ISBN 978-82-326-4744-6 (printed version)
ISBN 978-82-326-4745-3 (electronic version)
ISSN 1503-8181

Doctoral Thesis
Kristoffer Gryte

Doctoral theses at NTNU, 2020:198

Kristoffer Gryte

Precision control of fixed-wing UAV and robust navigation in GNSS-denied environments

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

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2020:198
NTNU

NTNU
Norwegian University of
Science and Technology

Kristoffer Gryte

Precision control of fixed-wing UAV and robust navigation in GNSS-denied environments

Thesis for the degree of Philosophiae Doctor

Trondheim, June 2020

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 978-82-326-4744-6 (printed version)
ISBN 978-82-326-4745-3 (electronic version)
ISSN 1503-8181

ITK-report: 2020-5-W

Doctoral theses at NTNU, 2020:198

Printed by Skipnes Kommunikasjon as

Summary

Unmanned aerial vehicles (UAVs) have been adopted to a wide range of applications, but fully autonomous flight is still utopian, especially for safety-critical applications. This thesis seeks to close this gap by mainly considering robust navigation and precision control.

Navigation systems of safety-critical unmanned aircraft need an alternative position aiding source to global navigation satellite system (GNSS), as it is vulnerable to naturally occurring disturbances, such as ionospheric scintillation, and to malicious disturbances. Two promising alternatives, are position measurements from ultra-wideband radio beacons and phased-array radio systems. Tight integration of ultra wideband (UWB) range measurements with real-time kinematic (RTK) aiding of inertial navigation for increased robustness to GNSS dropouts, is achieved using a double-differenced nonlinear observer. The results are verified using a simulated unmanned aerial vehicle (UAV) with realistic inertial, GNSS and UWB measurements. The phased-array-radio-based navigation system consists of a multiplicative extended Kalman filter that utilize these measurements, along with an exogenous altitude measurement, to provide corrections to the inertial navigation system, and performs measurement outlier rejection to mitigate effects of radio measurement reflections. In addition to providing a positioning solution, the navigation system make magnetometers and magnetic compasses made superfluous during flight, by providing heading estimates. The accuracy, reliability and versatility of the navigation system are demonstrated through three series of experiments. First, experimental results obtained from a multirotor and a fixed-wing UAV flight are presented, showing a 14 % improvement in root-mean-square error compared to previous results. Then an online guidance, navigation and control system, based on the navigation system is demonstrated in beyond-visual-line-of-sight flights in controlled airspace, with a fixed-wing UAV, without the use of GNSS.

More precise control can enable the use of fixed-wing UAVs in a wider range of applications, in particular landing, without the use of a skilled remote control pilot. Commercial-off-the-shelf autopilots are reliable, given their extensive use and testing, and some even provide automatic landing capabilities. They are, however, not sufficient for automatic recovery in moving arrest systems, such as nets, which are simple infrastructures that can be installed in crammed or remote locations, such as aboard ships. Under general assumptions on the autopilot, it is extended by

automatic plan generation, line-of-sight guidance along moving lines and enhanced RTK-GNSS positioning relative to the net, in a modular, non-intrusive manner to remain reliable. The system is demonstrated by two series of experiments; in a stationary net, to demonstrate controller performance, and in a manually moving net, to demonstrate compensation of arrest system motion.

Another landing strategy is through deep-stall, where the large increase in aerodynamic drag during stall is used to significantly reduce the speed, and thus impact, of the UAV during landing. One way to increase the precision compared to existing solutions, is through the presented nonlinear model predictive controller, which use a six degree of freedom model of the UAV to determine control surface deflections that will steer it to a desired landing location, along a predetermined decent angle, while minimizing speed and respecting dynamic and actuator constraints. An extensive simulation study shows that the controller handles a large span of wind speeds, but is sensitive to wind gusts.

To enable operation in a wider range of conditions, path following for fixed-wing UAVs should emphasize mitigation of wind effects, which at the very least implies a formulation using course angles. In this thesis, this is achieved by basing the guidance law on the coordinated turn equation, which is derived in in a general form for completeness, to ensure that no effects are unintentionally ignored. By cascading the coordinated-turn based controller with a line-of-sight guidance law, uniform semiglobal exponential stability of the cross-track error and course angle is proven. The controller is easy to tune, in terms of damping ratio and bandwidth. Through a comparative simulation study to the state-of-the-art, four guidance laws based on the analysis are shown to reduce the maximum cross-track error by 10 % in wind. The system shows merit through an experimental verification.

Mathematical models of UAVs can be used both to obtain higher precision control, through model-based control techniques, to improve robustness, through e.g. fault detection, and to cut development time, cost and risk, through a simulator. For this reason, a complete six degrees-of-freedom UAV model is presented, based on a combination of wind tunnel testing and numeric analysis for the aerodynamic model, wind tunnel testing for the propulsion model, and bifilar pendulum testing to identify moments of inertia.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU). The work that forms the foundation of this thesis has been carried out at the Department of Engineering Cybernetics and the NTNU Centre for Autonomous Marine Operations and Systems (NTNU AMOS). Professor Thor Inge Fossen has been the main supervisor of this work, while Professor Tor Arne Johansen and Associate Professor Torleiv Håland Bryne have been co-supervisors. The Research Council of Norway has co-funded this project through the Centres of Excellence funding scheme with grant number 223254.

Acknowledgments

When I started this project, in August 2015, I knew little the academic and psychological roller coaster I had signed up for. The journey has been filled with eager to learn more, deadlines that are a bit too close, new students who do not remember what I taught last years students, painful git merges, a few all-nighters, ... but most off all; a good time with great friends. I want to thank each and every one of you who shared this period of my life with me, sometimes carrying me through, both in life and in academics. Then there are a few persons that deserves an extra appreciation, as this journey would have had a different outcome without you.

First I would like to thank my office mate Håkon. If only we could publish half of the solutions to all the problems, both within academia, sports and dating, we have solved under four eyes, we would lead the publication statistics by a long shot. It could appear that Håkon, outdoor-guru Bjørn-Olav, Michelin-chef Erik, and Mathias, who always offers a different perspective, are stuck at with me NTNU. Let this be a symbol of how much we have enjoyed it!

Andreas, my partner in crime/lab/skiing: thanks for all the things you have introduced me to, that I now consider paramount, notably vim and Linux. If I ever end up in a war, I'm bringing you!

John Martin, my first friend in Trondheim: thank you for the long discussions on air/hydrofoils and their control, and for your inspiring, never ending quest for knowl-

edge. Thanks Jostein for all the insights and inspiration, which was enough to convince me to enter the world of cybernetics.

In academia you are measured in publications, so thanks to all my co-authors. Behind every publication, there are a lot of steps that are import, but easy to forget about. Thanks to Kristian, João and Frederik, for holding my hand through DUNE and open source development, and to Artur, the lord of all jacks-of-all-trade, who always has a solution. Martin, thanks for showing me the extra dimension when it comes to attention to details, resulting in investigative skills beyond compare that have been crucial for experiments. Thanks to the administrative staff at the department, for filtering out most of the bureaucracy from above, and for making me feel welcome and appreciated. This thesis involves flying vehicles. Thanks to Pål at the department, Lars and Carl Erik at Maritime Robotics, and Thomas at Nordic Unmanned for operating them. I'm sorry if the results of this thesis puts you out of business.

I have also had the opportunity to collaborate with several industrial partners; Jon Are Kolstad, Inge Aune Paulsen, Øystein Pedersen and Atle Sægrov at Radionor Communications, and Einar Nielsen at Petroleum Geo-Services (PGS). Collaboration with people with such talents in one field, who still appreciate and rely on inputs from my field, was truly inspiring. Thank you also for bringing me out of the academic bubble.

I have been lucky enough to be supervised by Thor Inge and Tor Arne, otherwise known as TnT (because together they are dynamite), T(h)or and Bob, etc. Thor; the world always seems more straight forward after a meeting in your office. I'm not sure if it is mostly caused by how quick you come up with solutions, or mostly caused by an inspirational discussion about skiing, but it has been much appreciated. Tor Arne; your academic and emailing undertakings are beyond compare, yet your people skills is what I admire the most. During my PhD I also became friends with then-PhD-candidate Torleiv, whom I now proudly call my co-supervisor. Torleiv; I attribute parts of my big guns to you. Thanks for the interesting discussions, on all topics, and for teaching me how papers are written.

Thanks to future UCI World Tour team, Team Nerd, i.e. Glenn, Haavard, Dirk, Fredrik and Tobias, and Trine at SiT Idrett for my physical (and thus mental) health. And I'm sorry, Inger and Gunhild, for all the coffee breaks I have missed.

Finally, I would like to thank my closest family for their unconditional support. Born by a scientist-of-sorts and a pseudo-engineer, perhaps this was my destiny. Mom; thank you for everlasting interest, for always being able to relate and for making me aware of my mindset. Dad; thank you for sharing your investigative skills, aspiring us to see how the world is connected, for pushing me, and for always having a solution to my practical problems. Magne; thanks for listening to all my science crap over the years. Seems like some of it stuck, though. You make me proud for following your dreams, especially when they require effort. My most important discovery during this PhD was you, Siri. Thank you for accepting, and even facilitating, this monstrous ego trip.

Contents

Chapter 1

Introduction

This thesis considers modeling, control, guidance and navigation of unmanned aerial vehicles (UAVs), in an effort to increase the level of autonomy of their operations. This chapter seeks to put the findings of the thesis in a broader context, by providing background and motivation for the work. In light of this, the main contributions are listed, along with the publication in which they are made. Finally, the structure of the rest of this thesis is presented.

1.1 Background and motivation

With a coastline of over 50 000 km, the sea has always played an important role for Norway. Today this is seen through the countrys two most revenue-bringing industries, seafood and the oil and gas sector. In 2018, the country exported seafood for 99 billion NOK[139], and oil and gas for 442 billion NOK in 2017. These earnings, and the historical importance of the sea, makes research in the maritime environment important for Norway, and the focus of many of its research institutions, such as the Center for Autonomous Marine Operations and Systems (NTNU AMOS), which includes the NTNU Unmanned Aerial Vehicle laboratory (UAVlab).

The miniaturization of electronics over the past two decades has enabled the cost effective use of small UAVs as a tool for gathering information from the maritime environment. In the *information pyramid*, UAVs are considered mid-level, as they cover medium sized areas at a medium resolution, when compared to high-area/low-resolution satellites and low-area/high-resolution surface vessels. Some UAV applications are detection of drifting, potentially damaging ice-bergs near critical infrastructure [105], estimation of oxygen-consuming algae near fish farms, and detection of floating debris along a ships path.

Many of these applications are continuous, everlasting, monitoring of environments that are largely feature-less, making them expensive and dull, so to increase the economic feasibility and safety, they should be automated, overthrowing the skilled

remote control pilot. They also require long endurance, as takeoffs and landings should be minimized to increase the time of effective operation. Therefore, fixed-wing UAVs capable of beyond-visual-line-of-sight flight are desirable.

From the title 'Precision control... and robust navigation...' it is clear that this thesis mainly seek to improve autonomy in two ways; by increasing precision and robustness.

Increased precision enable operations that are difficult to perform today, such as landing. Today, most landings in low-cost, civilian missions require a trained pilot to manually land the UAV. By increasing the precision, difficult operations, such as landing on highly moving platforms, can be automated, thus extending the operability. Another difficult operation, for some autopilots, is to consider wind, and thus flying exactly where the operator intended. Wind is much more dominating, and limiting, for small UAVs than for large, manned aircraft, since the airspeeds in which they operate are much closer to the wind speeds they encounter.

Increased robustness will enable UAVs to operate in a larger range of conditions, with lower risk. There are several conditions that render a UAV mission very risky, or even infeasible. The primary positioning sensor for UAVs is global navigation satellite systems (GNSS), such as GPS, which offers accurate position measurements at a low price to the end user. In the start of this work, low-cost GNSS receivers where mainly limited to single-frequency, single-constellation, such as the Ublox LEA/NEO-M8T, where the precision could be improved by third-party software RTKLIB [182], providing real-time kinematics (RTK) capabilities. The quality and ease of use has improved significantly during this period, with multi-frequency, multi-constellation receivers for the consumer market, such as the Ublox ZED-F9P, providing RTK *out-of-the-box*. For a typical mission at the NTNU UAVlab, the receiver now see around 30 satellites, compared to between 5 and 10 with the older receivers. Despite this improvement, GNSS is still vulnerable to radio frequency interference (RFI), either naturally occurring, such as ionospheric scintillations [199] and multipath, or intentional RFI, such as jamming [149] and spoofing [92, 165]. In the extreme, this has lead to hostile takeover of UAVs [80, 24]. Another potential drawback with GNSSs is that they are controlled by international bodies, which means that they are of the first services to become unavailable on the rise of a large international conflict. They are also complex, illustrated in July 2019 when the positioning and timing services provided by Galileo of the European Global Navigation Satellite System Agency (GSA) were unavailable for a full week before it was fixed [89]. This motivates the investigation into GNSS-free alternatives for navigation, of which there are many, each with its own limitation. Visual navigation, such as visual odometry [48, 132] and visual SLAM [43, 133] are limited to local, feature-rich environments, typically indoors, while using range measurements from other sources [40], such as ultra-wideband (UWB) [55, 112], relies on local infrastructure, and is limited by the range of the radios. Flying far, over the open, feature-less water, of which the majority of Norways economic interests are covered, has not been done without GNSS.

Another vulnerability to UAVs comes from conditions in the environments in which they operates. As mentioned, UAVs are more affected by wind than larger aircraft, which motivates research into online estimation of wind parameters [194] to be used in adaptive path planning [15]. Better models could give a better understanding of how the wind affects the UAV, and can be used in planning and guidance to increase performance, in a wider range of conditions. As with large aircraft, icing on the wings, propeller and control surfaces is also a problem for UAVs [177], with the ultimate consequence of losing the vehicle. However, the typically passive solutions to mitigate this problem used in large aircraft, such as spraying the aircraft with a chemical prior to takeoff, have proven difficult to scale for UAVs, which has given rise to new, active solutions [190], that are applied when needed. One way to detect faults, such as icing, is to compare the behavior of the UAV with the *expected* behavior, which requires an accurate model.

Such models can also be used in simulators, which is an important tool in development and testing of UAVs. Simulators can also be used in operator training by practicing for different scenarios, thus lowering operation risk and increasing the success rate. While there are many UAV simulator frameworks available, such as JSBSim [16] which is the standard software-in-the-loop simulator in the ArduPilot project [11], but the number of available UAV models is small, with the SIG Rascal 110 [83] being a notable exception. In particular, the UAVs used by the NTNU UAVlab lack available models.

1.2 Contributions

In the light of the above, this thesis has contributed to the research community by

- deriving a complete model of the Skywalker X8 fixed-wing UAV based on identification of aerodynamic, propulsion and inertial parameters.
- deriving the coordinated-turn equation using course angles, and considering wind effects
- deriving, proving and experimentally verifying a lateral, nonlinear line-of-sight guidance law for fixed-wing unmanned aerial vehicles in wind, based on the coordinated-turn relation
- designing and testing a modular system for landing of fixed-wing UAV in a moving arrest system, built around a general autopilot that accepts desired position commands and provides its own position, velocity and attitude
- redefining what is possible by demonstrating beyond-visual-line-of-sight flight in controlled airspace, 5 km away from the takeoff location, without the use of global navigation satellite systems, by instead relying on phased-array radio systems to aid the inertial navigation
- confirming that ultra-wideband radio beacons is another promising position sensor, particularly for landing and other close-range operations

- demonstrating that the precision of deep stall landing can be improved by nonlinear model predictive control

1.3 Publications

This thesis is based on the following articles published in peer-review international journals and conferences:

- [61] K. Gryte, T. H. Bryne, and T. A. Johansen, "Unmanned aircraft flight control aided by phased-array radio navigation", *Journal of Field Robotics*, 2020, Submitted
- [64] K. Gryte, T. A. Johansen, and T. I. Fossen, "Coordinated-turn based path following for fixed-wing unmanned aircraft", *Journal of Guidance, Control, and Dynamics*, 2020, Submitted
- [65] K. Gryte, M. L. Sollie, T. A. Johansen, and T. I. Fossen, "Control system architecture for automatic recovery of fixed-wing unmanned aerial vehicles in a moving arrest system", 2020, In progress
- [23] T. H. Bryne, K. Gryte, S. M. Albrektsen, and T. A. Johansen, "GNSS-free navigation of unmanned aerial vehicles based on phased-array radio systems", *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 1–15, 2020, Submitted
- [60] K. Gryte, T. H. Bryne, S. M. Albrektsen, and T. A. Johansen, "Field test results of GNSS-denied inertial navigation aided by phased-array radio systems for UAVs", in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 1398–1406. DOI: 10.1109/ICUAS.2019.8798057
- [63] 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
- [62] 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 *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2018, pp. 826–835
- [123] 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 SciTech*, 2016
- [27] E. M. Coates, A. Wenz, K. Gryte, and T. A. Johansen, "Propulsion system modeling for small fixed-wing UAVs", in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 748–757. DOI: 10.1109/ICUAS.2019.8798082

The following articles have been published during the PhD, but are not part of the thesis:

- [26] 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 *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–10. DOI: 10.1109/AERO.2018.8396534
- [121] S. G. Mathisen, F. S. Leira, H. H. Helgesen, K. Gryte, and T. A. Johansen, "Autonomous ballistic airdrop of objects from a small fixed-wing unmanned aerial vehicle", *Autonomous Robots*, Jan. 2020. DOI: 10.1007/s10514-020-09902-3
- [120] S. G. Mathisen, K. Gryte, S. Gros, and T. A. Johansen, "Precision deep stall landing of fixed-wing UAVs using nonlinear model predictive control", *Journal of intelligent and robotic systems*, 2020, Submitted
- [196] A. Winter, R. Hann, A. Wenz, K. Gryte, and T. A. Johansen, "Stability of a flying wing uav in icing conditions", *8th European conference for aeronautics and aerospace sciences (EUCASS)*, pp. 1–15, Jun. 2019. DOI: 10.13009/EUCASS2019-906
- [70] R. Hann, A. Wenz, K. Gryte, and T. A. Johansen, "Impact of atmospheric icing on uav aerodynamic performance", in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, Oct. 2017, pp. 66–71. DOI: 10.1109/RED-UAS.2017.8101645
- [156] D. Rotondo, A. Cristofaro, K. Gryte, and T. A. Johansen, "LPV model reference control for fixed-wing UAVs", *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 559–11 564, 2017, 20th IFAC World Congress. DOI: 10.1016/j.ifacol.2017.08.1640

1.4 Outline

The thesis is divided into three main parts: modeling, guidance and control, and navigation. But before this, chapter 2 introduces some common mathematical concepts and notation, that are used throughout the thesis.

