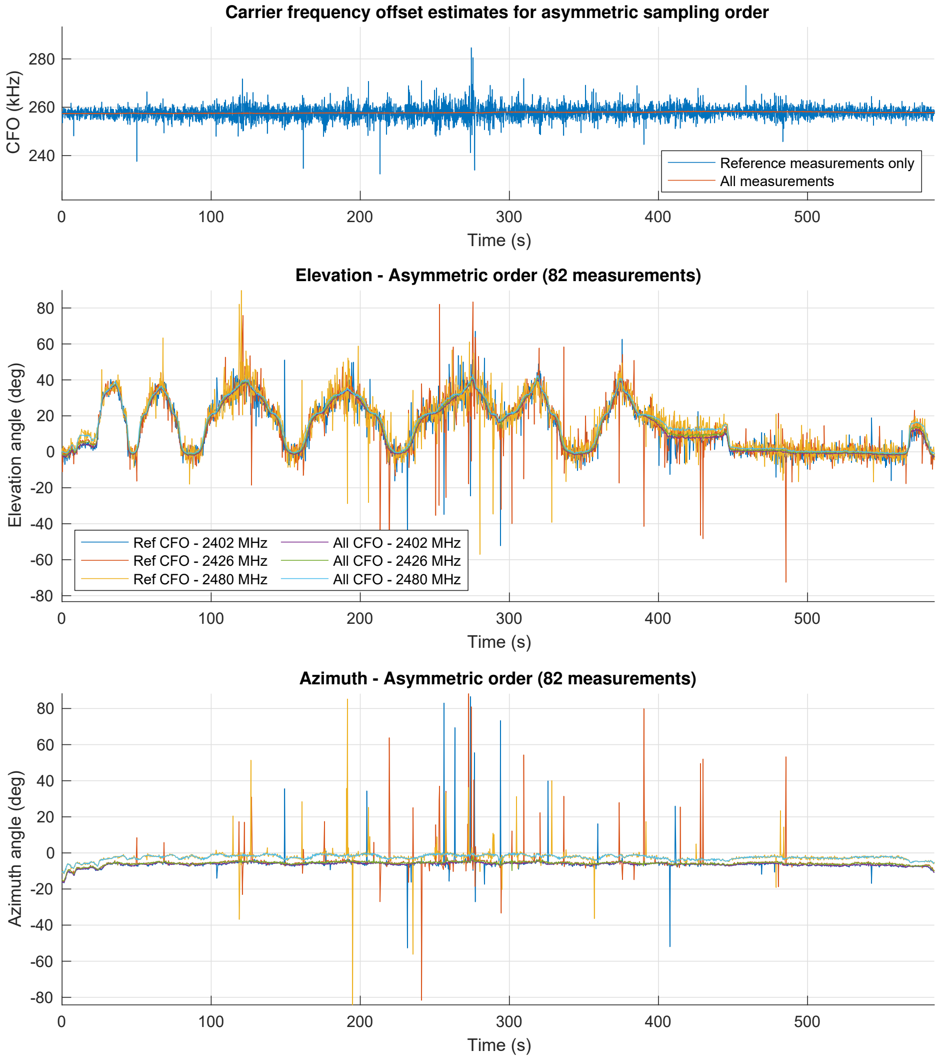


Figure 5.24: Estimates of CFO and direction for a moving transmitter, using the default non-symmetric sampling order. Direction was estimated using both a CFO estimate using all available measurements (82) and only the reference samples (8).

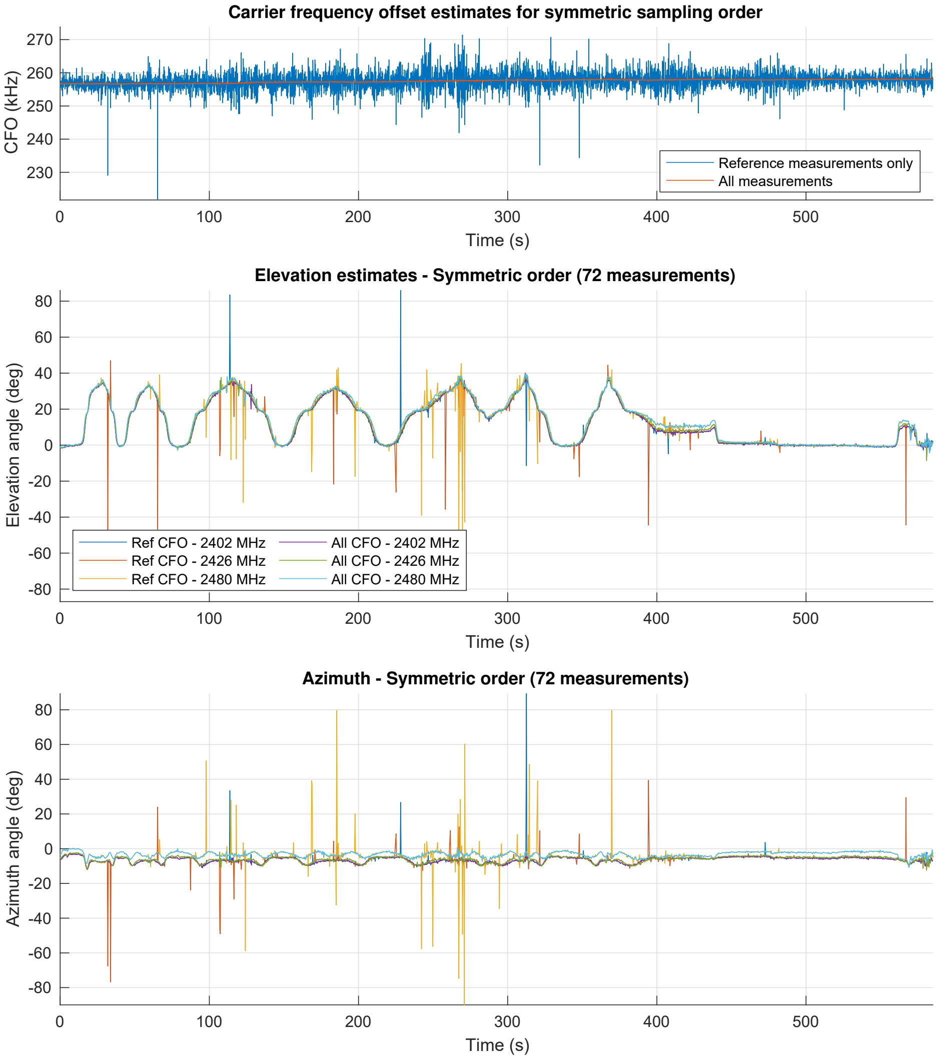


Figure 5.25: Estimates of CFO and direction for a moving transmitter, using the sampling order from Fig. 5.19. Direction was estimated using both a CFO estimate using all available measurements (82) and only the reference samples (8).

5.5 Conclusion

In this paper, the use of time-symmetric sampling sequences for the single-receiver switched sampling of the Bluetooth angle-of-arrival feature has been discussed and experimentally tested. Based on the analysis performed, it is clear that the considered time-symmetric sampling sequences are not able to avoid error in the direction estimate for the entire range of frequency offsets possible using Bluetooth, with allowed average modulation deviation in the 225-275 kHz range [15]. It is also clear that the cancellation of signal power to reduce the correlation between CFO error and direction estimation error comes at the cost of reducing the signal-to-noise ratio. Therefore obtaining the best possible CFO estimate is still desirable. For the effect of CFO to cancel completely under realistic conditions, the magnitude of measurements from each array antenna must be equal. Any inaccuracies in timing, e.g. due to clock drift during the CTE, will be a source of error. Despite this, it appears that a time-symmetrical sampling order can be beneficial in reducing the effect of residual CFO estimation errors on the direction estimate.

Chapter 6

Reducing Ground Reflection Multipath Errors for Bluetooth Angle-of-Arrival Estimation by Combining Independent Antenna Arrays

This chapter is based on the publication

- [134] M. L. Sollie, T. A. Johansen, K. Gryte, and T. H. Bryne, “Reducing Ground Reflection Multipath Errors for Bluetooth Angle-of-Arrival Estimation by Combining Independent Antenna Arrays,” *Submitted, in revision,* 2022

and considers the use of two antenna arrays to reduce the multipath error observed in Chapter 4.

6.1 Introduction

Multipath interference from ground reflections is a known problem for elevation determination in radar tracking of aircraft at low elevation angles [12, 160, 86]. This problem "has no simple solution and is generally minimized by using narrow-beam antennas" [125, p. 9.38], i.e. antenna reflectors or antenna arrays with large apertures and/or operating at high frequencies, yielding high angular resolution. The multipath effect also occurs for other direction finding systems, such as Bluetooth angle-of-arrival (AoA) determination. For Bluetooth the problem can be significant due to the use of arrays that are much smaller than those typically used for aircraft tracking radar, resulting in low angular resolution. Both the direct signal and the ground reflection can then be within the array main lobe for elevation angles of tens of degrees. The elevation error can be significant up to an elevation angle approximately equal to the one-way beamwidth of the array [125, Fig. 9.28], which corresponds to the Rayleigh resolution limit [152] for angular separation. Direction estimation algorithms such as beamformers [71, 152], Multiple Signal Classification (MUSIC) [117], or Estimation of Signal Parameters via Rotational Invariance Technique (ESPIRIT) [112] are not able to separate the direct and reflected signals unless they have enough angular separation, especially for coherent signals.

Modeling the reflection and taking it into account in the direction estimation is possible, but requires knowledge or assumptions on the reflection geometry, signal polarization and reflection coefficient. This was considered in [3], which assumed a flat surface and a known reflection coefficient using horizontal polarization. A vertical linear array with 20 elements and 1.5m element spacing was considered, yielding a large vertical aperture. The approach to model the reflection is not ideal if the reflection coefficient is unknown, the ground surface cannot be assumed flat (such as being uneven or having an unknown slope), or the reflection polarization depends on the angle of incidence, such as for circular polarization [138]. This is especially the case for portable equipment, which can be moved between different locations and therefore be subject to different reflection conditions.

Instead of designing and producing a single large array, an alternative method is to synchronize measurements from multiple separate arrays, and process the measurements together. This way, low-cost equipment can be assembled in a modular fashion. A challenge of using multiple separate receivers is that they each use their own reference oscillator for IQ sampling. This means that measurements cannot directly be processed as if they are the output from a single receiver, as the oscillators can run at slightly different frequencies and can not at any specific time be expected to have the same phase angle. In [18], synchronization of multiple independent sub-arrays for the Internet-of-Things DASH7 Alliance Protocol was demonstrated, increasing the accuracy of signal direction determination. Optimization was used for synchronization of signal frequency, timing and phase, with separate receiver channels used for each receiving antenna in the sub-arrays. Multipath interference was not considered, and experimental verification was performed only with a linear array for azimuth angle determination, thus not encountering the multipath elevation problem.

In the field experiments using Bluetooth AoA in Chapter 4, it was found that elevation estimation errors due to ground reflection multipath was the main error source at elevation angles below approximately 25°, using a $15 \times 15\,\mathrm{cm}$ array. The error depends on the properties of the ground material, including moisture levels which can vary over time. For an array placed high enough above the ground, multiple true elevation angles can map to the same elevation angle measurement. Due to this non-uniqueness of measurements, the effect is not easily removed by a calibration in all situations.

Synchronizing smaller, low-cost arrays, combined in a modular setup using vertical separation can increase the angular resolution of the combined elevation estimate, and thereby reduce the multipath effect. This way the error can be reduced without modeling or requiring knowledge of the reflection conditions. Compared to using a single large Bluetooth array, two arrays, each with its own receiver, will provide twice as many samples, and sample elements from both arrays in parallel. For a single large array with double the number of elements, we would get half the number of measurements from each element due to the maximum length of the signal sampled.

With the use of a single transmitter and two independent arrays, both arrays will receive the same Bluetooth packets with the same Constant Tone Extension (CTE). The use of independent receiver hardware gives rise to several sources of timing differences: Since the distance from the transmitter to each array is different, one array may receive the packet with a very small delay of approximately 3ns per meter compared to the other. Digital sampling occurring synchronously with the hardware clock will not be synchronized between arrays. Since each reference oscillator can run at slightly different frequencies, the frequency offset from baseband of the measurements may be different for each receiver. The difference in clock frequencies also causes the interval between the measurements to be different, since each receiver has a different understanding of the passage of time, with each receiver drifting relative to the other. After compensating the frequency offset for each array to transform measurements to baseband, measurements from the same antenna of the same array at different times during the CTE have approximately the same phase angle. After taking the difference in position of each array into account, this means that measurements from one array will have a phase bias compared to the other, which must be compensated to process the measurements together. Due to the short 160µs duration of the CTE used for direction determination, the direction from each array to the transmitter is assumed constant during the sampling of a single CTE.

6.1.1 Main contribution

The main contribution of this paper is the proposition and testing of a method to use two independent arrays together for direction estimation, in the same way as a single larger array. We demonstrate that a simple synchronization procedure can significantly improve the elevation angle estimates, and verify this in field experiments using two Bluetooth antenna arrays.

6.1.2 Organization

The paper proceeds as follows: In Section 6.2 we discuss the positioning of two arrays to enable phase synchronization in the presence of ground reflection multipath. Section 6.3 presents the measurement processing and direction estimation method, and presents simulation results for predicted results. Results from a field experiment are presented in Section 6.4, and finally, conclusions are drawn in Section 6.5.

6.2 Array positioning for synchronization in the presence of multipath

Large spacing between arrays would be beneficial for multipath error reduction. In the absence of multipath, receiving only a signal directly propagating from the transmitter to all array elements, finding the phase bias between the arrays would be straightforward, allowing easy synchronization. Consider for example the vertically spaced arrays in Fig. 6.1. The phase angle difference between antennas placed along a straight line would differ only due to differences in the distance along the line. Therefore, for arrays with element spacing d and vertical phase differences φ, the deviation between the phase difference measured across the gap d_g between arrays and the expected phase difference $\frac{d_g}{d}\varphi$ for the element gap would be the bias in the phase between arrays.

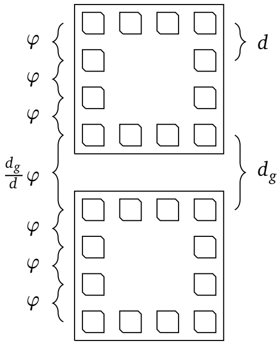


Figure 6.1: Vertically stacked arrays with gap. Without the effect of multipath, the uniformly spaced antenna pairs would measure the same phase difference, allowing the inter-array phase-bias to be determined.

In the presence of multipath, the measured phase difference between vertically uniformly spaced antennas is not equal but depends on the distance from the reflection surface, which in the following is assumed to be the ground. This effect can, in theory, be modeled, but in practice, the ground surface is often uneven, not perfectly level, and the height over the surface may not be exactly known. Determining the phase bias is therefore no longer trivial with a gap between the arrays. With a high number of antenna elements with a small spacing, the phase difference between pairs along the vertical would be smoothly varying, and it could be possible to predict the phase difference across the array gap by using curve fitting. The accuracy of this would be reduced as the gap between arrays increases.

By eliminating the gap between arrays, with the arrays overlapping in height with at least one element, multipath should no longer be an issue in determining the phase bias. Fig. 6.3 and Fig. 6.3 illustrate such setups. The elements at the same height for both arrays can be expected to be affected equally by ground reflection multipath interference. The difference in phase measurement between these after

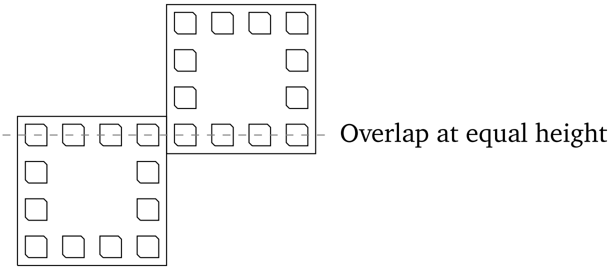


Figure 6.2: Stacked arrays with one row of elements overlapping in height.

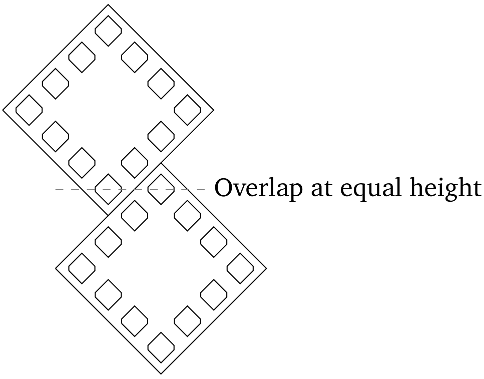


Figure 6.3: Vertically stacked diagonal arrays with one element overlapping in height.

azimuth angle correction can therefore provide information about the reference phase difference between the arrays, without needing to model the interference with the reflected signal.

6.3 Measurement processing and direction estimation

The method considered for AoA estimation for an in-phase and quadrature (IQ) complex measurement vector $\boldsymbol{x}$ from a single array, is to find the direction parameters Ψ, α maximizing the conventional beamformer [71, 152] spatial pseudo-spectrum

$$P(\Psi,\alpha) = |\boldsymbol{a}(\Psi,\alpha)^H \boldsymbol{x}|^2, \tag{6.1}$$

where $|\cdot|$ denotes complex magnitude and $(\cdot)^H$ denotes the conjugate transpose, with the steering vector

$$\boldsymbol{a}(\Psi,\alpha) = e^{\frac{2\pi j}{\lambda}\boldsymbol{P}^{a\top}\boldsymbol{l}^a(\Psi,\alpha)}. \tag{6.2}$$

λ is the signal wavelength, $\boldsymbol{P}^a$ is the matrix of array element positions in the array frame $\{a\}$, and $\boldsymbol{l}^a(\Psi,\alpha)$ is the line-of-sight vector corresponding to the direction parameters Ψ,α. For two arrays it is assumed that both sample the Constant Tone Extension (CTE) of the same Bluetooth packets, meaning that they sample at nearly identical times and using the same frequency channel. It is important to distinguish between only combining the signal power from both arrays, and combining the measurements as if we have a single large array with increased size. The former can be done by simply averaging the direction result from each array, or by combining the spectrum from each array,

$$P(\Psi,\alpha) = |\boldsymbol{a}_1(\Psi,\alpha)^H \boldsymbol{x}_1|^2 + |\boldsymbol{a}_2(\Psi,\alpha)^H \boldsymbol{x}_2|^2, \tag{6.3}$$

and finding the new combined peak. Since the phase angle of $\boldsymbol{a}_1(\Psi,\alpha)^H \boldsymbol{x}_1$ and $\boldsymbol{a}_2(\Psi,\alpha)^H \boldsymbol{x}_2$ do not influence the spectrum values, the relative positions of the arrays make no difference in the processing, and the element positions in each steering vector are independent and does not have to use the same origin. This essentially results in a weighted average of the two independent estimates, by combining the spectra of each, and does not result in increased angular resolution in any direction, as would be the case for an array of increased size. The multipath error is not reduced in the way processing the measurements as one large array would. The goal of processing the measurements together is to produce a better estimate than what can be obtained by combining the direction estimates from each array.

To use the measurements together for estimation, the relative position of the arrays should be known with high accuracy and precision. The steering vector for each array, including phase offsets φ_1 and φ_2, can be formulated as

$$\boldsymbol{a}_1(\Psi,\alpha,\varphi_1) = e^{\frac{2\pi j}{\lambda}\boldsymbol{P}_1^{a\top}\boldsymbol{l}^a(\Psi,\alpha)+\varphi_1 j}, \tag{6.4}$$

$$\boldsymbol{a}_2(\Psi,\alpha,\varphi_2) = e^{\frac{2\pi j}{\lambda}\boldsymbol{P}_2^{a\top}\boldsymbol{l}^a(\Psi,\alpha)+\varphi_2 j}, \tag{6.5}$$

where $\boldsymbol{P}_1^a$ and $\boldsymbol{P}_2^a$ should have the same origin. Considering only the phase bias of one array relative to the other, the steering vectors can instead be formulated as

$$\boldsymbol{a}_1(\Psi,\alpha) = \boldsymbol{a}_1(\Psi,\alpha,0) = e^{\frac{2\pi j}{\lambda}\boldsymbol{P}_1^{a\top}\boldsymbol{l}^a(\Psi,\alpha)}, \tag{6.6}$$

$$\boldsymbol{a}_2(\Psi,\alpha,\Delta\varphi) = e^{\frac{2\pi j}{\lambda}\boldsymbol{P}_2^{a\top}\boldsymbol{l}^a(\Psi,\alpha)+\Delta\varphi j}, \tag{6.7}$$

with the combined steering vector for all measurements being

$$\boldsymbol{a}(\Psi,\alpha,\Delta\varphi) = \begin{bmatrix} \boldsymbol{a}_1(\Psi,\alpha) \\ \boldsymbol{a}_2(\Psi,\alpha,\Delta\varphi) \end{bmatrix}. \tag{6.8}$$

If $\Delta\varphi$ is known, and the measurements from each array have been compensated for carrier frequency offset (CFO) to transform all measurements to baseband, the same method can be used for direction estimation for the combined array as for each array individually. By creating simulated measurements using a spherical-wave model and multipath as in Chapter 4 with the array configuration in Fig. 6.3 with 5 cm element spacing, the lowermost element 10 cm over the flat reflecting surface, and the combined array's boresight direction pointing horizontally, Fig. 6.4 shows the estimated elevation angle for different methods. The estimate error from the lowermost array has a lower spatial frequency than the top array, with both having significant deviation in the 5° to 15° range. Averaging the independent estimates

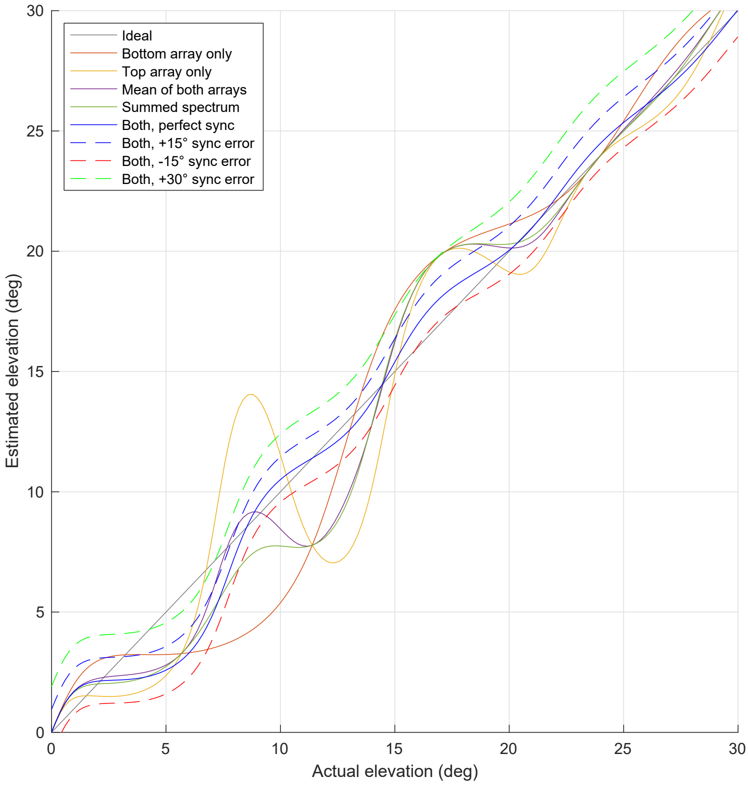


Figure 6.4: Comparison of processing methods for simulated measurements.

does not yield the same result as using (6.3), although they behave similarly. For elevation angles where both independent estimates are above or below the true value, both of these methods also result in an estimate above or below the true value, respectively. This is in contrast to the result from treating the array combination as a single array using perfectly synchronized measurements. An interesting observation is that even if the $\Delta\varphi$ assumed in processing is inaccurate, the elevation angle estimate may still be improved. A 15° offset in the phase bias results in approximately 1° offset in the elevation angle estimate for the array positioning used. Estimation error noise for $\Delta\varphi$ can therefore be expected to reduce the systematic elevation error at the expense of increased elevation noise. Since multipath error is primarily height-dependent, we can calculate an estimate of the phase bias between arrays by assuming that the difference in phase measurement between elements at the same height in each array, after compensation for azimuth angle error estimated using only a single array, must be caused by the phase offset between the arrays. The following method is proposed:

1. For each array individually, estimate the CFO and correct the measurements to baseband, see Chapter 4.

2. Estimate the signal line-of-sight vector $\hat{\boldsymbol{l}}^{a}$ using only the lowest array, with a coarse search and an NLP solver, Chapter 4. This will provide the steering vector to compensate for the difference between the phase of the elements at equal heights due to the azimuth angle to the transmitter.
3. Calculate the estimated phase offset

$$\widehat{\Delta\varphi} = \mathrm{Arg}\left(\boldsymbol{a}_{\mathrm{overlap}}(\Psi,\alpha)^{H}\sum_{i=1}^{n_1}\boldsymbol{x}_{1,i}\right) - \mathrm{Arg}\left(\sum_{i=1}^{n_2}\boldsymbol{x}_{2,i}\right) \tag{6.9}$$

   where

$$\boldsymbol{a}_{\mathrm{overlap}}(\Psi,\alpha) = e^{\frac{2\pi j}{\lambda}\begin{bmatrix}0 & d & 0\end{bmatrix}\hat{l}^{a}}$$

   is the compensation for azimuth angle for two elements at the same height, spread by the distance d. Arg$(\cdot)$ is the complex argument. A weakness of this simple approach is that azimuth estimation error will influence the elevation angle estimate.
4. Using coarse search and optimization, find Ψ, α by maximizing

$$P(\Psi,\alpha,\widehat{\Delta\varphi}) = \left|\begin{bmatrix}\boldsymbol{a}_1(\Psi,\alpha)\\ \boldsymbol{a}_2(\Psi,\alpha,\Delta\hat{\varphi})\end{bmatrix}^{H}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}\right|^{2} = |\boldsymbol{a}(\Psi,\alpha,\widehat{\Delta\varphi})^{H}\boldsymbol{x}|^{2}. \tag{6.10}$$

6.4 Field experiments

Two experimental reference design antenna arrays from Nordic Semiconductor were assembled as in Fig. 6.3. This was done by mounting the arrays on a thick and completely flat plate with drilled mounting holes and milled cutouts for the nRF52833 receiver boards on the back of the arrays. This way, the arrays were oriented the same way, with minimal deviation in the mounting planes of the arrays. The assembly is shown in Fig. 6.6, set up on a grass runway. The small two-dimensional level visible in the lower right of Fig. 6.6b was used for leveling, although this still leaves some uncertainty in the elevation angle of the array boresight directions.

A DJI S1000 multirotor UAV with the same Bluetooth transmitter payload and GNSS receiver as in Chapter 4 was used, with a GNSS receiver working as a real-time kinematic (RTK) base mounted on the top of the array mounting plate. The UAV performed a flight using waypoints along approximately a constant azimuth angle in front of the arrays, varying the distance and height. A plot of the UAV position is shown in Fig. 6.7. The elevation angle was mostly in the range of $5-40^{\circ}$. Fig. 6.8 shows the UAV flying in front of the arrays. IQ measurements from both arrays were logged for offline estimation.

It was found that if the transmitter broadcasts advertising messages on all the primary advertising channels in quick succession, e.g. advertising at 10Hz rate, and using all channels every time, the receivers would most often end up not sampling the same

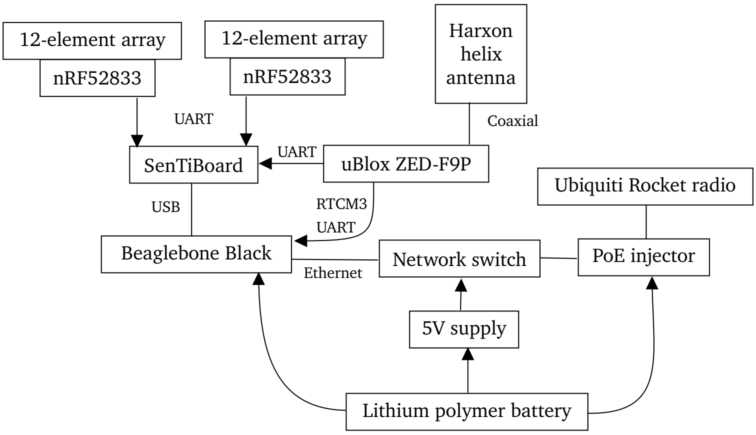
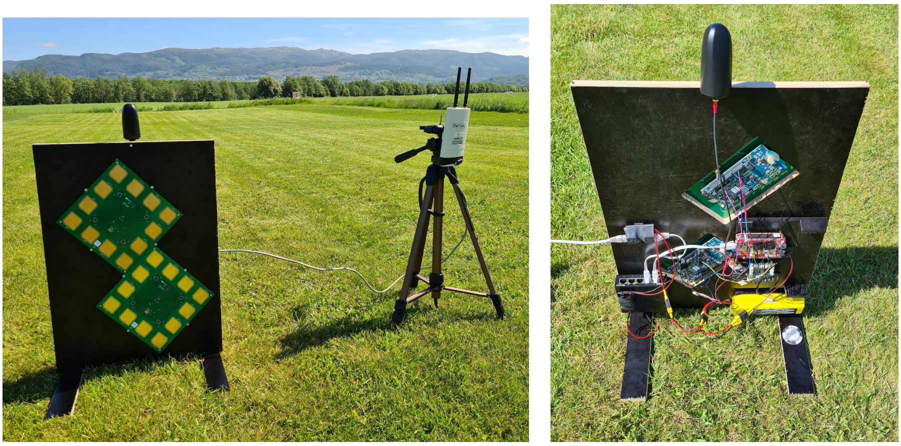


Figure 6.5: Ground hardware schematic



(a) Front view

(b) Back view

Figure 6.6: Setup of vertical array assembly with one element overlapping in height

CTE on the same channels. This effect was random, and by restarting one arrays enough times they would get "in sync", sampling the same channel. To remove this effect and simplify data collection, the beacon was reconfigured to broadcast only on the 2402 MHz channel, ensuring that the arrays sampled the same CTE.

The elevation angles were estimated both individually for each array, and as one array by synchronizing the measurements. Fig. 6.9 shows the CFO estimates for each array used to convert the measurements to baseband. The top array shows a significantly higher CFO than the bottom array. The array used on top in the assembly has consistently had a higher CFO also in previous measurement sets not presented here, likely due to variation between individual oscillators.

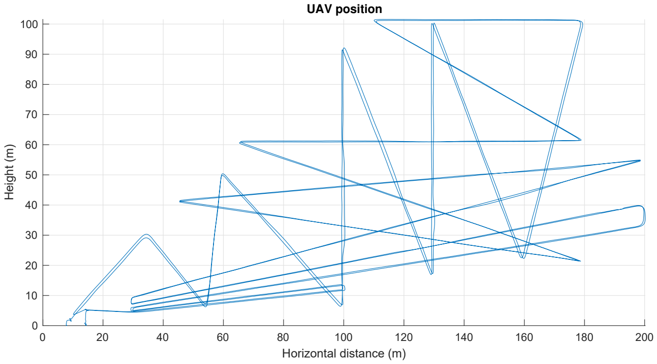


Figure 6.7: Side view of the flight path. The array is located at the origin, with the horizontal axis indicating the distance from the array.

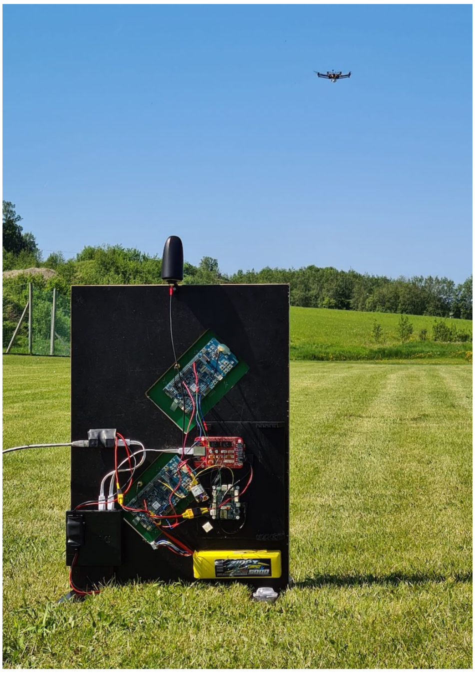


Figure 6.8: DJI S1000 multirotor UAV flyving in front of the Bluetooth arrays.

The elevation and azimuth angle estimates are plotted in Figs. 6.10 and 6.11. For RTK GNSS the height offset between the base antenna and the center point of the array assembly, and the offset between the Bluetooth transmitter antenna and GNSS receiver on the UAV was corrected when calculating the GNSS elevation angle. The variation in azimuth angle is likely the result of roll angle array leveling errors. The elevation angles are also plotted as a function of the GNSS elevation estimate in Fig. 6.12, showing the systematic error behavior. It is clear that the

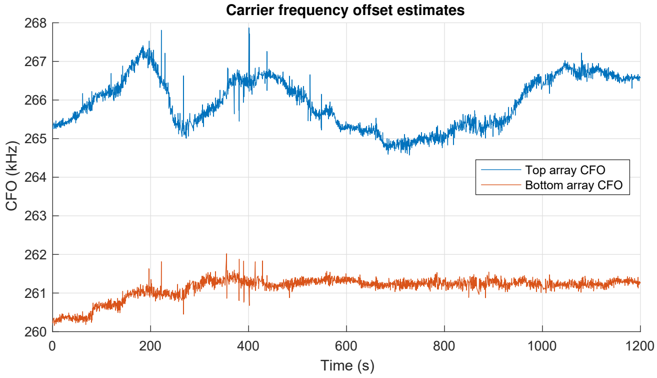


Figure 6.9: CFO estimates for the two arrays

combined processing significantly reduces the error in the elevation angle range 7° to 15°. For the lowest angles, below approximately 7°, the results are similar, with negative elevation angle estimates. Above 20 degrees elevation the results are also similar. The error pattern for each array resembles the one predicted from simulation in Fig. 6.4. Examples of calculated spatial spectra for a common time are shown in Fig. 6.13, where it is clear that the combined processing increases the angular resolution in the vertical direction, allowing separation of the direct and reflected signals as individual peaks. Since the top array is offset to the side compared to the lower array, the direction of maximum resolution in the combined spectrum is not exactly in the vertical direction, but at an angle.

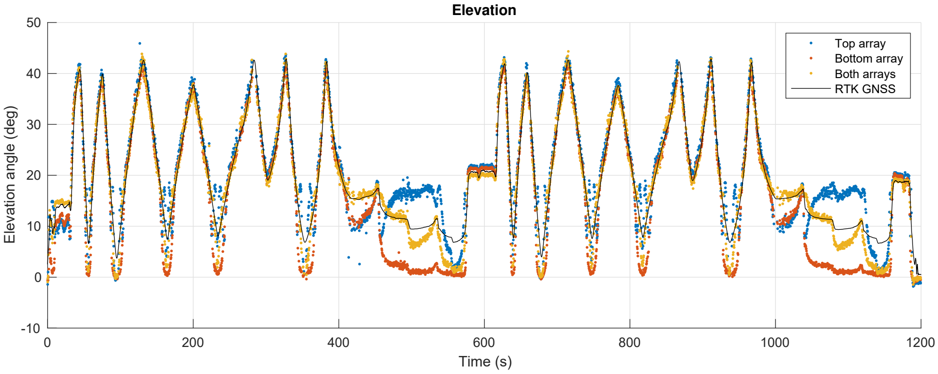


Figure 6.10: Elevation plot

The phase offset $\Delta\hat{\varphi}$ found using (6.9), used to synchronize every pair of measurements from the arrays is plotted in Fig. 6.14, appearing to be without any systematic behavior over time, with a uniform distribution. The angle of the azimuth correction in (6.9) is plotted on Fig. 6.15. The two array elements overlapping in height is spaced by 7.07 cm, and the $5-6°$ offset from the boresight direction of the arrays

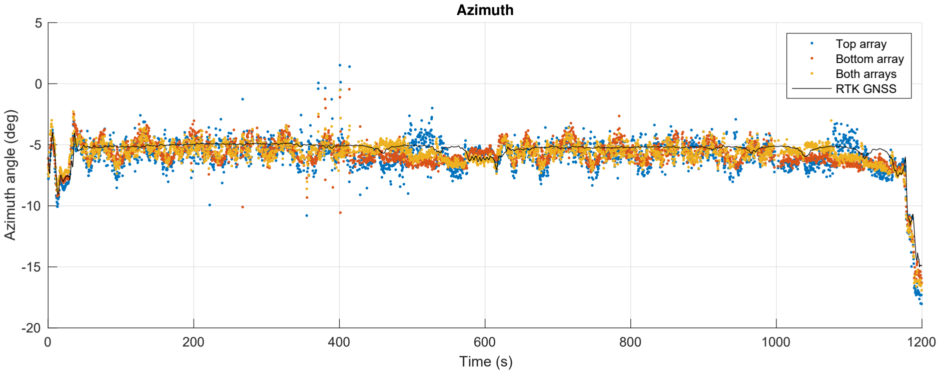


Figure 6.11: Azimuth plot. The GNSS values cannot be used to assess accuracy in this case, as the direction of the array boresight has been found by comparing the GNSS and Bluetooth estimates.

corresponds to approximately 20° phase offset. The Received Signal Strength Indicator (RSSI) for each array is plotted in Fig. 6.16 along with the distance from the array from RTK GNSS positions. In addition to the dependency of the RSSI on distance, there also appears to be an effect caused by the multipath interference. This is shown enlarged in Fig. 6.17. Note how the top array, which is furthest from the ground and therefore yields elevation estimates with a higher spatial frequency, shows RSSI variation which has a higher oscillating frequency than the lower array. The variation may be the result of the constructive and destructive interference also affecting the elevation estimates.