The modeling, in part I, presents a complete model of the Skywalker X8 flying-wing UAV. Chapter 4 presents the modeling of the aerodynamic forces and moments, based on wind tunnel testing and numerical analysis, while chapter 5 presents a propulsion model. To complete the UAV model, its inertial properties are identified in chapter 6, using data recorded from a bifilar pendulum in a nonlinear programming problem. Finally what is known as the coordinated turn equation is derived in a new way, in chapter 7.

Part II presents the guidance and control part, by considering three different problems. First, chapter 11 presents a solution to deep stall landing using nonlinear model predictive control, through an extensive simulation study, by relying on the model structures from chapter 4. The second problem, in chapter 9, presents a modular system for landing in a moving arrest system, based on commercial-off-the-shelf

component, demonstrated through physical experiments for both a stationary and a moving arrest system. A sub-component of the landing system, the lateral guidance controller, is investigated in chapter 10, by combining the coordinated turn equation from chapter 7 with line-of-sight-based path following. This is demonstrated through experimental verification and a simulation study, comparing with the state-of-the-art, in different wind conditions.

The last part, part III, considers navigation without the use of global navigation satellite systems, such as GPS, by aiding inertial navigation with two different position measurement sensors, using two different strategies: Ultra-wideband radio beacons in chapter 13 that use a nonlinear observer, and phased-array radio systems in chapter 14 that use a multiplicative extended Kalman filter.

Finally, the work is summarized in chapter 15, where also some future directions are laid out.

CHAPTER 2

PRELIMINARIES

2.1 Coordinate frames

The following frames are considered:

- The non-rotating north-east-down (NED) frame, $\{n\}$, where the x-axis is pointing north, the y-axis east and the z-axis down.
- The ground velocity frame, $\{P\}$, which is a NED frame rotated by the course angle χ around the z-axis, and by the flight path angle γ around the y-axis, such that the x-axis is aligned with the ground velocity vector. This frame is centered in the UAV.
- The tangential $\{t\}$ frame, which is a horizontal projection of the ground velocity frame.
- The body frame $\{b\}$, where the x-axis is pointing out the nose of the UAV, the y-axis out the right wing and the z-axis through the belly.
- The wind frame $\{w\}$, where the x-axis is pointing along the air velocity vector, the y-axis is rotated around the x-axis by the air relative bank angle μ_a from the horizontal, while the z-axis completes the right-hand system.
- Chapter 14 considers the PARS ground radio antenna coordinate frame, denoted $\{r\}$. This coordinate system is a rotated NED frame.
- Chapter 13 considers the Earth-Centered-Earth-Fixed (ECEF) frame, denoted $\{e\}$.

The relation between the different frames are illustrated in figs. 2.1 and 2.3.

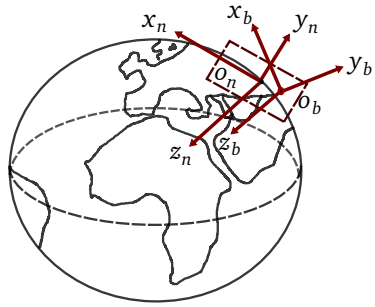


Figure 2.1: Definitions of the BODY and the NED coordinate frames.

2.2 Notation

The Euclidean vector norm is denoted $\|\bullet\|_2$, where $\bullet$ is the variable placeholder. The $n \times n$ identity matrix is denoted $\boldsymbol{I}_n$. The transpose of a vector or a matrix is denoted $(\bullet)^\intercal$, while $(\bullet)^\mathsf{H}$ denotes the conjugate transpose. Coordinate frames are denoted with $\{\cdot\}$. $\boldsymbol{S}(\bullet) \in SS(3)$ represents the skew symmetric matrix such that $\boldsymbol{S}(\boldsymbol{z}_1)\boldsymbol{z}_2 = \boldsymbol{z}_1 \times \boldsymbol{z}_2$ for two vectors $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^3$. $\boldsymbol{z} = (\boldsymbol{z}_1; \boldsymbol{z}_2; \ldots; \boldsymbol{z}_n)$ denotes a vector of stacked column vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots \boldsymbol{z}_n$. Furthermore, error variables are represented with with $\delta\bullet$. Partial derivatives are denoted with $\partial \bullet_1 / \partial \bullet_2$. In addition, $\boldsymbol{z}^{\zeta}_{\gamma\beta} \in \mathbb{R}^3$ denotes a vector $\boldsymbol{z}$, to frame $\{\beta\}$, relative $\{\gamma\}$, decomposed in $\{\zeta\}$. The $\text{diag}(\bullet, \ldots, \bullet)$ function places the n arguments on the diagonal of a square matrix with n rows and n columns. Moreover, $\varepsilon_\bullet \sim \mathcal{N}(0, \sigma^2_\bullet)$ denotes Gaussian white noise with standard deviation $\sigma_\bullet$ and $\mathbb{E}[\bullet]$ denotes the expected value. Furthermore, T_s represent the sampling time or step length in numerical integration methods. The imaginary unit is denoted j.

2.2.1 Attitude representations and relationships

An attitude representation used in this work is the unit quaternion, using the Hamiltonian representation, given as

$$\boldsymbol{q}^{\gamma}_{\beta} = \begin{bmatrix} q_s \\ \boldsymbol{q}_v \end{bmatrix} = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} \in \mathcal{Q} \tag{2.1}$$

where the set $\mathcal{Q}$ is defined according to [50] as

$$\mathcal{Q} := \{\boldsymbol{q}^{\gamma}_{\beta} \mid (\boldsymbol{q}^{\gamma}_{\beta})^\intercal \boldsymbol{q}^{\gamma}_{\beta} = 1, \boldsymbol{q}^{\gamma}_{\beta} = (q_s; \boldsymbol{q}_v), q_s \in \mathbb{R}^1, \boldsymbol{q}_v \in \mathbb{R}^3\} \tag{2.2}$$

Conforming to Gade's notation [53], the quaternion can be used to calculate rotation matrix, $\boldsymbol{R}_{ab} \in SO(3)$,

$$\boldsymbol{R}_{\gamma\beta}(\boldsymbol{q}^{\gamma}_{\beta}) = \left(q_s - \boldsymbol{q}_v^\intercal \boldsymbol{q}_v\right)\boldsymbol{I}_3 + 2q_s\boldsymbol{S}(\boldsymbol{q}_v) + 2\boldsymbol{q}_v\boldsymbol{q}_v^\intercal, \tag{2.3}$$

as in e.g. [115, Eq. (4)] and [45, App. D.2]. The notation $\boldsymbol{R}_c(d)$ emphasize that the transformation is a rotation of d degrees around the c-axis.

The Hamiltonian quaternion product, denoted $\otimes$, is given such that [175, Sec. 1.2.2]

$$\begin{aligned}\boldsymbol{q}_3 = \boldsymbol{q}_1 \otimes \boldsymbol{q}_2 &= [\boldsymbol{q}_1]_L \boldsymbol{q}_2 = [\boldsymbol{q}_2]_R \boldsymbol{q}_1 \\ &= \begin{bmatrix} q_{1_s} q_{2_s} - \boldsymbol{q}_{1_v}^\intercal \boldsymbol{q}_{2_v} \\ q_{1_s} \boldsymbol{q}_{2_v} + q_{2_s} \boldsymbol{q}_{1_v} + \boldsymbol{S}\left(\boldsymbol{q}_{1_v}\right) \boldsymbol{q}_{2_v} \end{bmatrix},\end{aligned} \tag{2.4}$$

where

$$[\boldsymbol{q}]_L = q_s \boldsymbol{I}_4 + \begin{bmatrix} 0 & -\boldsymbol{q}_v^\intercal \\ \boldsymbol{q}_v & \boldsymbol{S}(\boldsymbol{q}_v) \end{bmatrix} \tag{2.5}$$

is the left quaternion production matrix, while

$$[\boldsymbol{q}]_R = q_s \boldsymbol{I}_4 + \begin{bmatrix} 0 & -\boldsymbol{q}_v^\intercal \\ \boldsymbol{q}_v & -\boldsymbol{S}(\boldsymbol{q}_v) \end{bmatrix} \tag{2.6}$$

is the right quaternion production matrix. Moreover, the differential equation of a given unit quaternion $\boldsymbol{q}_1$ is given as

$$\dot{\boldsymbol{q}}_1 = \frac{1}{2} \boldsymbol{q}_1 \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2} [\bar{\boldsymbol{\omega}}]_R \boldsymbol{q}_1 = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}) \boldsymbol{q}_1 \tag{2.7}$$

where $\bar{\boldsymbol{\omega}} = (\boldsymbol{0}; \boldsymbol{\omega})$ and

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^\intercal \\ \boldsymbol{\omega} & -\boldsymbol{S}(\boldsymbol{\omega}) \end{bmatrix}. \tag{2.8}$$

Similarly, for a given unit quaternion $\boldsymbol{q}_2$, the differential equation is given as

$$\dot{\boldsymbol{q}}_2 = -\frac{1}{2} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\omega} \end{bmatrix} \otimes \boldsymbol{q}_2 = -\frac{1}{2} [\bar{\boldsymbol{\omega}}]_L \boldsymbol{q}_2 = -\frac{1}{2} \boldsymbol{\Gamma}(\boldsymbol{\omega}) \boldsymbol{q}_2, \tag{2.9}$$

where

$$\boldsymbol{\Gamma}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}^\intercal \\ \boldsymbol{\omega} & \boldsymbol{S}(\boldsymbol{\omega}) \end{bmatrix}. \tag{2.10}$$

Furthermore, the quaternion conjugate is denoted as

$$\boldsymbol{q}^* = \begin{bmatrix} q_s, & -\boldsymbol{q}_v \end{bmatrix}^\intercal, \tag{2.11}$$

and where conjugation of the quaternion product eq. (2.4) results in

$$\boldsymbol{q}_3^* = \left(\boldsymbol{q}_1 \otimes \boldsymbol{q}_2\right)^* = \boldsymbol{q}_2^* \otimes \boldsymbol{q}_1^*. \tag{2.12}$$

The Euler angles (roll, pitch and yaw) are given as

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi, & \theta, & \psi \end{bmatrix}^\intercal. \tag{2.13}$$

2.2.2 Attitude error

In this work, the attitude error is represented using four times the Modified Rodrigues Parameters (MRP)

$$\delta \boldsymbol{a} = 4\delta \boldsymbol{a}_{\text{mrp}} = 4\frac{\delta \boldsymbol{q}_v}{1+\delta q_s} \tag{2.14}$$

as given in [115]. This choice is founded on the knowledge that both the MRP and the four times MRP (4xMRP) are minimum representations of the attitude error, and that their singularities are outside the range of the attitude error. The 4xMRP is chosen over the MRP since its units are in rad, as opposed to $1/4$ rad.

2.2.3 Kinematics – Strapdown equations

The position and linear velocity of the BODY frame relative the NED frame is represented as $\boldsymbol{p}^n_{nb} \in \mathbb{R}^3$, $\boldsymbol{v}^n_{nb} = [v_N, v_E, v_D]^\intercal \in \mathbb{R}^3$. The rotation of the body about the NED frame is parameterized with the unit quaternion $\boldsymbol{q}^n_b$. The angular velocity of BODY w.r.t. to NED is given as $\boldsymbol{\omega}^b_{nb} \in \mathbb{R}^3$, while the gravity vector is given as $\boldsymbol{g}^n_b$. With this stated, the resulting stapdown equations follows

$$\begin{aligned} \dot{\boldsymbol{p}}^n_{nb} &= \boldsymbol{v}^n_{nb} \\ \dot{\boldsymbol{v}}^n_{nb} &= \boldsymbol{R}_{nb}(\boldsymbol{q}^n_b)\boldsymbol{f}^b_{nb} + \boldsymbol{g}^n_b \\ \dot{\boldsymbol{q}}^n_b &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b_{nb})\boldsymbol{q}^n_b \end{aligned} \tag{2.15}$$

based on the underlying assumption above that this NED frame is assumed to be inertial, where,

$$\boldsymbol{f}^b_{nb} = \boldsymbol{R}^\intercal_{nb}(\boldsymbol{q}^n_b)\left(\dot{\boldsymbol{v}}^n_{nb} - \boldsymbol{g}^n_b\right). \tag{2.16}$$

represent the specific force.

Rotated into the body frame, $\boldsymbol{v}^n_{nb}$ becomes

$$\boldsymbol{v}^b_{nb} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \boldsymbol{R}^b_n \boldsymbol{v}^n_{nb}. \tag{2.17}$$

The ground ground speed is given as $V_g = \|\boldsymbol{v}\| = \sqrt{v_N^2 + v_E^2 + v_D^2}$. The direction of the ground velocity vector, relative to NED, is a rotation χ around the z-axis followed by γ around the current y-axis, see fig. 2.3. χ is known as the course angle, while γ is the (flight) path angle.

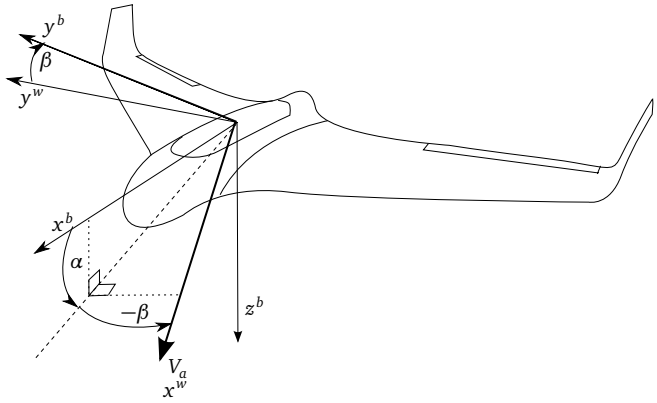


Figure 2.2: Wind and body axes

2.3 Aerodynamics

The air velocity vector is defined as

$$\boldsymbol{V}_a^b = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix}, \tag{2.18}$$

such that the airspeed is given as $V_a = \|\boldsymbol{V}_a\| = \sqrt{u_r^2 + w_r^2 + w_r^2}$, where the different components can be found from

$$\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix}, \tag{2.19}$$

$$\mathbf{V}_w^b = \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix} = R_n^b(\mathbf{q}) \begin{bmatrix} w_{n_s} \\ w_{e_s} \\ w_{d_s} \end{bmatrix} + \begin{bmatrix} u_{w_g} \\ v_{w_g} \\ w_{w_g} \end{bmatrix} \tag{2.20}$$

Then the angle of attack α and sideslip angle β can be defined as

$$\alpha = \tan^{-1} \frac{w_r}{u_r}, \quad \beta = \sin^{-1} \frac{v_r}{V_a}, \tag{2.21}$$

whose relation to body axes are seen in figs. 2.2 and 2.3. The direction of the air velocity vector, relative to NED, is a rotation χ_a around the z-axis followed by γ_a around the current y-axis, see fig. 2.3. χ_a is known as the air-mass-referenced course angle, while γ_a is the air-mass-referenced (flight) path angle. The relationship between the air velocity vector, ground velocity vector and the body axis, known as the wind triangle, is depicted in fig. 2.4.

A general model of the aerodynamic forces and moments, given in the stability frame, acting on a fixed-wing UAV is given in eq. (2.22). The aerodynamic moments l, m, n are about the x, y, z axes of the body frame. The drag force D, side force Y and lift force L acting on an object are typically modeled in terms of the dimensionless functions C_D, C_Y and C_L. Similarly, the roll, pitch and yaw moments, are modeled in terms of C_l, C_m and C_n, as well as the characteristic lengths b and c that represent wing span and mean aerodynamic chord.

$$\begin{aligned}\begin{bmatrix} D \\ Y \\ L \end{bmatrix} &= \frac{1}{2}\rho V_a^2 S \begin{bmatrix} C_D(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r) \\ C_Y(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r) \\ C_L(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r) \end{bmatrix} \\ \begin{bmatrix} l \\ m \\ n \end{bmatrix} &= \frac{1}{2}\rho V_a^2 S \begin{bmatrix} bC_l(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r) \\ cC_m(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r) \\ bC_n(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r) \end{bmatrix}\end{aligned} \tag{2.22}$$

Here, $\delta_a, \delta_e, \delta_r$ are the control inputs to the aileron, elevator and rudder. S is the area of the wing, and ρ is the air density. Equation (2.22) makes the common assumptions that Froude-, Mach-, Strouhal- and Reynolds number effects are small, and that the mass and inertia of the aircraft is dominating that of the surrounding air[98].