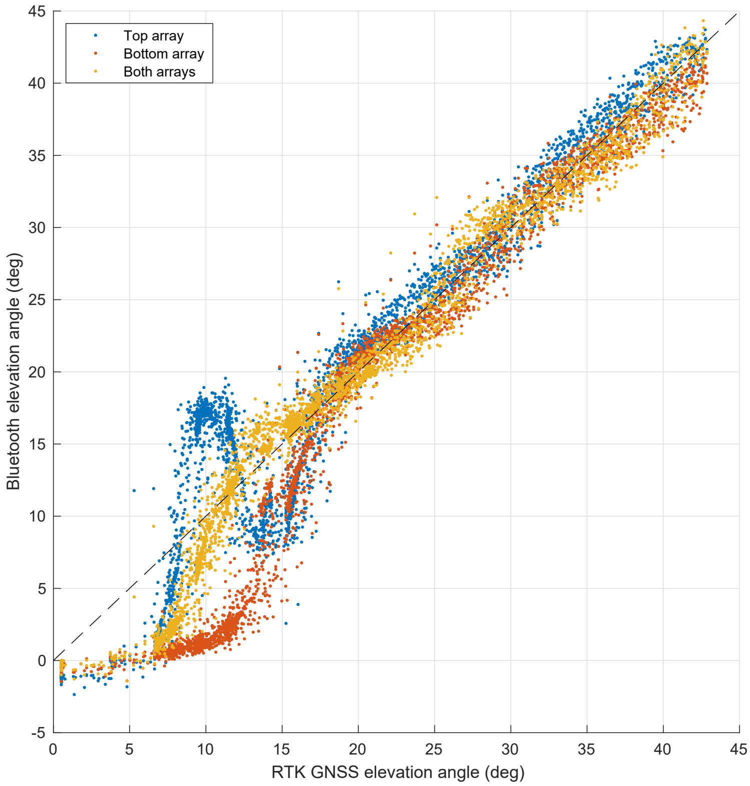
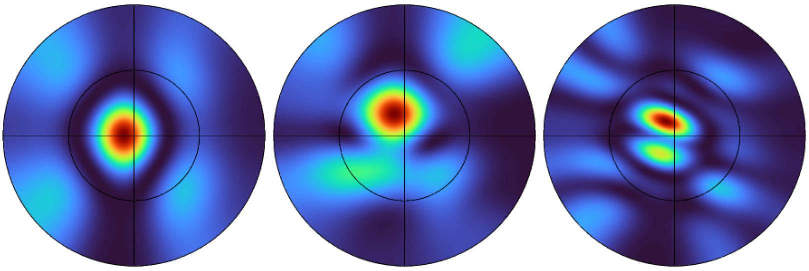


Figure 6.12: Bluetooth elevation angle estimates using each array individually, and processed together, plotted as a function of the RTK GNSS elevation angle.



(a) Bottom array (b) Top array (c) Combined array

Figure 6.13: Example spatial pseudo-spectrum for separate and combined arrays at the time 492.2s in Fig. 6.10 and Fig. 6.11, when the RTK GNSS elevation angle is 11.4°. Red indicates the maximum value. The spectrum covers the half-sphere in front of the array. The edges of the spectra are at 90° from the array boresight, with the top of each spectrum pointing towards the zenith, and the center pointing in the boresight direction which is aligned horizontally. The combined processing has increased angular resolution in the vertical direction, and is able to separate the direct and reflected signals

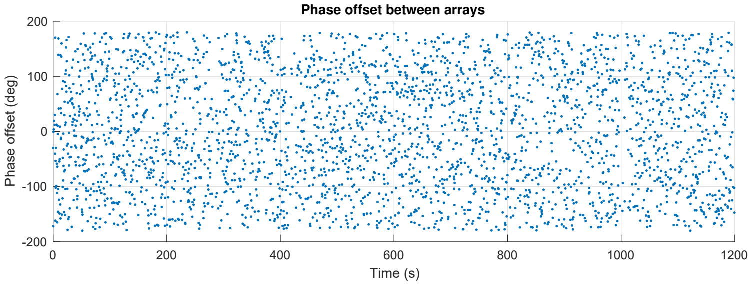


Figure 6.14: Phase offset $\hat{\Delta\varphi}$ used for array measurement synchronization.

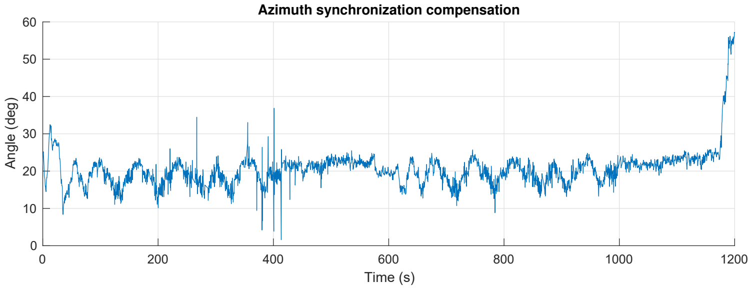


Figure 6.15: Angle of the azimuth correction $\boldsymbol{a}_{\text{overlap}}(\Psi, \alpha)^H$

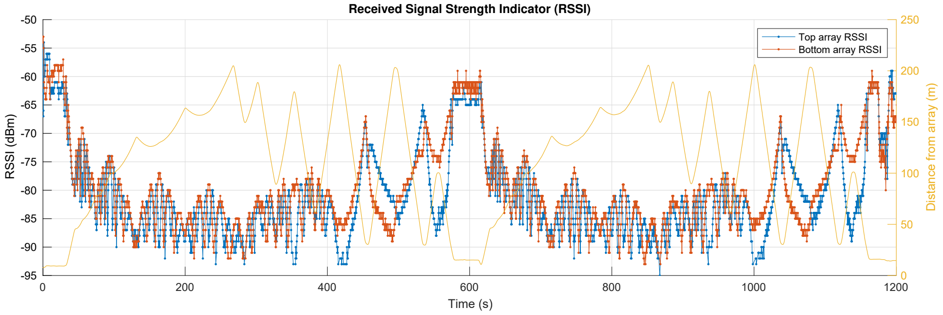


Figure 6.16: RSSI

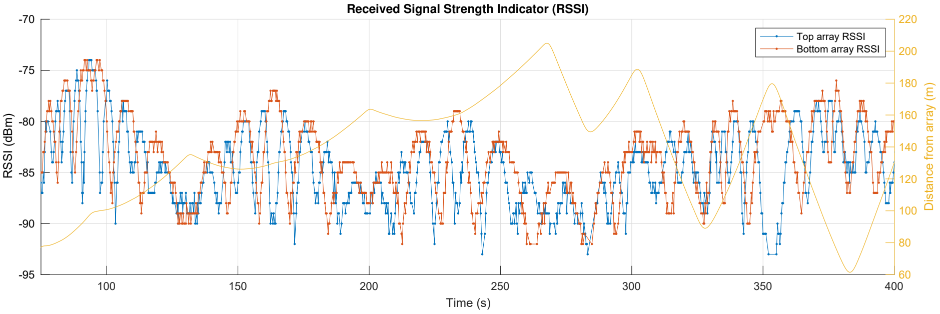


Figure 6.17: Closer view of the RSSI, showing oscillations with difference frequencies for each array.

6.5 Conclusion

This paper has considered the combined use of two independent arrays and proposed a measurement synchronization procedure to allow the measurements from both arrays to be used together for direction estimation as for a single larger array. The results from a field experiment show that the proposed method yields a significant reduction of elevation error due to multipath in the 7° to 15° range, while maintaining similar azimuth estimation performance.

Part II

Automatic arrest system recovery of fixed-wing UAV

Chapter 7

Control System Architecture for Automatic Recovery of Fixed-Wing Unmanned Aerial Vehicles in a Moving Arrest System

This chapter is based on the publication

- [47] K. Gryte, M. L. Sollie, and T. A. Johansen, “Control System Architecture for Automatic Recovery of Fixed-Wing Unmanned Aerial Vehicles in a Moving Arrest System,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 103, no. 4, 2021. DOI: 10.1007/s10846-021-01521-z

and considers automatic recovery of fixed-wing UAVs in a moving arrest system at sea.

7.1 Introduction

Fixed-wing unmanned aerial vehicles (UAVs) are typically superior to similar-sized rotary-wing UAVs using the same energy source when it comes to range, endurance and speed, and is thus the preferred option for many scenarios. While many missions can be flown automatically, possibly interacting with an operator at a ground control station, recovery of fixed-wing UAVs is often a manual task performed by a highly skilled pilot. In addition to the economic benefits of removing the human pilot from the control loop, this also enables operations with a smaller margin of error, as the sensing and control loops of a UAV autopilot are faster and capable of simultaneously monitoring more mission-critical conditions. This enables operations in rougher conditions, such as in strong and gusty winds, and landing in confined and moving locations, such as aboard ships.

Landing of UAVs on a moving platform is often limited by available space. One viable approach to enable recovery on a space-limited, moving platform is to design the operation around a fixed-wing UAV with vertical takeoff and landing (VTOL) capabilities, i.e. a rotary-wing/fixed-wing hybrid [104]. The increased maneuverability and hover capabilities associated with VTOL UAVs make them easier to land, but this comes at a cost of increased drag, mass and complexity, and decreased payload capacity. While fixed-wing VTOL UAVs and rotary-wing only require a flat surface to land on a moving platform, as in [111, 2, 116], conventional fixed-wing UAVs, which this publication focuses on, relies on arrest recovery systems to land on a space-limited, moving platform. Arrest recovery systems are herein defined as some mechanical system that seeks to remove the kinetic energy from the fixed-wing UAV and bring it to a standstill. This enables the design of the UAV to be focused on the main mission, which is usually what adds value for the end user. Arrest recovery systems, can be divided into the following categories:

Net recovery: flying into a tensioned, fixed net that absorbs the kinetic energy of the impact either vertically [126, 65, 171], horizontally mounted on the roof of a moving car [94, 99], or suspended between two multirotor UAVs [68].

Airbag recovery: flying into an inflated cushion [53], from any direction.

Hook recovery: attaching to a wire stretched between two points, e.g. horizontally [62], vertically [142] or between two multirotor UAVs [19].

One strategy to simplify recovery of fixed-wing UAVs on a moving platform is to control the platform itself, assisting or fully performing the alignment of the recovery system with the UAV. In both [68] and [19], where multirotor UAVs are used to capture the fixed-wing UAV, the accurate alignment of the recovery system with the flight path of the incoming UAV is performed by the multirotors, while the fixed-wing flies along a predetermined path. Another way to simplify the control for the recovery is to predict the motion of the platform [89] and when conditions are safe for landing[2]. In the ideal case, with perfect prediction, this simplifies the scenario to stationary landing, although in reality the prediction of e.g. ship motion is difficult [116, 49].

Recovery in an arrest system requires two types of navigation functions in the UAV; it has to *self-navigate*, i.e. keep track of its own position, velocity and attitude, while also keeping track of its position relative to the arrest system. While the self-navigation also is critical for the success of the main mission of the UAV, the relative navigation is only relevant for the recovery. Therefore, a large overlap in the hardware requirements for the two systems is ideal, to simplify avionics. UAV avionics typically consists of an inertial measurement unit (IMU) aided by GNSS position measurements, heading information from a magnetometer/compass, altitude information from a barometer or altimeter, and airspeed information from a pitot tube. These sensors are sufficient for the waypoint tracking involved in most missions, but the precision might not be sufficient for a recovery application. The required level of precision is largely governed by the error margins allowed by the geometry of the arrest system, and the dynamics of the moving arrest system compared to the agility of the UAV. Furthermore, as recovery is seen as a safety-critical phase of the operation, it may be required to add additional sensors to improve the robustness and resilience.

Visual navigation is a popular technique for relative navigation [70, 77]. What makes this approach tractable is the possibility to construct a self-contained system that does not rely on external communication or measurements, that delivers relative pose measurements at a high rate, with high precision when close to the arrest system, like e.g [65, 53, 144, 110]. Drawbacks include high processing requirements, risk of false detection, and sensitivity to visual conditions, such as light/weather conditions and distinctiveness of the arrest system relative to its background. The latter can to some extent be mitigated by using infrared (IR) cameras, either using natural landmarks [64, 165] or IR lamps in known locations [48].

The arrest system may also be equipped with position sensors such as GNSS receivers [94] or radio beacons, exemplified by ultra-wideband (UWB) [39, 79, 46], where the main advantage is low cost to the user, small footprint, all-weather availability and ease of use. This is especially true for GNSS, which is already part of most autopilot sensor suites. While the positioning accuracy of a single receiver without augmentation is in the order of meters, depending on whether one or more constellations and a single or dual frequency receiver is used [42], two independent receivers operating within a short distance will have atmospheric errors which are mostly common. This means that relative positioning accuracy between two receivers will generally be better than the absolute accuracy if both receivers track the same satellites and apply the same atmospheric corrections, although this also depends on the multipath situation for each receiver. Space-based augmentation systems (SBAS) can improve the positioning accuracy by transmitting corrections for satellite position errors, clock errors and atmospheric effects to the user from geostationary satellites [42]. Centimeter-level precision, between the UAV and a base station, can be achieved with real-time kinematic (RTK) GNSS [42], a technology that over the last decade has become available to the civilian market at a low cost. GNSS measurements are inherently absolute, so if the arrest system is moving, it must be fitted with a second receiver to obtain the relative position. This also calls for radio communication between the UAV and arrest system. Advantages with UWB, compared to GNSS, include robustness to interference, resistance to multipath [92], as well as high temporal resolution allowing for centimeter level precision of range measurements [78], but the ranges are typically only in the hundreds of meters. By positioning the UWB beacons to move with the arrest system, they can provide a relative navigation solution, possibly at the cost of weaker measurement geometry due to the limited size of the arrest system, leading to lower precision [39]. Drawbacks associated with GNSS include susceptibility to radio frequency interference (RFI), both natural RFI, such as ionospheric scintillations [169] and multipath, and intentional RFI, such as jamming [100] and spoofing [61].

Other relative navigation off-the-shelf options include the laser-based *Object Position* *and Tracking System* (OPATS) [113], GPS- and radar-based *Dual-Thread Automatic Takeoff and Landing System* (DT-ATLS) [123, 50], as well as the integrated navigation and control solution *UAV Common Automatic Recovery System* (UCARS) [124] for ship landing. Although these systems are well proven, they are also proprietary commercial systems with unknown algorithms and with little flexibility to make customizations.

To approach the arrest system from the correct direction, its (relative) heading must be found, from one of the seven ways to estimate heading [37]. The simplest is through a magnetometer/compass, which unfortunately is highly susceptible to magnetic anomalies and electromagnetic interference (EMI) [37]. With a camera, the relative heading angle can be found [65]. Another solution is to equip the arrest system with multiple position sensors to find the orientation of the baseline between them, see e.g. Chapter 2 or [51] that reports 0.27° precision for a baseline of about 0.5 m using GNSS. Depending on the dynamics of the arrest system and the precision requirements, a combination with inertial sensors may be required to provide smoother estimates at a higher rate.

Another important part of the recovery system is guidance and control. For an overview of different control algorithms for fixed-wing UAVs, see [90], and [139] for path following guidance algorithms. The navigation setup tends to dictate requirements for the guidance and control system, where visual servoing methods favor pure-pursuit guidance [65, 53], while with the relative navigation between the UAV and arrest system in an absolute frame, the guidance law can be chosen arbitrarily.

This paper seeks to investigate how precisely, accurately and reliably a fixed-wing UAV can land in a moving arrest system, using a control system architecture building modularly and non-intrusively on low-cost commercial off-the-shelf (COTS) hardware (HW) and software (SW). The main contribution is the design and implementation of the landing system SW and HW, in addition to extensive experimental testing. To our knowledge, there are no openly available publications that give a complete description of such a system, so we believe that the description and systematic evaluation herein is a solid foundation for an industrial implementation and future academic research. To be of any operational value, the system must be accurate and reliable enough that the operators have confidence in it, which boils down to repeatability and operability across a wide range of environmental conditions. It should also be precise enough to allow recovery in arrest systems that are of a manageable size. The presented system is based on COTS autopilot HW and SW as they are generally well tested, thus reliable, providing airworthiness and reducing the needs for implementation, possibly at the cost of performance, flexibility and licensing issues. However, they might not provide all the necessary features. Even though some commercially available autopilots are capable of automatic landing in fixed locations, such as [9], this does not suffice for a moving arrest system. Instead of adding the arrest system recovery functionality in a specific autopilot SW, by making possibly error-inducing changes to a working system, this work seeks to build on the existing interfaces of common autopilots by basing the extension on the very general assumption that all autopilots provide an estimate of its position, velocity and attitude, based on internal sensors and an external position measurement, while also providing a means to command the UAV to fly to a specific location. In this work the autopilot is provided with position measurements from RTK-GNSS, due to its simplicity and high precision, but it could in principle come from any position sensor with sufficient accuracy. In addition to non-intrusiveness, these assumptions also make the system modular so that it can be adapted to a wide range of autopilots and fixed-wing UAV configuration, through only a few tuning variables, although this work focus on the open source ArduPlane autopilot software, and is motivated by its current limitations. This is achieved by a line-of-sight (LOS) guidance controller, that ensures line-following of a virtual runway into the moving arrest system, by sending position commands to the autopilot.

The paper is split into two main parts. First, Section 7.2 describes the recovery system architecture for a general, idealized scenario. This includes the plan generation (Section 7.2.1), navigation (Section 7.2.2), motion prediction (Section 7.2.3), guidance and control (Section 7.2.4) and operator interface (Section 7.2.5) subcomponents. The second major part is Section 7.3, where the general architecture is adapted to a specific arrest system and a specific autopilot, where the implementation in a real-time system is discussed. The implemented system is experimentally validated in two experiments in Sections 7.4.1 and 7.4.2, first for a stationary arrest system and then for a moving arrest system mounted on a floating barge towed by a ship. Lastly, in Section 7.5.1 we discuss the results and mention possible improvements before drawing the conclusion in Section 7.5.2.

7.2 Recovery system architecture

The control system architecture is presented in this section, by considering each of the functional components that are needed to recover a fixed-wing UAV in a moving arrest system. The system creates the plan, seen in Fig. 7.1, from parameters set by the operator. As the arrest system is moving, so is the latter part of the plan, which is translated and rotated such that it lines up with the predicted pose of the arrest system at the time of impact. This pose is predicted from precision navigation, that includes compensation for predicted arrest system motion to maximize the chances of impacting near its center. To allow straight-line path-following, while being limited to only sending a position reference to the autopilot, the system is augmented with a line-of-sight guidance that finds an appropriate carrot-point reference that will give the desired behavior. Before impact with the arrest system, the motor is turned off, to avoid damage and severe entanglement.

7.2.1 Plan generation

A plan can be generated with different objectives in mind. The different objectives are usually a combination of minimizing risk and reducing the effect the recovery has

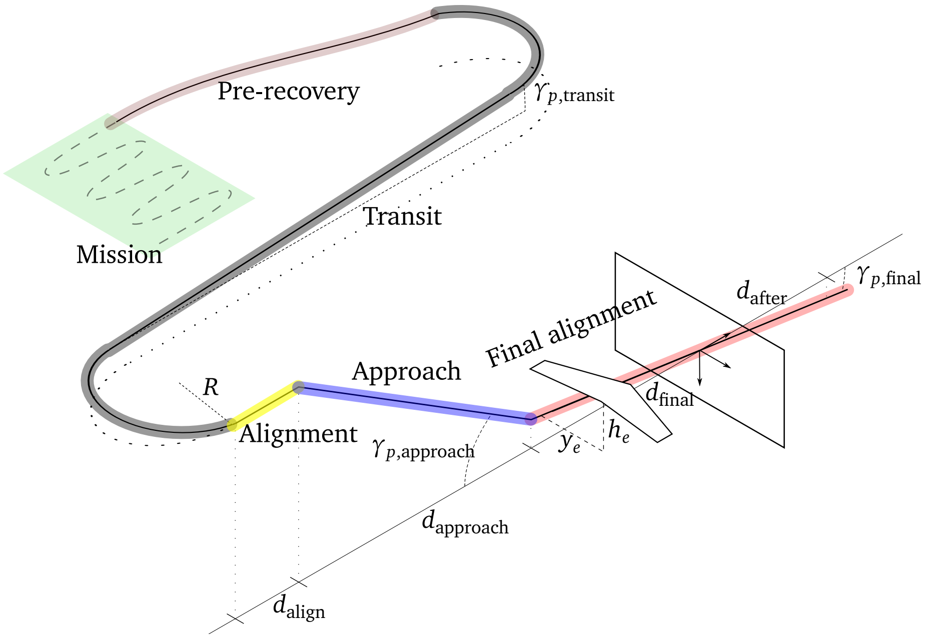


Figure 7.1: The geometry of the arrest system recovery plan, illustrated with a net

on the rest of the mission, being the primary objective. The presented solution seeks to minimize risk, primarily in three ways. First by maximizing the predictability of the UAV motion, by having the final stages be straight line segments. Secondly, the risk is minimized by delaying the reduction in height as much as possible, to increase the probability of succesful abort in case of an emergency. Lastly, the risk is minimized by reducing the relative speed between the UAV and the arrest system before impact, to not jeopardize the structural integrity of the UAV. However, this speed reduction must not come at the cost of a too low airspeed, to avoid stall, and to maintain enough speed to penetrate wind gusts and shear, as well as overcoming any forces needed to be captured by the arrest system. Based on these strategical decisions, the recovery is divided into the following phases, illustrated in Fig. 7.1.

Pre-recovery This is when the UAV has finished its mission, and is initiating recovery by flying to the start of the transit phase along an arbitrary path.

Transit The path toward alignment with the arrest system ends in a 2D Dubins path [26], to bring the UAV to the start of the descent with a correct course angle and altitude. This is an interconnection of circular arcs and straight lines, which, under the assumption of a maximum curvature, is shown to be the shortest path between two poses in 2D, thus minimizing the effect the recovery has on the rest of the mission. The altitude is simply a linear descent at a prescribed angle, $\gamma_{p,\text{transit}}$, again delaying the descent as long as possible. If the desired altitude at the end of the transit phase is unreachable at this descent rate, the final circular arc of the Dubins path is extended into a spiral to shed the excess altitude.

Alignment When exiting the Dubins path (transit phase), the course should be aligned with the arrest system, but to be sure this course is held for a distance d_{align}, to ensure that the UAV has a stable course.

Approach While maintaining alignment, the UAV descends with a flight path angle γ_{approach}, for a distance d_{approach}.

Final alignment The UAV is now on a virtual runway, starting a distance d_{final} before the arrest system. This runway is guiding the UAV into the arrest system center by continuously aligning itself with the orientation and position of the moving arrest system and the desired landing flight path angle γ_{final}. Speed is reduced to lower the impact, and the engine is turned off to avoid damage to the propeller or arrest system.

Catch After a successful recovery, with stopped/disarmed motors.

7.2.2 Navigation

The success of the recovery hinges on knowledge of where the UAV is, relative to the arrest system. For the presented approach, the self-navigation is assumed to be handled by the COTS autopilot, typically through a Kalman filter based on inertial navigation, aided by a GNSS receiver for position measurements, a compass/magnetometer for heading measurements, a barometer/altimeter for altitude measurements, and a pitot tube for airspeed measurements. These are considered as standard components, since they are part of most autopilot sensor suites.

Instead of only using a standalone GNSS receiver, this work uses Real-Time Kinematic (RTK) GNSS, which has been successfully utilized for similar applications [126, 68]. RTK GNSS works by continously sending all raw measurements including carrier phase measurements from a reference receiver, in addition to the reference receiver's own position estimate (as this may be changing over time), to the UAV. The UAV receiver then uses the measurements from both receivers, using a technique based on carrier phase interferometry, to estimate the baseline between them with high accuracy and precision. This works well as long as they are closer than about 20 km apart [31], as the atmospheric signal disturbances have a high degree of spatial correlation such that they are approximately equal for both receivers. It is important to note that the output format of the receiver is the same both when RTK is used and when it is used as a standalone receiver, and where there is a degradation of the RTK capability, e.g. GNSS signal strength drops so carrier phase measurements become unusable, this means that the precision and accuracy of the relative positioning is reduced. RTK GNSS was chosen based on the simplicity of its usage, availability and high precision. Whether the high precision of RTK GNSS is necessary, depends on the size of the UAV relative to the arrest system, but given the availability of low-cost RTK GNSS solutions, it seems tractable. These receivers output position and velocity estimates with high precision and accuracy directly into the autopilot, without the need for any additional computation. If using a widely supported output-data format, such as the NMEA standard, the receiver can easily be replaced, becoming a transparent source of high-precision and high-accuracy estimates to the autopilot.

7.2.2.1 Relative navigation setup

To reap the full potential of GNSS in terms of precision, it is important to make use of RTK processing, providing a precise *relative* position of the UAV and base. Therefore, the arrest system is equipped with one GNSS receiver acting as an RTK base station. For ship-based recovery in open waters, the only option is a moving-base configuration, where only the precision of the relative position is guaranteed. However, for recovery in stationary arrest systems, the base antenna position can be surveyed, allowing for both accurate and precise, global position measurements. Further, the arrest system is equipped with an additional RTK GNSS receiver, which is used to measure the orientation of the arrest system. During recovery, it is important that the position of the UAV and arrest system are reported in the same frame of reference, with the same origin. This also implies that a barometric pressure sensor onboard the UAV cannot be used as the only source of altitude measurements during the final stages of recovery, unless the arrest system is equipped with barometric pressure sensor that is calibrated to the same level as the onboard pressure sensor, pre-flight, as changes in ground level pressure would lead to drift in the altitude estimate during a flight, which ultimately would cause the UAV to aim above or below the physical arrest system target. Another option is to use a source of altitude measurements without long-term drift, such as RTK GNSS, either as a primary altitude sensor or to correct barometer drift over time.

The arrest system is instrumented with two GNSS antennas, with one antenna positioned in $\boldsymbol{p}^n_{\text{left}}$ on the left side as seen from the front, acting as the RTK base for the UAV and the second net antenna, which is placed in position $\boldsymbol{p}^n_{\text{right}}$ on the right side of the arrest system, as illustrated for a recovery net in Fig. 7.2. The position of the arrest system center in the NED frame $\{n\}$ with its origin at the position of the left antenna, $\boldsymbol{p}^n_{\text{arrest}}$, roll angle ϕ_{arrest} and heading angle ψ_{arrest} are calculated as

$$\psi_{\text{arrest}} = \operatorname{atan2}(-b^n_x, b^n_y), \tag{7.1}$$

$$\phi_{\text{arrest}} = \operatorname{atan2}\left(b^n_z, \sqrt{b^{n\,2}_x + b^{n\,2}_y}\right), \tag{7.2}$$

$$\boldsymbol{p}^n_{\text{arrest}} = \frac{1}{2}\boldsymbol{b}^n_{\text{arrest}} - \boldsymbol{R}^n_b \boldsymbol{p}^b_{\text{offset}}, \tag{7.3}$$

where $\boldsymbol{b}^n_{\text{arrest}} = \begin{bmatrix} b^n_x & b^n_y & b^n_z \end{bmatrix}^{\mathsf{T}} = \boldsymbol{p}^n_{\text{right}} - \boldsymbol{p}^n_{\text{left}}$ is the vector from the left antenna to the right antenna, estimated using moving-base RTK with the left antenna used as the base. $\{b\}$ here denotes the body frame of the arrest system, and the vector $\boldsymbol{p}^b_{\text{offset}}$ contains the position of the midpoint between the antennas relative the origin of $\{b\}$, allowing more flexibility in the mounting of the antennas if required, i.e. they can be positioned higher than the center or with different distance to the center on each side. With this antenna configuration, the pitch angle can not be calculated, forcing the approximation $\boldsymbol{R}_b^n \approx \boldsymbol{R}_b^n(\phi_{\text{arrest}}, \theta_{\text{nom}}, \psi_{\text{arrest}})$, where the pitch angle θ_{nom} is a constant nominal value. This encourages a small offset $\boldsymbol{p}_{\text{offset}}^b$ to minimize position errors. For excessive pitch motion, or large offsets in the xz-plane, an alternative would be to also estimate the pitch angle, using an IMU or a third antenna.

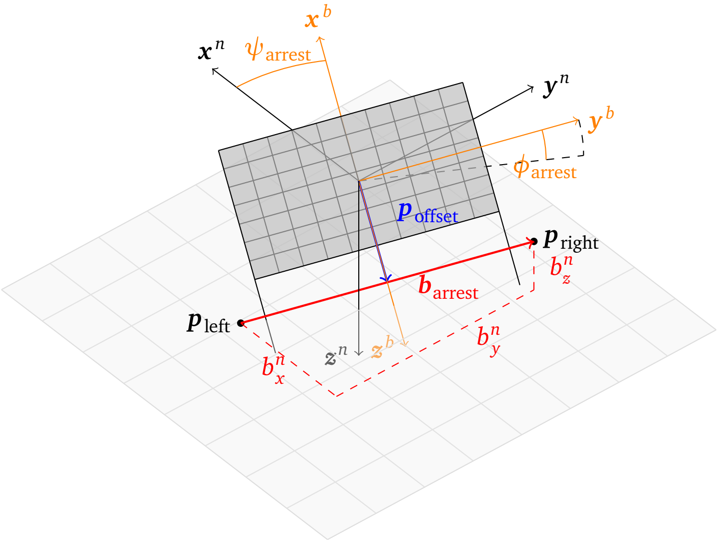


Figure 7.2: Illustration of recovery net with antenna positions. The $\{n\}$ origin is here shown coincidental with $\{b\}$ to illustrate the roll and yaw angles.

7.2.3 Motion prediction

In principle, for a successful recovery the UAV only needs to know the relative pose of the arrest system at the time of recovery. However, this can be difficult to predict for moving arrest systems, as the recovery location might not be determined uniquely and with certainty at the start of the recovery plan. Not only is this a chicken-and-egg problem, where the relative position of the arrest system, at the time of recovery, is needed to calculate the time it takes to fly to it (which again is needed to predict the relative position of the arrest system at the time of recovery), but it is also highly dependent on the dynamics of the arrest system. The arrest system can either have accurate, actively controlled motion, such as [68, 94, 99], which calls for synchronization between the UAV and arrest system, or be passively attached to a moving platform without accurate control, such as a moving ship [116]. Particularly for recovery in smaller arrest systems in space-restricted environments, using less agile UAVs with smaller error margins, good predictions of the arrest system motion will be more important, but the prediction quality naturally depends on how well the arrest system dynamics can be modeled. Given good predictions, the recovery controllers may be less reactive to rapid arrest system motion, thus allowing less agile UAV dynamics.

The proposed system only makes a rough prediction of the position of the arrest system at the time of impact, given its current velocity. The relative position of the UAV and arrest system is calculated as

$$\Delta \boldsymbol{p}^n = \boldsymbol{p}^n_{\text{arrest}} + \Delta \boldsymbol{p}^n_{\text{arrest}} - \boldsymbol{p}^n_{\text{UAV}}, \tag{7.4}$$

where $\boldsymbol{p}^n_{\text{arrest}}$ and $\boldsymbol{p}^n_{\text{UAV}}$ are current position estimates for the arrest system and UAV, respectively, and $\Delta \boldsymbol{p}^n_{\text{arrest}}$ is the predicted arrest system movement during the time t_{impact} remaining until impact, using an initial guess. Given this relative position, an estimate of the time until impact is found using the current UAV velocity and a simplifying assumption of straight line flight to the arrest system,

$$t_{\text{Impact}} = \frac{\|\Delta \boldsymbol{p}^n\|_2}{\|\boldsymbol{v}_{\text{UAV}}\|_2}, \tag{7.5}$$

which is then used to improve the prediction of arrest system movement,

$$\Delta \boldsymbol{p}^n_{\text{arrest}} = \boldsymbol{v}_{\text{arrest}} t_{\text{Impact}}. \tag{7.6}$$

Then, (7.4) is used to compute a new prediction of the relative position at the time of impact, and the process is iterated until sufficiently converged. $\Delta \boldsymbol{p}^n_{\text{arrest}}$ is initialized as 0, which corresponds to a stationary arrest system, but using the final result from the previous time step at the next time step can reduce the number of iterations needed for convergence.

In the above, no prediction or filtering of the attitude and heave motion of the arrest system is performed, as the time horizon for reliable wave-induced motion prediction for ships may be in the order of a few seconds, due to the stochastic nature of waves [49]. Furthermore, the motions of the arrest system are assumed small compared to the agility of the fixed-wing UAV.

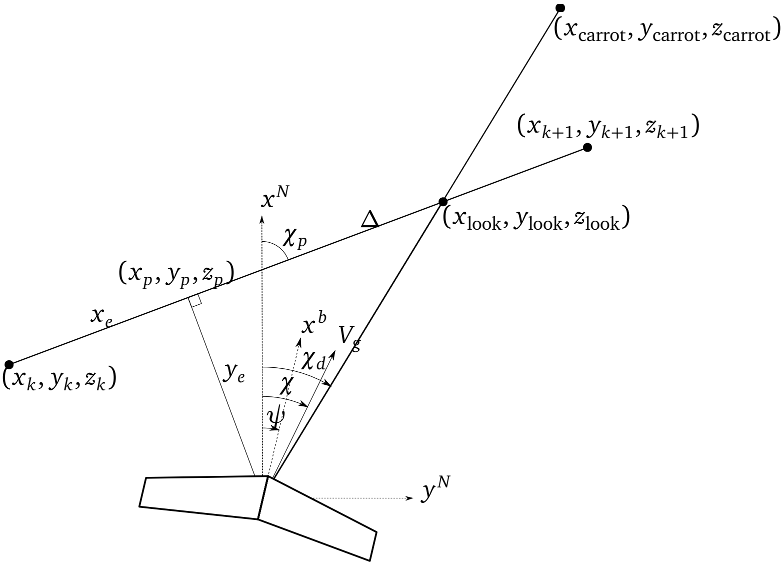
7.2.4 Guidance and control

To ensure that the UAV follows the final alignment stage of the recovery plan in a manner that is easy to predict by the operator, line-following guidance [139], such as line-of-sight (LOS), is applied in the approach and final alignment stages. LOS guidance [34] mimics an experienced navigator, by aiming to intercept the desired path a time-varying lookahead distance Δ ahead of the current position, see Fig. 7.3. The UAV, with position $\boldsymbol{p}^n_{\text{UAV}} = [x, y, z]^\top$ in the NED frame, follows the line segment that starts in waypoint $\boldsymbol{x}^n_k = [x_k, y_k, z_k]^\top$ and ends in waypoint $\boldsymbol{x}^n_{k+1} = [x_{k+1}, y_{k+1}, z_{k+1}]^\top$, illustrated in 2D in Fig. 7.3, while the different segments are illustrated in Fig. 7.1. As the virtual runway is moving with the position $\boldsymbol{p}^n_{\text{arrest}}$ and orientation ψ_{arrest}, the waypoints that define the start and end of its line segment are also moving. The lateral LOS guidance law

$$\chi_{\text{LOS}} = \text{atan}\left(\frac{-y_e}{\Delta} + K_i \int \left[\text{atan}\left(\frac{-y_e}{\Delta}\right) - \chi\right] \mathrm{d}t\right), \tag{7.7}$$

$$\chi_d = \chi_{\text{LOS}} + \chi_p, \tag{7.8}$$

where Δ is the lateral lookahead distance, where $y_e = -\sin(\chi_p)(x - x_k) + \cos(\chi_p)(y - y_k)$ is the lateral cross-track error, and where $\chi_p = \tan(\frac{y_{k+1} - y_k}{x_{k+1} - x_k})$ is the course angle of the line segment [34]. This guidance law is extended from [74, 35] to also include integral effect. Although the guidance law is formulated using the course angle, which inherently accounts for wind effects, integral effect is still needed to overcome stationary cross-track errors as a result of e.g. uncompensated misalignment of the navigation system with respect to the airframe. Specifically, there might be a small, uncompensated angular difference in how the navigation system is mounted compared to what roll and pitch angles correspond to trimmed level flight. This has the effect that to fly level, the autopilot should command a nonzero roll angle, which only can be achieved with a zero cross-track error if the integral term in (7.7) is nonzero. In (7.7), the integral term accounts for the error between the actual course χ and the integral-free desired LOS-angle.



$(x_{\text{carrot}}, y_{\text{carrot}}, z_{\text{carrot}})$
$(x_{k+1}, y_{k+1}, z_{k+1})$
x^N
Δ
$(x_{\text{look}}, y_{\text{look}}, z_{\text{look}})$
χ_p
(x_p, y_p, z_p)
x_e
x^b
V_g
χ_d
(x_k, y_k, z_k)
y_e
χ
ψ
y^N

Figure 7.3: Geometry of the lateral line-of-sight guidance

To ensure modularity of the system, and applicability to a wide variety of autopilot interfaces, the guidance controller set-point should be formulated as a desired position. For the lateral axis, this is achieved by considering the desired course χ_d as the direction of the vector from the UAV to the lookahead point $\boldsymbol{p}^n_{\text{look}}$, of length $\sqrt{y_e^2 + z_e^2 + \Delta^2}$, see Figs. 7.3 and 7.4, it is clear that the desired lateral position $\boldsymbol{p}^n_{h,\text{look}} = \left[x_{\text{h,look}}, y_{\text{h,look}}, z_{\text{h,look}},\right]^\top$ is found from the UAV position by

$$\boldsymbol{p}^n_{h,\text{look}} = \boldsymbol{p}^n_{\text{UAV}} + \boldsymbol{R}_z(\chi_d)\boldsymbol{R}_y(\gamma_d)\boldsymbol{R}_z(\chi_{\text{LOS}})\boldsymbol{R}_y(\gamma_{\text{LOS}}) \begin{bmatrix} \sqrt{y_e^2 + z_e^2 + \Delta^2} \\ 0 \\ 0 \end{bmatrix}, \tag{7.9}$$

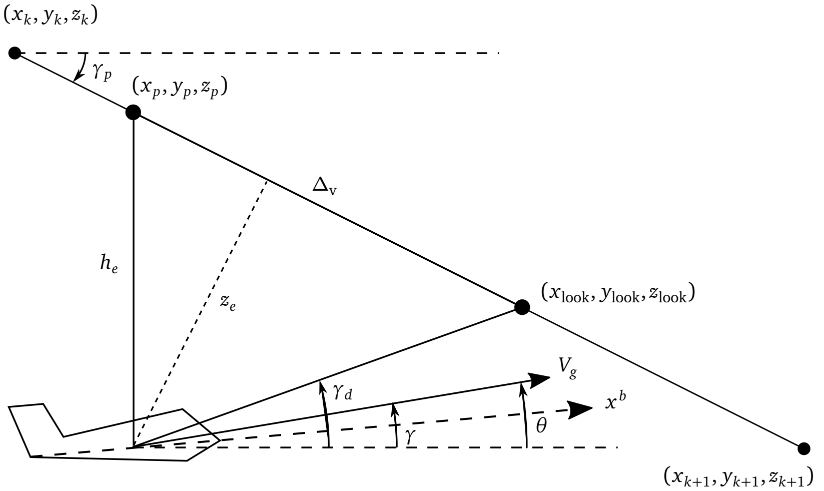


Figure 7.4: Geometry of the longitudinal line-of-sight guidance

where $\boldsymbol{R}_*$ is the rotation matrix representing rotation about the $*$-axis, and where γ_d, z_e and γ_p are defined in (7.10), (7.12) and (7.13). By making similar geometric considerations in the longitudinal plane, an analogous longitudinal LOS guidance law can be formulated as [172]

$$\gamma_d = \gamma_\mathrm{LOS} + \gamma_p = \operatorname{atan}\left(\frac{-z_e}{\Delta_\mathrm{v}}\right) + \gamma_p, \tag{7.10}$$

where the longitudinal cross-track error z_e and the flight path angle of the line segment γ_p are given by

$$\begin{aligned} z_e &= \cos(\chi_p)\sin(\gamma_p)(x - x_k) + \sin(\chi_p)\sin(\gamma_p)(y - y_k) + \cos(\gamma_p)(z - z_k) && (7.11)\\ &= \sin(\gamma_p)\sqrt{(x - x_k)^2 + (y - y_k)^2} + \cos(\gamma_p)(z - z_k), && (7.12)\\ \gamma_p &= \operatorname{atan2}\left(z_{k+1} - z_k, \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}\right), && (7.13) \end{aligned}$$

and where Δ_v is the longitudinal lookahead distance. To ensure modularity, the desired flight path angle is translated into a desired vertical position using the same LOS principles behind (7.10). The UAV still aims at a point a distance Δ_v ahead of the projection $\boldsymbol{p}^n_\mathrm{p} = \left[x_\mathrm{p}, y_\mathrm{p}, z_\mathrm{p}\right]^\top$, but considers the projection point to be on the the vector $\boldsymbol{l}^n = \boldsymbol{x}^n_{k+1} - \boldsymbol{x}^n_k$, which represents the line segment that the UAV is tracking, directly above or below the UAV $\boldsymbol{p}^n_\mathrm{UAV}$. Now, the height error h_e is vertical, as illustrated in Fig. 7.4, in contrast to the longitudinal cross-track error from (7.10), which is orthogonal to $\boldsymbol{l}^n$.

From the vertical component of the projection point, $z_\mathrm{p} = \frac{\boldsymbol{p}^n_\mathrm{h,UAV}\cdot\boldsymbol{l}^n_\mathrm{h}}{\boldsymbol{l}^n_\mathrm{h}\cdot\boldsymbol{l}^n_\mathrm{h}} l^n_\mathrm{v}$, where $\cdot$ represents the dot product, and where subscripts $_\mathrm{h}$ and $_\mathrm{v}$ indicate the horizontal and vertical components, respectively, the vertical lookahead position is computed as

$$z_{\mathrm{v,look}} = z_{\mathrm{p}} + \frac{\Delta_{\mathrm{v}}}{||\boldsymbol{l}^n||} l_{\mathrm{v}}^n \tag{7.14}$$

Similarly to (7.7), to account for possible steady-state vertical errors, the height component of (7.14) is extended with an integral term

$$\bar{z}_{\mathrm{v,look}} = z_{\mathrm{v,look}} + K_{v,i} \int h_e \mathrm{d}t, \tag{7.15}$$

where $h_e = z_p - z_{\mathrm{UAV}}$.

To make the guidance performance similar both in downwind and leewind, both the lateral and longitudinal lookahead distances should be functions of the ground speed, i.e.

$$\Delta = V_g \Delta_t \tag{7.16}$$

$$\Delta_{\mathrm{v}} = V_g \Delta_{\mathrm{v},t} \tag{7.17}$$

$$K_{v,i} = \frac{\bar{K}_{v,i}}{V_g}, \tag{7.18}$$

where $\Delta_t, \Delta_{\mathrm{v},t}$ are constant, tunable lookahead-time parameters, and where V_g is the estimated ground speed. The longitudinal lookahead time, $\Delta_{\mathrm{v},t}$, can be considered as a compensation for the response of the UAV, including communication delays and time constants in lower-level controllers and reference filters.

The guidance laws (7.8) and (7.10) can be implemented through different interfaces to the autopilot. As a consequence of the modular design goals, and under the assumption that all autopilots provide an interface to receive position references, the presented solution simply send the aggregate of the lateral and longitudinal lookahead points, $\boldsymbol{p}_{\mathrm{look}}^n = \left[x_{\mathrm{h,look}}, y_{\mathrm{h,look}}, \bar{z}_{\mathrm{v,look}},\right]^{\top}$, to the autopilot. However, if the autopilot provides an interface to receive e.g. desired course angle, desired flight path angle and desired airspeed, then χ_d and γ_d from the (7.8) and (7.10) can be used directly. Furthermore, if an interface that accepts desired roll angle, desired pitch angle and desired throttle, these values can be calculated on the basis of χ_d, γ_d and airspeed error [43, 172]. The lower-level control, regardless of the interface, is assumed provided by the autopilot.

7.2.4.1 Recovery prediction and detection

To avoid damage or entanglement in the arrest system, the motor should in some cases be stopped before it hits the arrest system. For a fixed-wing UAV in puller configuration, the propeller is the first thing that hits the arrest system, which forces the motor stop to be triggered by distance to the arrest system, not by impact detection. This implies a small risk of missing the arrest system while also deactivating the motor, which is mitigated by a starting a watchdog timer. If no impact is detected briefly after the deactivation of the motor, the recovery is deemed unsuccessful and the motor is re-activated, if possible. For an UAV using an internal combustion engine without a starter motor, reactivation in-air would not be possible and an alternative would be to set the engine to idle before recovery, although this could lead to propeller entanglement. Upon detection of impact the motor is disarmed. This impact is detected based on the longitudinal acceleration of the UAV, which is typically 5—10g during impact.

7.2.5 Operator interface

Generally, an increased level of autonomy decreases the requirements to the user interface for the operator. To allow for automatic recovery, the operator interface should let the operator initiate the recovery, while also enable performance monitoring and possibly intervention. This requires radio communication, such that the UAV and arrest system can report their states, and a user interface that displays the essence of this information, including cross-track errors, as a performance metric of the guidance controllers, desired values for the control, motion of the arrest system, the status of any automatic abort monitors, described in Section 7.2.5.1, as well as the ability to abort the landing.

There are many COTS UAV flight management graphical user interfaces (GUI) available that are used during normal UAV operations, and it is an advantage if the same interface is used in the recovery, as long as the recovery-specific requirements are met.

7.2.5.1 Aborted recovery framework

If the operator initiates abort, an emergency plan is executed. This is a simple dynamic plan, designed by the operator, that consists of a series of waypoints and a loiter, positioned relative to the current position of the arrest system. Thus, initiation of the emergency plan should bring the UAV to a loiter in a safe location, regardless of how the arrest system has moved.

In addition to being triggered by the operator, different abort triggers, that monitor different situations automatically, are implemented to relieve the burden on the operator. Examples of such situations, that make recovery impossible or highly risky, are

Loss of radio communication or loss of arrest system pose measurement: making it impossible for the UAV to know the pose of the arrest system.

Severe weather conditions: such as strong and/or unpredictable wind, increase the risk involved with recovery. A coarse wind estimate, e.g. [58], is typically monitored by the UAV autopilot, or can be implemented separately.

Poor recovery performance: The ultimate objective is to hit the arrest system, so the system predicts if this is not achievable. This is monitored by the UAV by considering its cross-track errors.

Large relative speed: from e.g. a strong tail wind can lead to high impact that jeopardizes the structural integrity of the UAV. This is monitored by the UAV, by comparing its own ground speed by that of the arrest system.

Missed catch: If the UAV passes the arrest system without registering a catch, its state is undefined, so the emergency plan is started.

However, not all situations allow for an abort [126]. Therefore, the abort framework also acknowledges the UAVs state *severity* level. This level is increased by another set of triggers e.g. if the UAV is too close to the arrest system to make a successful emergency maneuver, or if the fuel or battery is running so low that a go-around is impossible. An elevated state severity level causes the UAV to over-ride aborts, and continue the recovery regadless of some risks.

7.3 Implementation

To evaluate the arrest recovery control system architecture, the system was implemented for net recovery on a ship. A net was chosen since it occupies little space on a ship deck, which is the landing site of primary interest. Ship decks tend to be crammed, and recovery nets can be removed when not in use. Nets can also be held off the side of the ship by a crane, further reducing the space requirements and risk to the ship. The following subsections describe how the generic arrest recovery control system architecture from Section 7.2 is implemented for the specific scenario of net landing, and what adaptations have been made to accommodate a specific autopilot software and hardware.

7.3.1 Net hardware and software

The components of the net instrumentation are pictured in Fig. 7.5 and illustrated in the upper left part of Fig. 7.6. Both the base and the rover GNSS receivers are U-blox ZED-F9P, which are multi-constellation (configured to use the four global systems GPS, Galileo, GLONASS and BeiDou), multi-frequency receivers with built-in Real-time-kinematic (RTK) processing. The first UART connection of each receiver is configured to send position and velocity estimates to a SenTiBoard sensor interface and timing board [4] at 5 Hz rate, while the second UART is used for a RTCM3 correction data stream also with a rate of 5 Hz, needed for RTK. The only difference in configuration between the receivers is that the base outputs RTCM3 data on the second UART, while the rover uses it as an input. The base receiver sends the correction stream to a BeagleBone Black (BBB) single-board computer, which is set up to distribute this on the network using a TCP server, and to the rover receiver of the net over UART. The SenTiBoard sends the received estimates to the BBB over

USB for processing. On the BBB the position data is parsed, translated into net center position and heading according to (7.1) and (7.3). This is implemented as a task in the *DUNE Unified Navigation Environment* robotic middleware framework of the LSTS Toolchain [101], while the resulting net position and heading, are distributed over the network in terms of Inter-Module Communication (IMC) [84] protocol over UDP.

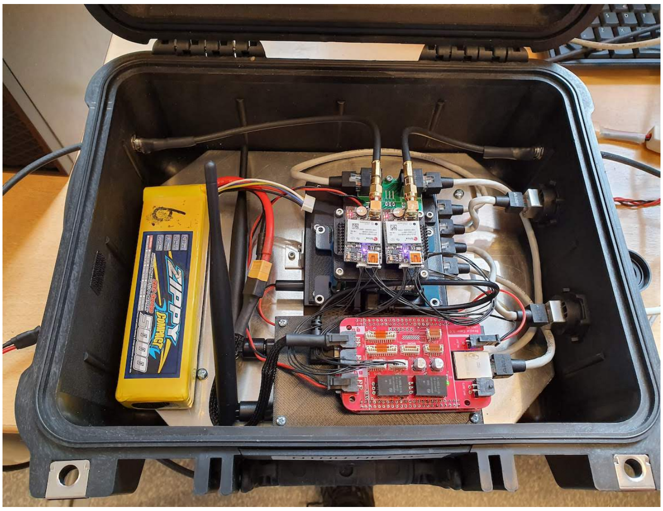


Figure 7.5: Picture of the instrumented net case, showing the two GNSS receivers ontop of the SenTiBoard and an ethernet switch. In the front is the embedded computer with a custom cape, ontop of the 5 GHz radio.

7.3.2 UAV hardware and software

The UAVs used in the experiments use ArduPlane 3.9.9 [9], running on a Pixhawk autopilot hardware in the X8 UAV, and on a Pixhawk 2.1 for the Dolphine UAV. The rest of the landing-specific payload is common to all the experiments, and is illustrated in the upper right of Fig. 7.6. In both situations, the autopilot is connected to a Ublox ZED-F9P GNSS receiver, that is used to aid its INS, and to both an ethernet switch and an Odroid XU4 single board computer over UART, to send and receive Mavlink telemetry data. The GNSS receiver receives the correction data from the net base receiver, through the network, via the Odroid, into the receivers second UART port. The GNSS receiver onboard the UAV essentially has the same configuration as the net rover receiver, but outputs a few other messages required by the autopilot.

In order to maintain consistent altitude estimates which are comparable between the UAV and the net, the UAV cannot rely on the internal barometer alone, which is the default behaviour in ArduPilot. The barometer is calibrated once at the time of launch, but the ground level pressure can change during a flight, leading to drift in the altitude estimate. In order to avoid this problem, ArduPlane is set to use the height from the RTK GNSS receiver as the primary altitude sensor.

The Odroid also runs DUNE, which includes the net motion prediction, plan generation, guidance, and interface to the autopilot using the MAVLink protocol, in addition to publishing the state of the UAV and net recovery system as IMC messages over UDP, making it available for the Neptus GUI (see Section 7.3.3).

The recovery plan, as described in Section 7.2.1, is generated upon request by the operator, according to the parameters, start location, and the location of the recovery system. While the transit phase is static, based on the initial estimate of the landing location[1] and heading, the remainder of the plan is dynamic, and will update as the UAV receives position updates from the net. As the heading of the arrest system has a significant effect on the location of the endpoint of the transit phase, which is static from the time it is generated, it is assumed that the heading of the arrest system does not change significantly during the transit phase. This is reasonable for short, ship-based recoveries, under the assumption that the ship will either be in transit, with a clearly defined heading, actively kept stationary, using dynamic positioning, or slowly drifting. To some extent, an increased uncertainty in the yaw motion of the arrest system can be accounted for by increasing the length of the final alignment stage. For simplicity, the Dubins path, which is computed using the Dubins path library provided in [158, 159], is represented as a sequence of waypoints. This makes the path piecewise linear, and thus not *flyable* according to [147], but by adjusting the parameter that sets the distance between the points, the performance is sufficient for this application. After it has been generated, the plan is stored in the plan database, see Fig. 7.7, and may be inspected by the operator. Upon initiation of recovery, the plan is loaded into the plan engine, which tracks the progress of the plan, and divides it into separate maneuvers. Each of the static waypoints of the transit phase are represented as a single maneuver, while the dynamic waypoints of the remainder of the plan is its own maneuver. Upon completion of one maneuver, the plan engine starts the next maneuver, by sending it to the maneuver handler. For static waypoints, the desired location is sent directly to the ArduPlane lateral L_1 guidance [7] and longitudinal TECS guidance [8] controllers, operating in GUIDED-mode. In AUTO-mode, the L_1 lateral guidance controller already supports line following. However, it is limited to static lines. So to achieve line following of dynamic lines, like the virtual runway, the ArduPlane guidance controllers are fed a desired location and an airspeed that is continuously updated by the *Fake LOS* block in Fig. 7.7.

Based on the current position of the net, and the UAVs progression along the dynamic part of the plan, the *Fake LOS* block calculates the desired destination for the UAV based on (7.8) and (7.10). However, when in GUIDED-mode, ArduPlane interprets a desired location as "go here, then loiter". As a consequence, when horizontally close to the desired location, closer than the distance set by the parameter `LOITER_RAD`, the UAV will turn to one side and start a loiter. To avoid this, `LOITER_RAD` is set

[1] For a slowly moving arrest system, or for short recovery plans, the location of the recovery system could be a reasonable estimate of the landing location.

low, and the horizontal components $x_{\mathrm{h,look}}$ and $y_{\mathrm{h,look}}$ of the desired location $\boldsymbol{p}^n_{\mathrm{look}}$ are extended in the direction χ_d to form a carrot point $\boldsymbol{p}^n_{\mathrm{carrot}}$ for the UAV to follow, when combined with the original desired height $\bar{z}_{\mathrm{v,look}}$, see Fig. 7.3. Essentially, the *Fake LOS* block transforms the desired position interface into a desired course and height interface.

The carrot point $\boldsymbol{p}^n_{\mathrm{carrot}}$ from the LOS guidance is converted into a WGS84 reference, consisting of latitude, longitude and height, before it is passed to ArduPlane. As ArduPlane does not do line following in this setup, the integral effect in the L_1 guidance controller is disabled. To reduce the need for the integral term in (7.7) and (7.15), it is advantageous to precisely determine the correct attitude misalignment of the autopilot, and account for this in the `AHRS_TRIM_X` and `AHRS_TRIM_Y` parameters to reduce the cross-track error.

From the desired height, and the desired airspeed, the ArduPlane TECS guidance controller calculates the desired pitch angle and throttle command based on the energy balance[2]. One important parameter is `TECS_SPDWEIGHT`, which weighs the importance of speed tracking against the importance of altitude tracking. During recovery, altitude tracking becomes relatively more important than airspeed tracking, compared to normal flight, so `TECS_SPDWEIGHT` is set to zero. In this configuration, airspeed is controlled by the slow throttle dynamics, while altitude is controlled by the fast elevator dynamics. Another important adjustment to TECS is to set `GLIDE_SLOPE_MIN` to zero. By default, ArduPlane smooths all jumps in altitude that are larger than this value, so by setting it to zero DUNE is given greater authority and less delay. In addition to these parameters, the L_1 and TECS controllers, and the lower level pitch and roll controllers, should also be tuned for a fast response, to compensate for rapid movement of the arrest system.

The recovery detection of Section 7.2.4.1 is implemented as a separate task in DUNE that subscribes to the distance to the net. Once this value is below a threshold, the ArduPlane parameter `THR_MAX` is set to zero, effectively cutting the electric motor. The threshold is set to be dependent on the speed of the UAV relative the net, to be invariant to wind. When setting this threshold, communication rates from the net to the UAV should be considered, so that the motor is stopped before the net impact even with slow communication.

7.3.3 Ground station software

In this prototype implementation, Neptus [25] was chosen as the basis for an arrest system recovery GUI module. Neptus is the ground station component of the open-source LSTS Toolchain, allowing communication with DUNE using the IMC protocol. Specially made plugins provide two main features. First, the operator is able to

[2] A bug was discovered in the ArduPlane TECS implementation, which lead to poor altitude tracking. It was fixed, and has been included in the ArduPlane 4.0.6 release. See `https://github.com/ArduPilot/ardupilot/pull/12822`

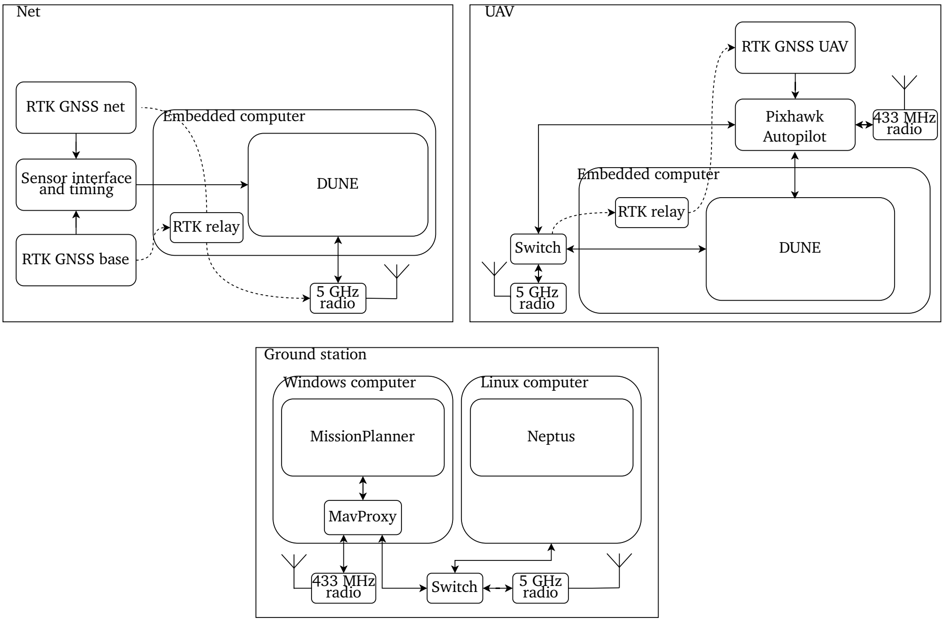


Figure 7.6: The different subcomponents of the UAV, the ground station and the instrumented net, that are relevant for the recovery

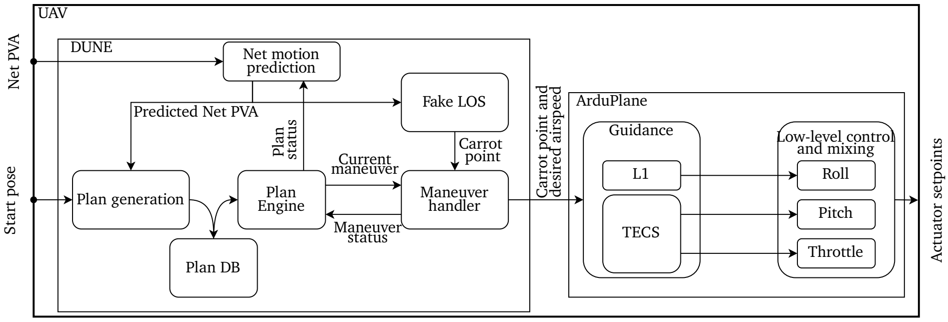


Figure 7.7: Block diagram of the UAV control architecture. Each block within DUNE roughly corresponds to a separate task, with IMC messages being passed between them.

decide the parameters that are used when the recovery plan is generated. This includes

- selecting the starting point for the recovery plan, or select that it should start from the current UAV position,
- select the desired recovery system, which enforces the UAV to only receive arrest system position messages from the selected recovery system,
- the different distances and angles illustrated in Figs. 7.1 and 7.2, with typical values given in Table 7.1.

This interface also presents the generated plan to the operator, to allow validation, see Fig. 7.8. Secondly, the GUI contains a display where the operator can monitor the progress of the UAV along the recovery plan, including cross-track errors and a prediction of where the UAV would hit the arrest system given its current position, course and flight path angle, see Figs. 7.9 and 7.10. Neptus also includes a button that will abort the recovery attempt.

Table 7.1: Parameters and typical values for small UAVs

Parameter	Typical range
R	50 m to 200 m
d_{align}	0 m to 100 m
$\gamma_{p,\text{transit}}$	3° to 10°
d_{approach}	100 m to 400 m
$\gamma_{p,\text{approach}}$	5° to 20°
d_{final}	100 m to 400 m
$\gamma_{p,\text{final}}$	3° to 10°
d_{after}	20 m to 150 m
$\boldsymbol{p}_{\text{offset}}$	0 m to 10 m in each axis
$\boldsymbol{p}_{\text{start}}$	(Geodetic) start position of the recovery plan

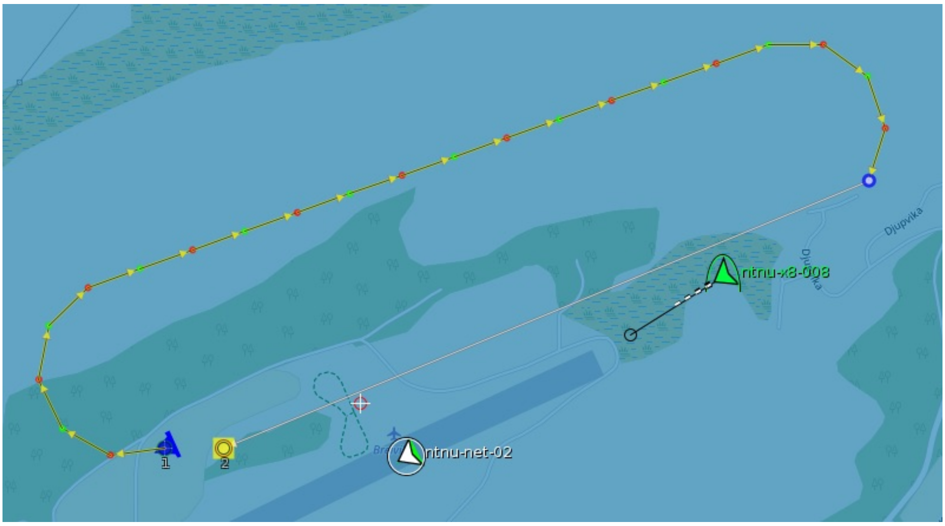
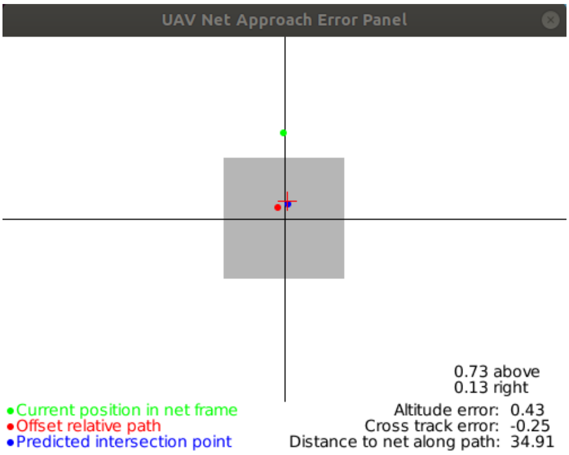


Figure 7.8: The recovery plan generation GUI, with the start position (blue), the arrest system (white), the UAV (green) and the plan inbetween. The small black circle ahead of the UAV corresponds to the carrot point, while the dynamic plan is not in the map view.


UAV Net Approach Error Panel

0.73 above
0.13 right

● Current position in net frame
● Offset relative path
● Predicted intersection point

```
          Altitude error:  0.43
      Cross track error:  -0.25
Distance to net along path:  34.91
```

Figure 7.9: The GUI recovery profile plugin, illustrating the net (here drawn with a size of 5 by 5 meters), current UAV position relative the path consisting of the errors y_e and z_e (red dot), predicted net impact point (blue dot) and current position in a NED-frame rotated around the z-axis to align with the net heading (green dot). The red cross marks the net impact point of the latest completed recovery.

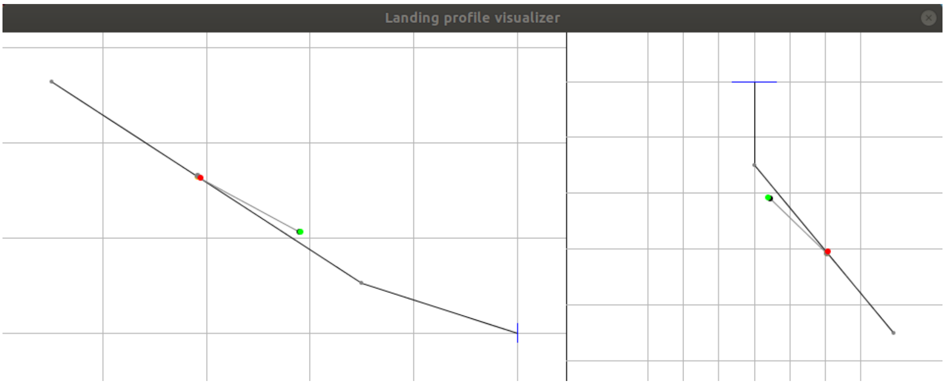
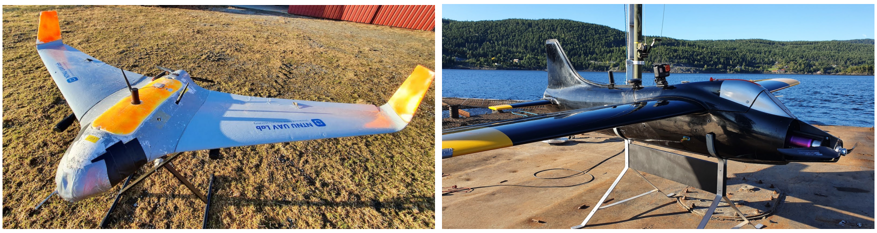


Figure 7.10: The arrest system approach path visualization plugin, illustrating the segments of the approach path, as well as the current UAV position $\boldsymbol{p}^n_{\text{UAV}}$ (red dot) and the carrot point $\boldsymbol{p}^n_{\text{carrot}}$ sent to the autopilot (green dot). Because of the integral effect in (7.7), the horizontal component of the carrot point does not have to lie on the line segment even if the cross-track error is zero. Similarly, the carrot-point height does not have to lie on the path due to differences in longitudinal and lateral lookahead distances, and lateral carrot point extension. In order to better show the errors in height and in cross-track, these are scaled independently of the horizontal distance towards the net, filling the available window space. The left side shows a vertical profile of the path, with grid marks with 100m spacing in the horizontal and 25m in the vertical direction. The right side shows a horizontal plane view, with grid marks with 4m spacing sideways and 100m in the direction towards the net. The arrest system, exemplified by a net, was rotated after the plan generation, to also show the lateral changes.

7.4 Experimental validation

Initial verification of the software running in DUNE during development was performed using simulation, with Ardupilot running as software-in-the-loop on a laptop and the net position and attitude coming from a simulated vehicle in DUNE. In simulation the net was tested both as stationary and moving, with playback of logged position and attitude from a seismic support vessel motion reference unit providing realistic movement. Two physical experiments are described in this section. The first demonstrates the performance of the system with a stationary net, to isolate the control performance from the motion of the net. The second series of tests involve recovery when the net is placed on a barge, towed behind a moving ship. The two tests also use two different airframes, to demonstrate the flexibility of the system. The first test used a Skywalker X8 styrofoam flying wing UAV with an electric motor and a pusher propeller, see Fig. 7.11a, that has a wingspan of 2.1 m, a takeoff weight of about 3.5 kg, and cruise speed of 18 m/s. The second test used a Maritime Robotics Dolphine, see Fig. 7.11b, with an electric motor and a puller propeller, elevator and ailerons. Its wingspan is 1.8 m, the takeoff weight is 9.3 kg, while its cruise speed is

26 m/s. The airframe, actuators and autopilot hardware of the two UAVs are different, while the hardware that the recovery software runs on is simply moved from one UAV to the other, which illustrate the modularity of the system. The same net instrumentation is used in all the experiments.



(a) The NTNU Skywalker X8 flying wing UAV (b) The Maritime Robotics Dolphine UAV

Figure 7.11: The fixed-wing UAVs used in the experiments

During both the experiments, the UAV and net are connected to a ground control station over a data link based on Ubiquiti Rocket radios, using the AirMax communication protocol. The ground station, depicted in the lower part of Fig. 7.6, consists of two computers; one running Ubuntu Linux and another running Windows 10. The Linux computer runs the Neptus ground control station, with the recovery GUI, which is used to visualize information and to interface the payload on both the UAV and the net. The Windows computer runs MissionPlanner, the ArduPlane ground control software, which is used as a backup for the Neptus GUI for communication with the UAV. In addition to communicating with the UAV using Mavlink messages over UDP, the Windows computer also communicates using a 433 MHz telemetry radio, for redundancy. The Mavlink messages are then fused using MavProxy.

7.4.1 Experiments with stationary net

A preliminary test of the system with stationary net instrumentation, but without a physical net catching the UAV, was performed with the X8 UAV shown in Fig. 7.11a. This allowed looping the recovery plan to perform multiple recovery attempts in a single flight with lower risk. 43 recovery maneuvers were performed with a 220 m long approach with 9° glideslope and a 190 m long final alignment with a 4° glideslope. Winds were calm without significant gusts. The position of the UAV for all attempts are shown in Fig. 7.12. The top plot showing the sideways movement of the UAV indicates weak oscillating motion which could be caused by too high integral gain or too short lookahead distance in combination with time delays in the communication between the UAV and DUNE[3].

[3] To illustrate the development progress and these preliminary results, see `https://youtu.be/nMON_udjtiE`

The position where the UAV would have impacted the net is depicted in Fig. 7.13, which shows a tendency to hit slightly below the target, but no clear tendency sideways. This is also supported by the average impact position, as reported by performance statistics in Table 7.2.