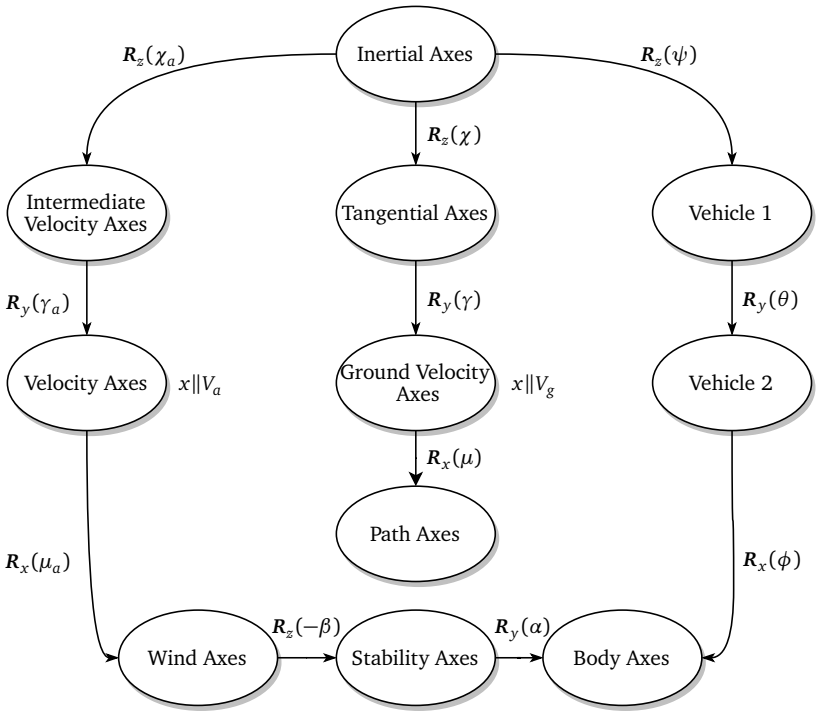


Figure 2.3: Transformations between the different axes, inspired by [179, fig. 2.2-7] and [14]

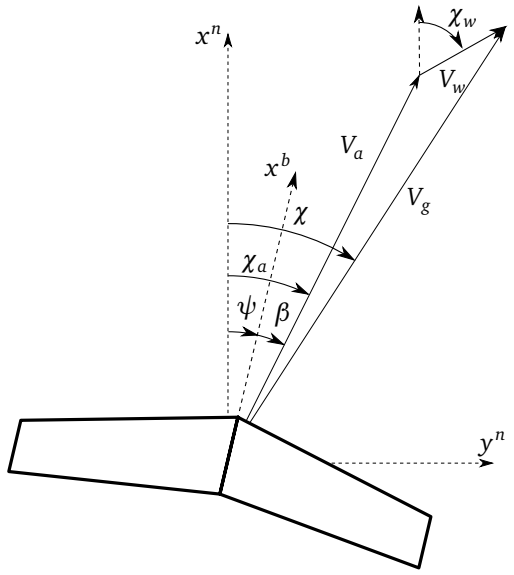


Figure 2.4: The wind triangle, assuming $\chi_a = \psi + \beta$.

Part I

Modelling

Chapter 3

Introduction

The use of mathematical models to describe a fixed-wing aircraft's motion is a well known field. There are many reasons why a mathematical description of the aircraft motion is useful. First and foremost, it can be used in the design process of the aircraft, by considering how different design changes affect the performance. Another important use for an accurate model is simulation of flight conditions, which provides vital aiding for both pilots and control engineers, by saving time and money spent in initial training and control system testing. However, the applicability of the tests are limited by the accuracy of the model. A mathematical description of an unmanned aerial vehicle (UAV) is also crucial in model based control strategies. One example is model predictive control (MPC), where the model is used to find the control input needed to achieve the control objective, by predicting the response of the UAV. See for example chapter 11, where MPC is used in deep stall landing. Similarly, model based estimation relies on a UAV model to propagate the estimates forward in time. See e.g. [20], where aerodynamic models are used in estimating the angle-of-attack and side-slip-angle of a UAV.

This chapter considers different parts of a UAV model. Chapter 4 considers aerodynamic modeling of the Skywalker X8 flying-wing UAV, based on wind tunnel testing and numerical analysis. However, the aerodynamic forces only tell a part of the story. Therefore, chapter 5 presents a propulsion model for the same UAV, based on data collected from another wind tunnel experiment. To be able to translate these forces and moments into linear and angular accelerations, through Newtons second law, the inertial properties of the UAV are required. They are identified in chapter 6, by means of a bifilar pendulum. Some applications, such as guidance or plan generation, might not require such detailed dynamic models. Therefore, chapter 7 presents a kinematic model for the coordinated turn.

Chapter 4

Aerodynamic modeling

The following chapter presents the aerodynamic model of the Skywalker X8 flying-wing UAV, based on the publication

- [62] 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 *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2018, pp. 826–835

It is also noted that the work presented in chapter 4 has been extended in

- [196] A. Winter, R. Hann, A. Wenz, K. Gryte, and T. A. Johansen, "Stability of a flying wing uav in icing conditions", *8th European conference for aeronautics and aerospace sciences (EUCASS)*, pp. 1–15, Jun. 2019. DOI: 10.13009/EUCASS2019-906

to include icing effects.

4.1 Introduction

Development of an aerodynamic model requires intimate knowledge of aerodynamics, in planing the wind tunnel tests and in analyzing the results, as well as knowledge in regression analysis, due to the large amounts of data produced by the tests.

One important tool in creating an aerodynamic model of a flying object is wind tunnel testing. Wind tunnels cause the air to move, instead of the flying object, so that the characteristics of how the flying object is affected by the air can be studied more easily. A problem with wind tunnel measurements, as with any measurement, is measurement errors. In the context of mathematical modeling, the objective of wind tunnel testing is to find the model, consisting of a structure and a set of parameters, that best describe the measured resulting aerodynamic forces and moments, given

the measured characteristics of the flow and the forces and moments on aircraft[98, 82]. One possibility in identifying the parameters of the underlying aerodynamic model is through regression analysis, such as the least squares method.

Another option for creating an aerodynamic model is to perform system identification on data from flight tests. This refers to the process of designing appropriate control inputs to excite the system, measuring the response, and determine the type of model, and fitting the data to it. The advantages of this method is its close resemblance of real flight; any motion experienced during flight can, in theory, be modeled. Attitude, velocities, accelerations and airspeed are estimated using standard UAV avionics, and can be used in the identification. The main challenges with identification from flight tests are disturbances and noise. As opposed to wind tunnel experiments, flight testing is not performed in a controlled environment, and will be affected by external disturbances such as wind. Wind is particularly problematic for slow flying UAVs, since the wind velocity easily becomes large compared to the airspeed of the UAV. The uncertainty in the estimation of the state of the UAV will affect the accuracy of the system identification. A further challenge is to achieve sufficient excitation of the UAV. Finally, since many aerodynamic effects are dependent on the angle-of-attack or the side-slip-angle, they should be measured, if seeking a model of the aerodynamic coefficients in the wind frame. Unfortunately, sensors that are both capable of measuring these quantities, and are small and light-weight enough to fit in a UAV payload, are expensive.

In recent decades, numerical models have been developed to aid aircraft design and wind tunnel testing. Numerical design can reduce the number of tests needed to run in a wind tunnel, and its associated cost. There are two main types of numerical models currently used. The first type, computational fluid dynamics (CFD), is simulating the entire flow field around the aircraft, in either 2D or 3D, by solving the governing Navier-Stokes equations. CFD simulations are computationally expensive, and require experience, careful tuning, and time to set up [19]. In contrast, the so called panel methods use semi-empirical equations to predict the velocity distribution along the surface of an airfoil, without resolving the flow field around it. These tools have low computational requirements and are easier to set up. They are by design limited to 2D calculations, although some model extensions also attempt to simulate 3D conditions with comparatively less accurate results. Examples of programs based on these methods are XFOIL [41], Surfaces Aircraft Design [47] and XFLR5 [37]. The latter is based on XFOIL and extends the model for 3D applications using lifiting line theory (LLT) and vortex lattice method (VLM), and is generally most accurate for thin lifting surfaces at small angles of attack. See also [119] for an extensive list of available software for aerodynamics and aircraft design, and [90] for a good introduction to panel methods.

However, all numerical models are approximations, and often the solution must be calibrated with the use of physical data from a wind tunnel. Thus it is reasonable, and important, to determine how realistic the data from the numerical software is, by relating it to experimental data [101].

In this chapter, both the Vortex Lattice Method (VLM) of XFLR5 and wind tunnel data are used to create an aerodynamic model of a fixed-wing UAV. The stability coefficients are not measured in the wind tunnel, due to time restrictions and equipment limitations, and are therefore solely based on the XFLR5 analysis. For this reason it is important to assess the quality of the XFLR5 data, by comparing the static coefficients of XFLR5 with the wind tunnel data.

There exists several aircraft models that are available to the general public. Of the more notable are the NASA General Transportation Model (GTM) [32], NASAs large envelope F-16 model[138, 159], and the SIG Rascal 110 model [83] which is the standard fixed-wing model for ArduPlane Software-in-the-loop simulations with JSBSim[11]. Reference [151] includes a model of the Zagi flying wing UAV, based on aerodynamic modeling in DARCorp Advanced Aircraft Analysis program[35], verified by wind tunnel testing. The Zagi model is also given in [14], along with a model of an Aerosonde UAV.

The fixed-wing UAV considered in this chapter is the Skywalker X8, produced by Skywalker Technology Co. Ltd. Its flying wing configuration gives a large payload area compared to the size of the UAV, and makes disassembly for transportation easier compared to conventional, tailed fixed-wing UAVs. These characteristics, in combination with its durability, low price point and availability, has lead the X8 to become very popular within the first person view (FPV) community, which was its original design purpose, and later also within various research facilities, which is the primary reason for the interest in an aerodynamic model of the X8. Some examples of its use is autonomous net landing[172], design of a model based longitudinal controller based on system identification[173] , platform for different camera-based experiments [160, 46, 44, 49, 188]. Reference [54] use the X8 in experiments with vision based roll angle determination, [125] installed fuel cells, while [178] use it in experiments on communication-aware path planning.

Validating CFD results with wind tunnel measurements is nothing new. Thus the focus of this chapter is the lessons learned in the process, and the presentation of the X8 aerodynamic model. The chapter is organized into 6 sections. Section 4.2 presents the structure of the aerodynamic model that lays the foundation for the identification process. The wind tunnel experiments, with its challenges, the design and the subsequent parameter estimation, is introduced in section 4.3, while the numerical modeling is presented in section 4.4. Sections 4.5 to 4.6 states and discusses the results from the wind tunnel and numerical calculations, upon which the concluding remarks of section 4.7 are drawn.

4.2 Aerodynamic model structure

Through Newtons laws of motion, the general six degrees-of-freedom equations of motion in eq. (4.1) relate the force F and moment M to linear and angular motion

of a rigid body of mass m and with an inertia matrix $\boldsymbol{I}$.

$$\begin{aligned} m\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times m\boldsymbol{V} &= \boldsymbol{F} \\ \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{V} &= \boldsymbol{M} \end{aligned} \tag{4.1}$$

Here, $\boldsymbol{V} = [u, v, w]$ are the velocities along the body x, y, z-axes, while $\boldsymbol{\omega} = [p, q, r]$ are the angular rates about the same axes. For a UAV, the force is often split into gravity, $\boldsymbol{F}_g$, thrust, $\boldsymbol{F}_t$ and aerodynamic forces, $\boldsymbol{F}_a$, as seen in eq. (4.2). The same is true for the moment $\boldsymbol{M}$, but it is assumed to act around the center of gravity, thus being independent of gravity.

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{F}_a + \boldsymbol{F}_g + \boldsymbol{F}_t \\ \boldsymbol{M} &= \boldsymbol{M}_a + \boldsymbol{M}_t \end{aligned} \tag{4.2}$$

Propulsion forces and moments are considered in chapter 5.

The aerodynamic forces are modeled in the wind frame, and are rotated into the body frame, see fig. 2.2.

$$\boldsymbol{F}_a = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \mathbf{R}_{bw}(\alpha, \beta) \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix}, \tag{4.3}$$

where, D, Y and L are the aerodynamic drag-, side- and lift forces, acting along the negative x, positive y and negative z axes of the wind frame[14], and where

$$\mathbf{R}_{bw}(\alpha, \beta) = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ -\sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta)\sin(\alpha) & \sin(\alpha)\sin(\beta) & \cos(\alpha) \end{bmatrix}, \tag{4.4}$$

is the rotation matrix from wind- to body frame.

A common, further simplification of eq. (2.22) for flight with small angles-of-attack is to assume that lift- and pitch moment coefficients are linear in α, q and δ_e, and independent of β, p, r, δ_r and δ_a. The roll- and yaw coefficients, on the other hand, are independent of α, q and δ_e, and linear in β, p, r, δ_r and δ_a. However, due to the drag force's quadratic nature in α, a linear model in drag is often deemed unfit [98].

The above decoupling between the lateral and longitudinal axes is questionable, but very common[179]. One coupling term for flight at low angles-of-attack is the drag force's dependence on side-slip-angle. Due to symmetry about the xz-plane, drag is an even function in the side-slip-angle.

In summary, the force and moment coefficients are expressed as in eq. (4.5). Metric units are used in the aerodynamic model. Specifically, angles, including control surface deflections, and angular rates are given in radians and radians per second. However, degrees are used for plotting.

$$\begin{bmatrix} C_D \\ C_Y \\ C_L \end{bmatrix} = \begin{bmatrix} C_{D_0} + C_{D_{\alpha 1}}\alpha + C_{D_{\alpha 2}}\alpha^2 + C_{D_{\delta_e}}\delta_e^2 + C_{D_q}\frac{c}{2V_a}q + C_{D_{\beta 2}}\beta^2 + C_{D_\beta}\beta \\ C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\ C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e \end{bmatrix}$$
$$\begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} = \begin{bmatrix} (C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r) \\ C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e \\ (C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r) \end{bmatrix} \tag{4.5}$$

4.3 Wind tunnel testing of the Skywalker X8

The accuracy of the data from a wind tunnel depends on the accuracy of the mass balance. Since this accuracy usually is given as a percentage of the nominal load for the mass balance, the nominal load should reflect the expected loads in the experiment.

An inherent problem with wind tunnel testing of small UAVs is that they operate at lower Reynolds numbers (Re)[1] than larger aircraft, for which most wind tunnels are designed. This means that the forces exerted by UAVs on the mass balance are very small. To maximize the signal-to-noise ratio of the measured forces in the experiment, there are two obvious solutions, each with its own weakness. The first option is to decrease the noise by utilizing a mass balance designed for measuring small forces. Such mass balances generally have a smaller range and are more expensive. The other solution is to maximize the forces exerted by the UAV on the mass balance, such that the mass balance accuracy becomes negligible in comparison. This translates into performing the tests at higher Reynolds numbers, since there is a direct relationship between force and Reynolds number. The Reynolds number can be increased by using a larger model of the UAV, by decreasing the viscosity, or by increasing the airspeed, the latter being the most practical. A drawback with this approach is the larger structural load it will put on the UAV. In addition to potentially damaging the UAV structurally, the deflection of the control surfaces will change if the servos can not hold the larger moment generated by the increased airspeed. Also, the Reynolds numbers should be representable for actual flight conditions.

The problem with low accuracy of the mass balance is particularly pronounced in drag, as these forces are typically one order of magnitude lower than the lift forces. This well known difficulty has been addressed in e.g. [174] and [76]. One way to increase accuracy is to use a wind rake to measure the airspeed reduction downstream of the wind, determine the loss of energy, and thereby the drag. While being accurate, this method suffers from the curse of dimensionality as it requires spatial sampling of the airspeed, making it less feasible for streams in three dimensions.

[1] The nondimensional Reynolds number is defined as $\mathrm{Re} = \frac{V_a L}{\nu}$, where L is the characteristic length of the UAV, i.e. the mean aerodynamic chord, while ν is the kinematic viscosity of air.

4.3.1 Test setup

The wind tunnel testing presented in this section were conducted at the Czech Aerospace Research Center (VZLU) in Prague, Czech Republic. In order to mount the UAV into the wind tunnel, a hole was made in the rear of the fuselage so that the mass balance could be firmly attached to the inside. This internally mounted multi-component strain-gauge mass balance is used to measure the forces and moments acting on the UAV. Table 4.1 gives its nominal forces/moments and accuracy, along with basic specifications for the wind tunnel[2]. The mass balance is further attached to a vertical rod, as seen in fig. 4.1. This rod can be rotated longitudinally and laterally, to change the angle-of-attack and side-slip-angle respectively. Before each test, the mass balance was calibrated for the current location of the center of gravity in the UAV, so that contributions from gravity on the measured moments can be corrected for. This was done by sweeping through the range of angles-of-attack or side-slip-angles while the air is not flowing, to calculate the moment contributions from gravity at each angle.

Table 4.1: Wind tunnel specifications

Wind tunnel type	Closed
Test section	Open
Test section diameter	3 m
Test section length	3 m
Maximum velocity	90 m/s
Max. deviation between local and mean velocity	<0.5 %
Characteristic turbulence intensity	0.3
Nominal X Force	825 N ±0.5%
Nominal Y Force	825 N ±0.5%
Nominal Z Force	2500 N ±0.5%
Nominal Mx Moment	150 Nm ±0.5%
Nominal My Moment	175 Nm ±0.5%
Nominal Mz Moment	150 Nm ±0.5%

To be able to adjust the control surfaces during the experiments, the UAV was equipped with a reduced version of its avionics [204] consisting of two Hitec HS-5085MG digital servos, a BeagleBone Black single board computer, and a small WiFi router for communication with the control room. As servos rely on a PWM command, not a deflection angle, a program was made to translate the commanded angle to a PWM signal, based on curve fitting of empirical data.

A challenge with the servos is that the angular output of the RC servos typically found in small UAVs are not consistent when approaching the same angle from different sides; commanding an angle of 0° yields a slightly different result when commanded from e.g. 10° than when commanded from e.g. −10°, due to the hysteresis of the

[2]For more information about the wind tunnel in question, see also `http://www.vzlu.cz/en/low-speed-wind-tunnels-c73.html`.

servo motor and gear. This is mitigated by always commanding the servo to go to the maximum PWM value before a new angle is commanded.

Figure 4.1: The Skywalker X8 mounted in the wind tunnel

4.3.2 Design of experiments

In addition to mitigating the challenges mentioned in the start of section 4.3, the experiments were designed to be as efficient as possible. The experiments focused on making a good model for the flight envelope that is used the most. This translates into focusing on the quality of the model around $\alpha = \beta = \delta_a = \delta_e = \delta_r = 0$ and Re corresponding to cruise speed, and spending time on verifying these data rather than exploring new regimes of the flight envelope.

The Reynolds numbers for the X8 during cruise speed of 18 m/s vary in the range 400 000 to 500 000 when the air temperature changes from −10 °C to 30 °C, which are both within the operational limits of the X8. Therefore, both these Reynolds numbers where included in the experiments.

In order to simplify the testing process, and drastically reduce the number of tests that need to be performed, a common assumption is that the longitudinal and lateral dynamics are decoupled. The following sections go into detail on the tests for the different axes.

4.3.2.1 Longitudinal dynamics

The longitudinal tests focused on expressing the lift, drag and pitch moment as functions of angle-of-attack and elevator deflection, and assumed linearity in the angle-of-attack up to the stall of the wing. The tests are performed by sweeping through the different angles-of-attack for different Reynolds numbers and elevator deflections, while keeping the aileron and side-slip-angle constant at zero, see table 4.2.