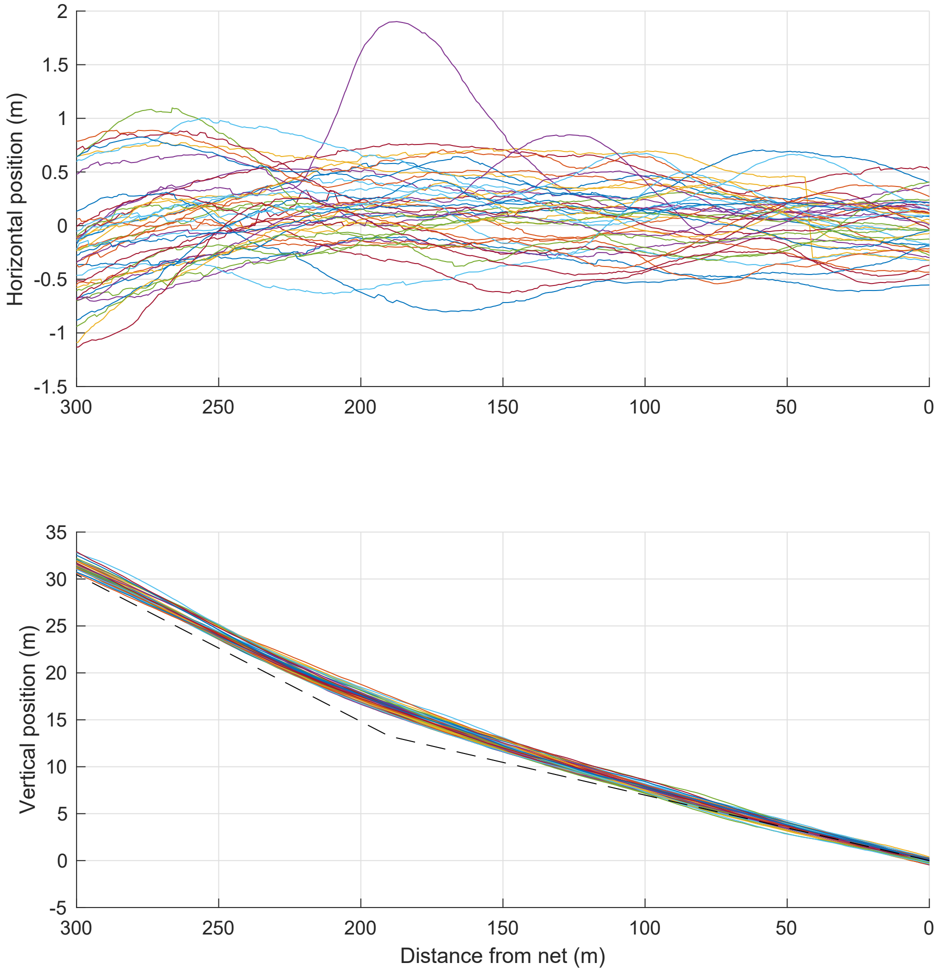


Figure 7.12: The position of the UAV in the arrest system frame while approaching the net, for all attempts. The planned descend profile is shown as a dotted line.

7.4.2 Experiments with moving net

To test the arrest recovery system in a more challenging and realistic environment, the net was mounted on a barge, towed behind a ship, depicted in Fig. 7.14. For these experiments a modified net rig with telescoping poles and fixed antenna mounting

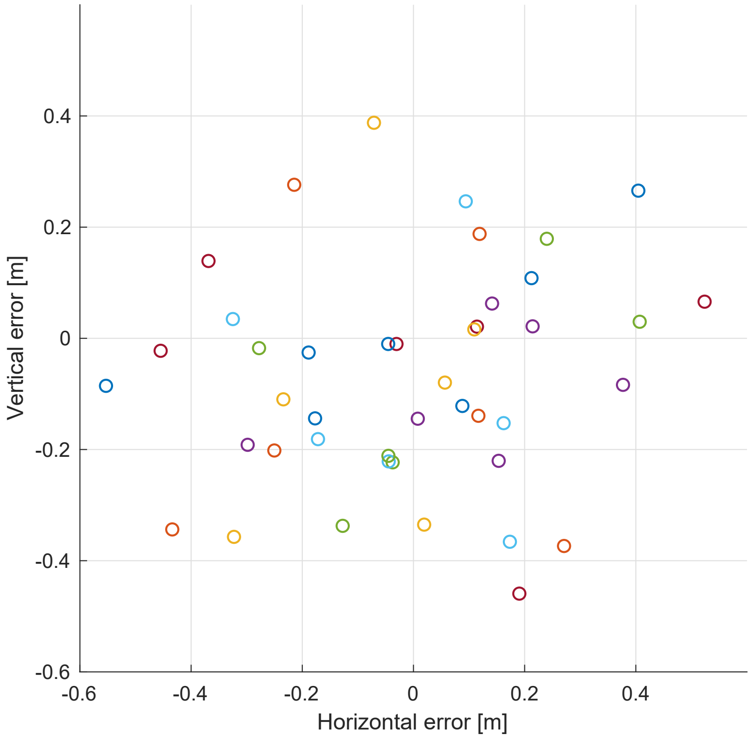


Figure 7.13: The position of the UAV when impacting the net, for all 43 attempts in a stationary net, as seen into the net from the approaching UAV.

Table 7.2: Net impact performance of 43 recovery maneuvers using a stationary net

	Vertical	Horizontal
Mean	-0.07m	-0.01m
RMS	0.21m	0.25m
Std. dev	0.20m	0.25m

points was used, to simplify the mounting on the barge. The barge is about 8 m wide and 5 m long. When towed by the ship, it has a maximum speed of about 2.5 m/s. By adjusting the ropes used in the towing, the angle between the net velocity vector and heading can be adjusted, to test different scenarios[4].

An overview of the different scenarios used in the 15 recoveries of the Dolphine UAV made in the barge-mounted net is given in Table 7.3, which lists the horizontal and vertical error in the point of impact, as seen from the UAV flying into the net, for the different recoveries. All the scenarios use an approach of 225 m at a flight path angle of 7° downward, while the final alignment is 225 m long, descending with 3°.

[4] A video showing the setup and some of the recoveries can be found at https://youtu.be/n4XhzcKLgm8

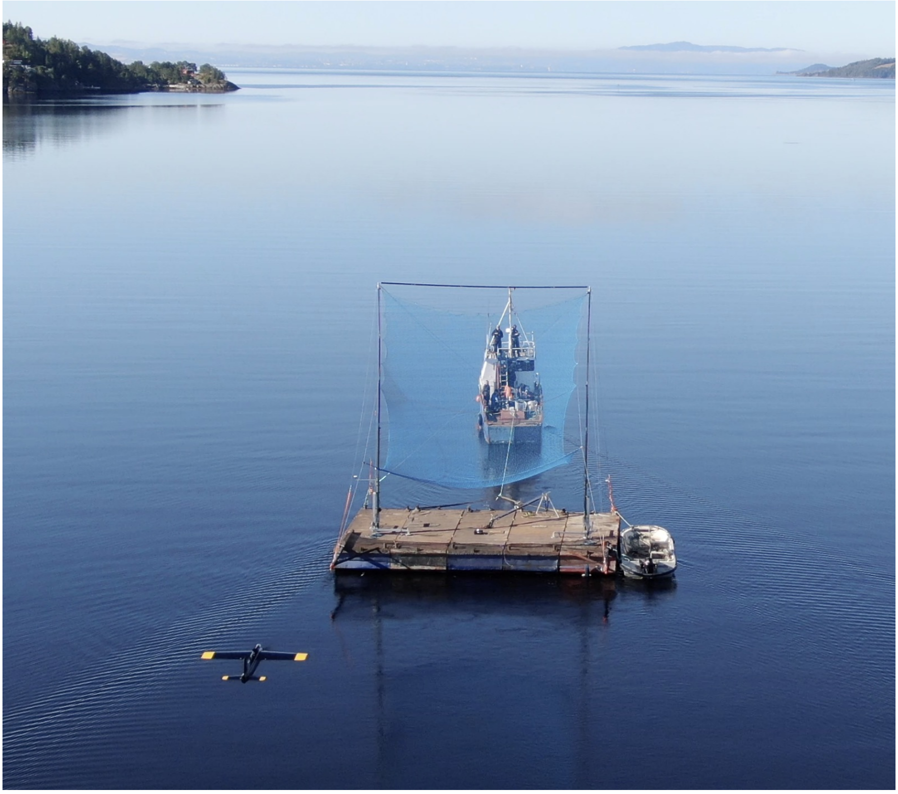


Figure 7.14: UAV approaching the barge-mounted net towed by the ship

To illustrate the environmental conditions for each recovery attempt, the speed of the net and its relative direction, as well as the wind speed and direction relative the net, are also given. In recovery 4, a yaw motion of the net of approximately 0.5° per second was initiated approximately 40 seconds before impact, meaning the UAV had to correct its approach course by 20 degrees while approaching the net. The same yaw rate was initiated approximately 20 seconds before impact in recovery 5. In recovery 15, the net was yawed by 5 degrees over a 10 seconds period starting 18 seconds before impact, then yawed back again before impact.

All the points-of-impact are plotted in Fig. 7.15, where the numbers correspond to the recovery attempt number given in Table 7.3. The vertical and horizontal trajectories of the UAV as it approaches the net are seen in Fig. 7.16, where the trajectories have been rotated by ψ_{arrest} around the down-axis to ease the comparison. As the trajectories in these figures are relative to the net, some of the error can be attributed to the movement of the net. This is particularly true for the recovery attempts with large sideways velocity, as only the approach and the final alignment stages utilize the prediction of the ship motion. This is materialized as a larger initial cross-track error to the right of the path, as the barge and net are moving to the left while the

Table 7.3: Results from moving net recoveries. Directions are relative ψ_{arrest}

	Net		Wind		Impact position error	
Recovery #	Speed (m/s)	Dir (°)	Speed (m/s)	Dir (°)	Horizontal (m)	Vertical (m)
1	0.9	−2	<1	-	0.13	−0.87
2	0.8	−2	<1	-	0.03	−1.32
3	0.8	−7	<1	-	0.03	−1.44
4	1.0	6	2	−157	−0.83	−0.79
5	0.9	1	4	10	−1.11	−1.00
6	1.0	−22	4	−28	0.53	−1.13
7	0.6	−25	4	−74	0.05	−1.38
8	0.9	−4	<1	-	0.15	−0.52
9	1.0	−2	<1	-	0.24	−0.69
10	0.9	3	<1	-	0.24	−0.57
11	2.5	−32	1	72	0.25	−0.91
12	1.1	2	3	6	0.08	−1.01
13	1.7	−85	4	−94	0.94	−1.03
14	1.7	−85	4	−99	0.63	−0.88
15	1.7	−33	2	−40	−0.52	−1.12
Average					0.06	−0.98

start point of the approach phase is fixed when the plan is generated, and is not considering net motion afterwards. The cross-track error in the final alignment phase for the recovery attempts with a large sideways velocity or yaw rate seem large and to the left of the path, as Fig. 7.16 consider the error relative to the *actual* position of the net, while the UAV aims at the *predicted* position for the time of impact. As the UAV approaches the net, these errors approach zero, as the predicted positions approach the actual positions. However, due to some communication delays that are not accounted for in the net motion prediction, the impact of the UAV ends up to the right of the net center, as it is lagging slightly behind the net. Furthermore, the simple net motion prediction does not account for any rotation of the net, which causes a larger error in scenarios 4, 5 and 15, where the a yawing motion was performed by the barge. In the straight approaches, the communication delays will also cause the UAV to believe that the net is closer than what it actually is, as the net has moved slightly during the communication delay. This could explain some of the height error. Another small contribution to the height error is the power-off of the motor before the impact. From the vertical position it would seem like there is a large error in the approach phase, ending 225 m before the net, as most of the trajectories approach at a much lower angle than the dashed line. This, however, is simply an artifact caused by the forward motion of the net, and the stationary Dubins path. As the net moves forward, the virtual runway moves with it, while the end of the Dubins path remains fixed. This causes the descent of the approach phase to be more gentle. As the height error seems systematic, it could possibly have been reduced by increasing the height integral effect or directly compensated for.

It is noted that the wind in Table 7.3 is based on the autopilot wind estimate, which is believed to give a reasonable representation of the average wind conditions during the approach and final alignment stages, but is not fast enough to accurately estimate wind gusts. From the results, there is no clear tendency in how this average wind affects the impact error, which is reasonable given the course-based guidance, and it is believed to be dominated by the effects of the communication delays. An example of this is the similar recoveries 7, 13 and 14, where the larger impact error in recoveries 13 and 14 are attributed to the larger sideways velocity of the net.

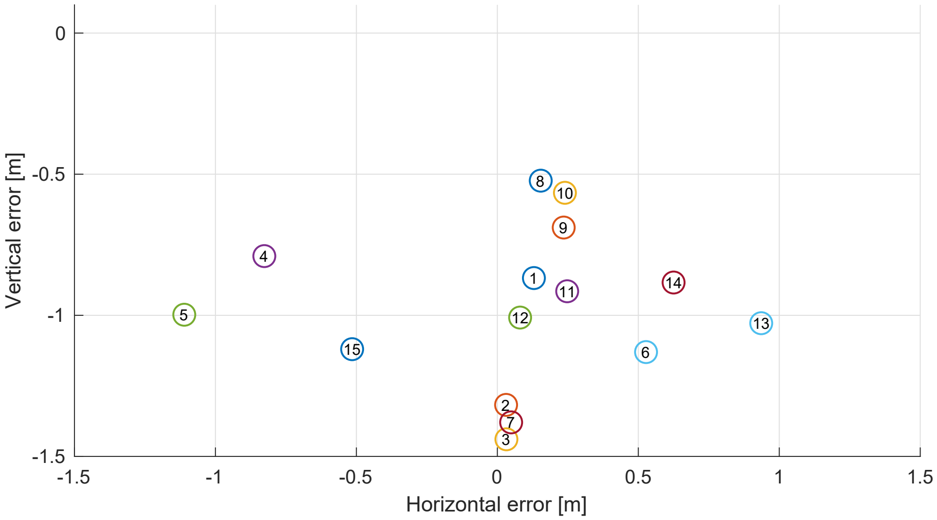


Figure 7.15: The position of the UAV when impacting the net, for all 15 attempts in a moving net, as seen into the net from the front.

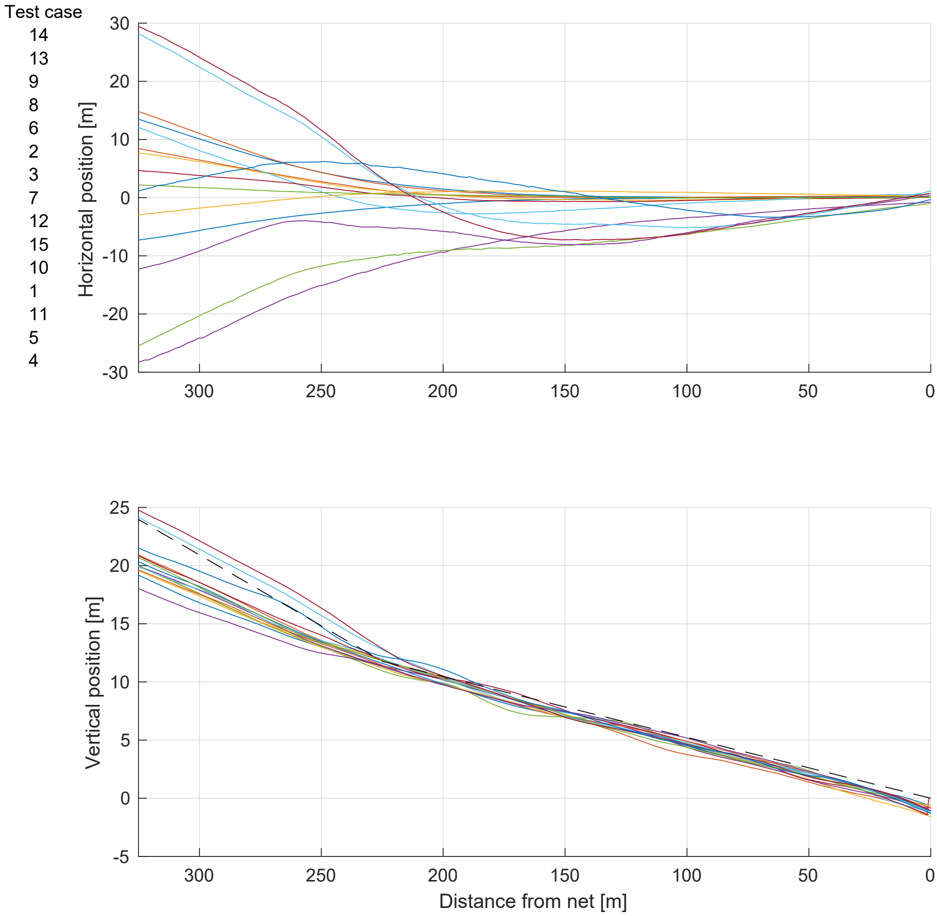


Figure 7.16: The position of the UAV in the arrest system frame while approaching the net, for all 15 attempts in a moving net. Positive horizontal position errors are to the right as seen from the UAVs perspective. For the horizontal position plots the test cases are numbered by the horizontal position each case has when entering the figure at 325m distance from the net, with the case having the largest positive position listed first.

7.5 Concluding remarks

7.5.1 Discussion

The results show that the presented recovery system is able to reliably recover the UAVs in an arrest system of a size that would fit on many ships, in the tested environmental conditions which had even winds without significant gusts, and small waves. A test on a ship in more challenging conditions would give better understanding of the limits of the system and the net size required to cover all reasonable flight conditions.

Industrialization of the proposed architecture would require changes to increase its robustness to equipment failure. Software or hardware failure of the computer running the recovery software or the serial communication link with the autopilot is not handled in the implemented system, as the autopilot mode used is intended for single "go here, then loiter"-behaviour, not inputs at a fixed rate. A loss of position input is therefore not considered a failure, although it could lead to unfortunate situations in our case, with the UAV starting a small loiter around the position last received. This could be mitigated by extending the autopilot to include a watchdog that triggers a pre-defined action in the event that it stops receiving setpoints.

In a case where the arrest system can move and yaw significantly during the transit phase, the transit phase should be made dynamic. Re-planning of the transit could be done either continuously or if the arrest system movements pass set thresholds.

The communication delays causing increased arrest system impact position errors should be compensated for by improving the timestamping and clock synchronization of the UAV and the net case computer, for example by using UTC timestamps from the GNSS receivers.

7.5.2 Conclusion

Two test campaigns, with two different UAV platforms, demonstrated the modularity, reliability and performance of the presented arrested recovery system, where the average error norm of 43 recovery attempts in a stationary net was 0.30 ± 0.14m, while 15 recoveries in a moving net had an average error norm of 1.10 ± 0.30m. Although the results are deemed sufficiently accurate for the presented setup, remaining error sources are mostly systematic, like the simplistic motion prediction and communication delays, were discussed, as correction for these will enable recovery of larger UAVs or use of smaller arrest systems.

Chapter 8

Automatic Recovery of Fixed-wing Unmanned Aerial Vehicle Using Bluetooth Angle-of-Arrival Navigation

This chapter is based on the publication

- [131] M. L. Sollie, K. Gryte, T. H. Bryne, and T. A. Johansen, “Automatic Recovery of Fixed-wing Unmanned Aerial Vehicle Using Bluetooth Angle-of-Arrival Navigation,” *Submitted, in revision*, 2022

and considers the use of Bluetooth direction finding from Chapter 4 as a navigation system for an automatic recovery system as demonstrated in Chapter 7.

8.1 Introduction

Fixed-wing unmanned aerial vehicles (UAVs) have great potential as a platform for tasks such as remote sensing or transporting goods to remote areas. A key to enabling fully automated UAV flights is the automatic recovery at the end of the mission, where a skilled pilot is typically used. Catapult launches and autopilots flying pre-determined routes are common today, while automatic recovery is not, especially without a runway. An arresting system such as a suspended net [126, 65, 47] or wire [62] can be used for this purpose, requiring only a small ground area. To perform the recovery, a system for navigation relative to the arrest system is required.

Camera-based relative navigation [70, 77] can be used for arrest system recovery [65, 53, 144, 110], measuring the directions from a visual sensor mounted on the

UAV to visible features on or close to the arrest system, or the direction from one or more sensors on the arrest system to the UAV. By utilizing visible features of known physical dimensions, the range can also be estimated using a single camera, allowing the estimation of position and not only direction. The downside of visual navigation is the requirement for visibility, with limited usefulness in conditions such as fog, snow, dust, or rain.

Another option is to use a local radio-navigation system. Large aircraft often use the Instrument Landing System (ILS) for guidance during runway approach. ILS is a radio navigation system that provides the deviation from the desired path both in the vertical and cross-track directions, allowing the autopilot to guide the aircraft onto the desired path irrespective of visibility conditions. ILS requires large and powerful transmitters and is therefore unsuitable for use outside large airports.

UAV flight using phased array radio for navigation, instead of GNSS, was demonstrated in [44], using a powerful long-range system operating on a licensed frequency band. The array was used to determine the azimuth angle from a known location to the UAV, flying over water. Due to issues with elevation angle estimation caused by multipath reflections from the sea surface, an alternative height estimate based on barometric height was used. To estimate the position, range measurements from the same system were utilized in combination with the azimuth angle, making beyond-line-of-sight flight possible.

Ultra-wideband (UWB) radio can be used for UAV positioning, using trilateration with range measurements to fixed independent anchors [22, 93]. To estimate position from these range measurements, the anchors should be positioned to create a sufficiently good measurement geometry, with a low dilution of precision (DOP), the same as is the case for GNSS pseudorange measurements. Such anchor positioning results in a larger infrastructure footprint than for antenna arrays, although it is also possible to use UWB for direction finding using arrays [20, 14].

Bluetooth direction finding using antenna arrays is another alternative for radio navigation, operating in the unlicensed 2.4 GHz industrial, scientific and medical (ISM) band. Bluetooth uses a single receiver and signal switches to sequentially connect the receiver to each element, while UWB arrays use separate receiver channels for every element. The cost for a UWB array with many elements will therefore likely be higher than for Bluetooth, although the high sampling rate of UWB can result in improved performance for multipath propagation conditions. A benefit of using Bluetooth is that the low cost of the equipment enables widespread use even by consumers. In Chapter 4 we demonstrated navigation using Bluetooth Angle-of-arrival (AoA) direction finding at up to 700 m distance. The angular direction estimate did not degrade significantly with range, even with received signal strength just over the receiver sensitivity threshold of −93 dBm, although packet loss did increase at longer ranges, especially beyond 500 m. Reflections from the ground interfering with the directly propagating signal, resulting in elevation angle estimate errors, were found to be the main error source, which was demonstrated using simulations and field experiments.

Automatic recovery of fixed-wing unmanned aerial vehicles in a moving net, using moving-base Real-Time-Kinematic (RTK) Global Navigation Satellite System (GNSS) receivers for navigation relative to the arrest system, was demonstrated in Chapter 7. RTK GNSS was used to find both the orientation of the recovery net and the very accurate and precise position of the UAV relative to the net. RTK processing requires good quality GNSS signal reception, and due to the low power of GNSS signals, unintentional or intentional signal interference can occur. It is therefore desirable to have alternatives for local navigation as an addition to GNSS for increased robustness, especially in the final phase of recovery where high precision is important.

Bluetooth direction finding can be used as an addition to GNSS for navigation relative to the arrest system, providing information of the offset from the desired path in a way similar to ILS, but with very compact ground equipment that is easy to set up. This would improve navigation resilience in case of GNSS dropouts, and allow safe vehicle recovery if GNSS has poor accuracy or is unavailable during recovery, as long as we have some method of initially getting the UAV back within Bluetooth signal range. Fixed-wing UAVs are typically able to maintain the airspeed, altitude, and pitch and roll angles required to stay airborne without an accurate position estimate, using an airspeed sensor, a barometer, and inertial sensors.

The recovery concept using Bluetooth involves placing an antenna array behind a recovery net, as illustrated in Fig. 8.1. Since the Bluetooth specification does not at present provide accurate range measurements, the position cannot be estimated without aiding from other sensors. The UAV, therefore, follows a path directly towards the array, enabling control based on direction, without the need for a distance estimate. A single array for direction estimation will yield a linearly increasing uncertainty in cross-track and vertical position along a glideslope towards the array as the range increases. For automatic recovery we are mostly concerned with intersecting a desired point on the arrest system just in front of the array, meaning that it is of little concern that the glideslope control is less precise at longer distances.

The main contribution of this paper is demonstrating the practical use of Bluetooth AoA navigation for control of a fixed-wing UAV, performing a maneuver that enables automatic recovery in a stationary arrest system. This is intended as a proof of concept and includes field experiments where AoA measurements are used in the control loop for a Skywalker X8 UAV at a range of up to 480m. Methods to handle ground reflection multipath are considered.

The paper proceeds as follows: Section 8.2 explains the mathematical notation used throughout the paper. Section 8.3 briefly explains how measurements from a Bluetooth array are used to estimate signal direction, and presents considerations for the effect of multipath signal propagation on the elevation angle estimate. Section 8.4 presents high-level path-following controllers that can be used with array direction estimates to command roll and pitch angles, and throttle, to a flight controller performing the low-level roll and pitch angle control. Section 8.5 presents the hardware and software used in the field experiments, while Section 8.6 presents the field experiments performed and their results. Section 8.7 concludes the paper with

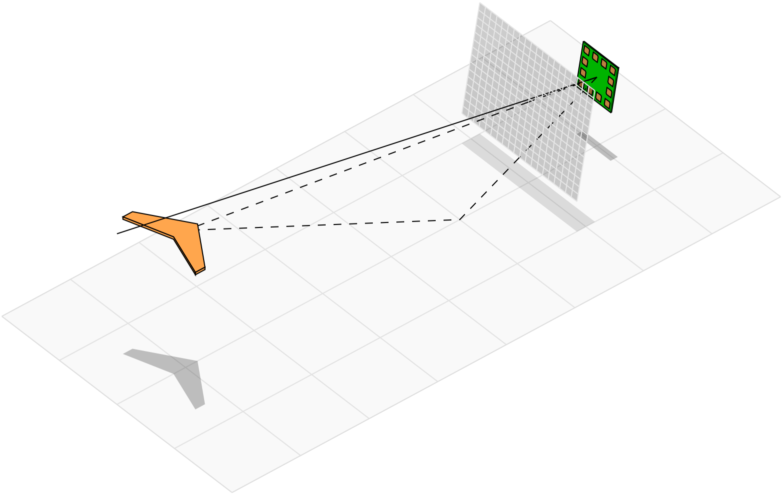


Figure 8.1: Illustration of the recovery concept: a net is placed in front of the array, and a desired path/glideslope (solid line) guides the UAV directly towards the array, through the net. The UAV, which is illustrated as flying slightly below the glideslope, transmits a signal to the array. The array measurements are used to estimate the direction from the array to the UAV, which is used by the UAV controllers to steer onto the desired path. The signal can propagate both directly to the array and as a reflection off the ground surface (dashed lines), which can affect the direction estimate.

suggestions for future work.

8.2 Preliminaries

The antenna array coordinate frame $\{a\}$ and the navigation coordinate frame $\{n\}$ are illustrated in Fig. 8.2. Frame $\{n\}$ has its origin coincident with $\{a\}$, but with axes pointing towards North-East-Down (NED). Directions in the antenna frame are parameterized using the polar angle α and the azimuthal angle Ψ. α is the angle of incidence with the array plane, which is 0 in the boresight direction and $\frac{\pi}{2}$ for a direction in the array xy-plane. Ψ is measured in the antenna xy-plane about $\boldsymbol{z}^a$ using the right-hand rule, with $\Psi = 0$ for the direction $\boldsymbol{x}^a$. In $\{n\}$ we have the azimuth angle Ψ_n measured relative to North and elevation angle α_n measured from the horizontal tangent plane. Estimated values are denoted $\hat{\Psi}_n$ and $\hat{\alpha}_n$. The UAV body frame is denoted $\{b\}$ and illustrated in Fig. 8.3. The rotation matrix from $\{b\}$ to $\{n\}$, $\boldsymbol{R}_b^n \in SO(3)$, is parameterized by the ZYX Tait-Bryan Euler angles $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top$. ϕ is the roll or bank angle, θ the pitch angle, and ψ is the heading or yaw angle. The direction of the UAV velocity vector in the horizontal plane is the course angle χ. The direction of a flight path in the horizontal plane is denoted χ_p. Subscript d, e.g. ϕ_d, denotes desired values for control.

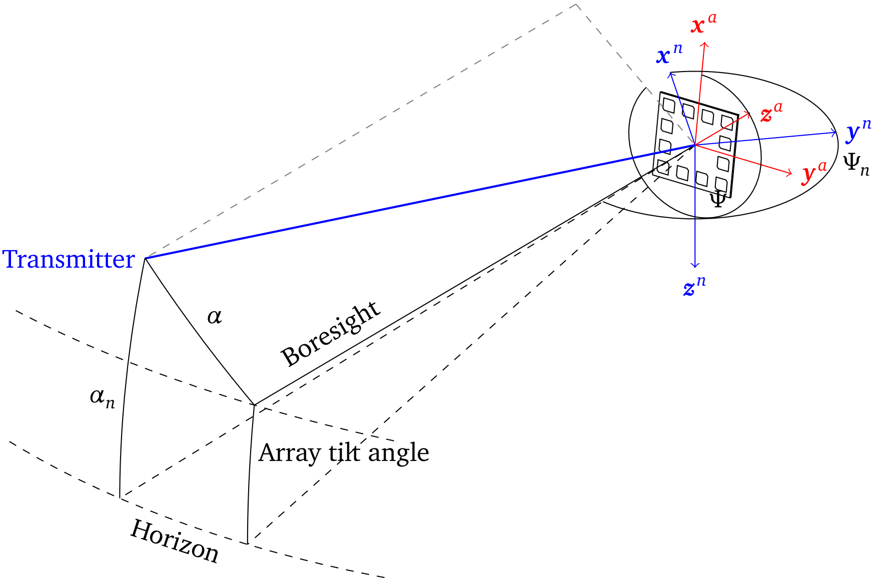


Figure 8.2: Illustration of array setup with coordinate frames $\{a\}$ and $\{n\}$, and the angle notation used for directions from the array. $-\boldsymbol{z}^a$ is the boresight direction, the direction where the antenna elements have the maximum gain.

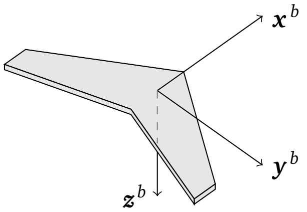


Figure 8.3: UAV body frame $\{b\}$ with the UAV seen from above.

8.3 Navigation using Bluetooth Angle-of-Arrival estimation

Direction finding using AoA estimation involves an antenna transmitting a signal, and an array of receiver antennas all receiving the same signal. For Bluetooth direction finding the signal transmitted is a pure sinusoidal carrier, the Constant Tone Extension (CTE) at the end of a Bluetooth packet. The phase difference between the received signal and an internal reference oscillator is found for all receiver antennas. Relative differences in the phase angles indicate differences in distance between the transmitter and each element. Fig. 8.4 illustrates the signal propagation geometry. The signal radiates spherically from the transmitter, but if the distance between array and transmitter is very large relative to the size of the array, the wavefront appears nearly planar when received, which simplifies processing by making it range-independent. Based on the measured distance differences for the antennas, the direction from which the signal arrives can be found. For details of the measurement processing for Bluetooth AoA estimation, see Chapter 4. The output of the complete estimation method is an antenna-frame direction α, Ψ. Using the known orientation of the array, this is transformed to navigation-frame direction parameters α_n, Ψ_n, which are used for the UAV control.

It was shown in Chapter 4 that a source of error in the elevation angle estimate α_n was the signal reflected from the ground surface and received by the array antennas

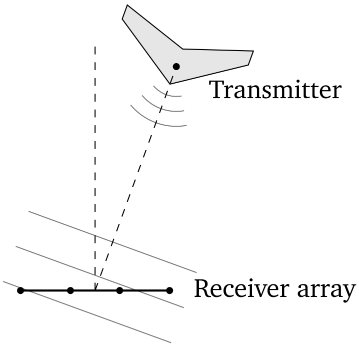
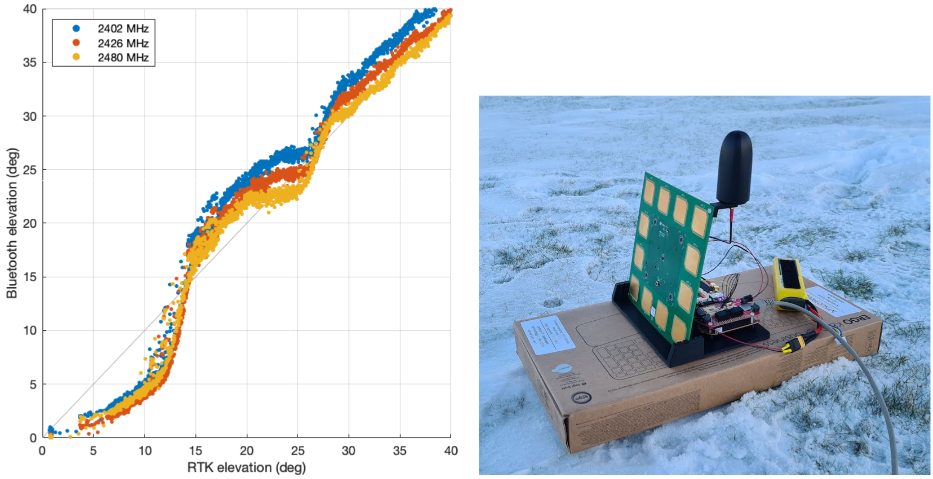


Figure 8.4: Angle-of-arrival geometry (not to scale)

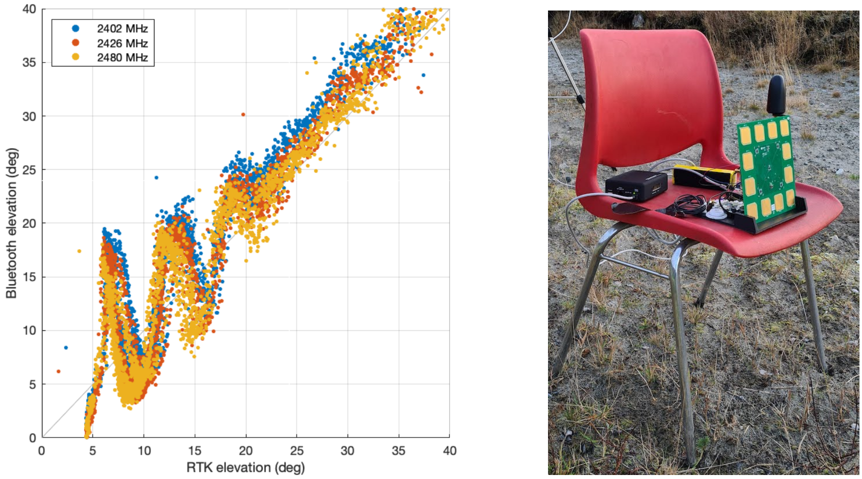


(a) Bluetooth elevation estimate plotted against RTK GNSS elevation

(b) Array setup

Figure 8.5: Effect of multipath on elevation estimate for array with center approximately 0.15m above the ground.

as an addition to the direct signal. The reflected signal causes interference, which results in an elevation-angle-dependent error in the elevation angle estimate. This error depends on several factors including the array size, its height over the ground, and the reflectivity of the ground surface. For uneven ground surfaces, different azimuth angles can also result in different elevation errors. It would be natural to assume that the use of Bluetooth direction finding near buildings or other vertical surfaces could result in the same type of error for azimuth angle Ψ_n, but as the field experiments have been conducted mostly in open areas, this has not been a significant issue. Removing the elevation error by calibration is not always possible, as the same measurement can in some cases be the result of more than one true elevation angle, i.e. the measurement for a given elevation angle is not necessarily unique. This phenomenon can be seen by plotting the elevation angle estimate as a function of the measured elevation from another system such as RTK GNSS, which provides relative positioning with errors on the centimeter level. Fig. 8.5a shows an example plot from a field experiment using the setup depicted in Fig. 8.5b, where the relationship between RTK elevation and Bluetooth elevation, for each of the three Bluetooth channels used, appears to have a one-to-one relation. A calibration correcting most of the systematic errors should then be possible. Fig. 8.6a on the other hand shows the plot from a different field experiment where the array was placed as depicted in Fig. 8.6b, further from the ground. For RTK-indicated elevation angles below 20°, a Bluetooth measurement cannot unambiguously be corrected to the true elevation angle in this case, at least when considering only a single measurement at a single frequency. This makes it challenging to use this setup for navigation at low elevation angles.