Table 4.2: Longitudinal axes tests for the X8

Test #	α[°]	Re	δ_e [°]
1	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	0
2	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	-20
3	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	-10
4	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	-5
5	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	20
6	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	10
7	$[-5, 15]$	$\{0.4, 0.5\} \times 10^6$	5

4.3.2.2 Lateral dynamics

The lateral tests focused on expressing the side force, roll moment and yaw moment as functions of side-slip-angle, aileron deflection, and assumed linearity in side-slip-angle up to the stall of the winglets. The tests were performed by sweeping through the different side-slip-angles for different Reynolds numbers and control surface deflections, while keeping the angle-of-attack constant at zero. In the interest of time, symmetry around the body x-axis was assumed. The lateral tests performed on the X8 are listed in table 4.3.

Table 4.3: Lateral axes tests for the X8

Test #	β[°]	Re	δ_a [°]
1	$[-15, 15]$	$\{0.25, 0.5\} \times 10^6$	0
2	$[-15, 15]$	$\{0.25, 0.5\} \times 10^6$	20
3	$[-15, 15]$	$\{0.25, 0.5\} \times 10^6$	10

4.3.2.3 Coupled dynamics

The questionable assumption that longitudinal and lateral dynamics are decoupled can be necessary in the interest of time efficiency. If using a 6 DOF mass balance, the drag component from varying side-slip-angle is already found from the lateral tests, so the inclusion of this coupling term does not require any more time. Other

important couplings between the lateral and longitudinal dynamics can vary from airframe to airframe, and a decision on what coupled dynamics should be included in the model relies on the UAV design and experience from flight tests or simulations. Other coupling terms, like the angle-of-attack dependency of the roll damping coefficient C_{l_p}, are not considered here since they are small and complicate the estimation procedure.

4.3.3 Parameter estimation

Once the data from the wind tunnel testing were recorded, they were fit to the model given by eq. (4.5), through reformulation into a linear regression model of the form $y_i = \boldsymbol{\varphi}_i^\intercal \boldsymbol{\Theta} + \epsilon_i$. Here, the measured variable y_i corresponds to the i'th quasi-measurement[3] of an aerodynamic coefficient $C_*(\alpha, \beta, p, q, r, \delta_a, \delta_e, \delta_r)$, $\boldsymbol{\varphi}_i$ is a vector of explanatory variables, $\boldsymbol{\Theta}$ is a parameter vector of the coefficient to be determined, and ϵ_i is an error term that accounts for the measurement noise as well as any unmodeled effects.

The goal of the parameter estimation is to find the value of the coefficients $\boldsymbol{\Theta}$ that explain the measurements y_i, given the explanatory variables $\boldsymbol{\varphi}_i$, in a way that minimize the error ϵ_i. This can be achieved by minimizing the squared error between the measurement y_i and the estimate $\hat{y}_i = \boldsymbol{\varphi}_i^\intercal \boldsymbol{\Theta}$, known as the least squares problem:

$$\begin{aligned} \hat{\boldsymbol{\Theta}} &= \arg\min_{\boldsymbol{\Theta} \in \mathcal{R}^d} V(\boldsymbol{\varphi}, \boldsymbol{\Theta}) \\ &= \arg\min_{\boldsymbol{\Theta} \in \mathcal{R}^d} \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \boldsymbol{\varphi}_i^\intercal \boldsymbol{\Theta} \right)^2, \end{aligned} \tag{4.6}$$

where d is the dimension of the parameter vector, while N is the number of measurements. Table 4.4 lists the explanatory variables and coefficients for the model eq. (4.5). For the tests with the X8, the angular rates p, q, and r where zero, due to the limitations of the wind tunnel facility.

There are many ways to find a solution to the least squares parameter estimation problem eq. (4.6). See e.g. [98] for an in-depth explanation on parameter estimation.

The parameter estimation procedure is greatly simplified by using the MATLAB curve fitting toolbox[4].

[3] The aerodynamic coefficients are considered *quasi*-measurements since they are not directly measured, but derived from the measurements of the forces and moments acting on the mass balance in the body frame.

[4] https://www.mathworks.com/products/curvefitting.html

Table 4.4: Explanatory variables and parameter vector for the different measurements

y	φ^T	Θ^T
C_D	$[1, \alpha, \alpha^2, \frac{c}{2V_a}q, \beta^2, \beta, \delta_e^2]$	$[C_{D_0}, C_{D_{\alpha 1}}, C_{D_\alpha 2}, C_{D_q}, C_{D_{\beta 2}}, C_{D_\beta}, C_{D_{\delta_e}}]$
C_Y	$[1, \beta, \frac{b}{2V_a}p, \frac{b}{2V_a}r, \delta_a, \delta_r]$	$[C_{Y_0}, C_{Y_\beta}, C_{Y_p}, C_{Y_r}, C_{Y_{\delta_a}}, C_{Y_{\delta_r}}]$
C_L	$[1, \alpha, \frac{c}{2V_a}q, \delta_e]$	$[C_{L_0}, C_{L_\alpha}, C_{L_q}, C_{L_{\delta_e}}]$
C_l	$[1, \beta, \frac{b}{2V_a}p, \frac{b}{2V_a}r, \delta_a, \delta_r]$	$[C_{l_0}, C_{l_\beta}, C_{l_p}, C_{l_r}, C_{l_{\delta_a}}, C_{l_{\delta_r}}]$
C_m	$[1, \alpha, \frac{c}{2V_a}q, \delta_e]$	$[C_{m_0}, C_{m_\alpha}, C_{m_q}, C_{m_{\delta_e}}]$
C_n	$[1, \beta, \frac{b}{2V_a}p, \frac{b}{2V_a}r, \delta_a, \delta_r]$	$[C_{n_0}, C_{n_\beta}, C_{n_p}, C_{n_r}, C_{n_{\delta_a}}, C_{n_{\delta_r}}]$

4.4 Numerical modeling of the Skywalker X8

Numerical modeling was chosen to be performed with XFLR5 as it is an easy and fast tool to use. The code performs reasonably well for viscid 2D and with limitations also in 3D[36]. Generally, the biggest limitations are on the 3D methods, as the best results are obtained for thin surfaces at low angles of attack. The deviations increase as the aircraft geometry differs from an idealized 2D infinite wing. Also, flows with large rotational components (especially turbulence) cannot be captured accurately. However, XFLR5 can be used to support wind tunnel testing data and is here used to determine the stability coefficients which were not obtained experimentally.

To model a fixed-wing aircraft in XFLR5, knowledge of its geometry is required, in particular the shape of the airfoils that are used. Unfortunately, the airfoil of the X8 is unknown, even after comparing it to commonly used flying wing airfoils. However, the X8 geometry was found from an openly available 3D-scan of the Skywalker X8[5]. The 3D model is in a raw format meaning that the surface of the aircraft is uneven and shows erroneous artifacts. When extracting 2D cross sections to be used in the XFLR5 code, the 2D airfoils will be edgy and containing an uneven curvature which leads to unrealistic predictions of the pressure and velocity distribution. In order to minimize this effect, the 2D cross sections have been smoothed manually using XFLR5. First, the geometry is approximated with a spline. Second, the inverse foil design tool is used on the spline to generate a smooth pressure distribution on the surface. The resulting airfoil has good confidence that the mean chamber line have been captured correctly, which is affecting lift. The airfoil curvature however, is not well captured as the input data was too inaccurate. Airfoil curvature affects the resulting pressure gradients which are driving the boundary layer transition processes. These are responsible for the generation of drag and pitch forces. Hence, it is expected that the XFLR5 data will perform well for lift and poor for drag and pitch. The final step of the modeling in XFLR5 is to assemble the airfoils into a wing, and possibly add a fuselage. The X8 wing and fuselage was modeled as a blended

[5]The 3D-scan of the X8 can be found at `https://diydrones.com/profiles/blogs/x8-3d-scanning`

wing, with no fuselage, since there is no clear distinction between the two.

XFLR5 can perform two types of analysis: a static analysis and a stability analysis[6]. The static analysis sweeps through a range of angle-of-attack or side-slip-angles for a range of Reynolds numbers, to generate polars that show how the forces and moments vary with angle-of-attack or side-slip-angle. The stability analysis calculates the damping terms and control input coefficients by approximating the partial derivatives of the aerodynamic forces and moments. The forward differentiation approximation is acheived by slightly perturbing the UAV by a slight change in attitude, velocity or control surface deflection, and noting how much the forces and moments change. A finite difference approximation is found by dividing this change in force/moment by the perturbation.

The data from the different XFLR5 analyses were then fit to the model eq. (2.22), in a similar manner as the wind tunnel data.

4.5 Results

From the geometry of the 3D model of the Skywalker X8, the wingspan and mean aerodynamic chord were found to be 2.1 m and 35.7 cm, respectively. This was used in the subsequent parameter identification, whose resulting aerodynamic coefficients are shown in table 4.5, along with the root mean square error (RMSE) and coefficient of determination (R^2) corresponding to the fitting of the wind tunnel data. The table also contains the data from the numerical analysis, both for comparison to the wind tunnel (WT) data and for complementing the wind tunnel data with stability coefficients. Figures 4.2a to 4.2f and fig. 4.3 visualize the fit of the model along with the measured data from the wind tunnel, for all the six axes. The differences in the coefficient, for the same control surface deflection angle and angle-of-attack/side-slip-angle, comes from running the tests at different Reynolds numbers. As seen in figs. 4.2a, 4.2c and 4.2e, the longitudinal fitting exclude data points with angle-of-attack of more than 12° and less than 0° to avoid the stall region.

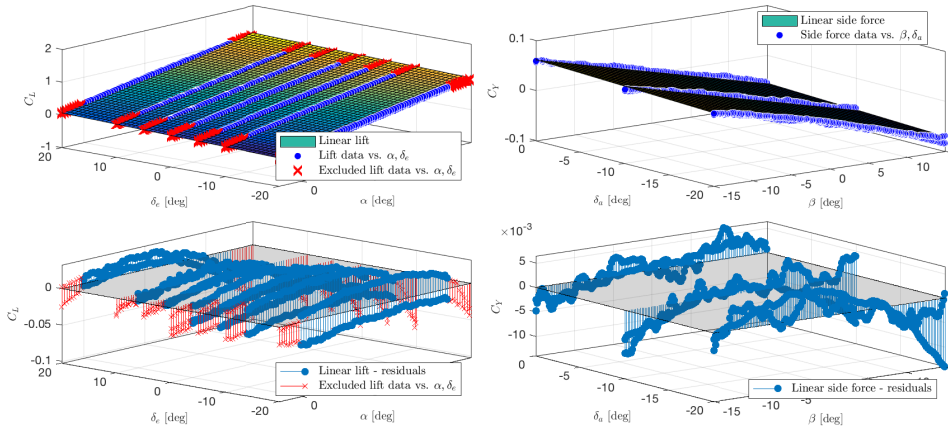
The R^2-column shows that most of the coefficients from the wind tunnel experiment fit well with the data. The exception is the lateral drag, C_{β_2} and C_{β_1}, which likely comes from the large relative difference between the data for different Re, compared to the magnitude of the coefficient, as shown in fig. 4.3. The fit did not improve for higher order polynomials.

Unfortunately, neither XFLR5 nor the wind tunnel testing is able to estimate the increase in drag from pitch rate, C_{D_q}.

[6]The static and stability analyses are referred to as *Analysis* and *Stability Analysis* in XFLR5

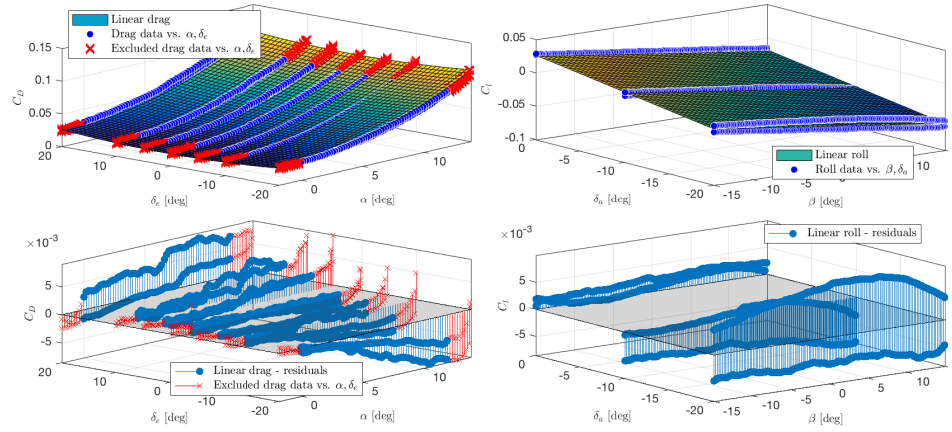
Table 4.5: Skywalker X8 aerodynamic coefficients

	WT	XFLR5	RMSE	R^2
C_{L_0}	0.0867	0.0477	0.0153	0.996
C_{L_α}	4.02	4.06	0.0153	0.996
$C_{L_{\delta_e}}$	0.278	0.7	0.0153	0.996
C_{L_q}	—	3.87	—	—
C_{D_0}	0.0197	0.0107	0.00262	0.982
$C_{D_\alpha}1$	0.0791	−0.00955	0.00262	0.982
$C_{D_\alpha}2$	1.06	1.1	0.00262	0.982
$C_{D_{\delta_e}}$	0.0633	0.0196	0.00262	0.982
$C_{D_{\beta_2}}$	0.148	0.115	0.00234	0.734
$C_{D_{\beta_1}}$	−0.00584	$-2.34E-19$	0.00234	0.734
C_{m_0}	0.0302	0.00439	0.00576	0.983
C_{m_α}	−0.126	−0.227	0.00576	0.983
$C_{m_{\delta_e}}$	−0.206	−0.325	0.00576	0.983
C_{m_q}	—	−1.3	—	—
C_{Y_0}	0.00316	$1.08E-08$	0.00326	0.991
C_{Y_β}	−0.224	−0.194	0.00326	0.991
$C_{Y_{\delta_a}}$	0.0433	0.0439	0.00326	0.991
C_{Y_p}	—	−0.137	—	—
C_{Y_r}	—	0.0839	—	—
C_{l_0}	0.00413	$1.29E-07$	0.00476	0.953
C_{l_β}	−0.0849	−0.0751	0.00476	0.953
$C_{l_{\delta_a}}$	0.12	0.202	0.00476	0.953
C_{l_p}	—	−0.404	—	—
C_{l_r}	—	0.0555	—	—
C_{n_0}	−0.000471	$1.51E-07$	0.000615	0.98
C_{n_β}	0.0283	0.0312	0.000615	0.98
$C_{n_{\delta_a}}$	−0.00339	−0.00628	0.000615	0.98
C_{n_p}	—	0.00437	—	—
C_{n_r}	—	−0.012	—	—



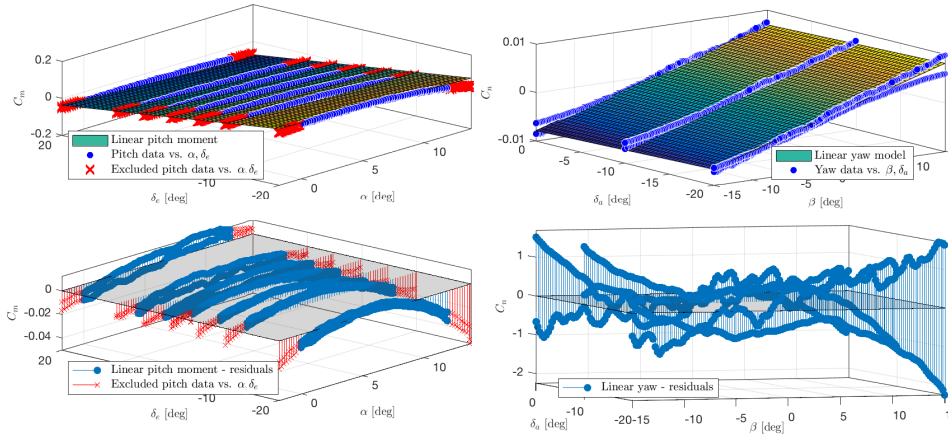
(a) Lift force coefficient fit

(b) Side force coefficient fit



(c) Drag force coefficient fit

(d) Roll moment coefficient fit



(e) Pitch moment coefficient fit

(f) Yaw moment coefficient fit

Figure 4.2: Curve fit

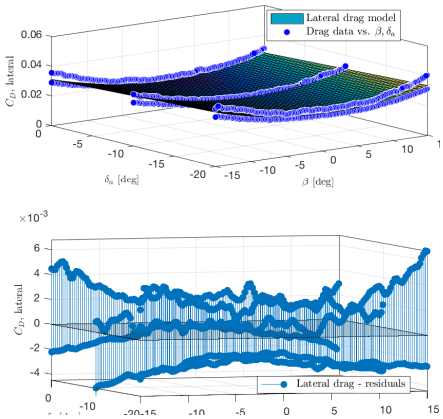


Figure 4.3: Lateral drag

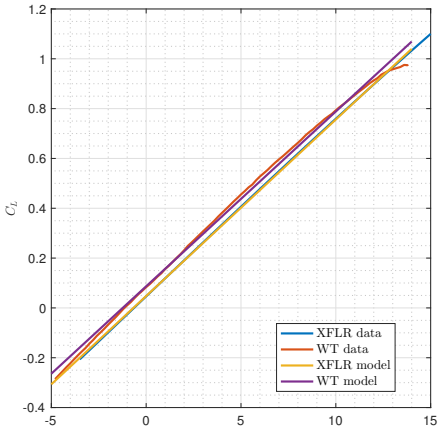


Figure 4.4: Lift coefficient vs. angle-of-attack

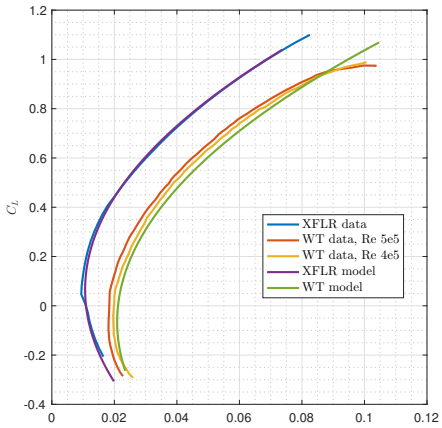


Figure 4.5: Lift coefficient vs. drag coefficient

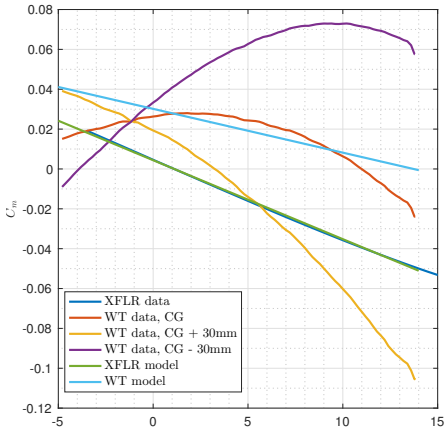


Figure 4.6: Pitching moment coefficient vs. angle-of-attack

4.6 Discussion

The coefficients from XFLR5 and from the wind tunnel testing match reasonably well, with some exceptions.

Figure 4.4 compares the angle of attack against the lift coefficient for all the models used. As expected, model and experiment show a good fit of the lift gradient. An constant offset of approximately $\alpha = 0.5°$ of the XFLR5 and wind tunnel data is observable which might be related to an offset in angle of attack during the wind tunnel testing. This is reasonable as the definition of the zero angle in the wind tunnel was difficult to achieve.

Figure 4.5 displays the aircraft performance; lift versus drag. The drag values show significant deviation from the wind tunnel results. This was expected and can be attributed to the deviations from the real airfoil geometry as well as the code limitations of XFLR5. The numeric method under-predicts the drag by approximately 50% in the linear region, which is probably caused by the difficulty in predicting viscous effects. As the drag forces are not captured accurately by XFLR5 this means that the pitching moment will also be subject to the same error.

Furthermore, fig. 4.5 shows that the main difference in drag is a constant offset. This could come from the fact that the theoretical background of the methods in XFLR5 are based on inviscid flow, and use an ad-hoc method for viscous calculations. Another error source for the drag from XFLR5 could be the trailing edge of the fuselage. It has a blunt edge, which is an inherent problem with panel methods. In addition to the measurement errors, the precision of the wind tunnel data is also compromised by blockage effects in the wind tunnel, which has been accounted for by VZLU, and by its finite width. The wind tunnel is 0.9 m wider than the wing span of the UAV, so wing tips might be affected by the boundary layer outside the airflow in the open section of the wind tunnel. Among the longitudinal control derivatives, $C_{m_{\delta_e}}$ from the wind tunnel and from XFLR5 are in the same orders of magnitude. Consistent with XFLR5s over prediction of the aircraft performance, $C_{L_{\delta_e}}$ from XFLR5 is far larger, while drag, $C_{D_{\delta_e}}$, is smaller.

In the pitch moment, fig. 4.6 reveals larger differences than one would expect from table 4.5. The wind tunnel data is very nonlinear in angle-of-attack, and does not fit well with neither the XFLR5- nor the wind tunnel models. It can also be seen that the wind tunnel data indicate instability for $\alpha < 2°$. Since the X8 is a flying wing, $C_m(\alpha)$ is small, and very sensitive to changes in the center of gravity, so the instability can come from using wrong center of gravity. Currently, the location of the center of gravity is 44.0 cm behind the nose of the UAV, and is based on experience from flights, but as shown in fig. 4.6, a slight shift of the center of gravity 30 mm to the front yields a large stability margin in the pitching moment. It is also shown that the zero moment for the CG-centered curve is at about $\alpha = 11°$, which is unrealistically high. The curve referenced 30 mm in front of the center of gravity yields zero pitching moment at $\alpha = 3.25°$, which is more in line with experience from flight testing. There

are some differences in C_{Y_0}, C_{l_0} and C_{n_0}, but these likely come from misalignment in the wind tunnel testing, as these coefficients should be zero for an airframe with symmetry about the xz-plane.

With these differences, it is difficult to conclude on the confidence in the stability coefficients from XFLR5, and this matter should be investigated further in flight test validation. However, it is clear that XFLR5 easily and quickly gives reasonable data that can be valuable for initial estimates of static aerodynamic coefficients. Estimation of stability coefficients in a wind tunnel require more advanced equipment, in the form of a rotary balance, and considerably more time.

For estimation and control purposes, wind gusts are considered model disturbances. Small UAVs are likely to encounter large wind gusts disturbances, that are difficult to measure. This indicates that regardless of the precision of the aerodynamic model, considerable amounts of noise will degrade the performance of the estimation or control algorithm that utilize the model.

Further, it is apparent from the residual in figs. 4.2a to 4.2f that the averaging of the coefficients over the different Reynolds numbers yields a good fit in the flight regime on average, as intended, but the precision could have improved by only fitting the model to the data from one Reynolds number. This is also true for the fitting to the control surface deflections. Figure 4.5 shows that the model differs from the measurements for both Reynolds numbers, when $\delta_e = 0$. However, the least squares estimation has ensured that the model reduces the estimation error over all Re and δ_e. In order to improve the fit at $\delta_e = 0$, weighted least squares could have been used instead.

Through the work with these tests, it has become apparent that wind tunnels are not ground truth, as there is a lot of measurement errors, especially in drag. This comes from the relative low accuracy of mass balances designed for larger forces than what is typically exerted by the air on small UAVs. Further, the importance of checking the measurements was demonstrated; the experiments presented in this chapter where performed twice, since the data from the first test was found to be corrupted by a faulty axis on the mass balance. Another costly lesson learned is that wings break when exposed to large airspeeds at large side-slip-angles. Considerable amounts of time can be saved by optimizing the test schedule, depending on the test facility. At the VZLU facilities, the angle-of-attack can be adjusted very easily. Adjusting the Reynolds number, i.e. the airspeed, takes more time since the flow has to stabilize. Adjusting the control surfaces is by far the more time consuming, since the center of gravity changes slightly with the deflection. Therefore, the model has to be tared again, by calculating the contributions from gravity at each angle-of-attack or side-slip-angle.

Other topics for future work include replacing XFLR5 with a Navier-Stokes-based CFD method to estimate the stability data with higher accuracy, integrating the presented model into the software-in-the-loop framework for ArduPlane, extending the model into the high angle-of-attack region, and adapting the model to icy conditions. The

model is available for download[7], and will be updated with the results from future work.

4.7 Conclusion

A mathematical model is always going to be an approximation of the physical world. This chapter has presented considerations that are important to make, both in the experiment design and in the analysis of the data, in order to make the model as close a representation as possible, using numerical modeling backed by wind tunnel testing. The case study of the Skywalker X8 fixed-wing UAV indicate a reasonable match between the numerical results and the wind tunnel, particularly in lift, but show some discrepancies that needs further investigation through validation with flight data. Notably, the drag is under-predicted by the numerical modeling, and the sensitivity to center of gravity is reflected in the pitch moment from the wind tunnel testing. Also, the stability coefficients from the numerical modeling needs validation with real flight data, as they were not tested in the wind tunnel. To promote the use of the X8 model presented in this chapter, it has been made publicly available for download.

[7] See `https://github.com/krisgry/x8`

Chapter 5

Propulsion system

In addition to the aerodynamic forces and moments, the propulsion system must be identified to have a complete picture of the forces and moments at play. This chapter briefly summarize the propulsion model identification of the UAVlabs standard setup for a Skywalker X8 from

- [27] E. M. Coates, A. Wenz, K. Gryte, and T. A. Johansen, "Propulsion system modeling for small fixed-wing UAVs", in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 748–757. DOI: 10.1109/ICUAS.2019.8798082

for completeness.

5.1 Propeller model

For estimating aerodynamic forces and moments, based on measured rotation speed ω_m of the motor, the battery voltage U_{dd} and current I_a, and the controlled pulse-width-modulation (PWM) duty-cycle ratio of the throttle $\delta_t = \frac{t_{\text{on}}}{t_{\text{period}}}$[1], the relevant equations are

$$\omega_m = \frac{U_{dd}\delta_t - RI_a}{k_E} \tag{5.1}$$

$$T = \frac{\rho D^4}{4\pi^2}\left(C_{T,0} + C_{T,1}J\right)\omega_m^2 \tag{5.2}$$

$$Q = \frac{\rho D^5}{4\pi^2}\left(C_{Q,0} + C_{Q,1}J\right)\omega_m^2 \tag{5.3}$$

[1] $t_{\text{on}} < t_{\text{period}}$ is the time, within a period t_{period} that the PWM signal is high

where $C_T(J) = C_{T,0} + C_{T,1}J$ and $C_Q(J) = C_{Q,0} + C_{Q,1}J$ are linear approximations of the thrust and torque coefficients, respectively, in terms of the advance ratio $J = \frac{2\pi V_a}{\omega D}$. Furthermore, k_E is the aggregate back-EMF constant for the motor, D is the propeller diameter, and ρ is the air density. If only the estimated thrust and torque needed, for e.g. fault detection, measured rotation speed could be used instead of eq. (5.1). However, for a control application, the complete cascade from input δ_t to resulting thrust/torque is desirable.

5.2 Identification results

From a series of wind tunnel experiments, at 7 different airspeeds in the range 5 m/s to 21 m/s, the unknown coefficients from eqs. (5.1) to (5.3) were identified, using nonlinear least squares, for the components given in table 5.1, which are the default propulsion setup for the Skywalker X8 at the NTNU UAVlab.

Table 5.1: Hardware Overview

Propeller	Aeronaut CamCarbon 14x8" $D = 0.35\,\text{m}$
Motor	Hacker Motor A40-12S V2 14-pin KV610
ESC	Jeti SPIN Pro 66
Battery	Zippy Compact 5000mAh 4S 25C LiPo

Table 5.2: Propulsion model parameters

Coefficient	Value
$C_{T,0}$	0.126
$C_{T,1}$	−0.1378
$C_{Q,0}$	0.0078
$C_{Q,1}$	−0.0058
R	$0.0587\,\Omega$
k_E	$0.0134\,\text{V s}$

Chapter 6

Inertia modeling

In order to translate the aerodynamic and propulsion forces and moments from chapters 4 and 5 into linear and angular acceleration, which again can be integrated to angular and linear velocity and position, the inertial properties of the UAV are needed, see eq. (4.1). The mass can simply be obtained with a scale, while identifying the inertia is more difficult. This chapter describes the process of identifying the products of inertia of a UAV, by hanging the UAV in a bifilar pendulum rig as seen in fig. 6.1, mainly based on [81]. Identification of inertia by pendulums is a well-known technique, also for larger airplanes, [127]. The literature consider many different types of pendulums to identify inertia [167], also by measuring the torque [34]. Another option is to use a 3D computer aided design (CAD) model, which requires an actual representation of the different components in the UAV and their mass distribution.

Figure 6.1: The Skywalker X8 mounted to oscillate around its x-axis as a bifilar pendulum

6.1 Equations of motion for a bifilar pendulum

This is based on Lagrangian mechanics, and follows [81], but is repeated for completeness.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}_I}\right) - \frac{\partial L}{\partial \theta_I} = Q \tag{6.1}$$

Here, $L = T - V$ is the Lagrangian function, given by the differnece between kinetic and potential energy, while Q is the sum of all non-conservative forces (damping). θ_I is the generalized coordinate, in this case the rotation angle of the UAV about the current axis of interest, with respect to its free-hanging angular position. For fig. 6.1, θ_I corresponds to the roll angle of the UAV.

$$T = \frac{1}{2} I \dot{\theta}_I^2 + \frac{1}{2} m \dot{z}^2 \approx \frac{1}{2} I \dot{\theta}_I^2 \tag{6.2}$$

where the approximation comes from arguing that the rotational energy is far greater than the translational, for typical pendulum dimensions. With $V = mgz$ and $z = h\left(1 - \sqrt{1 - \frac{1}{2}\frac{D}{h}^2(1-\cos\theta_I)}\right)$, the Lagrangian becomes

$$L = \frac{1}{2} I \dot{\theta}_I^2 - mgh\left(1 - \sqrt{1 - \frac{1}{2}\frac{D}{h}^2(1-\cos\theta_I)}\right) \tag{6.3}$$

where h is the pendulum height and D is the distance between the two pendulums. Assuming both aerodynamic and mechanic damping, the non-conservative forces are $Q = -K_d \dot{\theta}_I |\dot{\theta}_I| - C_d \dot{\theta}_I$, where K_d and C_d are nondimensional mechanical and aerodynamic damping coefficient. Then the equations of motion become

$$\ddot{\theta}_I + \left[\frac{K_d}{I}\dot{\theta}_I|\dot{\theta}_I| + \frac{C_d}{I}\dot{\theta}_I\right] + \frac{mgD^2}{4Ih}\frac{\sin\theta_I}{\sqrt{1 - \frac{1}{2}\frac{D}{h}^2(1-\cos\theta_I)}} = 0 \tag{6.4}$$

6.1.1 Added mass

[81] mentions that the generalized force of the added mass (for a flat plate traveling in a direction normal to the surface of the plate) can be expressed as

$$Q_a = \left[\frac{k^{'} \rho \pi c^2 b^3}{48} + \frac{k \rho \pi c^2 b l^2}{4} \right] \ddot{\theta}_I = I_a \ddot{\theta}_I \tag{6.5}$$

such that I in eq. (6.4) is the sum of inertial and added inertia: $I_{\text{measured}} = I + I_a$. The problem is that the scaling factors $k^{'}$ and k needs to be determined empirically for each test object. The measured inertia will also include vortex effects (in addition to the effects of the potential flow (added mass) and the mechanical effects (the "true" inertia)). These vortexes are very difficult to parameterize, but are small. Only the effect of the sum of the kinetic energy $T = T_{\text{mech}} + T_{\text{pot}} + T_{\text{vortex}}$ identify, but from Kelvin's minimum energy theorem [13] we know that they are all positive. We are therefore able to determine an upper bound on T_{mech}.

[104] claim that added mass effects can be as much as 25 % of the inertia. Rough estimates show that the order of magnitude for the added mass is believable [21]:

$$m_a = \frac{\rho \pi c^2}{4} b \approx \frac{1.225 * 3.14 * (0.35)^2}{4} \cdot 2.1 \approx 0.26\,\text{kg} \tag{6.6}$$

which is almost 10 % of the mass of the X8. It makes sense that added mass effects are more pronounced for small/slow UAVs than for manned aircraft: low speed, low wing loading, low mass compared to surface area, which makes the mass less dominant over the potential flow effects.

There are several ways to circumvent this problem:

- Use a spring-based pendulum. As long as a gravitational pendulum is used, it is not possible to separate the gravitational and the added mass effects. But oscilations in a spring only depends on the spring constant, and is independent of gravity.
- Perform each experiment twice, and add a known mass in the second test, to produce an additional inertia. This will yield two equations with two unknowns (the mechanical and the potential/added inertia).
- accept the possible errors

The latter is chosen, to not increase the complexity of the identification, and since the uncertainty of the effects of added mass will also be present during flight.

6.2 Parameter identification problem formulation

The identification problem has been formulated as an optimization problem, using the output error method [98], formulated as a nonlinear problem (NLP) through multiple shooting, in CasADi [8]:

$$\min_{\mathbf{x}} \quad \frac{1}{N}\sum_{k=0}^{N}(y_k - z_k)^\intercal(y_k - z_k) \tag{6.7}$$

Subject to:

$$\boldsymbol{x}_{k+1} - \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{p}, z_k) = 0 \tag{6.8a}$$
$$0 < I < 100 \tag{6.8b}$$
$$-\pi < \theta_{I,0} < \pi \tag{6.8c}$$
$$-\pi < \dot{\theta}_{I,0} < \pi \tag{6.8d}$$
$$0 < K_d < \infty \tag{6.8e}$$
$$0 < C_d < \infty \tag{6.8f}$$
$$-\pi < \theta_{I,\text{bias}} < \pi \tag{6.8g}$$

where $\boldsymbol{x}_k = \left[\theta_{I,k}, \dot{\theta}_{I_k}\right]^\intercal$, $y_k = \theta_{I,k}$, and $F(\boldsymbol{x}_k, \boldsymbol{p}, z_k)$ is the model eq. (6.4), discretized through Runge-Kutta's method of fourth order. z_k is the measured angle of the pendulim, while the parameter vector is given by $\boldsymbol{p} = \left[I, \theta_{I,0}, \dot{\theta}_{I_0}, K_d, C_d, \theta_{I,\text{bias}}\right]$.

6.3 Experimental setup

The above pendulum test can only be applied to a single axis at the time, so it has to be repeated four times to find the moments of inertia components I_{xx}, I_{yy} and I_{zz}, and the product of inertia component $I_{zx} = I_{xz}$ to produce the inertia matrix

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix} \tag{6.9}$$

To obtain I_{xz}, [98, App C] recommends to find the moment of inertia about an intermediate axis in the body xz-plane at an angle ϵ to the body x-axis, and translate it according to

$$I_{xz} = \frac{I_\epsilon - I_x \cos^2\epsilon - I_z \sin^2\epsilon}{\sin(2\epsilon)} \tag{6.10}$$

The UAV is hung in a motion capture lab, and equipped with markers, to identify the rotation angles directly, to avoid having to consider potential drift in inertial sensors, which is used in [81]. Before each experiment, the pendulum height, distance between the pendulums and the mass of the UAV and the rig were recorded. Each of the four experiments are repeated five times, to overcome measurement errors.