(a) Bluetooth elevation estimate plotted against RTK GNSS elevation

(b) Array setup

Figure 8.6: Effect of multipath on elevation estimate for array with center approximately 0.5m above the ground.

Even for equipment setups where calibration is possible, the calibration transform will be a function of ground reflectivity, meaning that surface material, such as grass, gravel, soil, snow, or water, and its moisture content, can make it necessary to calibrate for different conditions. For example, if it starts raining after a calibration performed in dry conditions, the error can change. However, the reflectivity only affects the magnitude of the error, so a slightly inaccurate calibration curve can be better than none, resulting in a smaller residual error than without calibration.

The glideslope angle for the descent of a fixed-wing aircraft is typically low to avoid excessive airspeed. From field experiments, it appears that the elevation error due to multipath is greatest at low elevation angles, without significant systematic error above approximately 20° to 25° for the array used. To impact the recovery arrest system, in this case a net suspended in front of the array, we want to fly along a straight-line path with a downward slope, using the Bluetooth direction measurements as the basis for guidance and control. The effect of multipath must be taken into account to make following the desired path with sufficient accuracy possible.

A straight-line path directly to the array will have a constant elevation angle α_n. The position of the net center, the glideslope angle, and the position of the array must be compatible, while also considering the multipath error the array position will produce. For a non-zero descent angle it is possible to combine a low array height with a reasonable net height, but the higher the desired net intersection point and lower the angle, the further behind the net the array must be placed. For example, a 4° net intersection slope as used for the X8 UAV in Chapter 7, with a net height of 3m and an array with its center 0.15m over the ground, requires the array to be placed approximately 41m behind the net. At this distance, the array orientation must be very accurately aligned to ensure that the glideslope intersects the desired net impact point, which is less critical with the array close to the net. Increasing the array height allows the array to be placed closer to the net, but may have undesirable effects on the elevation angle multipath error (see Chapter 4).

The measurement uniqueness requirement for calibration limits the height above ground the array can be placed, to make an elevation angle calibration possible. Therefore, we use an array mounted on a platform close to the ground in the experiments. To follow a glideslope with a constant elevation angle, the calibration only needs to be accurate at the angle chosen, with reasonable errors around it locally. The simplest such calibration is a constant offset for each of the frequency channels used, which is reasonable if the gradient of the measured elevation angle with respect to the true angle is close to 1.

8.4 Guidance and control

For control of the UAV, some measurements and systems are assumed to be available. We assume that the UAV flight controller estimates attitude using an IMU and a magnetometer, and uses these to handle low-level roll and pitch control, allowing us to command desired roll and pitch angles. A heading angle estimate is assumed available for the higher-level control, in addition to airspeed measurements from a pitot tube. To know the desired flight direction when exactly on the desired path, the array is mounted with a known azimuth angle and upwards pitch angle. The UAV's course angle χ is not assumed available, only the heading ψ. Wind perpendicular to the approach path causes the UAV to crab, with $\chi \neq \psi$. The crab angle of the UAV due to crosswind is compensated by using integral action in the controllers. To avoid integral wind-up for the initial path convergence the integrators were enabled below a specified distance from the array, with the distance calculated from elevation angle estimate and barometer height as $h/\tan(\alpha_n)$, although this could also have been handled using alternative methods.

8.4.1 Airspeed control

To maintain airspeed, the throttle is commanded using a PI controller with the measured airspeed $\hat{V}_a$ and the desired airspeed $V_{a,d}$, with the error $\tilde{V}_a = V_{a,d} - \hat{V}_a$,

$$T = k_{(T,V,p)}\tilde{V}_a + \int k_{(T,V,i)}\tilde{V}_a dt + T_{\text{trim}}, \tag{8.1}$$

where $k_{(T,V,p)}$ and $k_{(T,V,i)}$ are proportional and integral gain parameters, respectively, and T_{trim} is a throttle trim parameter.

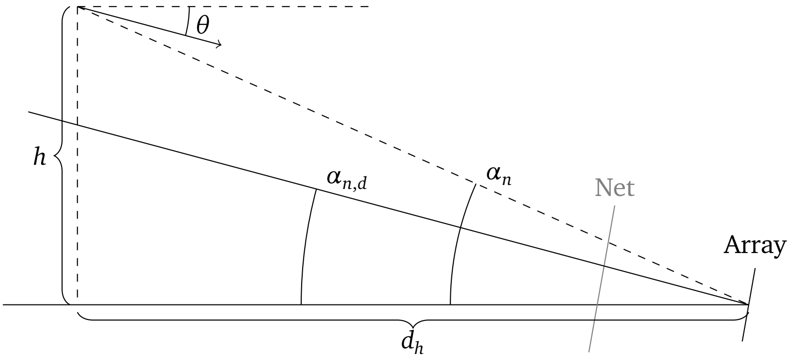


Figure 8.7: Longitudinal geometry

8.4.2 Angle-based line-following

Without knowledge of the distance between the array and UAV, we can follow a path towards the array with a desired glideslope using only angle measurements.

8.4.2.1 Longitudinal control

The longitudinal geometry is illustrated in Fig. 8.7. A pitch controller capable of following a glideslope is

$$\theta_d = -\alpha_{n,d} - k_{(\theta,\alpha,p)}(\hat{\alpha}_n - \alpha_{n,d}) - \int k_{(\theta,\alpha,i)}(\hat{\alpha}_n - \alpha_{n,d})dt + k_{(\theta,q,d)}\hat{q} + \theta_{\text{trim}}, \tag{8.2}$$

where θ_d is the pitch setpoint sent to the low-level pitch controller running on the flight controller. $\alpha_{n,d}$ is the desired elevation angle, $\hat{q}$ is the angular rate estimate in the $\boldsymbol{y}^b$ direction (essentially the pitch rate for zero roll angle), θ_{trim} is the angle-of-attack at trim flight condition, and $k_{(\theta,\alpha,p)}$, $k_{(\theta,\alpha,i)}$ and $k_{(\theta,q,d)}$ are proportional, integral and damping gain parameters.

8.4.2.2 Lateral control

The lateral geometry is illustrated in Fig. 8.8. The lateral control consists of two steps: first, the desired flight direction is determined, and second, the desired roll angle is computed from the error between the desired and actual flight direction. The desired course angle could be computed as

$$\chi_d = (\Psi_{n,d} - \pi) + k_{(\chi,\Psi_n,p)}(\hat{\Psi}_n - \Psi_{n,d}), \tag{8.3}$$

where $\Psi_{n,d}$ is the desired azimuth angle, the direction from which is assumed we want to fly towards the array. For $k_{(\chi,\Psi_n,p)} = 1$ we would aim directly towards the

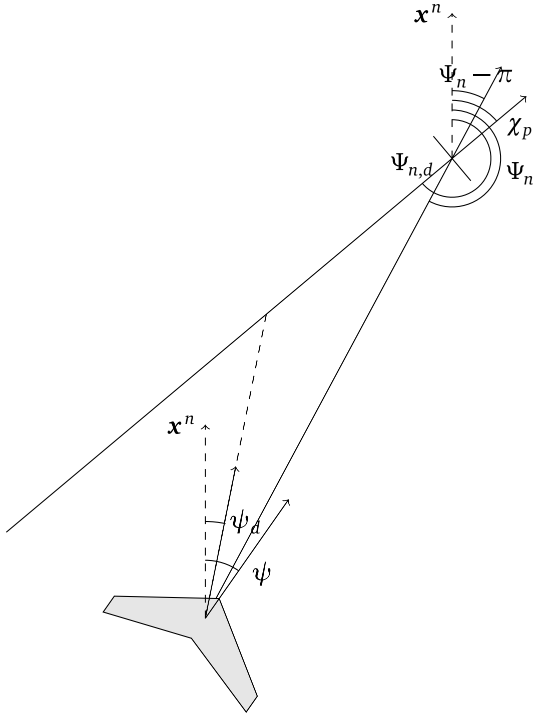


Figure 8.8: Lateral geometry

array without converging onto the desired path $\chi_p = \Psi_{n,d} - \pi$. Thus $k_{(\chi,\Psi_n,p)} > 1$ would be required for convergence. Ideally, the desired direction would be a course angle, but since we do not assume access to course angle estimates, the desired direction is a heading angle, where an integral is used to account for wind. The desired heading angle can be computed as

$$\psi_d = (\Psi_{n,d} - \pi) + k_{(\psi,\Psi_n,p)}(\hat{\Psi}_n - \Psi_{n,d}) + \int k_{(\psi,\Psi_n,i)}(\hat{\Psi}_n - \Psi_{n,d})dt \tag{8.4}$$

An issue with this very simple controller is that the proportional term $k_{(\psi,\Psi_n,p)}(\hat{\Psi}_n - \Psi_{n,d})$ can make the UAV fly away from the array for large initial azimuth errors or controller gain, $|k_{(\psi,\Psi_n,p)}(\hat{\Psi}_n - \Psi_{n,d})| > \frac{\pi}{2}$. Therefore, the controller gain would be limited by the maximum azimuth error allowed. A modification that avoids this problem is to instead use

$$\psi_d = (\Psi_{n,d} - \pi) + \tan^{-1}(k_{(\psi,\Psi_n,p)}\tan(\hat{\Psi}_n - \Psi_{n,d})) + \int k_{(\psi,\Psi_n,i)}(\hat{\Psi}_n - \Psi_{n,d})dt. \tag{8.5}$$

In this case the parameter $k_{(\psi,\Psi_n,p)}$ becomes a inverse fractional lookahead distance. For $k_{(\psi,\Psi_n,p)} = 2$, the desired heading points towards the point of the desired path where the distance along the path is half the remaining distance. For $k_{(\psi,\Psi_n,p)} = 3$ it would be $\frac{1}{3}$ of the remaining distance, and so on, and can therefore not turn away from the array even for high gains and azimuth errors.

The effect of using control based on azimuth and elevation angles is that the controller becomes more aggressive as the range decreases, when considering distance errors from the desired point on the path. For a given cross-track error, the commanded roll at twice the distance is halved, as the azimuth error is halved. The result is that the controller response is initially slow, but becomes more aggressive as the UAV approaches the array.

Using the desired heading angle ψ_d and the estimated heading angle ψ from the flight controller, a desired roll angle ϕ_d can be computed as

$$\phi_d = \text{atan}(k_{(\phi,\psi,p)}(\psi_d - \hat{\psi})), \tag{8.6}$$

where $k_{(\phi,\psi,p)}$ is a tuning parameter.

8.5 Implementation

The electronics hardware used is the same as in Chapter 4, but will be briefly reintroduced here.

8.5.1 Ground antenna equipment

Fig. 8.9 show a schematic for the hardware components placed on the ground. The array is a Nordic Semiconductor experimental reference design using 12 truncated corner right-handed circular polarization (RHCP) patch antennas in a square 15x15cm pattern, with 5cm antenna spacing, controlled and sampled by an nRF52833 board. The advertising channels with frequencies 2402 MHz, 2426 MHz and 2480 MHz are used in a connectionless configuration. Bluetooth was only used for direction estimation and not as a vehicle telemetry and command link. The nRF52833 board is connected to a SentiBoard [4], which works as a USB sensor interface for the Beaglebone Black single-board computer. The data is parsed in DUNE [73], and forwarded to the payload computer on the vehicle, where the direction estimation runs.

A uBlox ZED-F9P GNSS receiver with a helix antenna is used as a Real-Time-Kinematic (RTK) base, for use in evaluation of the positioning performance of the Bluetooth system. The components are attached to a plastic base, visible on the left side in Fig. 8.10, holding the array with a 10° upwards pitch angle. A bubble level on the plastic base is used to level the assembly. The Beaglebone Black is connected to a Ubiquiti Rocket M5 radio using ethernet, for communication with the UAV and a ground station computer used for system monitoring. The ground equipment was battery-powered, allowing remote placement.

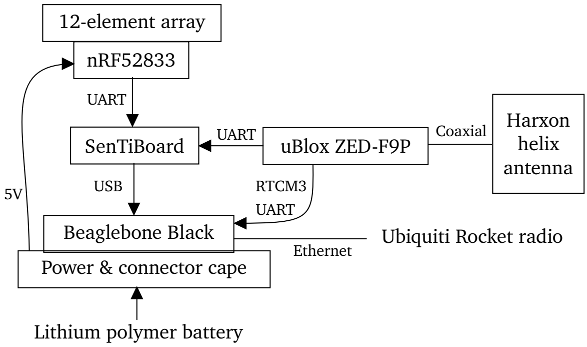


Figure 8.9: Ground hardware schematic

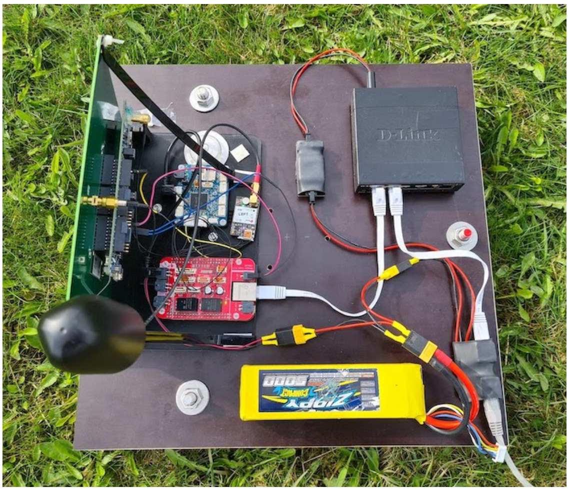


Figure 8.10: Ground hardware mounted to leveling platform. The network switch visible in the top-right corner of the platform allows connecting a laptop if needed but was not strictly necessary during normal operation. The small black shrink-wrapped device in the bottom right corner is a power-over-ethernet (PoE) injector for the Ubiquiti Rocket radio, running directly off a 4S lithium-polymer battery.

8.5.2 UAV payload

Fig. 8.11 shows a schematic with the relevant hardware components onboard the vehicle. A directional TrueRC Canada X-AIR 2.4GHz RHCP antenna is used. Fig. 8.12 shows the antenna mounted in the nose of a Skywalker X8 UAV pointing forwards. The antenna is specified to have a gain of 8 dB, a −3 dB beamwidth of 75° and performance equal to an omnidirectional antenna in a 120° beam [146]. The antenna is connected to the nRF52833 transmitter board using a coaxial cable. A uBlox ZED-F9P GNSS receiver with a helix antenna is used on the UAV, receiving RTCM3-format reference measurements from the base antenna mounted on the ground. This is only

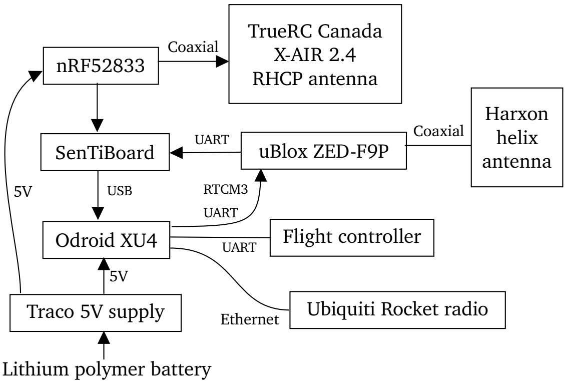


Figure 8.11: UAV payload hardware schematic

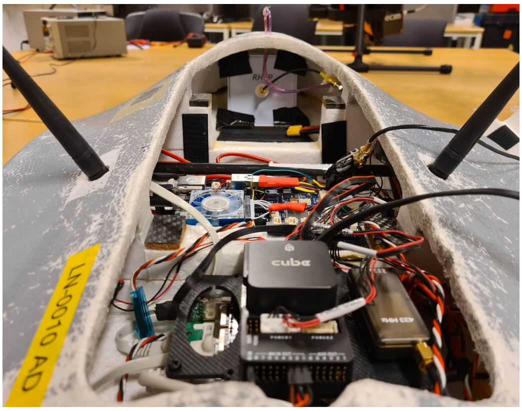


Figure 8.12: UAV payload: the RHCP antenna is visible in the front of the fuselage. The payload hardware is in the center and the Cube Black flight controller at the bottom.

used for calibration of the Bluetooth system and evaluation of Bluetooth navigation performance, not for control. The RTK GNSS setup yields very accurate and precise estimates of the UAV's position relative to the GNSS antenna on the ground, with position errors on the centimeter level.

The transmitter broadcasts advertising packets with a CTE at a rate of 10 Hz. The SentiBoard outputs measurements from all connected sensors to the Odroid XU4 computer where they are both logged for later analysis and parsed for real-time use. The Odroid and SentiBoard are shown in Fig. 8.13, where they are mounted on top of the nRF52833.

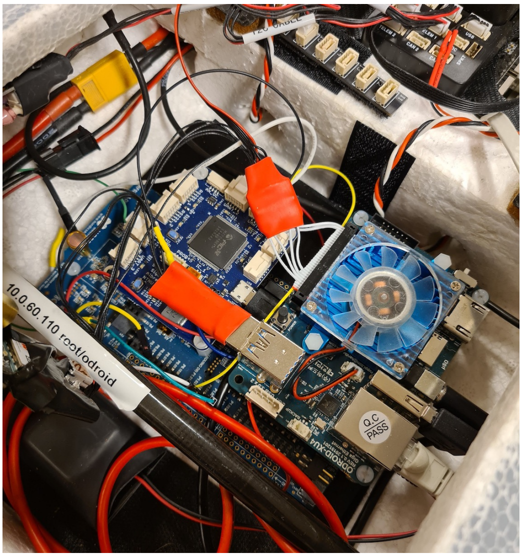


Figure 8.13: UAV payload: Odroid XU4 (right, with fan) and SenTiBoard (left, blue) on top of the nRF52833 Bluetooth transmitter board.

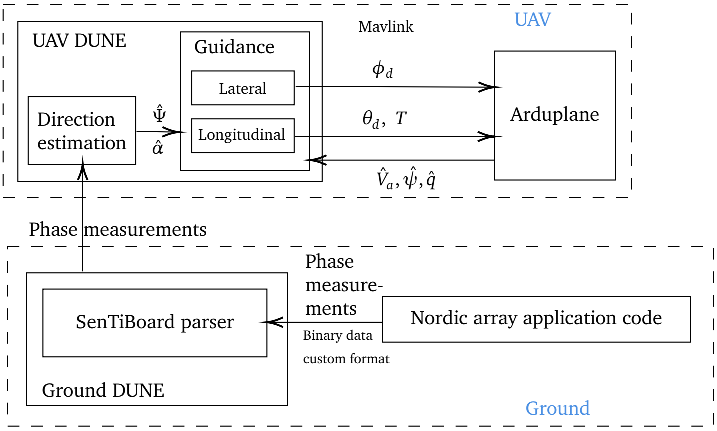


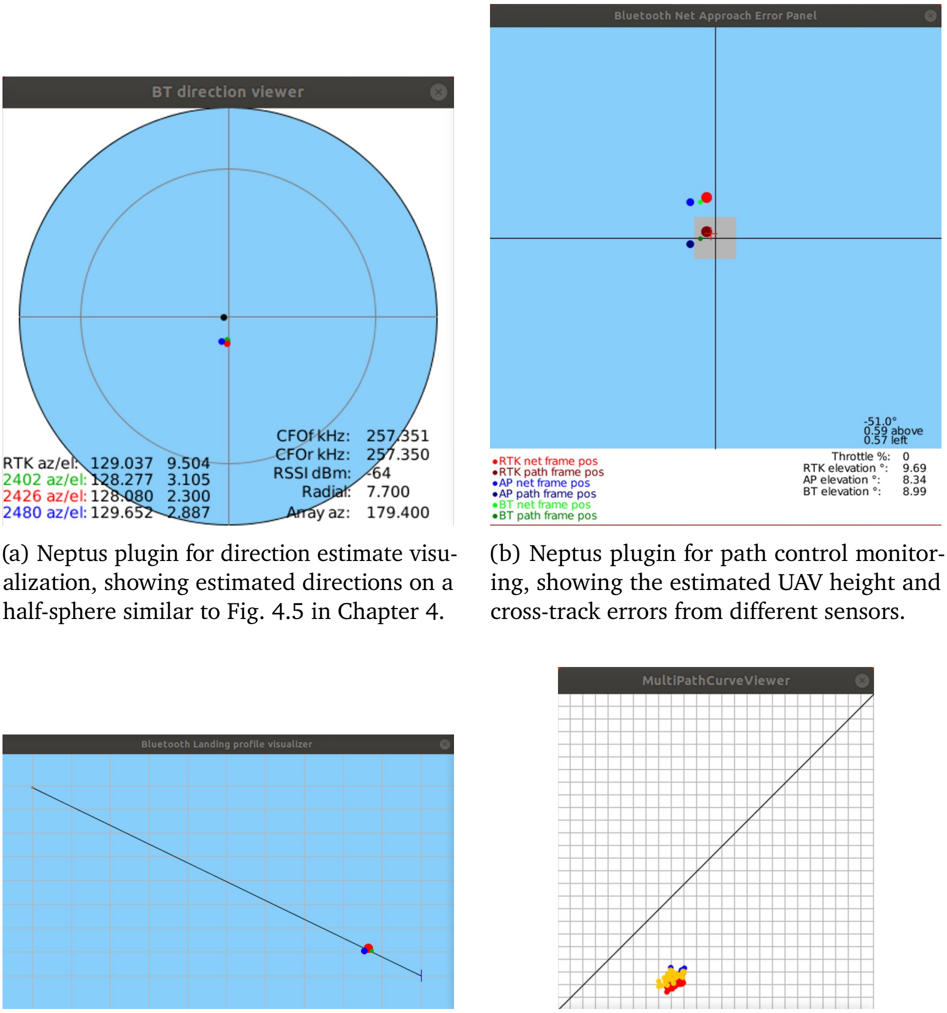
Figure 8.14: Main software components for Bluetooth net recovery

8.5.3 Software architecture

The UAV net recovery involves software running on and communicating between several different hardware components. DUNE [73], a part of the LSTS Toolchain, is used as a robotic middleware for implementing the algorithms used in a modular fashion. Fig. 8.14 shows a schematic illustrating the flow of data between the components and the most important tasks running in DUNE on the UAV and ground computers.

The packets from the receiver board are parsed in DUNE on the Beaglebone Black and forwarded to the UAV. Direction estimation using the conventional beamformer method is then performed on the Odroid XU4, using a resolution of 0.1° for Ψ and 0.05° for α. The direction estimation result is sent to the guidance and control task where the lateral and longitudinal controllers run, sending desired roll and pitch angles, as well as a throttle command, using the SET_ATTITUDE_TARGET Mavlink message to the Cube Black flight controller, which handles the low-level control of the elevon control surfaces.

Neptus, the ground station component of the LSTS Toolchain, was used on the ground to monitor the direction estimation and UAV controllers running in DUNE. The Neptus plugins implemented in Chapter 7 were adapted for use with Bluetooth, and two new plugins were implemented for visualizing the Bluetooth direction estimates. The plugins are shown in Fig. 8.15.



BT direction viewer
Bluetooth Net Approach Error Panel
Bluetooth Landing profile visualizer
MultiPathCurveViewer

RTK az/el: 129.037 9.504
2402 az/el: 128.277 3.105
2426 az/el: 128.080 2.300
2480 az/el: 129.652 2.887

CFOf kHz: 257.351
CFOr kHz: 257.350
RSSI dBm: -64
Radial: 7.700
Array az: 179.400

-51.0°
0.59 above
0.57 left

●RTK net frame pos
●RTK path frame pos
●AP net frame pos
●AP path frame pos
●BT net frame pos
●BT path frame pos

Throttle %: 0
RTK elevation °: 9.69
AP elevation °: 8.34
BT elevation °: 8.99

(a) Neptus plugin for direction estimate visualization, showing estimated directions on a half-sphere similar to Fig. 4.5 in Chapter 4.

(b) Neptus plugin for path control monitoring, showing the estimated UAV height and cross-track errors from different sensors.

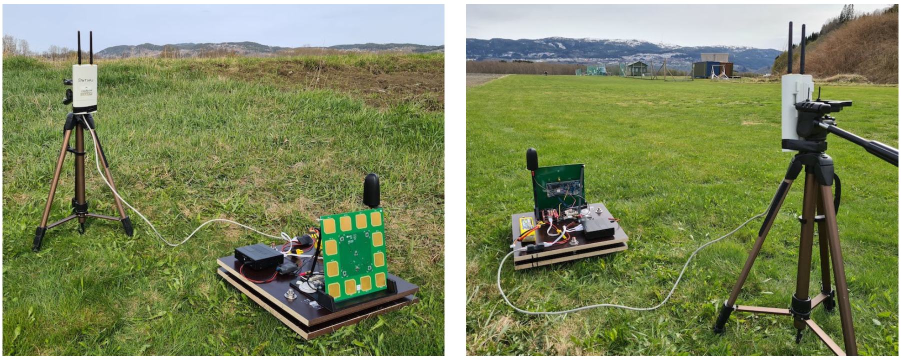
(c) Neptus plugin for path control monitoring, showing estimates for distance from and height over the array.

(d) Neptus plugin plotting Bluetooth elevation angle against RTK GNSS elevation angle for a chosen number of latest measurements, for each Bluetooth channel, similar to Fig. 8.5a and Fig. 8.6a. The grid has a resolution of 1°.

Figure 8.15: Neptus plugins for monitoring the UAV recovery approach.

8.6 Field experiments

In order to perform a large number of recovery maneuvers efficiently, tests are performed without a physical net and with the approach aborted before reaching



(a) Front view

(b) Back view

Figure 8.16: Setup of antenna array for fixed-wing UAV recovery

the array, similar to the stationary net tests in Chapter 7. In this way, the UAV can pass through the location where the net would be suspended without having to re-launch the UAV for every attempt. RTK GNSS positions were logged to assess the location relative to the virtual net center where the UAV would have impacted the net when using Bluetooth direction estimates for control.

Fig. 8.16 shows the ground hardware setup at the end of a grass runway. The bottom part of the array platform was pushed into the soil for stability, and the top plate was leveled using the bubble level on the array holder. After this, it was assumed that the array had an upwards pitch of 10 degrees and zero roll angle. For array azimuth angle calibration, The UAV was placed on the runway on a stand keeping it raised from the ground, with the nose of the UAV, containing the transmitter antenna, pointed towards the array. The azimuth angle assumed in software was then adjusted until azimuth angles for RTK GNSS and Bluetooth measurements matched, which was for an array boresight direction of 128° relative to North in this case. The three Bluetooth channels showed minor azimuth angle disagreement, but this was not accounted for in the experiments.

Elevation angle offsets were identified initially on the ground by holding the UAV in front of the array at an RTK-indicated elevation angle of 9°, corresponding to the desired glideslope angle, and then reading the Bluetooth elevation estimates for each channel. The elevation values from RTK GNSS and Bluetooth were also visualized in the GUI shown in Fig. 8.15d. The offset values were further adjusted while flying, although adjusting while performing approach maneuvers affects the actual elevation angle flown and thereby the offset required for correction, making it an iterative process. The Skywalker X8 UAV used for experiments is shown in Fig. 8.17.

An example plan uploaded to the UAV, as shown on the ground station monitor, is shown in Fig. 8.18. The plan consists of several waypoints and a maneuver for Bluetooth recovery. The angle chosen for the glideslope was set at 9°, which is the same

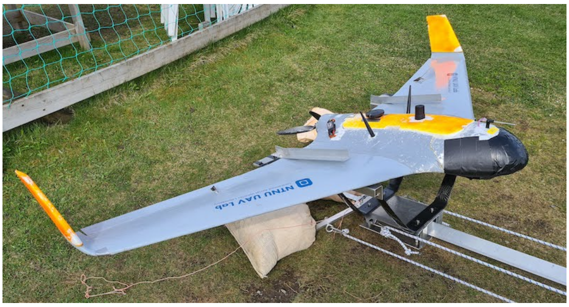


Figure 8.17: Skywalker X8 UAV. The GNSS antenna used for the RTK GNSS rover receiver is the cylinder-shaped component on top of the fuselage. The Bluetooth transmitter antenna is hidden inside the UAV behind the black canvas tape.

angle used in the initial descent of the experiments with the same UAV in Chapter 7. Once the recovery maneuver completes, the plan loops back to the beginning for another repetition. Several versions of this plan were used, with different placement of the waypoints before the start of the recovery maneuver, to verify the convergence onto the desired glideslope. Different initial heights, and start positions both on the desired horizontal path and on either side of it, were tested. Different initial heading angles were also tested. The wind was low, mostly about 1 m/s, but varying in direction, with testing performed both in headwind, crosswind, and tailwind.

Distance from GNSS was used to determine when the recovery maneuver was deemed completed. A distance of 40 m was chosen to maintain a safe distance margin over the ground, corresponding to a net center position 6.3 m over the array. Fig. 8.19 shows the UAV shortly before completion for one of the approach maneuvers. In total, 39 maneuvers were flown using Bluetooth navigation. The controller parameters used in the experiments are listed in Table 8.1.

The RTK GNSS positions in the horizontal and vertical for all maneuvers are shown in Fig. 8.20. RTK GNSS provides positions relative to the base antenna in the $\{n\}$-frame, which is transformed to the values plotted using the assumed array heading. The position where the UAV would have impacted the net according to RTK GNSS is shown in Fig. 8.21. The spread in the impact position is not significantly worse than that achieved in stationary net experiments using RTK GNSS in Chapter 7. A reasonably sized net, e.g. 5 × 5m, would have been hit for every attempt, with an RTK-indicated standard deviation of 0.41 m horizontally and 0.32 m vertically. The mean RTK GNSS impact position 0.45 m left of the desired point is likely the result of residual array azimuth calibration error, corresponding to a 0.67° error in the assumed azimuth angle. It does appear that the glideslope followed is slightly above the desired 9°, likely because of residual elevation calibration errors, resulting in impacts above the desired position. The position along the path calculated from Bluetooth direction and RTK GNSS distance is shown in Fig. 8.22. The noise in the direction estimate translates to a larger position noise for increasing range. The "sawtooth"-like pattern in the horizontal position is due to direction disagreement

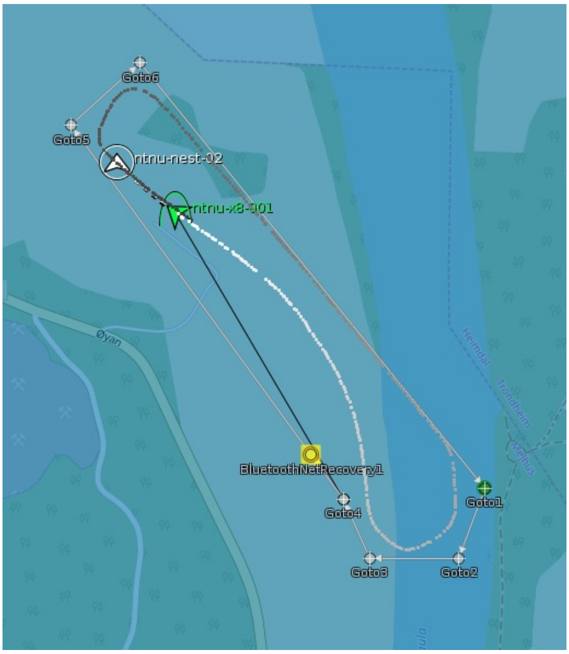


Figure 8.18: Recovery plan on the Neptus map. `ntnu-x8-001` is the fixed-wing UAV, and `ntnu-nest-02` is the ground equipment with the antenna array. The plan consists of a sequence of waypoints flown using GNSS, `Goto1`-`Goto4`, then a recovery maneuver `BluetoothNetRecovery1` flown using the controllers from Section 8.4. After reaching the net location a specified distance from the array, the approach is considered completed, and the UAV climbs towards waypoints `Goto5`-`Goto6`, before looping back to the start of the plan.

between the three Bluetooth channels used. A similar pattern would have appeared in the vertical direction if the same elevation compensation had been used for all channels, instead of the three different values in Table 8.1. It can be seen that the calculated vertical position based on Bluetooth elevation is higher than for GNSS at the longest distances, which can partially be attributed to the constant offset elevation angle compensation overcompensating at lower elevation angles, although the difference in measured elevation for different maneuver starting heights is smaller than expected. The elevation angles from RTK GNSS and Bluetooth direction finding are plotted against each other in Fig. 8.23, showing the multipath effect on elevation angle estimation for the array setup used. Note the increased spread of the elevation measurements at elevation angles below the 9° glideslope. When the maneuver is initialized, the UAV is below the glideslope, as seen in Fig. 8.20. From the time when the maneuver is initialized until the UAV has reached the desired path the azimuth angle changes, which also changes the point on the ground where the signal reflects. Since the grass surface is not perfectly flat, the multipath effect on elevation can change in this period, resulting in the measurement spread observed.

Figure 8.19: Skywalker X8 approaching the antenna array

Table 8.1: Maneuver parameters

Parameter	Value
$k_{(T,V,p)}$	$10\,\frac{\%}{\mathrm{m/s}}$
$k_{(T,V,i)}$	$2\,\frac{\%/\mathrm{s}}{\mathrm{m/s}}$
T_{trim}	45%
$k_{(\theta,\alpha_n,p)}$	12
$k_{(\theta,\alpha_n,i)}$	$0.25\,\frac{\mathrm{rad/s}}{\mathrm{rad}}$
$k_{(\theta,q,p)}$	0.05s
θ_{trim}	1.5°
$k_{(\phi,\psi,p)}$	$1.5\,\frac{\tan(\mathrm{rad})}{\mathrm{rad}}$
$k_{(\psi,\Psi_n,p)}$	3.5
$k_{(\psi,\Psi_n,i)}$	0.3
Distance to enable integrators	300 m
Elevation offset 2402 MHz	6.0°
Elevation offset 2426 MHz	6.6°
Elevation offset 2480 MHz	6.1°

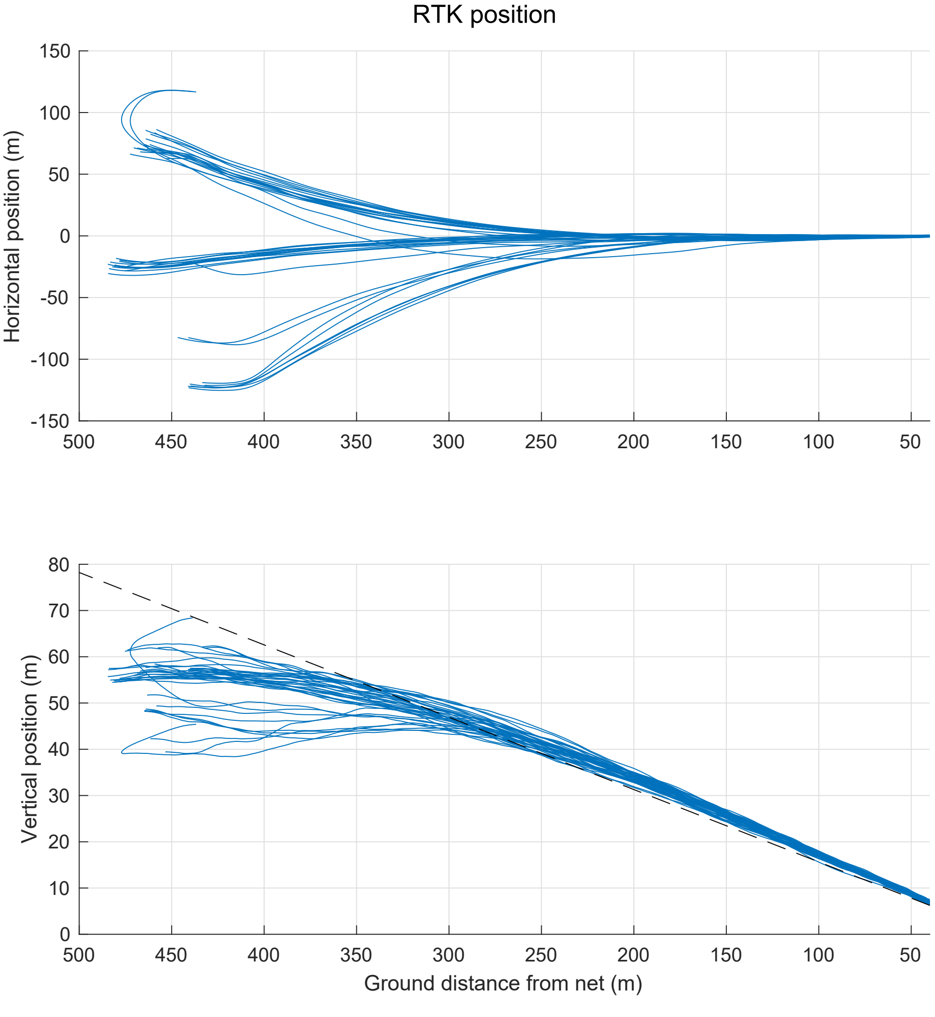


Figure 8.20: UAV position plots from RTK GNSS for the 39 approaches. For some of the approaches the maneuver is initialized with a course angle error over 90°, such that the distance from the array is increasing.