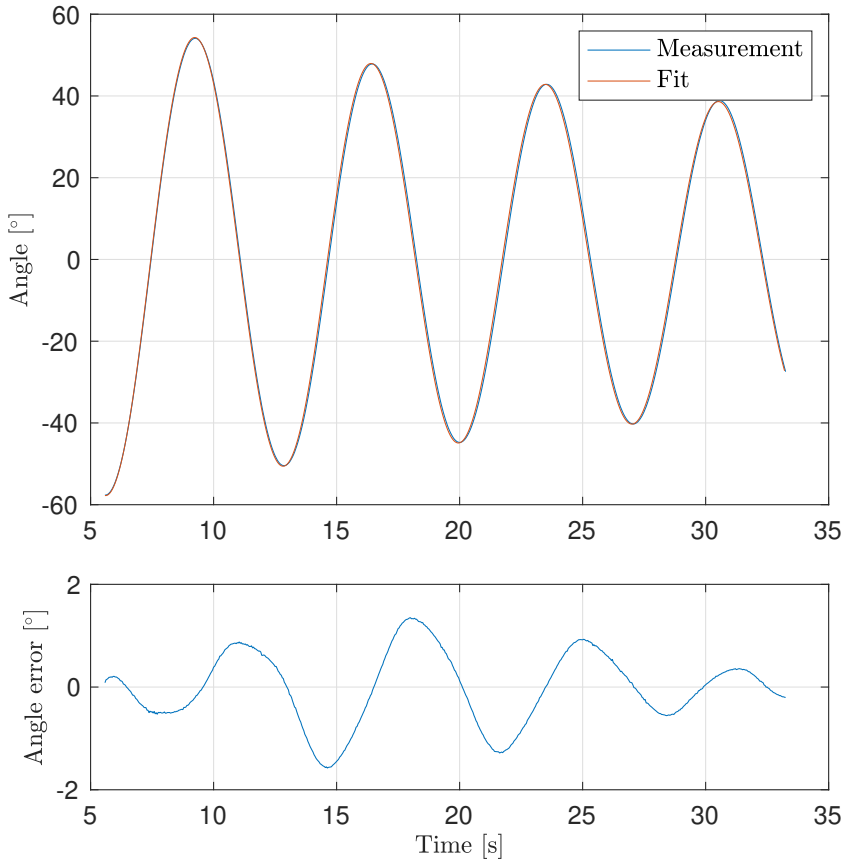


Figure 6.2: The measured angle plotted against the angle from the NLP

6.4 Results

The identified components of the inertia matrix for the Skywalker X8 are given in table 6.1. To illustrate the performance of the NLP, and to justify the results, the identified angle from the NLP is plotted against the measured angle from the motion capture system in fig. 6.2 for one recorded dataset, while fig. 6.3 shows how the inertia and the other parameters of the NLP change over the iterations, for the same dataset.

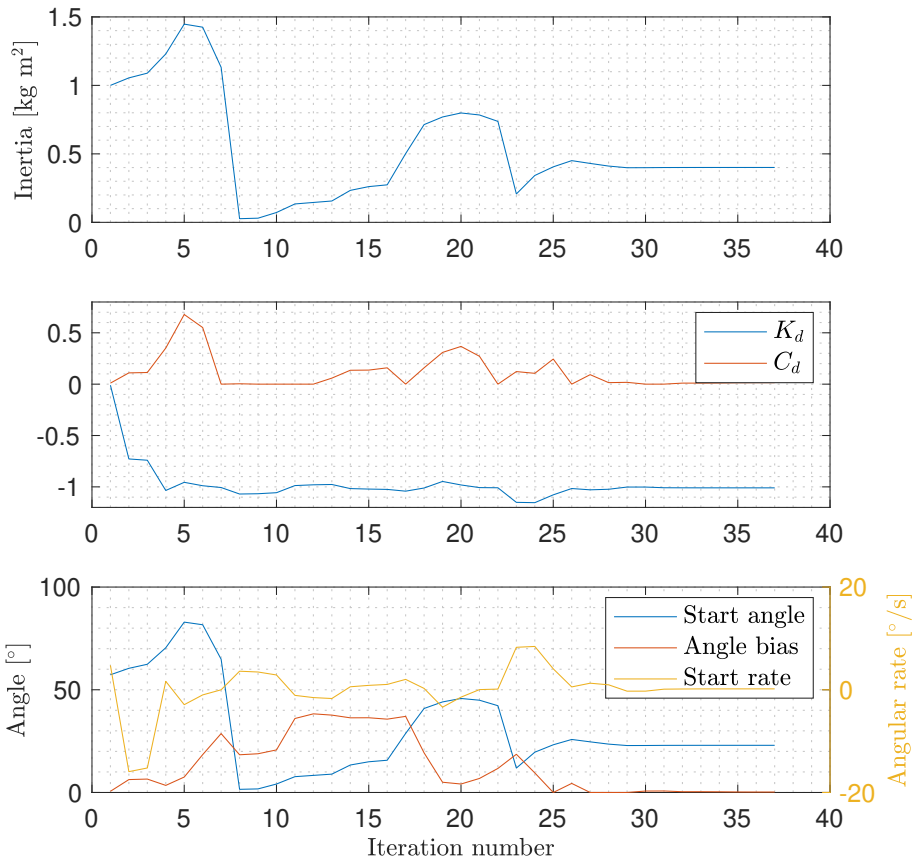


Figure 6.3: Convergence of the parameters in the NLP

Table 6.1: Inertia for the Skywalker X8

Parameter	Value kg/m^2
I_{xx}	0.335
I_{yy}	0.140
I_{zz}	0.40
I_{ϵ}	0.337
$I_{zx} = I_{xz}$	-0.029

Chapter 7

The coordinated turn

A *coordinated* turn is a turn where the aircraft flies with no sideslip, and with no acceleration along the body y-axes. This makes it a desirable way of turning, both in terms of passenger comfort, for commercial aircraft, and in terms of performance, as drag depends on sideslip. The literature contains many different derivations of the coordinated turn relation. Many of them, like [180, 155, 148], only consider yaw rate, while [14, 161, 162] consider course rate. Although their resulting equations give reasonable results, it is difficult to understand the validity, as few assumptions are stated. This chapter seeks to derive the coordinated turn relation expressed in course angle, while explicitly stating all assumptions and simplifications that are made. In the derivation, which considers how the different forces must be balanced out to fly along a circular/spiral path, the following assumptions are made

- No dynamics of the lift force are considered. It is simply assumed to be a force along the negative z-axis in the wind frame that balances the force of gravity, in the inertial frame. In reality, the lift force is a function of a number of quantities, most notably angle of attack, sideslip angle, airspeed and control surface deflection, which may vary.
- The lift force dominates the drag, which therefore is neglected
- All the linear velocity is directed tangential to the circle/spiral, such that $a = \frac{v^2}{R} = v\omega$ holds for the centripetal acceleration.
- The air-relative path angle, which is the vertical direction of the airspeed vector $\boldsymbol{V}_a$, can be approximated as $\gamma_a = \theta - \alpha$. This is a common assumption, but comes from assuming wings-level flight, which is not the case [179, eq 2.2-38]
- The air-relative course angle, which is the horizontal direction of the airspeed vector $\boldsymbol{V}_a$, can be approximated as $\chi_a = \psi + \beta$. This is a common assumption, but comes from assuming horizontal airspeed, which is not necessarily the case [179, eq 2.2-39]
- The angle of attack α and the sideslip angle β, are assumed to be zero. This is not the case, but they vary throughout the flight, and can be difficult to measure or estimate accurately for small UAVs [194]. If sideslip is measured

and is controllable for the given airframe, it is controlled to zero to reduce drag and side forces, to achieve turn coordination.

7.1 Derivation

In the tangential frame, the lift force has to counteract the gravitational force and the centripetal acceleration, implying the force balance $\boldsymbol{F}_L^t = m\boldsymbol{a}^t$, which can be extended into

$$\boldsymbol{F}_L^t = \boldsymbol{R}_{tw}\begin{bmatrix}0\\0\\-f_L\end{bmatrix} = m\vec{a} = \begin{bmatrix}0\\m\frac{v_x^{t\,2}}{R}\\mg\end{bmatrix} = \begin{bmatrix}0\\mV_g\cos(\gamma)\dot{\chi}\\mg\end{bmatrix}, \tag{7.1}$$

where $\frac{v_x^{t\,2}}{R} = V_g\cos(\gamma)\dot{\chi}$, since all the velocity is tangential to the circle, and since $V_g\cos\gamma$ is the horizontal component of

$$\boldsymbol{v}^t = \boldsymbol{R}_{tp}\begin{bmatrix}V_g\\0\\0\end{bmatrix} = \boldsymbol{R}_y(\gamma)\begin{bmatrix}V_g\\0\\0\end{bmatrix} = \begin{bmatrix}V_g\cos\gamma\\0\\V_g\sin\gamma\end{bmatrix} \tag{7.2}$$

It is noted that the force along the tangential x-axis does not actually evaluate to zero on the left hand side in eq. (7.1), but this is assumed to be canceled by drag effects.

7.2 Rotation between tangential and wind frame

The rotation matrix $\boldsymbol{R}_{tw}$ in eq. (7.1) rotates a vector from the wind frame to the tangential frame, which is rotated an angle of χ about the z-axis, and is considered inertial. Following the procedure in [179, fig 2.2-7], but also accounting for wind, the process of defining the rotation matrix involves the following series of rotations, which all are about the current frame.

1. $-\mu_a$ about the current x axis
2. $-\gamma_a$ about the current y axis
3. $-\chi_a$ about the current z axis, to arrive in the inertial frame
4. χ about the current z axis

As the two last rotations are about the same axis, they are combined, resulting in the following rotation matrix

$$\boldsymbol{R}_{tw} = \boldsymbol{R}_z(\chi-\chi_a)\boldsymbol{R}_y(-\gamma_a)\boldsymbol{R}_x(\mu_a) \tag{7.3}$$

7.3 Desired bank angle

By division of the y-component by the z-component on both sides of the force balance eq. (7.1), inserting the rotation matrix eq. (7.3), and inserting the assumptions $\gamma_a = \theta - \alpha$ and $\chi_a = \psi + \beta$, the balance reduce to

$$\frac{\cos(\beta - \chi + \psi)\sin(\mu_a) - \sin(\alpha - \theta)\sin(\beta - \chi + \psi)\cos(\mu_a)}{\cos(\mu_a)\cos(\alpha - \theta)} = \frac{V_g \cos(\gamma)\dot{\chi}}{g} \tag{7.4}$$

Further re-arranging, and assuming that $\alpha = \beta = 0$, the two equivalent expressions for the bank and course angles are found

$$\mu_a = \operatorname{atan}\left(\frac{V_g \cos\theta \cos\gamma}{g\cos(\chi - \psi)}\dot{\chi} - \sin(\theta)\tan(\chi - \psi)\right) \tag{7.5}$$

$$\dot{\chi} = \frac{g\cos(\chi - \psi)}{V_g \cos\gamma \cos\theta}(\tan\mu_a + \sin\theta\tan(\chi - \psi)). \tag{7.6}$$

It is noted that given good knowledge of the V_w, V_a and V_g, the angle $\chi - \chi_a$ can be found directly, see fig. 2.4, avoiding the assumptions $\chi_a = \psi + \beta$ and $\beta = 0$. However, the wind speed V_w is inherently stochastic, and difficult to estimate without making similar assumptions.

By comparing eq. (7.6) to [14, eq. 5.15], the resemblance is clear, despite a few apparent differences:

- While it is a small number, $\sin\theta\tan(\chi - \psi)$ is not accounted for by [14]
- $\cos\gamma$ in the denominator is canceled by [14], where the vertical component of the gravitational force is assumed to be $mg\cos\gamma$.
- $\cos\theta$ in the denominator is not included. This is however included in the ardupilot implementation of [144], albeit outside the atan-function.

It should be noted that these differences are small for many flight conditions, rendering [14, eq. 5.15] a reasonable approximation.

7.4 Translation to roll

Seeking to express eq. (7.5) in terms of the roll angle ϕ, a translation from bank angle μ_a must be found. Following a similar procedure as [179, eq 2.2-36b], which does not consider wind, the following relationship is apparent:

$$\boldsymbol{R}_z(\chi_a)\boldsymbol{R}_y(\gamma_a)\boldsymbol{R}_x(\mu_a) = \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi)\boldsymbol{R}_x(\alpha)^\intercal\boldsymbol{R}_z(-\beta)^\intercal \tag{7.7}$$

Equating element (3,3) in the matrices on both sides in the above relation, and assuming that $\gamma_a = \theta - \alpha$ and $\chi_a = \psi + \beta$, results in

$$\mu_a = \operatorname{acos}\left(\frac{\sin(\alpha)\sin(\theta) + \cos(\alpha)\cos(\phi)\cos(\theta)}{\cos(\alpha - \theta)}\right), \tag{7.8}$$

which simplifies to $\mu_a = \phi$ for $\alpha = \beta = 0$. Inserting this into eqs. (7.5) and (7.6) results in eqs. (10.11) and (10.12).

Part II

Guidance and control of fixed-wing UAVs, with applications to landing

Chapter 8

Introduction

Current autopilots for fixed-wing UAVs have an impressive skill set, enabling *click-and-fly*-style autonomous operations and flight between multiple waypoints at a price that has dropped significantly over the last decades. Some autopilots even provide automatic takeoff and landing capabilities. However, there are still some holes in the autonomy landscape, that this part attempts to fill.

Increased UAV operation in areas with limited infrastructure or available space, such as remote locations or aboard ships, has increased the interest in solutions for UAV takeoff and landing in confined spaces. The perhaps more popular choice is to design the operation around a UAV with vertical takeoff and landing (VTOL) capabilities, such as a rotary-wing UAV or some rotary-wing/fixed-wing hybrid [153]. The increased maneuverability and hover capabilities associated with VTOL UAVs make them easier to land, but this comes at a cost of reduced operational range and endurance, which makes some operations infeasible. A possible solution for these operations is to rely on an arrest landing systems, herein defined as some mechanical system that seeks to remove the kinetic energy from the fixed-wing UAV and bring it to a standstill. Then, the design of the UAV can be focused on the main mission, which is usually what adds value for the end user. Chapter 9 presents a modular system architecture for automatic recovery in such arrest systems, based on non-intrusive additions to available solutions, by making quite general assumptions on the interfaces.

Another way to remove the kinetic energy from the fixed-wing UAV prior to landing, is to dissipate energy by increasing the drag. In a deep stall, where the angle of attack is typically 40° to 60°, the drag forces are significantly larger than during flight below stall, as the drag coefficient typically is an order of magnitude larger in post-stall. Chapter 11 tries to exploit this increase through a nonlinear model predictive controller that tries to minimize the landing velocity, while also landing near the designated landing spot.

For many emerging UAV applications, such as precision drop of sensors or other payload [121], precision agriculture, automatic landing, deliberate or reactive sampling

and mapping, high precision guidance is an important component for mission success. Precise guidance can improve the quality and precision of the recorded data, make the mission more effective in terms of reducing time and fuel consumption, while also ensuring operations in a wider range of conditions and environments, such as ship landing in strong winds. To improve this precision, chapter 10 derives a line-of-sight-based guidance law that considers wind, by building on the coordinated-turn relation from chapter 7.

Chapter 9

Automatic arrest system landing

This chapter considers automatic recovery of fixed-wing UAVs in a moving arrest system, based on the publication

- [65] K. Gryte, M. L. Sollie, T. A. Johansen, and T. I. Fossen, "Control system architecture for automatic recovery of fixed-wing unmanned aerial vehicles in a moving arrest system", 2020, In progress

9.1 Introduction

Arrest landing systems, can be divided into the following categories:

Net landing: flying into a tensioned, fixed net that absorbs the kinetic energy of the impact either vertically [172, 96, 200], horizontally mounted on the roof of a moving car [134], or suspended between drones [97]

Airbag landing: flying into an inflated cushion [79], from any direction

Hook landing: attaching to a wire stretched between two points, e.g. horizontally [94] or vertically [184]

Landing in an arrest system requires two types of navigation from 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 landing. Therefore, a large overlap in the hardware requirements for the two systems is ideal, to simplify avionics. A minimal UAV payload 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 landing application. The required level of precision is largely governed by the relative size of the UAV compared to the arrest system, and the dynamics of the moving arrest system compared to the agility of the UAV. Furthermore, as landing is seen as a more 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 [99, 110]. 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 position measurements at a high rate, with high precision when close to the arrest system, like e.g [96, 79]. Cameras can also be used to measure the roll and pitch of the UAV by considering the horizon [185], and to measure the yaw if the horizon is sufficiently feature rich and a digital surface model is available [154]. Drawbacks include high processing requirements, danger 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 [95, 197] or IR lamps in known locations [66]. Further, stereo vision [185] or recognizing an object with known dimensions [134] is necessary to provide accurate depth information. As the camera must be forward-pointing, which is of little use for most other UAV missions, a gimball could be useful. However, precise knowledge of the orientation of the camera relative to the UAV body axis is paramount [193].

The arrest system may also be equipped with position sensors such as GNSS receivers [134] or ultra-wideband (UWB) beacons, see [55, 112] and chapter 13, 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 autopilots. 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 [59], two independent receivers operating within a short distance will have ionospheric and tropospheric 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, but 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 and clock errors and atmospheric effects to the user from geostationary satellites [59]. Centimeter-level precision, between the UAV and a base station, can be achieved with real-time kinematic (RTK) GNSS [59], a technology that over the last decade has become available to the civilian market at a low cost. Ground based augmentation systems (GBAS) are systems of ground reference stations in use at some airports, transmitting GNSS measurement corrections to commercial aircraft operating in the nearby area using VHF radio [88]. This can be used as an alternative to the radio-based Instrument Landing System (ILS) for approach and landing in difficult visibility conditions with lower infrastructure cost and greater operational flexibility than ILS. 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. Additional advantages with UWB include robustness to interference, resistance to multipath [130], as well as good time resolution allowing for centimeter level precision of range measurements [111], 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, leading to lower precision [55]. Drawbacks associated with GNSS include susceptibility to radio frequency interference (RFI), both natural RFI, such as ionospheric scintillations [199] and multipath, and intentional RFI, such as jamming [149] and spoofing [92].

Other relative navigation off-the-shelf options include the laser-based Object Position and Tracking System (OPATS) [158], GPS- and radar-based dual-thread automatic takeoff and landing system (DT-ATLS) [168, 74], and UAV common automatic recovery system (UCARS) [169] for ship landing.

To approach the arrest system from the correct direction, its (relative) heading must be found, from one of the seven ways to estimate heading [52]. The simplest is through a magnetometer/compass, which unfortunately is highly susceptible to magnetic anomalies and electromagnetic interference (EMI) [52]. With a camera, the relative heading angle can be found [96]. Another solution is to equip the arrest system with multiple position sensors to find the orientation of the baseline between them, see e.g. [176] or [75] 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 landing system is guidance and control. For an overview of different control algorithms for fixed-wing UAVs, see [126], and [181] for path following guidance algorithms. The navigation setup tends to dictate guidlines for the guidance and control system, where visual servoing methods favor pure-pursuit guidance [96, 79], 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, in a control system architecture building modularly and non-intrusively on low-cost commercial off-the-shelf (COTS) hardware (HW) and software (SW). 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 landings in arrest systems that are of a manageable size. COTS autopilot HW and SW 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 the all the necessary features. Even though some commercially available autopilots are capable of automatic landing in fixed locations, such as [11],

this does not suffice for a moving arrest system. Instead of adding the arrest system landing 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. 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. 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.