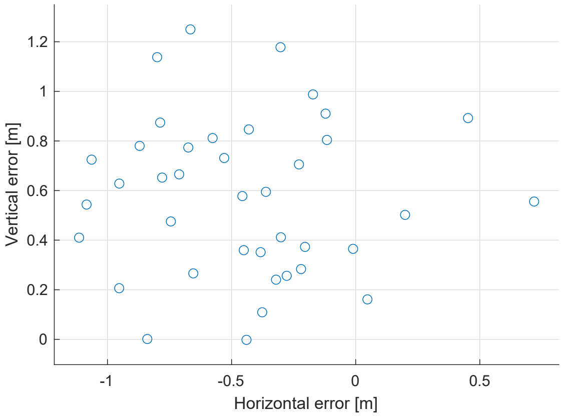


Figure 8.21: Impact positions as seen from the glideslope towards the array, calculated from RTK GNSS position with the assumption of 128° azimuth and 10° elevation angle for the array orientation.

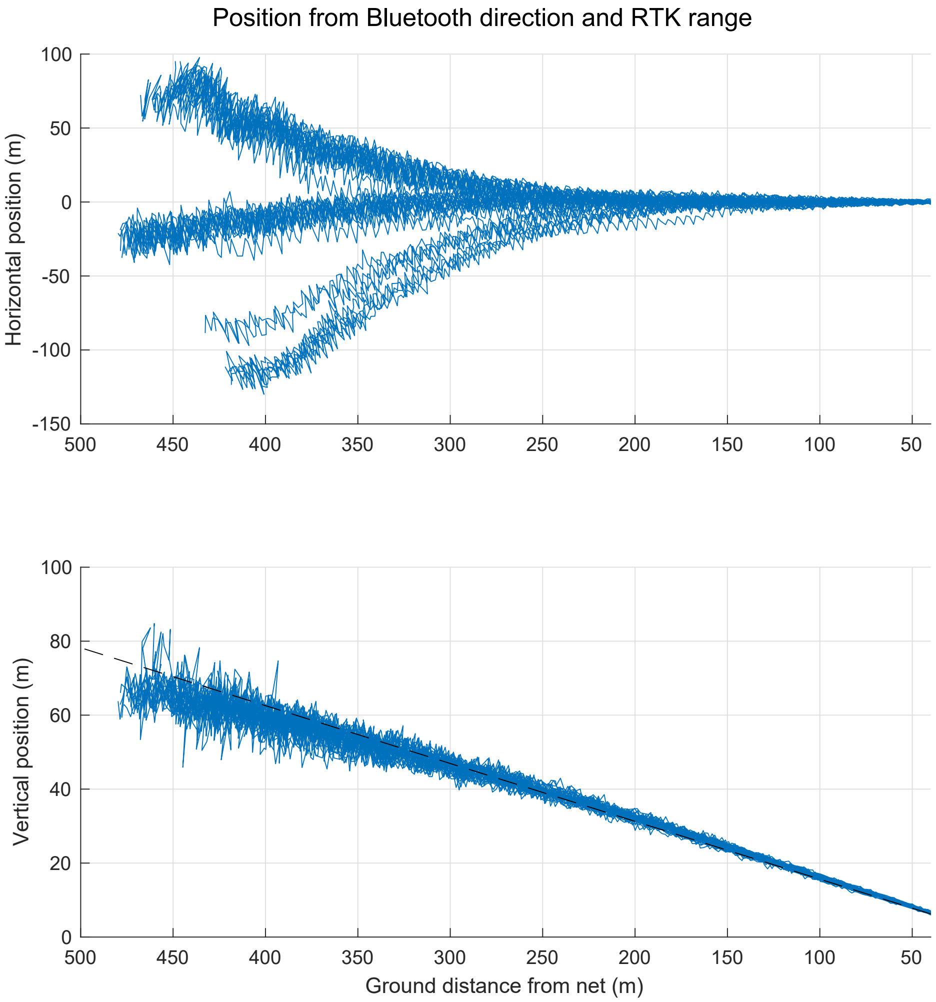
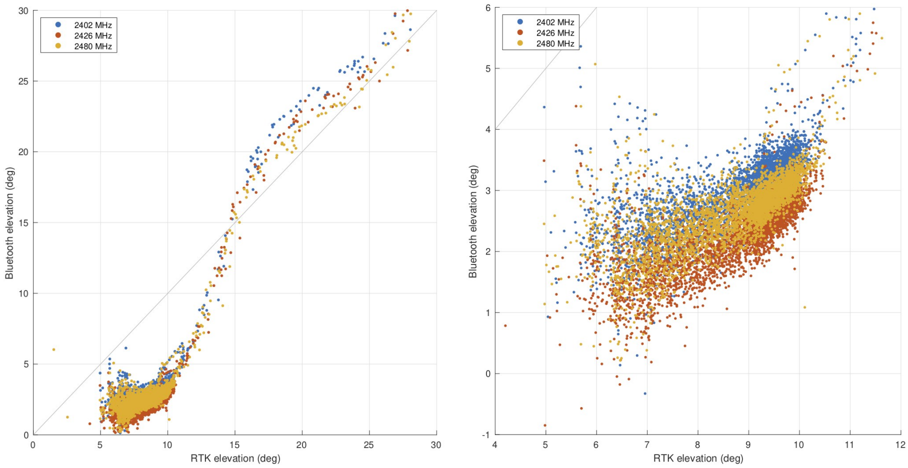


Figure 8.22: UAV position plots from Bluetooth direction and RTK GNSS distance for the 39 approaches.



(a) 0 — 30° elevation

(b) Values around the glideslope

Figure 8.23: Bluetooth elevation angle estimate plotted against elevation angle calculated from RTK GNSS positioning.

8.7 Conclusion

This paper has demonstrated the use of Bluetooth direction finding as a navigation system for automatic fixed-wing UAV recovery, suitable for arrest systems such as a suspended net. The multipath issue was taken into account, and the system has been shown capable of reliably hitting a stationary net without fusion with other sensors for navigation.

A weakness of the method demonstrated is the limited range of the system, making it unable to perform the complete recovery of a UAV operating far away if GNSS, as used under normal conditions, is lost during a flight. An alternative coarse navigation system would then be needed to guide the UAV within Bluetooth signal range for the final recovery maneuver. The error in the measured elevation was compensated by constant offsets that were manually adjusted. Based on the results, they could have been chosen better or replaced by an improved calibration such as a linear transform. The determination of parameters could have been performed using an automated procedure.

Chapter 9

Planning Approach Trajectories to Enable Late Aborts for Fixed-wing UAV Recovery on Ships

This chapter is based on the publication

- [133] M. L. Sollie and T. A. Johansen, “Planning approach trajectories to enable late aborts for fixed-wing UAV recovery on ships,” in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 311–320. doi: 10.1109/ICUAS51884.2021.9476706

and considers a method for planning an approach trajectory for a recovery system as in Chapter 7, where safely aborting the recovery approach at a late stage is possible.

9.1 Introduction

The use of unmanned aerial vehicles (UAVs) for remote sensing at sea has great potential for many operations, such as search and rescue, and detection and tracking of icebergs, oil spills, abandoned fishing equipment or other floating objects. Fixed-wing UAVs have the advantage of greater range and endurance than rotary-wing UAVs and can therefore cover larger areas in less time, with the disadvantage that recovery of the vehicle is more difficult and involves greater risk. Manually controlling the fixed-wing UAV into an arrest system such as a net [66, 127, 95] as shown in Fig. 9.1, stretched wire [63], inflated cushion [54] or other recovery system using a remote controller is a difficult task even for highly skilled UAV pilots. Autopilot controllers can respond faster to changing conditions such as wind gusts, and can monitor the health of communication links and navigation sensors to quickly abort a recovery attempt in case it is the safer option. Automatic landing systems can therefore make the recovery safer and more reliable.

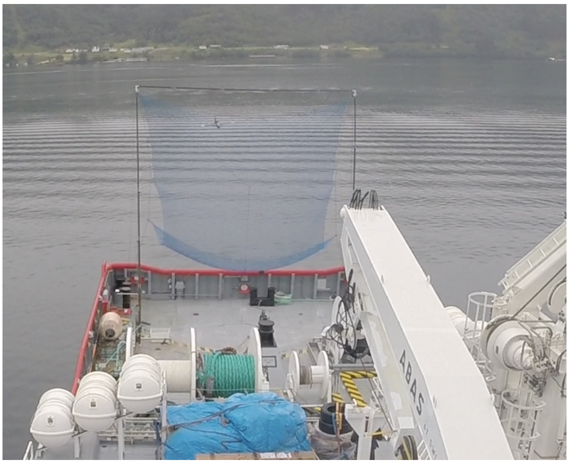


Figure 9.1: Recovery of fixed-wing UAV using a net mounted on a ship deck

Many different navigation sensors can be used for automatic landing systems. Inertial navigation systems is most often the basis for UAV navigation, with external aiding from one or more positioning systems. Global Navigation Satellite System (GNSS) receivers, camera-based navigation in the visual or infrared spectrums, barometers, lidar/radar altimeters, phased-array radio, radar positioning or local radio beacons such as Ultra-wideband (UWB) are possible options. For long duration UAV flights over several hours, the weather can change while the UAV performs its mission, making it necessary to recover the UAV in different weather conditions than it was launched in. This is a potential vulnerability for camera-based navigation systems, as fog, rain, snow, darkness and sun glint can reduce their range. GNSS signals can be interrupted by radio frequency interference such as jamming, due to its low signal power, meaning that position measurements cannot be guaranteed to persist until recovery is completed. Communication links required to transmit navigation data from the ship to the UAV can fail, making any navigation system which relies on data transmitted from the ship potentially unavailable. For short range positioning or direction-finding systems such as UWB or camera-based systems in conditions with reduced visibility, the navigation precision and accuracy required to hit the arrest system may only be available in the final part of the approach, meaning that the decision to complete or abort the landing may have to be done close to the ship. In this case it is important that we are able to abort the recovery at a distance from the ship which is less than the expected range of the required navigation system, in case it does not work as expected. For GNSS receivers the loss of position fix during the approach may make it desirable to abort, although there is a risk that the problem may persist such that it is not possible to retry the recovery. The action to take then depends on the economic value of losing the UAV, and not being able to use it in later flights, compared to potential damages from an unsuccessful recovery. In case the measurements of the UAV position relative the ship are lost during the approach, prediction of the ship and UAV positions can make it possible to complete the recovery if the predicted states are sufficiently accurate over the time remaining until recovery. This can be done using inertial navigation onboard the UAV, and also on the ship if communication with it is available. If communication is lost, ship motion prediction onboard the UAV using a model of the ship can be used over a short time period based on previous measurements. This is especially important if measurements are lost after the point at which the recovery cannot be safely aborted. It is desirable that when it is too late to abort, the recovery can be completed regardless of loss of ship state measurements. That is, it should always be possible to either proceed with the recovery with an expectation of success, or to abort in a safe manner. The ship motion prediction needed for these final seconds of the approach mostly depend on prediction of the oscillating heave, roll and pitch motions of the ship, altough prediction the movement of the ship over a longer horizon from its speed and course is useful for recovery planning even if all sensor and communication systems are working properly.

Due to vehicle inertia a UAV has limitations on how quickly it can turn in order to avoid obstacles. For this reason aborting a recovery can naturally not be performed arbitrarily late if it involves impacting an arrest system such as a net or a cushion, although methods such as a horizontally suspended wire on the ship, with a vertical wire released from the UAV just before recovery, can make late decisions easy. For the net and cushion methods, allowing aborts only until a predetermined time before impact or a mimimum distance to the ship are possible options, but this can also prevent aborts that are still feasible since wind velocity and airspeed may not be considered, leading to conservative margins. This motivates a look into how optimization techniques can be used to decide whether an abort can be performed or not based on a model of the vehicle dynamics. In addition, this could be used to plan a recovery approach that maximizes the possibility of late aborts. In order to allow aborts at a late stage in the recovery process, both the placement of the arrest system on the ship, the ship heading relative the wind direction and the choice of approach path are of significance. One possible approach is to run online optimization onboard the UAV, either using closed loop model predictive control (MPC) or open loop optimization to generate a reference for another controller. Another is to use offline optimization as a guide to help determine how approaches can be planned better when using simple methods in order to reduce complexity, how to select the best location on a ship to place the arrest system or how to orient the ship relative the wind. The airflow around a ship can also make some recovery system locations better than others, as shown in [161] where a simulation study of UAV recovery on a large ship is performed, including computational fluid dynamics (CFD) simulations of the airflow around the ship, to assess a given autopilot's ability to hit the desired position for different UAV and ship speeds, wind conditions and recovery system locations.

Optimization can be used in various forms in autonomous operations of UAVs, such as in trajectory generation [87], guidance [137, 85], cooperative vehicle rendezvous [99] and in low level control. [5] considers trajectory planning for avoiding obstacle collisions while attempting to track a known global trajectory. Of course, using optimization in real-time online control or decision-making for vehicles with fast dynamics, in the form of MPC, requires that the formulated problem can be solved quickly. Alternatively, the optimization can be performed at a lower rate to create references for low-level controllers, as long as the flight conditions, e.g. wind, does not change too quickly.

In the existing literature, some work has been done on decision making of UAV aborts, focusing mostly on runway landings where no obstacles need to be considered, or aborting missions. [55] designs a controller for runway landing, and determines a position region from which the landing is considered feasible with the controller used. If the UAV ends up outside this region during the landing, an abort is triggered and the UAV is directed to retry the approach. [72] describes a self-monitoring algorithm that can automatically abort a runway landing based on UAV position and velocity, and status of sensors and communication links, to perform a go-around to a new landing attempt. In [76] a mission abort policy for a vehicle exposed to external impacts is described, where the impacts increase the risk of system failure, and a tradeoff between probabilites for system survivability and mission success is considered. [166] continues this work, attempting to design a mission abort strategy that minimizes the total expected economic loss considering both the cost of a failed mission and vehicle failure, for a UAV operation where early-warning signals such as overheating, vibrations and intermittent loss of communications can indicate problems that increase the risk of system failure. To the best of our knowledge, no work on aborts of UAV recoveries on ships considering collision avoidance is found in existing literature.

This paper seeks to use optimization to find recovery approaches that allow aborts as late as possible. The main contribution is the planning concept with implemented examples for demonstration. The paper proceeds as follows: two simple and illustrative two dimensional examples, assuming an object moving at constant speed with a limited turn rate, show how approach trajectories can affect the ability to abort later while still being able to avoid collision. A three dimensional optimization problem for a recovery on a ship is then formulated, using a simple bank-to-turn UAV model, with assumptions for the UAV, ship and wind conditions. These results are then compared to simpler approaches, generated using the same implementation with additional constraints. The use of an aerodynamic model for forces and moments, ship motion, wind, the choice of optimality critera and implementation considerations are then discussed. Finally, concluding remarks are presented with suggestions for future work.

9.2 Illustrative example

To illustrate the effect that approach trajectories can have on the ability to perform a late abort, consider first a flat surface in two dimensions with a target placed on it. We want an object moving with a constant speed and a limited turn rate to hit the target with velocity perpendicular to the surface, and find the latest point the approach can be aborted without passing into the surface. The speed and maximum turn rate corresponds to a turning radius r. For a straight line approach as shown in Fig. 9.2, the latest point at which the approach can be aborted is obviously at a distance of r, with both possible aborts being symmetrical. If the approach is done

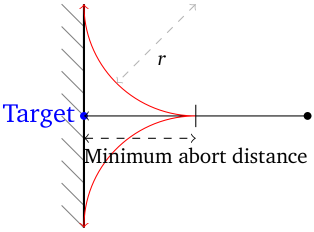


Figure 9.2: Straight line approach normal to the surface: the abort must start a distance r from the target to avoid passing the surface.

at an angle of 45 degrees as illustrated in Fig. 9.3, with a turn at the very end to hit the target perpendicular to the surface, an abort can be triggered at latest when the remaining distance is $\frac{\pi}{6}r \approx 0.52r$, which is significantly closer than in the straight line approach. This case is no longer symmetrical, and the shorter abort distance is only possible in one direction. An important point to make here is that as the abort is started during the final turn into the target, the minimum abort distance will be the same for any approach with an angle deviating 30 degrees or more from the surface normal, and straighter approaches than this will increase the minimum abort distance until reaching r as for the previous straight approach.

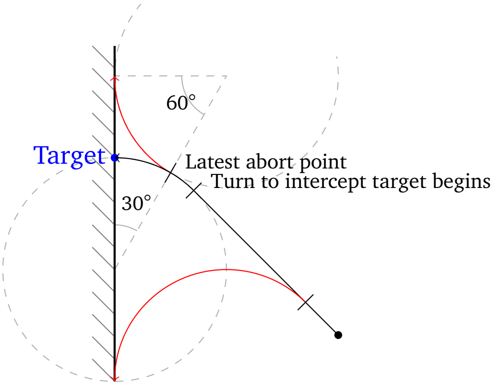


Figure 9.3: Straight line approach at a 45 degree angle: aborting to the left must begin very early, while the more reasonable right abort must begin when the remaining distance along the path to the target is $\frac{\pi}{6}r \approx 0.52r$.

As a second example consider a recovery net with width $0.8r$, which either should be hit perpendicularly in the center, or an abort can be triggered which should avoid impact with the net. For a straight approach as illustrated in Fig. 9.4, the minimum abort distance is $0.8r$. With the same 45 degree approach as in the previous example, with the recovery net as illustrated in Fig. 9.5, aborting to the left can be done significantly closer than for the surface case, and the right abort can be done until the remaining distance is approximately $0.48r$, again significantly closer than for the straight approach. The improvement resulting from a final turn in this case depends on the ratio between the turn radius and the net width. If the net width is $2r$ or greater then the result is the same as for the surface case, while a very narrow net compared to the turn radius will reduce the benefit of a late turn, although a late turn is always able to reduce the minimum abort distance[1].

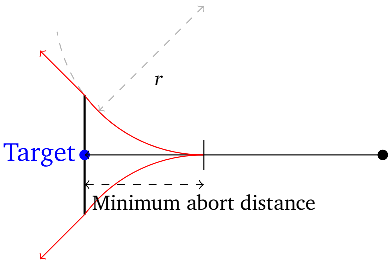


Figure 9.4: Straight line approach normal to the net with width $0.8r$. Minimum abort distance is then $0.8r$.

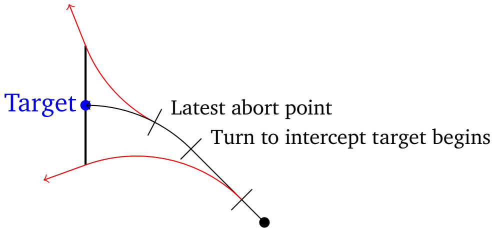


Figure 9.5: Straight line approach at a 45 degree angle with net width $0.8r$: minimum abort distance along path to target $0.48r$.

In these simple examples, finding the point where an abort is no longer possible is easy. In a more realistic case however, this is not necessarily the case. When wind is included, ground speed will depend on the flight direction if airspeed is maintained constant, meaning that tighter turns can be possible in one direction, and determining the optimal ship heading can be included in the planning. The shape of the volume which should be avoided to prevent collision with the ship can also be more complex. In three dimensions, which may allow aborts above the

[1] It can be shown that as the net width approaches 0, the ratio between the minimum abort distance for a straight approach and an approach with a final turn approaches $\frac{1}{\sqrt{2}} \approx 0.7$, meaning a reduction in the distance by a minimum of approximately 30% is possible for any combination of turn radius and net width.

ship and recovery system, the problem is also more difficult. For aborts done early, optimization can give a preferred trajectory based on weighing different aspects such as wing loading and energy usage.

An important observation in these examples is that the possibility to turn either left or right, to achieve the goal of avoiding collision, means that different local optimal solutions for aborts exists, and can be found by the solver. That is, the examples presented here are non-convex problems. A local solver may thus provide a solution that turns in a way that is a local, but not a global optimum, e.g. the left turn in Fig. 9.5, depending on the problem formulation and how the solver is initialized.

In practice, flying in a straight line does not require responsive or precise trajectory control. Guidance schemes such as Line-of-Sight (LOS) guidance is commonly used for straight line tracking, where the UAV attempts to fly towards a position on the line slightly in front of its current location. The examples in Fig. 9.3 and Fig. 9.5 would probably require better guidance and control, to enable the UAV to turn at the right time and impact the desired target position accurately. Highly complex trajectories where the UAV is operating on its limits or where fast maneuvers are present will in general require high performance controllers.

9.3 Example of concept application

The following example illustrates how the proposed concept can be implemented to generate an approach trajectory for a simple model of a UAV moving in three dimensions.

9.3.1 Assumptions

The example presented here does not consider uncertainty in the state of the ship or the time-varying and complex nature of airflow around it. The ship is assumed stationary at a known location, with a known heading, and the arrest system at a known ship-frame position and orientation.

In order to plan aborts a definition of *safe* in the context of aborts is needed. The method demonstrated here is to consider an ellipsoid placed around the ship, where the inside of the ellipsoid is considered off-limits for aborts. Any additional safety margins for how close to the ship the UAV is allowed to travel is applied by increasing the ellipsoid size from the minimum size required for the ship to fit inside it. Abort trajectories that pass inside of the ellipsoid are not considered safe. The physical size of the UAV is not considered in the implemented trajectory generation, meaning that only the center of mass of the UAV is required to stay outside the ellipsoid, but this can also be accounted for by increasing the ellipsoid dimensions, or to constrain the position of e.g. wingtips instead of the center of mass.

Both the approach and abort trajectories are constructed as polynomials, which allow discretizing the states and inputs at the desired rate after they have been calculated. This also has the advantage of allowing the abort to start at any point of the approach, not limited to certain discrete points in time, without including any integer decision variable in the optimization problem.

9.3.2 Differential flatness

For a system with state vector $\boldsymbol{x}$, input vector $\boldsymbol{u}$ and chosen output vector $\boldsymbol{y}$, the system is differentially flat [32] if the states and inputs can be expressed as algebraic functions of the output $\boldsymbol{y}$ and its derivatives $\dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}}, \ldots$. This is useful for trajectory planning because if the model satisfies this condition, then planning can be conducted in output space, and the inputs and states found using an algebraic mapping afterwards. In the absense of state and input constraints, states and inputs that satisfy the model differential equation can be found for any sufficiently smooth choice of output. Any such constraints can however be included in the optimization to find a valid output trajectory, as will be shown. The use of differential flatness for trajectory planning has been demonstrated for both multirotors [87] and fixed-wing UAV models [5].

9.3.3 Vehicle model

Consider a vehicle state

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}^{n\top} & \psi \end{bmatrix}^\top = \begin{bmatrix} x^n & y^n & z^n & \psi \end{bmatrix}^\top, \tag{9.1}$$

where $\boldsymbol{p}^n$ is the position in the North-East-Down (NED) frame $\{n\}$ and ψ is the yaw angle, and the input

$$\boldsymbol{u} = \begin{bmatrix} V_a & \phi & \theta \end{bmatrix}^\top, \tag{9.2}$$

consisting of the airspeed V_a, roll angle ϕ and pitch angle θ, repectively. ϕ, θ, ψ are the Euler angles representing the attitude of the vehicle body frame relative $\{n\}$. A simplified three dimensional bank-to-turn model is

$$\dot{\boldsymbol{x}} = \mathbf{F}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{x}^n \\ \dot{y}^n \\ \dot{z}^n \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} V_a \cos\theta \cos\psi + v^n_{w,x} \\ V_a \cos\theta \sin\psi + v^n_{w,y} \\ -V_a \sin\theta + v^n_{w,z} \\ \frac{g}{V_a} \tan\phi \end{bmatrix}, \tag{9.3}$$

where g is the gravitational acceleration and $\boldsymbol{v}^n_w = \begin{bmatrix} v^n_{w,x} & v^n_{w,y} & v^n_{w,z} \end{bmatrix}^\top$ is the wind velocity, which is assumed known and constant. The last relation $\dot{\psi} = \frac{g}{V_a} \tan\phi$ is the equation for a coordinated turn at constant altitude valid in the presence of wind [13]. Note that this model assumes zero sideslip and angle-of-attack, and is not really valid for steep climb or descent, but should suffice to demonstrate the concept.

More realistic models for fixed-wing UAVs have been shown to be differentially flat and can be used similarly [83, 5]. Improved controller-augmented dynamics can be used by identifying the closed-loop dynamics of the UAV and flight controller in question, such as in [137], but this only affects the trajectory generation if additional states or inputs are included in constraints.

For the model used here, the output is chosen as the position component of the state vector, $\boldsymbol{y} = \boldsymbol{p}^n$, and the trajectory is written as a polynomial of order m for each direction,

$$\begin{bmatrix} x^n(t) \\ y^n(t) \\ z^n(t) \end{bmatrix} = \begin{bmatrix} c_{x,m}t^m + c_{x,m-1}t^{m-1} + \cdots + c_{x,1}t + c_{x,0} \\ c_{y,m}t^m + c_{y,m-1}t^{m-1} + \cdots + c_{y,1}t + c_{y,0} \\ c_{z,m}t^m + c_{z,m-1}t^{m-1} + \cdots + c_{z,1}t + c_{z,0} \end{bmatrix}. \tag{9.4}$$

with t being the time parameter and coefficients $c_{\cdot,i}$ for $i = 1, \dots, m$ for each direction. To shown that the states and inputs can be written as functions of the position and its derivatives, and in this case the known constant wind, we write

$$\begin{aligned} V_a &= \|\dot{\boldsymbol{p}}^n - \boldsymbol{v}_w^n\| \\ &= \sqrt{(\dot{x}^n - v_{w,x}^n)^2 + (\dot{y}^n - v_{w,y}^n)^2 + (\dot{z}^n - v_{w,z}^n)^2}, \end{aligned} \tag{9.5}$$

$$\theta = \arcsin - \frac{\dot{z}^n - v_{w,z}^n}{V_a}, \tag{9.6}$$

$$\psi = \arctan \frac{\dot{y}^n - v_{w,y}^n}{\dot{x}^n - v_{w,x}^n}, \tag{9.7}$$

$$\phi = \arctan \frac{\dot{\psi} V_a}{g}, \tag{9.8}$$

with

$$\dot{\psi} = \frac{\ddot{y}^n(\dot{x}^n - v_{w,x}^n) - (\dot{y}^n - v_{w,y}^n)\ddot{x}^n}{(\dot{x}^n - v_{w,x}^n)^2 + (\dot{y}^n - v_{w,y}^n)^2}. \tag{9.9}$$

The only way the model (9.3) affects the trajectory generation where coefficients for (9.4) are found, is in the case of constraints on the attitude or airspeed.

An optimization problem which can be solved to find a trajectory impacting a target with a given position $\boldsymbol{p}_T^n$ and velocity $\boldsymbol{v}_T^n$ is

$$\begin{aligned} \min \quad & \int_0^{t_f} \left\| \frac{d^k}{dt^k} \boldsymbol{p}^n(t) \right\|^2 dt \\ \text{s.t.} \quad & \boldsymbol{p}^n(t_f) = \boldsymbol{p}_T^n, \\ & \dot{\boldsymbol{p}}^n(t_f) = \boldsymbol{v}_T^n, \end{aligned} \tag{9.10}$$

where we minimize the integral of the squared norm of a chosen derivative of the position. The starting time of the interval is set to 0 without loss of generality. In this case we choose to minimize the acceleration, using $k = 2$. The cost function in (9.10) can then be written as

$$\int_0^{t_f} \left(\frac{d^2}{dt^2}x^n(t)\right)^2 + \left(\frac{d^2}{dt^2}y^n(t)\right)^2 + \left(\frac{d^2}{dt^2}z^n(t)\right)^2 dt. \tag{9.11}$$

Inserting the position component polynomials from (9.4), taking the derivatives and performing the integral, it can be shown that this can be written as

$$\mathbf{c}_x^\top \mathbf{H}\mathbf{c}_x + \mathbf{c}_y^\top \mathbf{H}\mathbf{c}_y + \mathbf{c}_z^\top \mathbf{H}\mathbf{c}_z, \tag{9.12}$$

where $\mathbf{H}$ depends on m, k and t_f, with

$$\boldsymbol{c}_x = \begin{bmatrix} c_{x,m} & c_{x,m-1} & \cdots & c_{x,1} & c_{x,0} \end{bmatrix}^\top \tag{9.13}$$

and the equivalent for $\boldsymbol{c}_y$ and $\boldsymbol{c}_z$. The simple polynomials (9.4) used here can make $\mathbf{H}$ ill-conditioned for high m and large t_f, which can be improved by using alternative trajectory descriptions such as Legendre polynomials [88] and nondimensionalized time [87], but this is not considered a focus here.

Note that the initial position $\boldsymbol{p}^n(0)$ and derivatives are not constrained to allow the starting position and attitude of the approach trajectory to be chosen in the optimization process. If both the initial and final states are constrained, then determining the optimal end time t_f is a problem itself, which can be avoided if the starting position is unconstrained. For example, if $\boldsymbol{p}^n(0)$ is placed very close to the ship and t_f is large, then the UAV will have to make extra turns in order to "waste" time not needed to reach the target. With an unconstrained start position the distance from the ship will adapt to the time interval specified.

9.3.4 Constraint sampling

In (9.10) the time variable is eliminated from the cost function by integration over the time interval, as the time variable is not considered by the solver in the optimization. In order to add constraints for the states and inputs over the duration of the trajectory, constraints can be included at specified sampled points. This means that the constraints are not guaranteed to hold at any point of the trajectory, but by sampling with a sufficiently small time interval, the constraint will at least be close to holding between the sampled points due to the smoothness of the trajectory and its derivatives. The number of constraints that can be used is however limited by the number of decision variables, which depend on the polynomial order, in order for the problem to be solvable. By sampling at n_c uniformly spaced constraint points in time

$$t_c = \left\{0, \frac{t_f}{n_c - 1}, \ldots, t_f - \frac{t_f}{n_c - 1}, t_f\right\}, \tag{9.14}$$

constraints can be included using $\boldsymbol{p}^n(t_c)$, $V_a(t_c)$, $\phi(t_c)$, $\theta(t_c)$, $\psi(t_c)$ and their derivatives, which results in constraints on the polynomial coefficients.

9.3.5 Abort trajectory and safety constraint

An abort will be defined as a separate trajectory $\boldsymbol{p}_a^n = \begin{bmatrix} x_a^n(t_a) & y_a^n(t_a) & z_a^n(t_a) \end{bmatrix}^\top$ described in the same way as the approach (9.4), as a function of abort time $t_a = t - t_{as} \in [0, t_{af}]$, where t_{af} is the abort trajectory length and t_{as} is the time of the approach where the abort begins. The approach and abort are constrained to be continuous in position, $\boldsymbol{p}^n(t_{as}) = \boldsymbol{p}_a^n(0)$, and its three first derivatives velocity, acceleration and jerk, which means that not only is the attitude continuous, but also the angular rates.

The target position $\boldsymbol{p}_T^n$, which is the position of the arrest system, is considered to be placed on a stationary ship with center $\boldsymbol{p}_{\text{ship}}^n$ and heading ψ_{ship}. The target position is constant $\boldsymbol{p}_T^s$ in the ship frame {s}, with the relation

$$\boldsymbol{p}_T^n = \boldsymbol{p}_{\text{ship}}^n + \mathbf{R}_s^n(\psi_{\text{ship}})\boldsymbol{p}_T^s, \tag{9.15}$$

while the target velocity is $\boldsymbol{v}_T^n$, determined by the arrest system orientation and desired impact speed. Speed over ground is used as target instead of airspeed as this is more relevant for impact in an arrest system, where a low velocity relative the arrest system is desirable to minimize the impact and reduce the risk of damage, while some arrest systems on the other hand require a minimum impact velocity to trigger a mechanical capture of the UAV, i.e. to prevent it from falling out of a net.

Around the ship, a safe zone which should be avoided in case of an abort can be chosen as an ellipsoid, defined by

$$\frac{(\Delta x^s)^2}{a^2} + \frac{(\Delta y^s)^2}{b^2} + \frac{(\Delta z^s)^2}{c^2} < 1, \tag{9.16}$$

where a, b and c are the principal semi-axes of the ellipsoid, which will depend on the size of the ship, and Δx^s, Δy^s and Δz^s are the components of the position of the UAV in the ship frame,

$$\Delta \boldsymbol{p}^s = \begin{bmatrix} \Delta x^s \\ \Delta y^s \\ \Delta z^s \end{bmatrix} = \mathbf{R}_s^n(\psi_{\text{ship}})^\top(\boldsymbol{p}^n - \boldsymbol{p}_{\text{ship}}^n). \tag{9.17}$$

Defining the scaling matrix

$$\mathbf{M} = \begin{bmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & \frac{1}{c^2} \end{bmatrix}, \tag{9.18}$$

(9.16) can be written as

$$\Delta \boldsymbol{p}^{s\top} \mathbf{M} \Delta \boldsymbol{p}^s < 1. \tag{9.19}$$

An assumption made here is that the target is placed inside the ellipsoid in such a way that it can be reached in a safe manner as long as it is approached with a direction close to the target direction, i.e. the space in front of the arrest system is free from obstacles, but additional constraints can be added if this condition should be stricter. One way to prevent feasible aborts from entering the safe zone, while enabling detection of infeasible aborts, is to constrain sampled points on the abort trajectory to the set

$$\Delta\boldsymbol{p}^{s\top}\mathbf{M}\Delta\boldsymbol{p}^{s} \geq 1 \tag{9.20}$$

Expanding (9.10) with the matrix form (9.12) and adding cost for the abort trajectory and the start time of the abort, and weighting constants for the horizontal and vertical cost on the approach and abort, and the abort time, gives an optimization problem which can be solved to find an approach and the latest possible abort,

$$\begin{aligned}
\min_{\substack{\boldsymbol{c}_x,\boldsymbol{c}_y,\boldsymbol{c}_z,t_{as}\\ \boldsymbol{c}_{ax},\boldsymbol{c}_{ay},\boldsymbol{c}_{az}}} \quad & \mu_h\boldsymbol{c}_x^\top\mathbf{H}\boldsymbol{c}_x + \mu_h\boldsymbol{c}_y^\top\mathbf{H}\boldsymbol{c}_y + \mu_v\boldsymbol{c}_z^\top\mathbf{H}\boldsymbol{c}_z - \mu_t t_{as} + \\
& \mu_{ah}\boldsymbol{c}_{ax}^\top\mathbf{H}\boldsymbol{c}_{ax} + \mu_{ah}\boldsymbol{c}_{ay}^\top\mathbf{H}\boldsymbol{c}_{ay} + \mu_{av}\boldsymbol{c}_{az}^\top\mathbf{H}\boldsymbol{c}_{az} \\
\text{s.t.} \quad & \boldsymbol{p}^n(t_f) = \boldsymbol{p}_T^n, \\
& \dot{\boldsymbol{p}}^n(t_f) = \boldsymbol{v}_T^n, \\
& \Delta\boldsymbol{p}_a^{s\top}(\tfrac{t_a}{n_{\text{ell}}-1}l_{\text{ell}})\mathbf{M}\Delta\boldsymbol{p}_a^{s}(\tfrac{t_a}{n_{\text{ell}}-1}l_{\text{ell}}) \geq 1, \\
& \boldsymbol{p}^n(t_{as}) = \boldsymbol{p}_a^n(0), \\
& \dot{\boldsymbol{p}}^n(t_{as}) = \dot{\boldsymbol{p}}_a^n(0), \\
& \ddot{\boldsymbol{p}}^n(t_{as}) = \ddot{\boldsymbol{p}}_a^n(0), \\
& \dddot{\boldsymbol{p}}^n(t_{as}) = \dddot{\boldsymbol{p}}_a^n(0), \\
& 0 \leq t_{as} \leq t_f
\end{aligned} \tag{9.21}$$

In addition, these constraints were chosen here:

$$\begin{aligned}
-45^\circ \leq{}& \phi(\tfrac{t_f}{n_\phi-1}l_\phi) \leq 45^\circ & -50^\circ \leq{}& \phi_a(\tfrac{t_a}{n_\phi-1}l_\phi) \leq 50^\circ \\
-20^\circ \leq{}& \theta(\tfrac{t_f}{n_\theta-1}l_\theta) \leq 30^\circ & -20^\circ \leq{}& \theta_a(\tfrac{t_a}{n_\theta-1}l_\theta) \leq 30^\circ \\
16\text{m/s} \leq{}& V_a(\tfrac{t_f}{n_V-1}l_V) \leq 24\text{m/s} & 16\text{m/s} \leq{}& V_a(\tfrac{t_a}{n_V-1}l_V) \leq 24\text{m/s} \\
& z^n(\tfrac{t_f}{n_z-1}l_z) \leq -1\text{m} & & z_a^n(\tfrac{t_a}{n_z-1}l_z) \leq -1\text{m}
\end{aligned}$$

for $l_\star = 0,\ldots,n_\star - 1$ for the different constraints. μ_h, μ_v, μ_{ah} and μ_{vh} can be used to penalize vertical and horizontal acceleration differently for both the approach and abort. In order to find the latest possible abort, μ_t should be large enough that the UAV reaches the limits of its flight envelope, meaning that pitch, roll and/or airspeed limits should be active in the solution.

An example solution of this problem with $\boldsymbol{p}_{\text{ship}}^n = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^\top$, $\psi_{\text{ship}} = 0$, $\boldsymbol{p}_T^s = \begin{bmatrix}-15 & 8 & -5\end{bmatrix}^\top$ m, $m = 10$, ellipsoid dimensions $a = 50$, $b = 10$, $c = 15$, trajectory lengths $t_f = t_a = 3\text{s}$, sampled constraints $n_\phi = n_\theta = n_V = 20$, $n_z = 10$, $n_{\text{ell}} = 30$, $\mu_h = \mu_{ah} = 1$, $\mu_v = \mu_{av} = 3$, $\mu_t = 10000$, wind $\boldsymbol{v}_w^n = \begin{bmatrix}3 & 0 & 0\end{bmatrix}\frac{\text{m}}{\text{s}}$ and ground speed target of $20\frac{\text{m}}{\text{s}}$ horizontally into the target towards West is shown in Fig. 9.6 with

attitude i [156]
is used to own as
red and b to be
$t_{as} = 2.42$

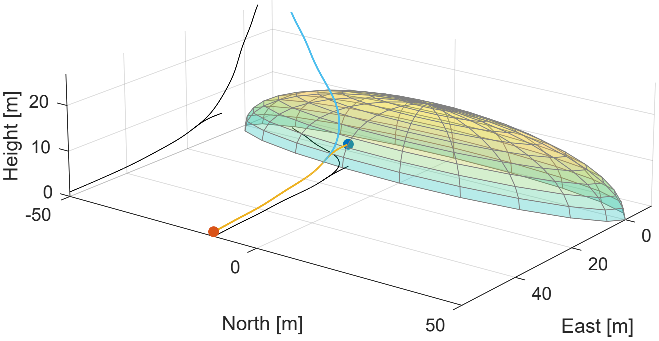


Figure 9.6: Approach and abort found by solving (9.21). The trajectories are also shown as projections in black.

An alternative to using an ellipsoid to create a safe volume is to define it by several flat intersecting planes, which describe a polyhedron such as a box. This would however result in multiple constraints where only one would need to hold at a time, because if the UAV is outside only one of the defining planes, then it cannot possibly be inside the volume. The results of this is either-or constraints, which introduces binary decision variables in the problem, and would require a different type of solver to solve the resulting mixed-integer nonlinear programming (MINLP) problem. Alternatively, the ellipsoid inequality (9.16) can be modified to get closer to a hexahedron by increasing the exponents from 2 to a higher even value, which would still not need the use of integers.

To better illustrate the approach and aborts that can be generated, a few different cases will be considered for comparison with Fig. 9.6. If the pitch of the abort is given a smaller upper limit of 15 degrees in the abort, the result is shown in Fig. 9.8. It is clear that the aborts in the new optimal now turn more sideways instead of flying directly over the ellipsoid, which moves the latest possible abort time back to $t_{as} = 2.27$s, and also plans an approach trajectory turning more sideways to improve the possibility of a sideways abort like illustrated in Fig. 9.3.

Changing the wind velocity to $\boldsymbol{v}_w^n = \begin{bmatrix} -3 & 0 & 0 \end{bmatrix} \frac{\text{m}}{\text{s}}$ gives a different result as seen in Fig. 9.9, showing that knownledge of the wind can be utilized by the approach planner to abort in the best direction. In this case, the abort time is $t_{as} = 2.38$s, which is slightly earlier than with the original wind and reasonable given that the target position is south of the ship center, making it harder to abort with a turn towards north. For the model (9.3) used here, a smaller ground speed gives a smaller turn radius for the same roll angle ϕ, which makes it better in this case to turn into headwind.

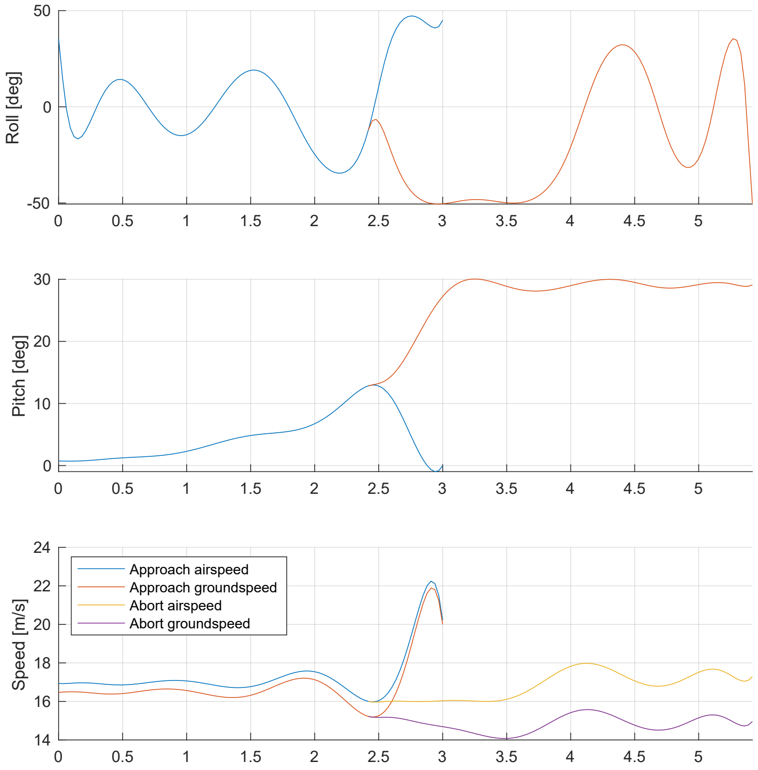


Figure 9.7 roach
and abor rates,
giving ag

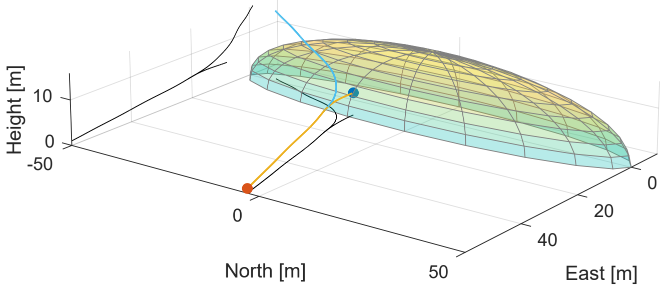


Figure 9.8: Approach trajectory with reduced upper pitch limit for the abort.

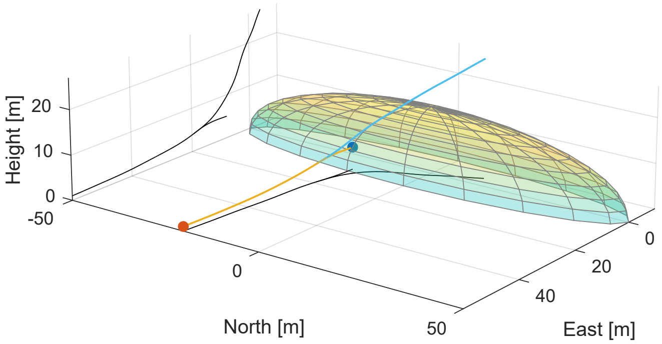


Figure 9.9: Approach trajectory with a $3\,\frac{\mathrm{m}}{\mathrm{s}}$ wind from the North.

9.4 Comparison with low complexity approaches

In order to compare the resulting trajectories in Section 9.3 with simpler methods,
the optimization problem was solved again, but with additional constraints. This
allows th s. Con-
straining exam-
ple in Fig) gives
the result 2.18s,
which is l n that
the solver

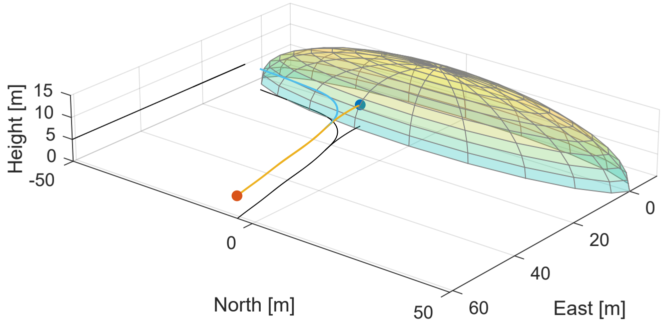


Figure 9.10: Approach trajectory with aborts constrained to lie in the horizontal plane.

A further erated
by additic target
position a own in
Fig. 9.11.

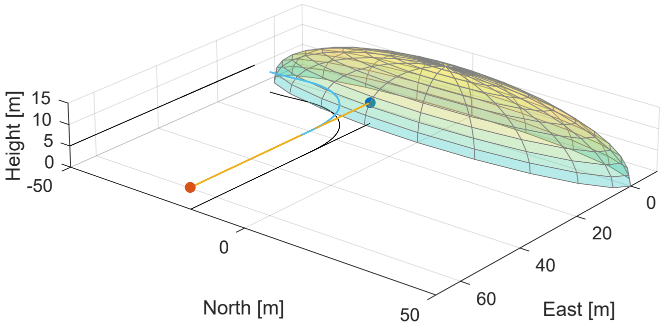


Figure 9.11: Simple straight approach with constant height, and constant height abort.

Replacing the requirement that the abort is in the horizontal plane with a straight

abort ove ... sult in
Fig. 9.12.

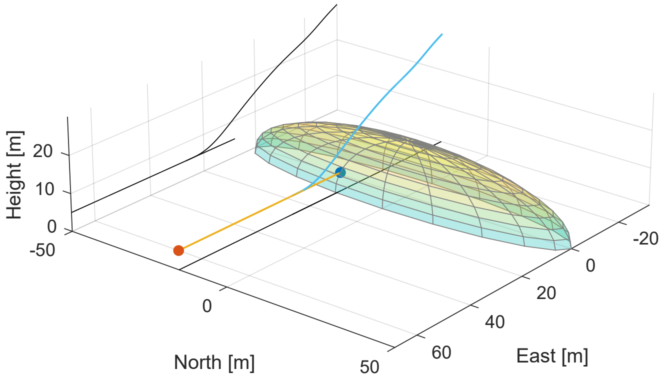


Figure 9.12: Simple straight approach with constant height, and straight abort over ellipsoid.

Another simple abort tested for comparison was similar to the late turns into the target illustrated in Fig. 9.3 and Fig. 9.5, beginning with a straight approach and ending with a constant rate turn. While trajectories exactly like this are not actually
flyable [1 ... e turn
would rec ... zontal
approach ... , 1.33s,
and const ... for the
transition ... ted in
Fig. 9.13. ... of the

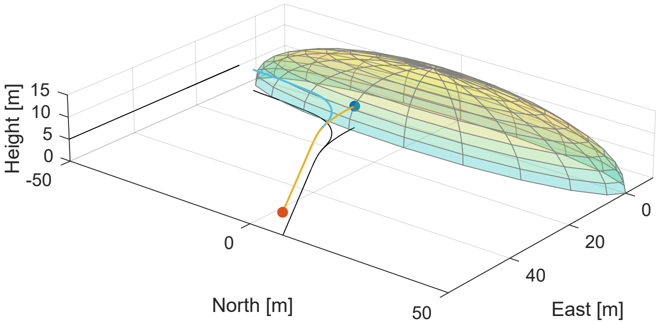


Figure 9.13: Approach which begins straight, and ends with a final turn, and constant height abort.

tested cases with the resulting abort times and corresponding time before recovery is given in Table 9.1.

Table 9.1: Result summary

Description	Latest abort time	Time before recovery	Fig. no.
Baseline example	2.42s	0.58s	6
Reduced upper pitch limit 15°	2.27s	0.73s	8
Reversed wind direction	2.38s	0.62s	9
Constant height approach and abort	2.18s	0.82s	10
Straight and constant height approach, constant height abort	1.78s	1.22s	11
Straight and constant height approach, straight abort	2.06s	0.94s	12
Initially straight approach with a final turn, constant height abort	2.14s	0.86s	13

9.5 Discussion

9.5.1 Optimality and objective function

For many problems, the choice of objective function is obvious, but that is not the case here. When safety is involved, the minimization of effort is not what we really aim for, as long as we are within the operational limits of the vehicle. For the ship crew overlooking the operations, predictable movements and a straight-line approach may be more desirable, making optimality a subjective question. This makes it hard to justify the computational requirements and complexity of the method over simpler methods consisting of waypoints and Line-of-sight guidance schemes. Uncertainties and risk assessment will realistically imply that conservative margins for the minimum distance between the UAV and ship in case of an abort will be used. A UAV flying quickly just a few meters from the side of a ship requires a high degree of confidence in the technology in order to make the ship crew comfortable when operating the system.

9.5.2 Implementation considerations

In the demonstrated examples where height is constrained to the target height, there are two possible solutions, as mentioned in Section 9.2: a left turn and a right turn. These are both local minima for the optimization problem, and unless the problem is completely symmetric, only one is the global minimum, meaning that the problem is non-convex. Since a local solver is used, the solution returned depends on initialization. If the solution is initialized to turn left, e.g. by setting an initial velocity or acceleration in this direction, the solver can be expected to give a solution turning left. Since in this case the number of possible solutions is only two, running the solver once using each initialization and picking the best of the two solutions can be done to find the global solution. On the other hand, even for the less contrained examples, different initializations can give significant differences in the computational time used to converge, and some initializations can make the solver struggle to converge in a reasonable time at all.

When using equality constraints at chosen discrete points in time as described in Section 9.3.4 as a method to globally enforce constraints, the number of equality constraints can exceed the number of decision variables, preventing the solver from finding a solution to the problem. For example, to constrain the height to a constant value, the polynomial coefficients $c_{z,m}, \dots, c_{z,1}$ can be constrained to 0, while $c_{z,0}$ is set from the desired height. This gives a total of $m+1$ equality constraints. If the height is instead constrained at more than $m+1$ discrete points in times, the number of equality constraints will be greater than the number of variables, giving the solver problems even though a trivial solution exists. Some constraints cannot be formulated directly using the coefficients, and the number of sampled constraint points cannot be chosen arbitrarily large, which is a disadvantage of the method.

As shown in Fig. 9.7, the trajectories found have some oscillating motion, especially in the roll angle and velocity, which are directly linked to oscillations in the acceleration. The polynomial parametrization in (9.4) is simple, but not a particularly great choice. Besides the mentioned issues with ill-conditioned cost matrices in (9.21) for high polynomial orders, a high polynomial order is needed to provide flexibility for the trajectory. While the resulting oscillations may seem unreasonable, the solver is utilizing all the available degrees of freedom of the parametrization, meaning that the result is reasonable for the problem as it is formulated. In order to improve this, alternative trajectory descriptions such as basis splines with control points, collocation methods or Legendre polynomials can be used.

The continuous-time polynomial formulation used here has the advantage that there are fewer decision variables when compared to a discrete-time method using e.g. Runge-Kutta integration, where the solver considers states and inputs directly. The availability of states at any time for use in constraints also allows directly solving for an optimum abort time. On the other hand, working with discrete states and inputs as decision variables allows any feasible trajectory to be generated, not being constrained by the possibilities given by the basis functions used in the continuous time trajectory. Solving for an abort time in a discrete formulation would require an integer index to be found for the states included in continuity constraints for the initial part of the abort, which makes the problem more complex as a different solver or an additional searching algorithm must be used.

9.5.3 Ship motion and wind

In the previous examples, the position of the ship has been assumed known and constant. In a real scenario involving UAV operations from large ship, the UAV needs to be able to land while the ship is en route from one place to another, or while other tasks are performed, for reasons of cost of operating the ship itself. This further complicates the problem by needing the optimization to avoid colliding with the time-varying ship position. The ship motion prediction is decoupled from the planning and it may for the optimization be assumed that the predicted state of the vessel for a time horizon is available when solving the problem, although the increasing uncertainty of the ship position should increase the margins used. Repeated replanning would improve the planned solutions as time progresses, and warm-starting the solver can help reduce the computational time required for replanning.

In the examples presented, the ship heading was assumed to be predetermined. Allowing the solver to determine the optimal ship heading could utilize the wind to reduce the minimum abort distance further. This would naturally depend on how realistic the modelling of the airflow around the ship is, and how accurately the effect of wind is included in the UAV model. If a path is generated that only serves as a reference for lower level controllers, the time-varying wind can be seen as a disturbance which should be rejected. Wind can be assumed measured using sensors onboard the ship, but the airflow around a ship can be complex and difficult to model or predict using CFD simulations. CFD analysis of a ship in different freestream wind speeds and directions relative the ship could be used to generate a table or fitted continuous functions for airflow, to take the airwake around the ship into account in a more accurate problem formulation. The slowly varying wind component can also be estimated onboard the UAV [57].

9.5.4 Using a real aerodynamic model

An aerodynamic model using identified aerodynamic parameters can be used to calculate forces and moments acting on the airframe [45]. The control surfaces and motor speed(s) of the UAV can be used as inputs to the model directly. It is important that the model structure is accurate, as the solver can utilize modeling errors in ways that are unexpected, but optimal for the given model. For example, the linearized force model in [13],

$$\boldsymbol{f}^b_{\text{aero}} = \frac{1}{2}\rho||\boldsymbol{v}_a||_2^2 S \begin{bmatrix} C_X + C_{X_q}\frac{cq}{2||\boldsymbol{v}_a||_2} + C_{X_{\delta_e}}\delta_e \\ C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{bp}{2||\boldsymbol{v}_a||_2} + C_{Y_r}\frac{br}{2||\boldsymbol{v}_a||_2} + C_{Y_{\delta_a}}\delta_a \\ C_Z + C_{Z_q}\frac{cq}{2||\boldsymbol{v}_a||_2} + C_{Z_{\delta_e}}\delta_e \end{bmatrix}, \tag{9.22}$$

where $C_\star$ are aerodynamic coefficients, $||\boldsymbol{v}_a||_2$ is airspeed, β is sideslip, ρ is air density, c is the mean cord length, b is the wingspan, S is the planform area of the wing, p, q and r are components of the angular rate $\boldsymbol{\omega}^b_{nb} = [p\ q\ r]^\top$ and δ_e, δ_a are the deflections of the elevator and aileron control surfaces, respectively. In this model the x-component should be a source of drag, but it can be seen that as $C_{X_q}\frac{c}{2||\boldsymbol{v}_a||_2}$ and $C_{X_{\delta_e}}$ have constant signs, q and δ_e are able to produce positive thrust if the values have a large enough magnitude, with the correct sign. If motor thrust is given a cost, while angular rate and control surface deflections are not, this can be exploited by the solver if a feasible trajectory utilizing these exist. Nonlinear effects can therefore be important in order to get meaningful optimal solutions, unless constraints are included to stay close to the linearization point, although this reduces the ability of the UAV to operate at the limits of its flight envelope.

For aborts that pass over the ship, there is a difference between the steepest ascent the UAV can manage over a long period, where propulsion must be sufficient to maintain airspeed, and a short and very steep ascent the UAV can perform while accepting a temporary reduction in airspeed. A UAV model that allows total energy considerations to be included in the problem could be benefitial to allow very steep climbs to pass over the ship.

9.6 Conclusion

This paper has presented a concept for planning of automatic recovery approaches for fixed-wing UAVs operating from ships to increase the feasibility of late aborts. While the usefulness of the concept depends on the ship, recovery system and UAV at hand, it does show potential in the presented examples. Better, more complex models of the ship movement and airflow around the ship could improve this further, at the cost of additional computational time.

Part III

Concluding Remarks

Chapter 10

Conclusion & Future Possibilities

This thesis has contributed to increased capabilities of fixed-wing UAVs using low-cost sensors, including increased state estimation performance with multiple independent GNSS receivers, and increased robustness to GNSS dropouts using Bluetooth direction finding. It has also contributed to increasing autonomy for UAVs with automatic arrest system recovery.

Tightly coupled integration of inertial navigation and two low-cost GNSS receivers with raw measurement output showed that the use of carrier phase interferometry can improve the heading estimate compared to the use of a single receiver, but that the different time synchronization methods for carrier phase measurements yielded nearly identical results. A multiplicative extended Kalman filter was used with measurement models derived for the raw GNSS observables. It was, however, difficult to evaluate the complete results without a higher accuracy reference estimate, and a comparison with RTKLIB baseline estimates was used to evaluate the heading. Rigorous handling of the covariance in the MEKF error-reset was not considered in this estimator but was considered separately with a review of previous considerations on this topic. The conclusion was that the MEKF covariance should be transformed using a rotation matrix from the previous to the new estimated body frame.

Navigation using the recently introduced direction finding feature of the Bluetooth specification was demonstrated at a range of up to 700 m, with limited packet loss until 500 m. In an open outdoor environment, using a square antenna array with 12 elements, the azimuthal performance was found to be very consistent with range, with noise standard deviation typically around 1°. The elevation was found to be significantly affected by multipath at lower elevation angles. It was shown that the choice of sampling sequence has an influence on the correlation between frequency estimation error and error in direction, and that time-symmetric sampling sequences can reduce this effect. It was demonstrated that using two arrays positioned vertically, where measurements are synchronized and processed together as if they came from a single large array, increased the elevation angle resolution. This allowed the separation of the direct and reflected signal at lower elevation angles than with a single array, significantly reducing the elevation error in the 7° to 15° range, where the error was largest when using a single array. The answer to the research question of whether Bluetooth direction finding equipment can be used for UAV navigation outdoors with a useful accuracy, precision, and range is definitively yes.

A system for automatic recovery in a moving arrest system was demonstrated in field experiments. The architecture built modularly and non-intrusively on a commercial-off-the-shelf autopilot, using RTK GNSS to find the arrest system orientation and for navigation relative to the arrest system. 15 recoveries in an net mounted on a barge towed by a ship were demonstrated, all of which were successful. Working towards automatic recovery without relying on GNSS, it was demonstrated in field experiments for a stationary scenario that Bluetooth direction finding can be used as a navigation system for arrest system recovery. This was performed using direction-based control, without the need for measurements of the distance from the array to the UAV. For 39 approach repetitions the resulting impact positions 40m in front of the array had standard deviations of 0.41 m horizontally and 0.32 m vertically. This shows both that an automatic recovery system for a moving arrest system can be designed with non-intrusive additions to a common autopilot, and that Bluetooth direction finding has the potential to increase resilience to GNSS signal loss in such systems.

Finally, a concept for path planning for recovery approaches towards a ship showed potential in the simulated examples presented, allowing the approach to be aborted later than paths from simple methods, if needed.

Future improvements

Extensions to the work presented in the thesis, that can further improve the results and capabilities, are

- The Bluetooth arrest system recovery method should be tested with a physical net, both to verify that the net does not significantly influence the direction finding, and to find a method for alignment of array and net.
- Integration of Bluetooth direction finding with intertial navigation. The net recovery using Bluetooth was performed with feedback directly from the direction estimates, and a combination with inertial navigation can provide a reduction in the noise of the feedback to the controllers.
- Combine the Bluetooth array with a motion reference unit to extend the Bluetooth-based arrest system recovery to a moving platform.
- The Bluetooth direction finding feature should be combined with Bluetooth ranging, allowing estimation of position. Methods for range estimation com-

patible with the Bluetooth standard exists, and ranging appears to be planned as a feature of the Bluetooth specification in the future [16].

- The automatic arrest system recovery was tested on a towed barge, but weather conditions were very calm, resulting in insignificant roll, pitch and heave motion of the arrest system. The system should be tested in adverse weather conditions, and on a ship.

Appendices

APPENDIX A

BLUETOOTH

A.1 Timing method for synchronizing array samples with inertial measurement units or other sensors

To improve both navigation robustness and control based on Bluetooth direction finding, fusion with inertial sensors in an inertial navigation system would be a natural extension to the direct feedback from processed Bluetooth measurements. The main challenge is that Bluetooth measurements are not sampled on the vehicle, but on the array on the ground. This means that the Bluetooth IQ samples and the sensors on the vehicle are not timestamped using a common clock. A feature that has been implemented but not yet used in field experiments, is a method for time synchronization of the Bluetooth measurements with other sensors carried on the moving vehicle. The flow of data is illustrated Fig. A.1, and the steps of the synchronization are:

1. The Bluetooth transmitter broadcasts a packet with a CTE. The value of an incrementing counter, called the CTE ID, is included in the packet data payload, and used as a data sequence number for identifying the packet. A general-purpose input/output (GPIO) pin is configured to toggle at a specific time of the CTE transmission (e.g. at the start or end), known as the Time Of Validity (TOV), by the use of a transmitter radio event interrupt. The GPIO pin is connected to a SentiBoard [4] which registers the time of the toggle event. Immediately after the packet is sent and the pin has toggled its output voltage, the same CTE ID as included in the Bluetooth payload is also transmitted using a data link to the SentiBoard. From this, the computer connected to the SentiBoard knows the time a specific CTE ID was broadcast by the transmitter.

2. The Bluetooth antenna array on the ground receives the CTE ID from the packet payload and performs IQ sampling of the CTE.

3. The array data, including the CTE ID, is parsed and forwarded to the vehicle computer. This can be done using Bluetooth or other communication links,

using the same hardware containing the Bluetooth receiver, or a separate computer.

4. On the vehicle computer, the Bluetooth IQ measurements are matched to the TOV using the CTE ID value. By connecting other sensors to the SentiBoard, we have a common clock used for timestamping of all measurements.

Since packet loss due to Bluetooth transmission conflicts, signal loss, or other reasons, can occur, the structure containing the CTE ID and TOV values from the transmitter should be cleared of older values if it is found that no data was received from the array for a packet.

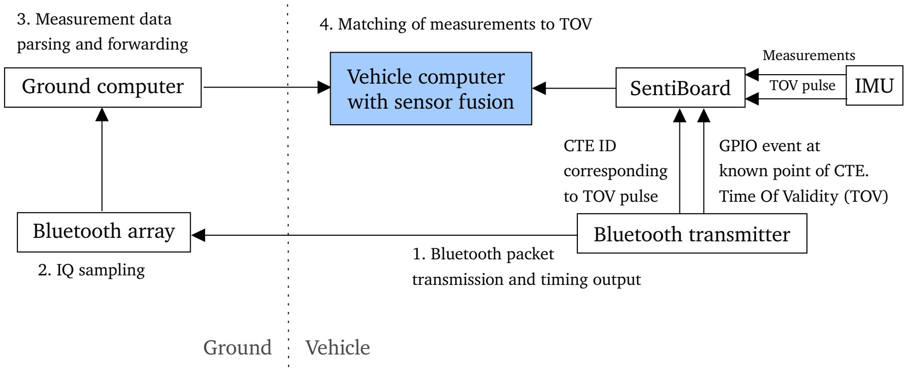

3. Measurement data parsing and forwarding
4. Matching of measurements to TOV
Ground computer
Vehicle computer with sensor fusion
SentiBoard
Measurements
TOV pulse
IMU
CTE ID corresponding to TOV pulse
GPIO event at known point of CTE. Time Of Validity (TOV)
Bluetooth array
Bluetooth transmitter
2. IQ sampling
1. Bluetooth packet transmission and timing output
Ground
Vehicle

Figure A.1: Diagram illustrating the communication used to synchronize Bluetooth measurements with inertial or other sensors on the vehicle

A.2 Pseudospectrum peak search using CasADi

To efficiently determine the peak of the direction pseudospectrum, CasADi [6] was used in Matlab, using the CasADi *Opti stack* syntax. As the use of complex numbers is not supported by CasADi, the complex calculations are performed manually with the real and imaginary components separate.

```
opti = casadi.Opti();
options = struct;
options.print_time = false; % Disable prints for running this in a loop
options.ipopt.print_level = 0; % Disable prints for running this in a loop
opti.solver('ipopt',options); % set numerical backend

psi = opti.variable(1);
alpha = opti.variable(1);
los =  [sin(alpha)*cos(psi);
        sin(alpha)*sin(psi);
        cos(alpha)];

a_angle = -2*pi*los'*ant_pos/wl;
a_real = cos(a_angle)';
a_imag = sin(a_angle)';

% Initial values from coarse search
opti.set_initial(alpha,alpha_0);
opti.set_initial(psi,psi_0);

% Run beamforming for CFO-corrected measurement vector X
X_real = real(X);
X_imag = imag(X);
corr_real = a_real'*X_real - a_imag'*X_imag;
corr_imag = a_real'*X_imag + a_imag'*X_real;
P = corr_real^2 + corr_imag^2;

opti.minimize(-P);
sol = opti.solve(); % actual solve
psi_opt = sol.value(psi);
alpha_opt = sol.value(alpha);
```

Listing A.1: Matlab code for direction estimation using CasADi.

A.3 Method for computing direction error for small CFO errors

From the steering vector (4.4) in Chapter 4, for a small residual frequency error $f_{\text{error}} = f_{\text{actual}} - f_{\text{estimated}}$ in the corrected measurements, the estimate $\hat{\Psi}, \hat{\alpha}$ and the true direction Ψ, α relate as

$$\frac{1}{\lambda_e} \boldsymbol{P}^{a\top} \hat{\boldsymbol{l}}^a(\hat{\Psi}, \hat{\alpha}) = \frac{1}{\lambda} \boldsymbol{P}^{a\top} \boldsymbol{l}^a(\Psi, \alpha) + f_{\text{error}} \boldsymbol{t}, \tag{A.1}$$

where $\boldsymbol{t}$ is the vector of measurement times for the antenna positions in $\boldsymbol{P}^a$, λ is the actual wavelength and λ_e is the assumed wavelength. Note that for this equation to be invariant of the origin of $\boldsymbol{t}$, $\boldsymbol{P}^a$ must be chosen such that the mean position (the centroid) is at the origin of the $\{a\}$-frame. It also assumes that the pseudo-spectrum contains a single distinct peak for any frequency error, which is not necessarily the case for for large errors. This can be solved for $\hat{\boldsymbol{l}}^a(\hat{\Psi}, \hat{\alpha})$ using the Moore-Penrose pseudoinverse, denoted $(\cdot)^+$,

$$\hat{\boldsymbol{l}}^a(\hat{\Psi}, \hat{\alpha}) = \left(\boldsymbol{P}^{a\top}\right)^+ \left(\frac{\lambda_e}{\lambda} \boldsymbol{P}^{a\top} \boldsymbol{l}^a(\Psi, \alpha) + \lambda_e f_{\text{error}} \boldsymbol{t}\right), \tag{A.2}$$

yielding a way to analyze the effect of frequency error on the direction estimate. Note that if the array is flat with $p^a_z = 0$ for all elements, this will output $\hat{l}^a_z = 0$ as this component cannot be estimated. The error direction and angle away from the boresight is

$$\hat{\Psi} = \operatorname{atan2}\left(\hat{l}^a_y, \hat{l}^a_x\right), \tag{A.3}$$

$$\hat{\alpha} = \sin^{-1}\left(\sqrt{(\hat{l}^a_x)^2 + (\hat{l}^a_y)^2}\right). \tag{A.4}$$

As an example, Fig. A.2 shows the direction the estimate will move for an increasing (positive) frequency error for a signal source in the boresight direction for a specific sampling order. Consider $\Psi = 0$, $\alpha = 0$, $f_{\text{error}} = 1000\text{Hz}$, $\lambda = \lambda_e$ corresponding to 2480 MHz, the array element position matrix $\boldsymbol{P}^a$ with origin at the array center and ordered as shown in Fig. A.2, and $\boldsymbol{t} = \begin{bmatrix} 0 & 2\mu\text{s} & 4\mu\text{s} & \dots & 22\mu\text{s} \end{bmatrix}$. Using (A.2) to (A.4), this results in a direction deflection away from the boresight along the arrow shown in Fig. A.2 by 0.75°. In the frequency estimate using only reference samples shown in Fig. 4.7, large amounts of noise with an amplitude of several kilohertz is visible, and a 5 kHz error would yield an error of approximately 3.77°, which shows the importance of an accurate and precise frequency deviation estimate for such a sampling order. If the receiver or external processing software assesses the validity of measurements, and can mark them as invalid resulting in their exclusion from the estimation, the amount of measurements additional to the reference samples is reduced, which can increase the estimate noise level.

The array element sampling order influences the correlation between errors in the frequency used for measurement correction and the resulting errors in the direction estimate. This means that some sampling orders results in a direction estimate that

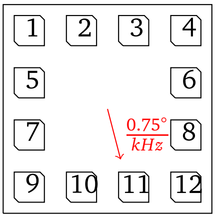


Figure A.2: Array viewed from the front. Consider that one IQ measurement for each array element is sampled at fixed intervals using the sample order indicated by the numbers. If the true CFO is higher than the estimate used for correction, a signal from a transmitter in the boresight direction will result in the estimated direction being deflected away from the boresight direction along the direction indicated by the arrow. This is essentially the direction we are "scanning" over the array, and direction changes along this direction correlate highly with frequency offset.

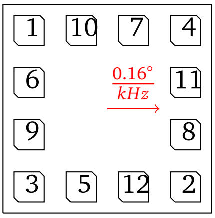


Figure A.3: Alternative sampling order with reduced effect of frequency error.

is more sensitive to errors in the CFO estimate than others. By using a sampling order where the effect of CFO estimation error on the direction estimate is low, the importance of having an accurate and precise CFO estimate is reduced. A different sampling order that is less affected is illustrated in Fig. A.3. Using (A.2), it can be found that repeating a sampling order multiple times does not change the effect of the frequency error. If more than one sampling pass is performed, the effect can however be reduced by combining different orders. If the second pass is a reversal of the first, this method shows that the resulting error is zero, meaning that frequency error and direction error no longer correlate.

Bibliography

[1] H. Abeida, Q. Zhang, J. Li, and N. Merabtine, “Iterative Sparse Asymptotic Minimum Variance Based Approaches for Array Processing,” *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 933–944, 2013. DOI: 10.1109/TSP.2012.2231676.

[2] S. Abujoub, J. McPhee, and R. A. Irani, “Methodologies for landing autonomous aerial vehicles on maritime vessels,” *Aerospace Science and Technology*, vol. 106, p. 106 169, 2020. DOI: https://doi.org/10.1016/j.ast.2020.106169.