In section 9.2, the main functional components that are needed for the landing system are discussed, to justify the design choices made. This includes the plan generation (section 9.2.1), navigation (section 9.2.2), motion prediction (section 9.2.3), guidance and control (section 9.2.4) and operator interface (section 9.2.5). The presented landing system is implemented in a real-time system, as discussed in section 9.3, and experimentally validated in two experiments sections 9.3.2 and 9.3.3.

9.2 Problem description

The system architecture illustrated in figs. 9.8 and 9.9 are presented in this section, by considering each of the different functional components that are needed to recover a fixed-wing UAV in a possibly-moving arrest system. The system creates the plan, seen in fig. 9.1, 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 is lines up with the pose of the arrest system. This pose is calculated from precision navigation, that includes compensation for arrest system motion to maximize the chances of impacting near its center. To allow line-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 entanglement.

9.2.1 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 landing has 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

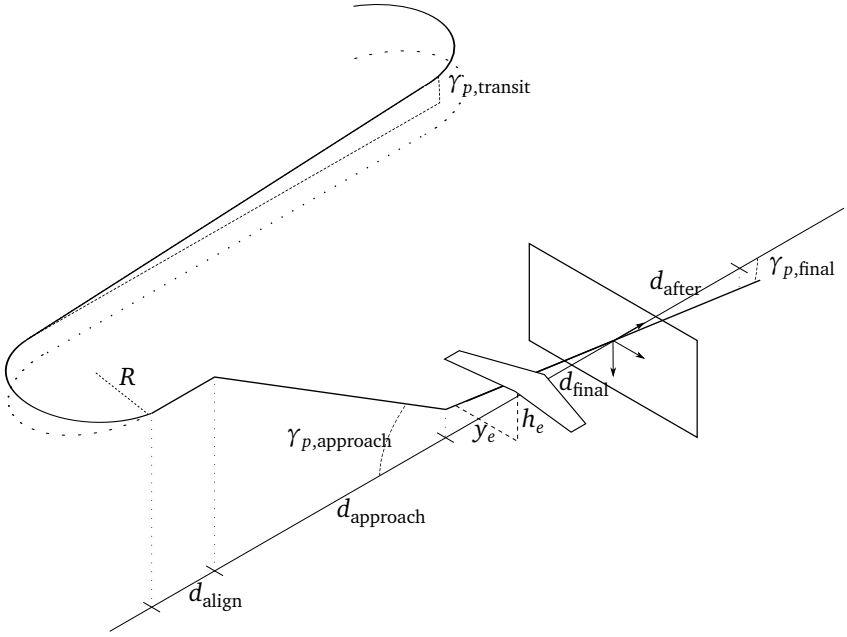


Figure 9.1: The geometry of the arrest system recovery plan, illustrated with a net

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 recovery 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 sensitivity to wind gust and shear.

Based on these strategical decisions, the recovery is divided into the following phases, illustrated in fig. 9.1.

Pre-landing This is when the UAV has finished its mission, and is initiating landing.

Transit The path to the vicinity of the arrest system is constructed using a 2D Dubins path [42]. 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 landing has on the rest of the mission. The altitude is simply a linear descent at an angle, $\gamma_{p,\text{transit}}$. If the desired altitude at the end of the transit phase is unreachable at this descent rate, the final circular arc of the Dubin path is extended into a spiral to shed the excess altitude.

Alignment When exiting the Dubins path, 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 s stable course.

Approach While maintaining alignment, the UAV descends with an 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 arrest system and the desired landing angle γ_{final}. Speed is reduced to lower the impact.

Catch indicates a successful landing, with stopped/disarmed motors, to avoid damage to the propeller or arrest system.

9.2.2 Navigation

The success of the arrest system landing hinges on knowledge of where the UAV is, in relation to the arrest system. For the presented approach, the self-navigation is assumed to be handled by the COTS autopilot, through e.g. 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 standard components, as they are part of most autopilots.

Instead of only using a standalone GNSS receiver, this work uses real-time kinematic (RTK) GNSS, which has been successfully utilized for similar applications [172, 97]. RTK GNSS works by continously sending all raw measurements including carrier phase 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 [45], 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 a loss of the RTK capability, i.e. if GNSS signal strength drops so carrier phase measurements become unusable, only means that the precision and accuracy of the relative positioning is reduced. RTK GNSS was chosen based on the simplicity of its usage when using receivers that support it, availability and high precision. Whether the 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, which involve little extra overhead compared to normal GNSS, it seems tractable. Lately, GNSS receivers with internal processing of the RTK solution have become available at a low cost. These receivers can output high precision and accuracy position and velocity estimates directly into the autopilot, without the need for any additional computation. If using a widely supported format, such as a National Marine Electronics Association (NMEA) standard, the receiver can easily be replaced, becoming a transparent provider of high-precision and high-accuracy estimates to the autopilot.

9.2.2.1 Relative navigation setup

The arrest system is also equipped with RTK GNSS, for the same reasons as above. To reach the full potential of RTK GNSS in terms of precision, it is important that the base station is either surveyed, thus increasing its accuracy, if the arrest system is stationary in a surveyed position, or that the base is moving along with the arrest system, as RTK GNSS only yields a highly accurate and precise *relative* position of the UAV and base. For ship landing in open waters, the only option is a moving-base configuration, so it is therefore mounted on the arrest system. During landing, it is important that the position of the UAV and arrest system are reported in the same frame of reference, with the same origin, such as latitude, longitude and height with respect to the ellipsoidal representation of the Earth, given in the world geodetic system standard of 1984, WGS-84 [45]. 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 a reference sensor such as a GNSS receiver, 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 net center. It is therefore necessary 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 net is instrumented with two GNSS antennas, with one antenna positioned in p^n_{left} on the left side of the net, as seen from the front, acting as the RTK base for the UAV and the second net antenna, which is placed in p^n_{right} on the right side of the net. The position of the net center in the NED frame $\{n\}$ with its origin at the position of the left antenna, $\boldsymbol{p}^n_{\text{net}}$, roll ϕ_{net} and heading ψ_{net} are calculated as

$$\psi_{\text{net}} = \text{atan2}(-b^n_x, b^n_y) \tag{9.1}$$

$$\phi_{\text{net}} = \text{atan2}(b^n_z, \sqrt{{b^n_x}^2 + {b^n_y}^2}) \tag{9.2}$$

$$\boldsymbol{p}^n_{\text{net}} = \frac{1}{2}\boldsymbol{b}^n_{\text{net}} - \boldsymbol{R}_{nb}\boldsymbol{p}^b_{\text{offset}} \tag{9.3}$$

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

9.2.3 Motion prediction

In principle, for a successful landing the UAV only needs to know the relative position of the arrest system at the time of landing. However, this can be difficult to predict for moving arrest systems, as the landing location might not be determined uniquely and with certainty at the start of the landing plan. Not only is this a chicken-and-egg problem, where the relative position of the arrest system, at the time of landing, 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 landing), but it is also highly dependent on the dynamics of the arrest system. The arrest system can either have actively controlled motion, such as [97, 134], which calls for synchronization between the UAV and arrest system, or be passively attached to a moving platform without active control, such as a moving ship [164]. How challenging the prediction is, i.e. how far into the future a reliable pose for the arrest system can be obtained, depends on the stochastic nature of the moving platform. Given good predictions, far into the future, the landing controllers may be less reactive to abrupt arrest system motion, thus allowing slower UAV dynamics.

The proposed system only makes a rough estimate of the position of the arrest system at the time of impact, given its current course and speed, and an initial guess of the time of impact. Given this position, a better estimate of the impact time is found, and the process is re-iterated until sufficiently converged. Particularly for recovery in smaller arrest systems in space-restricted environments, using larger UAVs, 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. For the specific system demonstrated in section 9.3, no prediction or filtering of the attitude and heave motion of the arrest system is performed, as the time horizon for reliable motion prediction for ships may be in the order of a few seconds, due to the stochastic nature of waves and wind [22]. Furthermore, the dynamics of the arrest system are assumed small compared to the agility of the fixed-wing UAV.

9.2.4 Guidance and control

To ensure that the UAV follows the final stages of the recovery plan in a manner that is easy to predict by the operator, line-following guidance [181], such as line-of-sight (LOS), is applied in the approach and final alignment stages. LOS guidance [50] mimics an experienced navigator, by aiming to intercept the desired path a positive, time-varying lookahead distance $\Delta(t)$ ahead of the current position, see fig. 9.2. The UAV, with position $\boldsymbol{p}^n_{\text{UAV}} = [x, y, z]^\intercal$ in the NED frame, follows the line segment that starts in $\boldsymbol{x}^n_k = [x_k, y_k, z_k]^\intercal$ and ends in $\boldsymbol{x}^n_{k+1} = [x_{k+1}, y_{k+1}, z_{k+1}]^\intercal$, illustrated in 2D in fig. 9.2, while the different segments are illustrated in fig. 9.1. The horizontal LOS

guidance law

$$\chi_{\text{LOS}} = \text{atan}\left(\frac{-y_e}{\Delta(t)}\right) \tag{9.4}$$

$$\chi_d = \chi_{\text{LOS}} + \chi_p, \tag{9.5}$$

where $\Delta(t)$ is the horizontal lookahead distance, where $y_e = -\sin(\chi_p)(x - x_k) + \cos(\chi_p)(y - y_k)$ is the horizontal cross-track error, and where $\chi_p = \tan(\frac{y_{k+1}-y_k}{x_{k+1}-x_k})$ is the course of the path [50], has strong stability properties [51]. To overcome stationary errors in the cross-track, as a result of e.g. a misalignment of the navigation system with respect to the airframe, integral effect is needed. This can be acheived by extending eqs. (9.4) and (9.5), by replacing the desired LOS angle χ_{LOS} in eq. (9.4) with one that also accounts for the error between the actual course χ and the integral-free desired LOS-angle:

$$\chi_{\text{LOS}} = \text{atan}\left(\frac{-y_e}{\Delta(t)} + K_i \int \text{atan}\left(\frac{-y_e}{\Delta(t)}\right) - \chi \, \mathrm{d}t\right) \tag{9.6}$$

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 $\sqrt{y_e^2 + \Delta(t)^2}$, see fig. 9.2, it is clear that the desired horizontal position $\boldsymbol{p}^n_{h,\text{look}} = \left[x_{\text{h,look}}, y_{\text{h,look}}, z_{\text{h,look}}, \right]^\intercal$ can be from the UAV position by

$$\boldsymbol{p}^n_{h,\text{look}} = \boldsymbol{p}^n_{\text{UAV}} + \boldsymbol{R}_z(\chi_d) \begin{bmatrix} \sqrt{y_e^2 + \Delta(t)^2} \\ 0 \\ 0 \end{bmatrix}, \tag{9.7}$$

where $\boldsymbol{R}_z$ is the rotation matrix representing rotation about the z-axis.

By making similar geometric considerations in the vertical plane, an analogous vertical LOS guidance law can be formulated as [201]

$$\gamma_d = \text{atan}\left(\frac{-z_e}{\Delta_v(t)}\right) + \gamma_p, \tag{9.8}$$

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

$$z_e = \cos(\chi_p)\sin(\gamma_p)(x - x_k) + \sin(\chi_p)\sin(\gamma_p)(y - y_k) + \cos(\gamma_p)(z - z_k) \tag{9.9}$$

$$= \sin(\gamma_p)\sqrt{(x - x_k)^2 + (y - y_k)^2} + \cos(\gamma_p)(z - z_k), \tag{9.10}$$

$$\gamma_p = \text{atan2}\left(z_{k+1} - z_k, \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}\right), \tag{9.11}$$

and where $\Delta_v(t)$ is the vertical lookahead distance. However, to send height commands instead of desired flight path angle, the same LOS principles behind eq. (9.8) are used to formulated longitudinal guidance in terms of height. The UAV still aims at a point a distance $\Delta_v(t)$ ahead of the projection $\boldsymbol{p}^n_{\text{p}} = \left[x_{\text{p}}, y_{\text{p}}, z_{\text{p}}\right]^\intercal$, of $\boldsymbol{p}^n_{\text{UAV}}$ onto

the vector $\boldsymbol{l}^n = \boldsymbol{x}^n_{k+1} - \boldsymbol{x}^n_k$, which represents the line segment that the UAV is tracking. From the projection point, $\boldsymbol{p}^n_{\mathrm{p}} = \frac{\boldsymbol{p}^n_{\mathrm{UAV}} \cdot \boldsymbol{l}^n}{\boldsymbol{l}^n \cdot \boldsymbol{l}^n} \boldsymbol{l}^n$, where $\cdot$ represents the dot product, the vertical lookahead position $\boldsymbol{p}_{v,\mathrm{look}} = \left[x_{\mathrm{v,look}}, y_{\mathrm{v,look}}, z_{\mathrm{v,look}}\right]^\intercal$ can be computed as

$$\boldsymbol{p}^n{}_{v,\mathrm{look}} = \boldsymbol{p}^n_{\mathrm{p}} + \Delta_v \frac{\boldsymbol{l}^n}{\|\boldsymbol{l}^n\|} \tag{9.12}$$

$$= \left(\frac{\boldsymbol{p}^n_{\mathrm{UAV}} \cdot \boldsymbol{l}^n}{\boldsymbol{l}^n \cdot \boldsymbol{l}^n} + \frac{\Delta_v}{\|\boldsymbol{l}^n\|} \right) \boldsymbol{l}^n. \tag{9.13}$$

Similarly to eq. (9.6), to account for possible steady-state vertical errors, the height component of eq. (9.13) is extended with an integral term

$$\bar{z}_{\mathrm{v,look}} = z_{\mathrm{v,look}} + K_{v,i} \int h_e \mathrm{d}t, \tag{9.14}$$

where h_e is the height error between the current UAV position p_{UAV} and the height at the projection point p_{p}.

To make the guidance performance invariant to changes in wind, both the horizontal and vertical lookahead distances should be functions of the ground speed V_g, i.e.

$$\Delta(t) = V_g \Delta_t \tag{9.15}$$

$$\Delta_v(t) = V_g \Delta_{v,t} \tag{9.16}$$

$$K_{v,i} = \frac{\bar{K}_{v,i}}{V_g}, \tag{9.17}$$

where $\Delta_t, \Delta_{v,t}$ are tunable lookahead-time parameters. The vertical lookahead time, $\Delta_{v,t}$, can be considered as a compensation for the response of the UAV.

The guidance laws eqs. (9.5) and (9.8) 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 horizontal and vertical lookahead points, $\boldsymbol{p}^n_{\mathrm{look}} = \left[x_{\mathrm{h,look}}, y_{\mathrm{h,look}}, \bar{z}_{\mathrm{v,look}}, \right]^\intercal$, to the autopilot. However, if the autopilot provides an interface to receive e.g. desired course angle, desired path angle and desired airspeed, χ_d and γ_d from the eqs. (9.5) and (9.8) can be used directly. Furthermore, if an interface that accepts desired roll, desired pitch and desired throttle, these values can be calculated on the basis of χ_d, γ_d and airspeed error, similar to chapter 10 and [201]. The lower-level control, regardless of the interface, is assumed provided by the autopilot.

9.2.4.1 Recovery detection

To avoid propeller damage or entanglement in the arrest system, the motor should be stopped before it hits the arrest system. For a fixed-wing UAV in puller configuration,

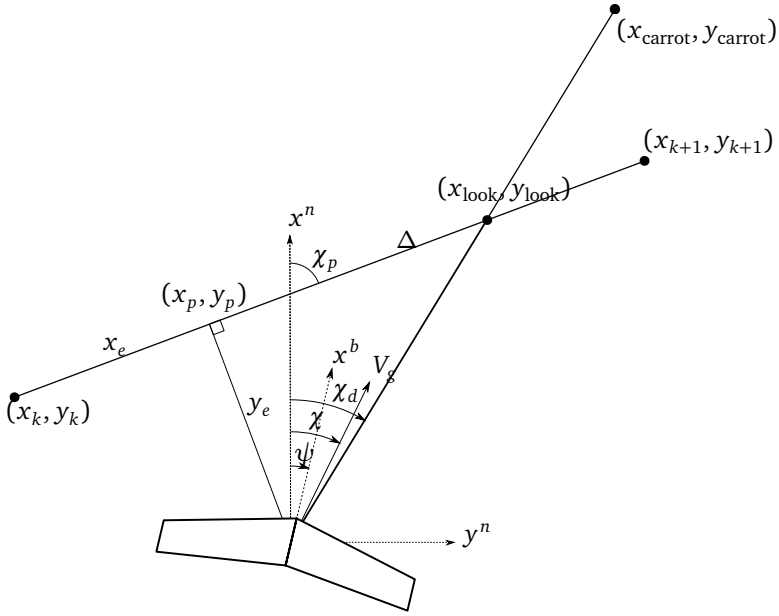


Figure 9.2: Geometry of the horizontal line-of-sight guidance

the propeller is the first thing that hits the arrest system, which forces the stop to be triggered by distance to the arrest system. 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 landing is deemed unsuccessful and the motor is re-activated. Upon detection of impact the motor is permanently disarmed. This impact is determined based on the longitudinal acceleration of the UAV.

9.2.5 Operator interface

Generally, an increased level of autonomy, decreases the requirements for the user interface for the operator.

The solution presented here allows for automatic recovery, but still requires the operator to monitor performance and possibly intervene. This requires radio communication, such that the UAV can report its state, and a user interface that displays the essence of this information. There are many COTS UAV ground control station (GCS) graphical user interface (GUI) available, that are used during normal UAV operations.

Neptus [38] was chosen as the basis for an arrest system recovery GUI module. Specially made plugins provide two main features. First, the operator is able to decide the parameters that are used when the recovery plan is generated. This

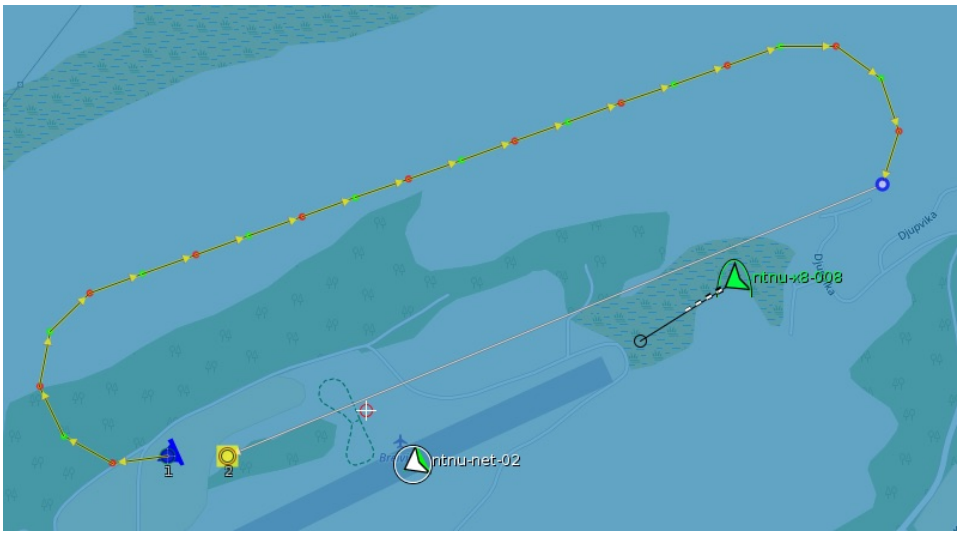


Figure 9.3: The recovery plan generation GUI, with the start (blue), the arrest system (white), the UAV (green) and the plan inbetween. The small black circle ahead of the UAV corresponds to the carrot point, while the dynamic plan is not in the map view.

includes

- selecting the starting point for the recovery plan, or select that it should start from the current UAV position
- select the recovery location. This enforces the UAV to only receive arrest system position messages from the selected entity.
- the different distances and angles illustrated in fig. 9.1

This interface also presents the generated plan to the operator, to allow validation, see fig. 9.3.

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. 9.4 and 9.5. This also includes a button that will abort the landing attempt, in case of unforeseen or undesirable events.

9.2.5.1 Aborted recovery framework

If the operator clicks the abort button, 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,

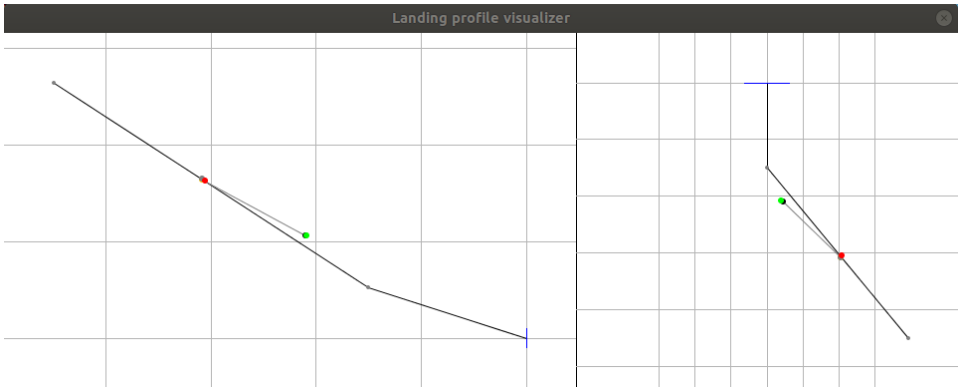


Figure 9.4: The arrest system approach path visualization plugin, illustrating the segments of the approach path, as well as the current UAV position (red dot) and commanded position sent to the autopilot (green dot). In order to better show the errors in height and in cross-track, these are scaled independently 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 net was rotated after the plan generation, to also show the lateral changes.

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 landing impossible or highly risky, are

Loss of radio communication: making it impossible for the UAV to know the position of the arrest system

Severe weather conditions: such as strong and/or unpredictable wind, increase the risk involved with landing

Severe arrest system motion: caused by e.g. waves also increase the risk involved with landing

Poor landing performance: The ultimate objective is to hit the arrest system, so the system predicts if this is not achievable, by e.g. considering the bearing angle from the arrest system to the UAV

Large relative speed: from e.g. a strong tail wind can lead to a hard landing that jeopardize the structural integrity of the UAV

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

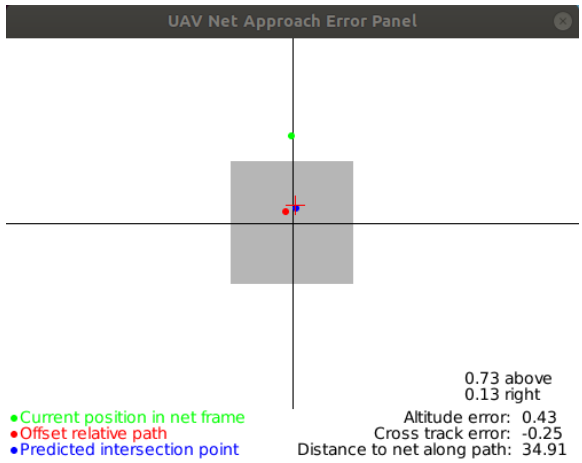


Figure 9.5: The GUI landing profile plugin, illustrating the net (here drawn with a size of 5 by 5 meters), current UAV position relative the path (red dot), predicted net impact point (blue dot) and current position in a NED-frame rotated around the z-axis to align with the net heading (green dot). The red cross marks the net impact point of the previously completed landing.

However, not all situations allow for an abort [172]. Therefore, the abort framework also acknowledge the UAVs *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 go-around, and if the fuel or battery is running so low that a go-around is impossible. An elevated severity level causes the UAV to ignore aborts, and continue the recovery.

9.3 Experimental validation

To evaluate the arrest landing system architecture, a series of net landing experiments are performed. A net is chosen since it occupies very 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 first experiment demonstrate the performance of the system on a stationary net, to isolate the control performance from the motion of the net. The second series of tests involve landing when the net is manually moved, first by a yawing motion and then by height oscillations.

Figure 9.6: The NTNU Skywalker X8 flying wing UAV

9.3.1 Instrumentation

The tests use a Skywalker X8 styrofoam flying-wing UAV with pusher propeller, see fig. 9.6, that has a wingspan of 2.1 m, a takeoff weight of about 3.5 kg, and cruise speed of 18 m/s.

The airframe, actuators and autopilot hardware of the two UAVs are different, while the hardware that the landing 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.

During all the experiments, the UAV and net are connected to a ground control station over a data link based on Ubiquiti Rocket M5, using AirMax. The ground station, depicted in the lower part of fig. 9.9, consists of two computers; one running Ubuntu Linux and another running Windows 10. The Linux computer runs the Neptus ground control station, with the landing GUI, which is used to visualize information and to interface DUNE on both the UAV and the net, through IMC messages. 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.

Figure 9.7: Picture of the instrumented net suitcase, showing the two GNSS receivers ontop of the SenTiBoard and an ethernet switch. In the front is the onboard computer with a custom cape, ontop of the 5 GHz radio.

9.3.1.1 Net hardware and software

The components in the net instrumentation are illustrated in the upper left part of fig. 9.9, and pictured in fig. 9.7. 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 support for Real-time-kinematic (RTK) positioning. The first UART connection of each receiver is configured to send position and velocity estimates to the SenTiBoard at 5 Hz rate [4], while the second UART is used for a RTCM3 correction data stream also with a rate of 5 Hz, needed to run 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 over a UDP stream on the local network, and to the rover receiver of the net over UART. The SenTiBoard timestamps the received estimates and sends them to the BBB over USB for processing. On the BBB the position data is parsed, translated into net center position and heading according to eqs. (9.1) and (9.3). This is implemented as a task in the *DUNE Unified Navigation Environment* robotic middleware framework [150], while the resulting net position and heading, are distributed over the network in terms of inter-module communication (IMC) [117] protocol over UDP.

9.3.1.2 UAV hardware and software

The UAVs used in the experiments use ArduPlane 3.9.9 [11], running on a Pixhawk autopilot hardware in the X8, and on a Pixhawk 2.1 for the Dolphine. The rest of the land-specific payload is common to all the experiments, and is illustrated in the upper right of fig. 9.9. 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 compensate for this, the ArduPlane parameter `OGN_HGT_MASK` is set to use the height from the RTK GNSS receiver to correct the barometer drift. Alternatively, the RTK GNSS receiver could have been set as the primary altitude sensor.

The Odroid also runs DUNE, which includes the net motion prediction, plan generation, guidance, and interface to the autopilot, in addition to publishing the state of the UAV and net landing system as IMC messages over UDP.

The landing plan, as described in section 9.2.1, is generated upon request by the operator, according to the parameters, start location, and the predicted location of the landing. While the transit phase is static, based on the initial estimate of the landing location, the remainder of the plan is dynamic, and will update as the UAV receives position updates from the net. For simplicity, the Dubins path, which is computed using the Dubins path library provided in [191, 192], is represented as a sequence of waypoints. This makes the path piecewise linear, and thus not flyable according to [189], 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. 9.8, and may be inspected by the operator. Upon initiation of landing, 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 and longitudinal TECS guidance 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. 9.8.

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 eqs. (9.5) and (9.8). However, when in GUIDED mode, a desired location is interpreted 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 low, and the horizontal components of the desired location $\boldsymbol{p}^n_{\text{look}}$ are extended in the direction χ_d to form a carrot point $\boldsymbol{p}^n_{\text{carrot}}$ for the UAV to follow, when combined with the original desired height, see fig. 9.2. Essentially, the *Fake LOS* block transforms the desired position interface into a desired course and height interface.

The aggregate lookahead point $\bar{\boldsymbol{p}}^n{}_{\text{look}} = \left[x_{\text{h,look}}, y_{\text{h,look}}, \bar{z}_{\text{v,look}}, \right]^\intercal$ 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 are disabled. To reduce the need for the integral term in eqs. (9.6) and (9.14), it is important to precisely determine the correct roll misalignment of the autopilot, and enter this in the `AHRS_TRIM_X` parameter to reduce the cross-track error.

From this desired height, and the desired airspeed, the ArduPlane TECS guidance controller calculates the desired pitch angle and throttle command based on the energy balance[1]. 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 TECS controller, 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 desired roll and pitch values from the guidance controllers are sent into the low-level roll and pitch PID controllers in ArduPlane.

The recovery detection of section 9.2.4.1 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 throttle. The threshold is set to be speed dependent, 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 even with slow communication.

[1] A bug was discovered in the ArduPlane TECS implementation, which lead to poor altitude tracking. It was fixed, but has not been merged into ArduPlane at the time of writing. See `https://github.com/ArduPilot/ardupilot/pull/12822`

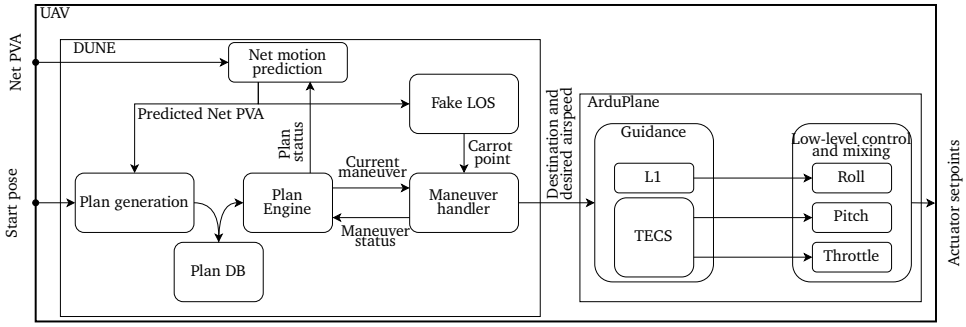


Figure 9.8: Overview of the UAV control architecture. Each block within DUNE roughly corresponds to a separate task, with IMC messages being passed between them.

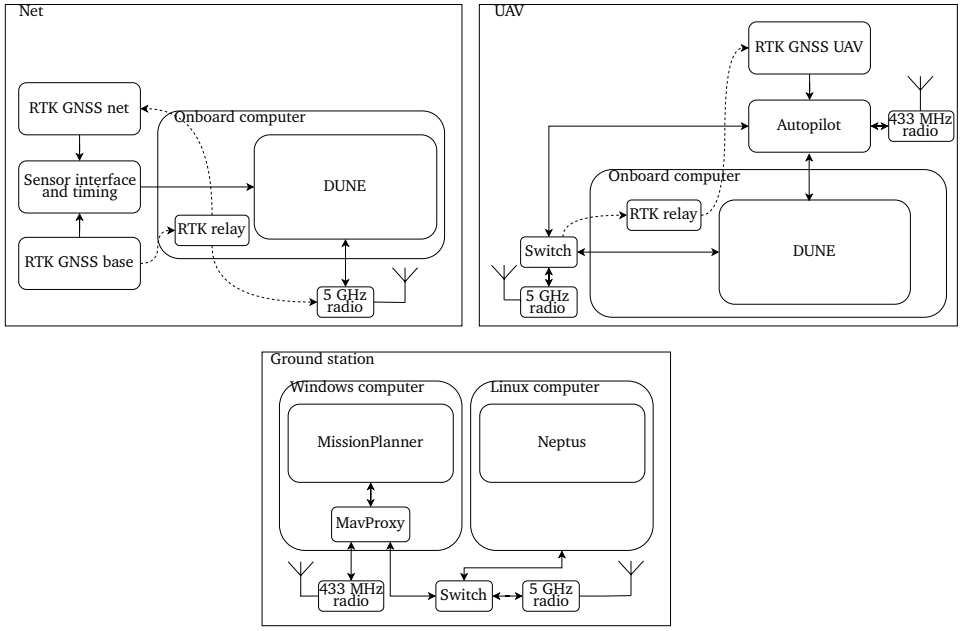


Figure 9.9: The different subcomponents of the UAV, the ground station and the instrumented net, that are relevant for landing

9.3.2 Experiments with stationary net with flying-wing

A 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. 9.6. This allowed looping the landing plan to perform multiple landing attempts in a single flight with lower risk. The setup of the net GNSS antennas is shown in fig. 9.10. 43 recovery maneuvers were performed with a 220 m long, 9° glideslope in the approach and

Figure 9.10: Picture of the UAV flying through the net rig, without a physical net. The GNSS-antennas are mounted on tripods on each side of the net, while the net instrumentation suitcase is in the center.

a 190 m long, 4° final alignment. Winds were calm without significant gusts. The position of the UAV in the net frame for all attempts are shown in fig. 9.11, also showing the planned descend profile shown as a dotted line. The top plot showing the sideways movement of the UAV may indicate 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. The rapid, short sideways movements is likely caused by the limited resolution of the net heading used to transform the UAV position into the net frame.

The position where the UAV would have impacted the net is shown in fig. 9.12, with performance numbers in table 9.1. Parts of the error are quantization errors from the UAV position Mavlink message.

9.3.3 Experiments with moving net

Net movement was simulated by manually moving the net GNSS antennas while the UAV was approaching. fig. 9.13 shows a test where the net heading was changed by

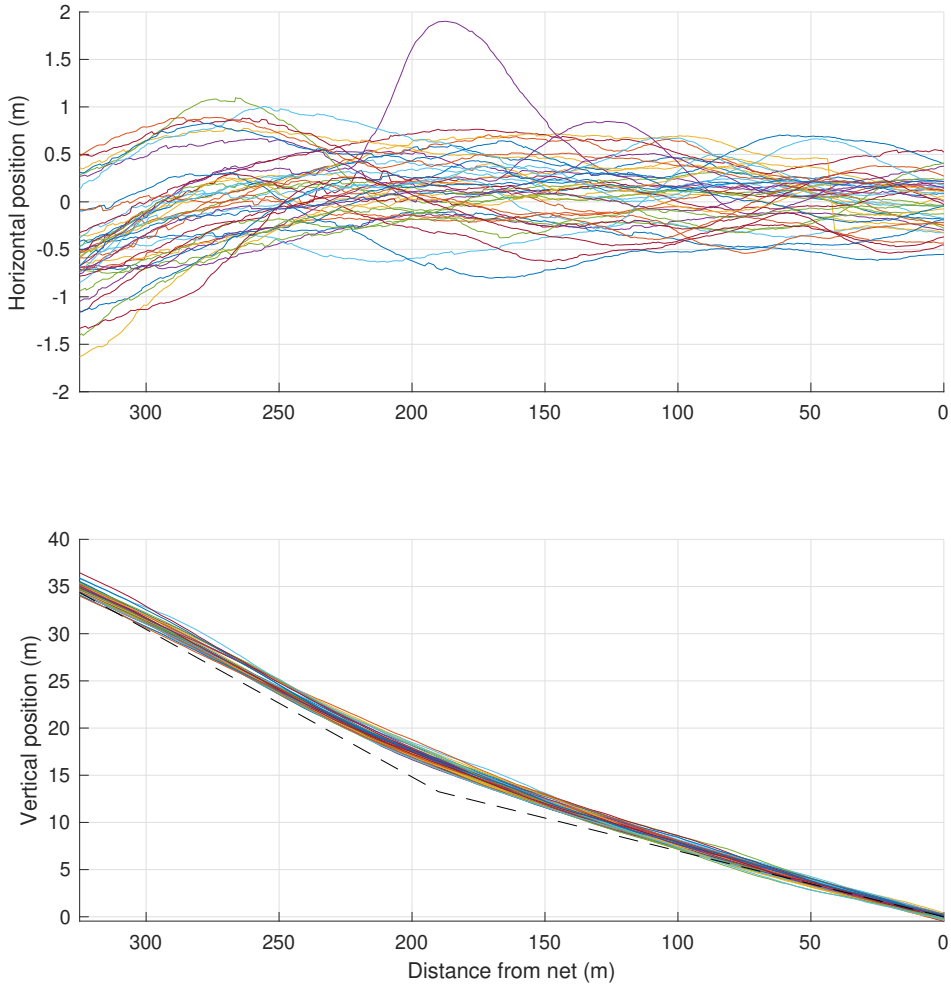


Figure 9.11: The position of the UAV in the arrest system frame while approaching the net, for all attempts.

moving the right antenna away from the UAV towards north east. The dynamic plan, being continuously updated, changed accordingly such that the UAV made a turn to align itself with the net. The UAV hit 0.02 m right and 0.32 m above the net center for this particular landing. fig. 9.14a and fig. 9.14b shows two tests where both net antennas were lifted in a wave-like motion, with a different frequency in each test, although both frequencies are probably higher that what would be expected on a large ship. The yellow desired height steps when the active segment changes, which happens when the lookahead-point reaches the end of a line segment. The UAV height can be seen to oscillate like the desired height, but with a phase shift caused by the response of the TECS-controller. The impact was about 0.05 m above and 0.5 m below the net center in the two tests.