[3] S. Akhavan, M. A. Bahabadi, Y. Norouzi, and H. Lolaee, “Direction of arrival estimation using array of antennas for low-altitude targets in multi-path environment,” *IET Radar, Sonar and Navigation*, vol. 10, no. 8, pp. 1439–1445, 2016. DOI: 10.1049/iet-rsn.2015.0529.

[4] S. M. Albrektsen and T. A. Johansen, “User-Configurable Timing and Navigation for UAVs,” *Sensors*, vol. 18, no. 8, 2018. DOI: 10.3390/s18082468.

[5] H. Alturbeh and J. F. Whidborne, “Real-time obstacle collision avoidance for fixed wing aircraft using B-splines,” *2014 UKACC International Conference on Control, CONTROL 2014 - Proceedings*, no. October 2018, pp. 115–120, 2014. DOI: 10.1109/CONTROL.2014.6915125.

[6] J. A. E. Andersson, J. Gillis, G. Horn, and J. B. Rawlings, “CasADi - A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019. DOI: 10.1007/s12532-018-0139-4.

[7] ArduPilot community, *Arduplane L1 guidance*, 2013. [Online]. Available: https://github.com/ArduPilot/ardupilot/pull/101 (visited on Dec. 3, 2019).

[8] ——, “Arduplane TECS controller.” (2019), [Online]. Available: https://github.com/ArduPilot/ardupilot/blob/ArduPlane-3.9.11/libraries/AP_TECS/AP_TECS.cpp (visited on Dec. 2, 2020).

[9] ——, “Ardupilot.” (2009–2022), [Online]. Available: http://ardupilot.org (visited on Jul. 18, 2022).

[10] K. Atteya and M. Amer, “Indoor Positioning - Bluetooth Angle of Arrival,” Master thesis, Lund University, 2019.

[11] P. Avital, G. Chardon, and J. Picheral, "Design of switching sequences for sine parameters estimation on switched antenna arrays," *Signal Processing*, vol. 188, no. 108244, 2021. DOI: 10.1016/j.sigpro.2021.108244.

[12] D. K. Barton, "Low-Angle Radar Tracking," *Proceedings of the IEEE*, vol. 62, no. 6, pp. 687–704, 1974. DOI: 10.1109/PROC.1974.9509.

[13] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft - Theory and Practice*. Princeton University Press, 2012.

[14] P. Bia *et al.*, "Compact Interferometric Antenna Array for UWB Direction Finding," in *2021 34th General Assembly and Scientific Symposium of the International Union of Radio Science, URSI GASS 2021*, Rome, Italy: URSI, 2021. DOI: 10.23919/URSIGASS51995.2021.9560254.

[15] Bluetooth SIG, *Bluetooth Core Specification v5.3*, 2021. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/ (visited on Jul. 19, 2022).

[16] ——, *Bluetooth Technology Overview*, 2022. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/ (visited on Jul. 20, 2022).

[17] ——, *Radio Frequency Physical Layer (RFPHY) - Bluetooth® Test Suite, rev. RFPHY.TS.p18*, 2022. [Online]. Available: https://www.bluetooth.com/specifications/qualification-test-requirements/ (visited on Jul. 19, 2022).

[18] N. BniLam, J. Steckel, and M. Weyn, "Synchronization of multiple independent sub-array antennas for IoT applications," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 2, pp. 1223–1232, 2018. DOI: 10.1049/cp.2018.0484.

[19] M. F. Bornebusch and T. A. Johansen, "Autonomous recovery of a fixed-wing UAV using a line suspended between two multirotor UAVs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 90–104, 2021. DOI: 10.1109/TAES.2020.3009509.

[20] L. Botler, M. Spork, K. Diwold, and K. Romer, "Direction Finding with UWB and BLE: A Comparative Study," *Proceedings - 2020 IEEE 17th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2020*, pp. 44–52, 2020. DOI: 10.1109/MASS50613.2020.00016.

[21] J. R. Carpenter and C. N. D'Souza, "Navigation filter best practices," NASA, Technical Publication (TP) NASA/TP–2018–219822, 2018. [Online]. Available: https://ntrs.nasa.gov/api/citations/20180003657/downloads/20180003657.pdf (visited on Jul. 19, 2022).

[22] K. Cisek, K. Gryte, T. H. Bryne, and T. A. Johansen, "Aided inertial navigation of small unmanned aerial vehicles using an ultra-wideband real time localization system," in *IEEE Aerospace Conference Proceedings*, vol. 2018-March, 2018, pp. 1–10. DOI: 10.1109/AERO.2018.8396534.

[23] S. D. Cloudt, "Bluetooth Low Energy Direction Finding on Embedded Hardware by Mitigating Carrier Frequency Offset and Multipath Fading," Master thesis, Eindhoven University of Technology, 2021.

[24] M. Cominelli, P. Patras, and F. Gringoli, "Dead on arrival: An empirical study of the Bluetooth 5.1 positioning system," in *WiNTECH '19: Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, Los Cabos, Mexico, 2019, pp. 13–20. DOI: 10.1145/3349623.3355475.

[25] P. S. Dias *et al.*, "Neptus - a framework to support multiple vehicle operation," in *Europe Oceans 2005*, IEEE, vol. 2, Jun. 2005, pp. 963–968. DOI: 10.1109/OCEANSE.2005.1513187.

[26] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957. DOI: 10.2307/2372560.

[27] T. W. R. East and K. L. S. Gunn, "The microwave properties of precipitation particles," *Quarterly Journal of the Royal Meteorological Society*, vol. 80, no. 346, pp. 522–545, 1954. DOI: 10.1002/qj.49708034603.

[28] G. Emel'yantsev, E. Dranitsyna, A. Stepanov, B. Blazhnov, and I. Vinokurov, "Tightly-Coupled GNSS-Aided Inertial System with Modulation Rotation of Two-Antenna Measurement Unit," in *2017 DGON Inertial Sensors and Systems (ISS)*, Karlsruhe, Germany: IEEE, 2017. DOI: 10.1109/InertialSensors.2017.8171503.

[29] G. I. Emel'yantsev, A. P. Stepanov, B. A. Blazhnov, D. A. Radchenko, I. Y. Vinokurov, and P. Y. Petrov, "Using satellite receivers with a common clock in a small-sized GNSS compass," *2017 24th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2017 - Proceedings*, vol. 116, no. 73, 2017. DOI: 10.23919/ICINS.2017.7995619.

[30] G. Emel'yantsev *et al.*, "Dual-mode GNSS gyrocompass using primary satellite measurements," in *ICINS 2018*, St. Petersburg, Russia: IEEE, 2018. DOI: 10.23919/ICINS.2018.8405842.

[31] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill, 2008.

[32] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995. DOI: 10.1080/00207179508921959.

[33] T. S. Fong, "An Interference Nulling Algorithm for a Single Receiver Phased Array," *IEEE Transactions on Antennas and Propagation*, vol. 38, no. 6, pp. 951–953, 1990. DOI: 10.1109/8.55599.

[34] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[35] T. I. Fossen and K. Y. Pettersen, “On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws,” *Automatica*, vol. 50, no. 11, pp. 2912–2917, 2014. DOI: 10.1016/j.automatica.2014.10.018.

[36] J. D. Fredrick, Y. Wang, and T. Itoh, “A smart antenna receiver array using a single RF channel and digital beamforming,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 12, pp. 3052–3058, 2002. DOI: 10.1109/TMTT.2002.805150.

[37] K. Gade, “The Seven Ways to Find Heading,” *Journal of Navigation*, vol. 69, no. 5, pp. 955–970, 2016. DOI: 10.1017/S0373463316000096.

[38] J. G. Garcia, P. I. Mercader, and C. H. Muravchik, “Use of Carrier Phase Double Differences,” *Latin American Applied Research*, vol. 35, pp. 115–120, 2005.

[39] S. Gezici *et al.*, “Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks,” *IEEE signal processing magazine*, vol. 22, no. 4, pp. 70–84, Jul. 2005. DOI: 10.1109/MSP.2005.1458289.

[40] Government of Norway, *Ambitious offshore wind initiative - Press Release*, 2022. [Online]. Available: https://www.regjeringen.no/en/aktuelt/ambitious-offshore-wind-power-initiative/id2912297/ (visited on Jul. 18, 2022).

[41] GPS Directorate, *GPS Interface Specification IRN-IS-200H-003*, 2015.

[42] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Artech House, 2013.

[43] K. Gryte, “Precision control of fixed-wing UAV and robust navigation in GNSS-denied environments,” Ph.D. dissertation, Norwegian University of Science and Technology (NTNU), Jun. 2020. DOI: 11250/2657113. [Online]. Available: https://hdl.handle.net/11250/2657113 (visited on Jul. 19, 2022).

[44] K. Gryte, T. H. Bryne, and T. A. Johansen, “Unmanned aircraft flight control aided by phased-array radio navigation,” *Journal of Field Robotics*, vol. 38, no. 4, pp. 532–551, 2021. DOI: 10.1002/rob.22002.

[45] K. Gryte, R. Hann, M. Alam, J. Roháč, T. A. Johansen, and T. I. Fossen, “Aerodynamic modeling of the Skywalker X8 Fixed-Wing Unmanned Aerial Vehicle,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 826–835. DOI: 10.1109/ICUAS.2018.8453370.

[46] K. Gryte, J. M. Hansen, T. Johansen, and T. I. Fossen, “Robust navigation of UAV using inertial sensors aided by UWB and RTK GPS,” in *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics (AIAA), 2017, pp. 1–16. DOI: 10.2514/6.2017-1035.

[47] K. Gryte, M. L. Sollie, and T. A. Johansen, “Control System Architecture for Automatic Recovery of Fixed-Wing Unmanned Aerial Vehicles in a Moving Arrest System,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 103, no. 4, 2021. DOI: 10.1007/s10846-021-01521-z.

[48] Y. Gui *et al.*, "Airborne vision-based navigation method for UAV accuracy landing using infrared lamps," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 2, pp. 197–218, 2013. DOI: 10.1007/s10846-013-9819-5.

[49] H. S. Halvorsen, H. Øveraas, O. Landstad, V. Smines, T. I. Fossen, and T. A. Johansen, "Wave motion compensation in dynamic positioning of small autonomous vessels," *Journal of Marine Science and Technology*, vol. 26, pp. 693–712, 2020. DOI: 10.1007/s00773-020-00765-y.

[50] R. Harker and J. Gilligan, "Dual thread-automatic takeoff and landing system (DT-ATLS)," in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, 2006, pp. 1146–1150.

[51] Hemisphere. "Vector VR500 Smart Antenna." (2019), [Online]. Available: https://www.hemispheregnss.com/wp-content/uploads/2019/02/hemispheregnss_vr500_userguide_875-0375-0_a4.pdf (visited on Mar. 14, 2020).

[52] P. Henkel and M. Iafrancesco, "Tightly coupled position and attitude determination with two low-cost GNSS receivers," *2014 11th International Symposium on Wireless Communications Systems, ISWCS 2014 - Proceedings*, no. 2, pp. 895–900, 2014. DOI: 10.1109/ISWCS.2014.6933480.

[53] S. Huh and D. H. Shim, "A vision-based automatic landing method for fixed-wing uavs," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, p. 217, 2009. DOI: 10.1007/s10846-009-9382-2.

[54] ——, "A vision-based automatic landing method for fixed-wing UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 57, no. 1-4, pp. 217–231, 2010. DOI: 10.1007/s10846-009-9382-2.

[55] P. Jetley, P. B. Sujit, and S. Saripalli, "Safe Landing of Fixed Wing UAVs," *Proceedings - 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, DSN-W 2017*, pp. 2–9, 2017. DOI: 10.1109/DSN-W.2017.43.

[56] X. Ji, C. Yu, W. Chen, and D. Dong, "GNSS 3D Attitude Measurement System Based on Dual-Antenna Receiver with Common Clock," in *CPGPC 2017 Forum on Cooperative Positioning and Service*, Harbin, China, 2017, pp. 223–227. DOI: 10.1109/CPGPS.2017.8075129.

[57] T. A. Johansen, A. Cristofaro, K. Sørensen, J. M. Hansen, and T. I. Fossen, "On estimation of wind velocity , angle-of-attack and sideslip angle of small UAVs using standard sensors," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, Colorado, USA: IEEE, 2015, pp. 510–519. DOI: 10.1109/ICUAS.2015.7152330.

[58] T. A. Johansen, A. Cristofaro, K. L. Sørensen, J. M. Hansen, and T. I. Fossen, "On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors," in *International Conference on Unmanned Aircraft Systems*, Denver, 2015. DOI: 10.1109/ICUAS.2015.7152330.

[59] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960. DOI: 10.1115/1.3662552.

[60] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and applications*, Second. Norwood, MA, USA: Artech House, 2006.

[61] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, “Unmanned aircraft capture and control via GPS spoofing,” *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014. DOI: 10.1002/rob.21513.

[62] S. Khantsis and A. Bourmistrova, “UAV controller design using evolutionary algorithms,” in *AI 2005: Advances in Artificial Intelligence*, S. Zhang and R. Jarvis, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1025–1030. DOI: 10.1007/11589990_134.

[63] ——, “UAV Controller Design Using Evolutionary Algorithms,” in *AI 2005: Advances in Artificial Intelligence*, S. Zhang and R. Jarvis, Eds., Sydney, Australia: Springer Berlin Heidelberg, 2005, pp. 1025–1030. DOI: 10.1007/11589990_134.

[64] S. Khattak, C. Papachristos, and K. Alexis, “Keyframe-based thermal–inertial odometry,” *Journal of Field Robotics*, vol. 37, no. 4, pp. 552–579, 2020. DOI: 10.1002/rob.21932.

[65] H. J. Kim *et al.*, “Fully autonomous vision-based net-recovery landing system for a fixed-wing UAV,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1320–1333, Aug. 2013. DOI: 10.1109/tmech.2013.2247411.

[66] ——, “Fully autonomous vision-based net-recovery landing system for a fixed-wing UAV,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1320–1333, 2013. DOI: 10.1109/TMECH.2013.2247411.

[67] J. Kim, J. Guivant, M. L. Sollie, T. H. Bryne, and T. A. Johansen, “Compressed pseudo-SLAM: Pseudorange-integrated compressed simultaneous localisation and mapping for unmanned aerial vehicle navigation,” *Journal of Navigation*, vol. 74, no. 5, pp. 1091–1103, 2021. DOI: 10.1017/S037346332100031X.

[68] K. Klausen, T. I. Fossen, and T. A. Johansen, “Autonomous recovery of a fixed-wing uav using a net suspended by two multirotor uavs,” *Journal of Field Robotics*, vol. 35, no. 5, pp. 717–731, 2018. DOI: 10.1002/rob.21772.

[69] J. A. Klobuchar, “Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 3, pp. 325–331, 1987. DOI: 10.1109/TAES.1987.310829.

[70] W. Kong, D. Zhou, D. Zhang, and J. Zhang, “Vision-based autonomous landing system for unmanned aerial vehicle: A survey,” in *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, Sep. 2014, pp. 1–8. DOI: 10.1109/MFI.2014.6997750.

[71] H. Krim and M. Viberg, “Two decades of array signal processing research: the parametric approach,” *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, Jul. 1996. DOI: 10.1109/79.526899.

[72] M. E. Kügler and F. Holzapfel, "Online self-monitoring of automatic take-off and landing control of a fixed-wing UAV," *1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017*, vol. 2017-Janua, pp. 2108–2113, 2017. DOI: `10.1109/CCTA.2017.8062764`.

[73] Laboratório de Sistemas e Tecnologia Subaquática - Faculdade de Engenharia da Universidade do Porto, *DUNE Unified Navigation Environment*. [Online]. Available: `https://www.lsts.pt/toolchain/dune` (visited on Feb. 1, 2022).

[74] A. M. Lekkas and T. I. Fossen, "Minimization of cross-track and along-track errors for path tracking of marine underactuated vehicles," in *Control Conference (ECC), 2014 European*, IEEE, 2014, pp. 3004–3010. DOI: `10.1109/ECC.2014.6862594`.

[75] L. K. Lepley and W. M. Adams, "Reflectivity of Electromagnetic Waves at an Air-Water Interface for Pure and Sea Water," Water Resources Research Center, University of Hawaii at Manoa, Tech. Rep., 1968, p. 21.

[76] G. Levitin and M. Finkelstein, "Optimal Mission Abort Policy for Systems Operating in a Random Environment," *Risk Analysis*, vol. 38, no. 4, pp. 795–803, 2018. DOI: `10.1111/risa.12886`.

[77] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-spatial Information Science*, vol. 21, no. 1, pp. 21–32, 2018. DOI: `10.1080/10095020.2017.1420509`.

[78] G. MacGougan, K. O'Keefe, and R. Klukas, "Tightly-coupled GPS/UWB integration," *Journal of Navigation*, vol. 63, no. 1, pp. 1–22, 2010. DOI: `10.1017/S0373463309990257`.

[79] M. R. Mahfouz, C. Zhang, B. C. Merkl, M. J. Kuhn, and A. E. Fathy, "Investigation of high-accuracy indoor 3-D positioning using UWB technology," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 6, pp. 1316–1330, Jun. 2008. DOI: `10.1109/TMTT.2008.923351`.

[80] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*, 1st ed., ser. Space Technology Library. Springer-Verlag New York, 2014, vol. 33. DOI: `10.1007/978-1-4939-0802-8`.

[81] F. L. Markley, "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311–317, 2003. DOI: `10.2514/2.5048`.

[82] F. Markley, "Lessons learned," *Journal of the Astronautical Sciences*, vol. 57, pp. 3–29, 2009. DOI: `10.1007/BF03321491`.

[83] P. Martin, "Aircraft Control Using Flatness," in *IMACS/IEEE-SMC Multiconference CESA'96 - Symposium on Control, Optimization and Supervision*, Lille, France, 1996, pp. 194–199.

[84] R. Martins, P. S. Dias, E. R. B. Marques, J. Pinto, J. B. de Sousa, and F. L. Pereira, "IMC: A communication protocol for networked vehicles and sensors," in *OCEANS 2009-EUROPE*, May 2009, pp. 1–6. DOI: `10.1109/OCEANSE.2009.5278245`.

[85] S. H. Mathisen, K. Gryte, T. I. Fossen, and T. A. Johansen, "Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing," in *AIAA Infotech @ Aerospace*, San Diego, California, USA: AIAA, 2016. DOI: 10.2514/6.2016-0512.

[86] R. J. McAulay, "Optimum Elevation Angle Estimation in the Presence of Ground Reflection Multipath," *Massachusetts Institute of Technology Lincoln Laboratory - Technical Note*, vol. 1976-11, 1976.

[87] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.

[88] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 477–483, 2012. DOI: 10.1109/ICRA.2012.6225009.

[89] Y. Meng, W. Wang, H. Han, and J. Ban, "A visual/inertial integrated landing guidance method for UAV landing on the ship," *Aerospace Science and Technology*, vol. 85, pp. 474–480, 2019. DOI: 10.1016/j.ast.2018.12.030.

[90] M. G. Michailidis, M. J. Rutherford, and K. P. Valavanis, "A survey of controller designs for new generation uavs: The challenge of uncertain aerodynamic parameters," *International Journal of Control, Automation and Systems*, 2019. DOI: 10.1007/s12555-018-0489-8.

[91] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Ganga-Jamuna Press, 2012.

[92] S. Monica and G. Ferrari, "An experimental model for UWB distance measurements and its application to localization problems," in *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*, Sep. 2014, pp. 297–302. DOI: 10.1109/ICUWB.2014.6958996.

[93] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *IEEE International Conference on Robotics and Automation*, Seattle, Washington: IEEE, 2015, pp. 1730–1736. DOI: 10.1109/ICRA.2015.7139421.

[94] T. Muskardin *et al.*, "A novel landing system to increase payload capacity and operational availability of high altitude long endurance uavs," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2, pp. 597–618, 2017. DOI: 10.1007/s10846-017-0475-z.

[95] ——, "A Novel Landing System to Increase Payload Capacity and Operational Availability of High Altitude Long Endurance UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 88, no. 2-4, pp. 597–618, 2017. DOI: 10.1007/s10846-017-0475-z.

[96] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006. DOI: 10.1007/978-0-387-40065-5.

[97] Nordic Semiconductor, *Direction Finding nWP-036 White Paper v1.1*, 2020.

[98] C. O'Driscoll, "GNSS Solutions," *Inside GNSS*, no. July/August, pp. 18–22, 2010.

[99] L. Persson, T. Muskardin, and B. Wahlberg, "Cooperative rendezvous of ground vehicle and aerial vehicle using model predictive control," in *2017 IEEE 56th Annual Conference on Decision and Control, CDC 2017*, 2017, pp. 2819–2824. DOI: 10.1109/CDC.2017.8264069.

[100] A. Pinker and C. Smith, "Vulnerability of the GPS signal to jamming," *GPS Solutions*, vol. 3, no. 2, pp. 19–27, 1999. DOI: 10.1007/PL00012788.

[101] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. B. de Sousa, "The LSTS toolchain for networked vehicle systems," in *2013 MTS/IEEE OCEANS-Bergen*, 2013, pp. 1–9. DOI: 10.1109/OCEANS-Bergen.2013.6608148.

[102] M. E. Pittelkau, "An analysis of the quaternion attitude determination filter an analysis of the quaternion attitude determination filter," *Journal of the Astronautical Sciences*, vol. 51, pp. 103–120, 2003. DOI: 10.1007/BF03546317.

[103] ——, "Rotation vector in attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 6, pp. 855–860, 2003. DOI: 10.2514/2.6929.

[104] K. S. Pratt, R. Murphy, S. Stover, and C. Griffin, "CONOPS and autonomy recommendations for VTOL small unmanned aerial system based on hurricane katrina operations," *Journal of Field Robotics*, vol. 26, no. 8, pp. 636–650, 2009. DOI: 10.1002/rob.20304.

[105] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 3rd ed. Prentice-Hall, 1996.

[106] PX4 Dev team, *Autopilot Hardware - Pixhawk series*, 2019. [Online]. Available: https://docs.px4.io/en/flight_controller/pixhawk_series.html (visited on Feb. 11, 2019).

[107] J. Y. Qu, J. Y. Zhang, and Z. G. You, "Direction finding based on single receiving channel," in *International Conference on Signal Processing Proceedings, ICSP*, vol. 1, Guilin, China, 2006, pp. 6–8. DOI: 10.1109/ICOSP.2006.344546.

[108] R. G. Reynolds, "Asymptotically optimal attitude filtering with guaranteed convergence," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 114–122, 2008. DOI: 10.2514/1.30381.

[109] S. Rinaldi, P. Ferrari, E. Sisinni, A. Depari, and A. Flammini, "An evaluation of low-cost self-localization service exploiting angle of arrival for industrial cyber-physical systems," in *2021 IEEE AFRICON*, IEEE, 2021. DOI: 10.1109/AFRICON51333.2021.9570985.

[110] C. D. Rodin, T. A. Johansen, and A. Stahl, “Skyline based camera attitude estimation using a digital surface model,” in *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, 2018, pp. 306–313. DOI: 10.1109/AMC.2019.8371108.

[111] A. Rodriguez-Ramos, C. Sampedro, H. Bavle, P. De La Puente, and P. Campoy, “A deep reinforcement learning strategy for UAV autonomous landing on a moving platform,” *Journal of Intelligent & Robotic Systems*, vol. 93, pp. 351–366, 2019. DOI: 10.1007/s10846-018-0891-8.

[112] R. Roy and T. Kailath, “ESPRIT-Estimation of signal parameters via rotational invariance techniques,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989. DOI: 10.1109/29.32276.

[113] RUAG. “Object position and tracking system (OPATS).” (2020), [Online]. Available: https://www.ruag.com/en/opats (visited on Jul. 26, 2022).

[114] M. N. O. Sadiku, “Refractive index of snow at microwave frequencies,” *Applied Optics*, vol. 24, no. 4, p. 572, 1985. DOI: 10.1364/ao.24.000572.

[115] P. Sambu and M. Won, *An Experimental Study on Direction Finding of Bluetooth 5.1: Indoor vs Outdoor*, 2021. [Online]. Available: http://arxiv.org/abs/2103.04121 (visited on Jul. 19, 2022).

[116] J. L. Sanchez-Lopez, J. Pestana, S. Saripalli, and P. Campoy, “An approach toward visual autonomous ship board landing of a VTOL UAV,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1, pp. 113–127, 2014. DOI: 10.1007/s10846-013-9926-3.

[117] R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986. DOI: 10.1109/TAP.1986.1143830.

[118] C. M. S. See, “High resolution DF with a single channel receiver,” in *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, Singapore, 2001, pp. 520–523. DOI: 10.1109/ssp.2001.955337.

[119] ——, “A Single Channel Approach to High Resolution Direction Finding and Beamforming,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, 2003, pp. 217–220. DOI: 10.1109/icassp.2003.1199907.

[120] Sensonor, “Datasheet STIM300 Inertia Measurement Unit TS1524 rev.22,” 2018.

[121] V. Sesma and V. Egorov, “Improved Accuracy for Indoor Positioning with Bluetooth 5.1: From Theory to Measurements,” Master thesis, Lund University, 2020.

[122] M. D. Shuster, “A survey of attitude representations,” *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, 1993.

[123] Sierra Nevada Corporation. “Dual-thread automatic takeoff and landing system (DT-ATLS).” (2010), [Online]. Available: https://www.sncorp.com/media/2000/dual-thread-product-sheet2010.pdf (visited on Mar. 14, 2020).

[124] ——, “Unmanned aerial vehicle common automatic recovery system (UCARS) - version 2.” (2013), [Online]. Available: https://www.sncorp.com/media/1998/ucars-v2product-sheet.pdf (visited on Mar. 14, 2020).

[125] M. Skolnik, *Radar Handbook, Third Edition*. McGraw-Hill, 2008.

[126] R. Skulstad, C. Syversen, M. Merz, N. Sokolova, T. Fossen, and T. Johansen, “Autonomous net recovery of fixed-wing UAV with single-frequency carrier-phase differential GNSS,” *Aerospace and Electronic Systems Magazine, IEEE*, vol. 30, no. 5, pp. 18–27, May 2015. DOI: 10.1109/MAES.2015.7119821.

[127] ——, “Autonomous net recovery of fixed-wing UAV with single-frequency carrier-phase differential GNSS,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 5, pp. 18–27, 2015. DOI: 10.1109/MAES.2015.7119821.

[128] J. Solà, *Quaternion kinematics for the error-state kalman filter*, Nov. 2017. DOI: 10.48550/arXiv.1711.02508. [Online]. Available: https://arxiv.org/abs/1711.02508v1 (visited on Jul. 19, 2022).

[129] M. L. Sollie, T. H. Bryne, and T. A. Johansen, “Pose Estimation of UAVs Based on INS Aided by Two Independent Low-Cost GNSS Receivers,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 1425–1435. DOI: 10.1109/ICUAS.2019.8797746.

[130] ——, “On the Error-Reset Covariance Transform for the Multiplicative Extended Kalman Filter,” *Submitted, in revision*, 2022.

[131] M. L. Sollie, K. Gryte, T. H. Bryne, and T. A. Johansen, “Automatic Recovery of Fixed-wing Unmanned Aerial Vehicle Using Bluetooth Angle-of-Arrival Navigation,” *Submitted, in revision*, 2022.

[132] ——, “Outdoor Navigation Using Bluetooth Angle-of-Arrival Measurements,” *IEEE Access*, vol. 10, pp. 88 012–88 033, 2022. DOI: 10.1109/ACCESS.2022.3199772.

[133] M. L. Sollie and T. A. Johansen, “Planning approach trajectories to enable late aborts for fixed-wing UAV recovery on ships,” in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 311–320. DOI: 10.1109/ICUAS51884.2021.9476706.

[134] M. L. Sollie, T. A. Johansen, K. Gryte, and T. H. Bryne, “Reducing Ground Reflection Multipath Errors for Bluetooth Angle-of-Arrival Estimation by Combining Independent Antenna Arrays,” *Submitted, in revision*, 2022.

[135] ——, “Sampling Sequence Considerations for Bluetooth Angle-of-Arrival Estimation,” *Submitted, in revision*, 2022.

[136] M. L. Sollie, “Estimation of UAV Position, Velocity and Attitude Using Tightly Coupled Integration of IMU and a Dual GNSS Receiver Setup,” Master thesis, Norwegian University of Science and Technology, 2018. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2561061.

[137] T. Stastny and R. Siegwart, "Nonlinear Model Predictive Guidance for Fixed-wing UAVs Using Identified Control Augmented Dynamics," *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 432–442, 2018. DOI: `10.1109/ICUAS.2018.8453377`.

[138] W. L. Stutzman, *Polarization in Electromagnetic Systems*, 2nd ed. Norwood: Artech House, 2018.

[139] P. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, 2014. DOI: `10.1109/MCS.2013.2287568`.

[140] T. Takasu, *RTKLIB ver.2.4.3 demo5 b29*, 2017. [Online]. Available: `http://www.rtklib.com/rtklib.htm`.

[141] P. J. G. Teunissen, "Least-Squares Estimation of the Integer GPS Ambiguities," in *Invited Lecture for Section IV "Theory and Methodology" at the General Meeting of the International Association of Geodesy*, Beijing, China, 1993, pp. 1–16.

[142] The Boeing Company. "Scaneagle product page." (2020), [Online]. Available: `https://www.boeing.com/defense/autonomous-systems/scaneagle/index.page` (visited on Jul. 18, 2022).

[143] The Observatory of Economic Complexity, *2020 Yearly Trade - Historical Data*. [Online]. Available: `https://oec.world/en/profile/country/nor#yearly-trade` (visited on Jul. 18, 2022).

[144] S. Thurrowgood, R. J. D. Moore, D. Soccol, M. Knight, and M. V. Srinivasan, "A biologically inspired, vision-based guidance system for automatic landing of a fixed-wing aircraft," *Journal of Field Robotics*, vol. 31, no. 4, pp. 699–727, 2014. DOI: `10.1002/rob.21527`.

[145] F. A. Toasa, L. Tello-Oquendo, C. R. Penafiel-Ojeda, and G. Cuzco, "Experimental demonstration for indoor localization based on AoA of bluetooth 5.1 using software defined radio," *2021 IEEE 18th Annual Consumer Communications and Networking Conference, CCNC 2021*, pp. 18–21, 2021. DOI: `10.1109/CCNC49032.2021.9369638`.

[146] Truerc.ca, *X-AIR 2.4*. [Online]. Available: `https://www.truerc.ca/shop/2-4ghz-2/receiver-long-range-2-4ghz-2/x-air-2-4` (visited on Jan. 31, 2022).

[147] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*. Wiley, 2011.

[148] D. C. Tsui, *Angle of arrival (AOA) solution using a single receiver (patent US5497161A)*, 1994.

[149] U-Blox, *U-Blox website*, 2019. [Online]. Available: `https://www.u-blox.com/en` (visited on Feb. 11, 2019).

[150] U-blox, *u-blox 8 / u-blox M8 Receiver Description Including Protocol Specification*, 2021. [Online]. Available: https://content.u-blox.com/sites/default/files/products/documents/u-blox8-M8_ReceiverDescrProtSpec_UBX-13003221.pdf (visited on Jul. 19, 2022).

[151] C. F. Van Loan, "Computing Integrals Involving the Matrix Exponential," *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 395–404, 1978. DOI: 10.1109/TAC.1978.1101743.

[152] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, USA: John Wiley and Sons, Inc., 2002. DOI: 10.1002/0471221104.

[153] N. Vasilyuk, M. Vorobiev, and D. Tokarev, "Heading and attitude determination system with low-cost IMU embedded inside one of multiple antennas," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA, USA, 2018, pp. 267–274. DOI: 10.1109/PLANS.2018.8373390.

[154] Vectornav, *VN-300*, 2019. [Online]. Available: https://www.vectornav.com/purchase/product/vn-300 (visited on Feb. 11, 2019).

[155] S. Verhagen, B. Li, and P. Teunissen, *LAMBDA - Matlab implementation, version 3.0*, 2012. [Online]. Available: https://www.researchgate.net/publication/236213370_LAMBDA_software_package_Matlab_implementation_Version_30 (visited on Jul. 19, 2022).

[156] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006. DOI: 10.1007/s10107-004-0559-y.

[157] B. G. Wahlberg, I. M. Y. Mareels, and I. Webster, "Experimental and Theoretical Comparison of som Algorithms for Beamforming in Single Receiver Adaptive Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 1, pp. 21–28, 1991. DOI: 10.1109/8.64430.

[158] A. Walker, "Hard real-time motion planning for autonomous vehicles," Ph.D. dissertation, Swinburne University, 2011.

[159] ——, "Dubins-Curves GitHub repository." (2013), [Online]. Available: https://github.com/AndrewWalker/Dubins-Curves (visited on Mar. 14, 2020).

[160] W. D. White, "Low-Angle Radar Tracking Tracking in the Presence of Multipath," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-10, no. 6, pp. 835–852, 1974. DOI: 10.1109/TAES.1974.307892.

[161] F. C. Wong, F. Veilleux, R. Lee, and E. Gagnon, "Study of factors affecting the successful recovery of fixed-wing UAVs onboard surface vessels," *AUVSI Unmanned Systems North America Conference 2009*, vol. 2, no. February, pp. 1176–1191, 2009.

[162] M. Woolley, *Bluetooth Direction Finding - A Technical Overview v1.0.3*, 2021. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/Files/developer/RDF_Technical_Overview.pdf (visited on Jul. 19, 2022).

[163] J. G. Worms, "RF direction finding with a reduced number of receivers by sequential sampling," in *IEEE International Conference on Phased Array Systems and Technology*, Dana Point, CA, USA, 2000, pp. 165–168. DOI: 10.1109/past.2000.858932.

[164] W. Wu, C. C. Cooper, and N. A. Goodman, "Switched-element direction finding," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1209–1217, 2009. DOI: 10.1109/TAES.2009.5259194.

[165] O. A. Yakimenko, I. I. Kaminer, W. J. Lentz, and P. Ghyzel, "Unmanned aircraft navigation for shipboard landing using infrared vision," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 4, pp. 1181–1200, 2002. DOI: 10.1109/TAES.2002.1145742.

[166] L. Yang, Q. Sun, and Z. S. Ye, "Designing mission abort strategies based on early-warning information: Application to UAV," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 277–287, 2020. DOI: 10.1109/TII.2019.2912427.

[167] Q. Yang, K. Wang, S. H. Li, Y. Liu, and Q. W. Fu, "A Low-cost Attitude Determination Algorithm Based on MEMS IMU dual-antenna GNSS Receiver for UAVs," in *ICINS 2018*, St. Petersburg, Russia: IEEE, 2018. DOI: 10.23919/ICINS.2018.8405847.

[168] L. Yao, "Bluetooth Direction Finding," Master thesis, Delft University of Technology, 2018.

[169] K. C. Yeh and C.-H. Liu, "Radio wave scintillations in the ionosphere," *Proceedings of the IEEE*, vol. 70, no. 4, pp. 324–360, 1982. DOI: 10.1109/PROC.1982.12313.

[170] H. Y. Yen, Z. T. Tsai, Y. C. Chen, L. H. Shen, C. J. Chiu, and K. T. Feng, "I/Q Density-based Angle of Arrival Estimation for Bluetooth Indoor Positioning Systems," *IEEE Vehicular Technology Conference*, vol. 2021-April, pp. 4–8, 2021. DOI: 10.1109/VTC2021-Spring51267.2021.9448850.

[171] S. Yoon, H. J. Kim, and Y. Kim, "Spiral landing trajectory and pursuit guidance law design for vision-based net-recovery UAV," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5682. DOI: 10.2514/6.2009-5682.

[172] D. I. You, Y. D. Jung, S. W. Cho, H. M. Shin, S. H. Lee, and D. H. Shim, "A guidance and control law design for precision automatic take-off and landing of fixed-wing UAVs," in *AIAA Guidance, Navigation, and Control Conference*, 2012. DOI: 10.2514/6.2012-4674.

[173] P. Zand, J. Romme, J. Govers, F. Pasveer, and G. Dolmans, "A high-accuracy phase-based ranging solution with Bluetooth Low Energy (BLE)," in *IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2019. DOI: 10.1109/WCNC.2019.8885791.

ISBN 978-82-326-6247-0 (printed ver.)
ISBN 978-82-326-6547-1 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology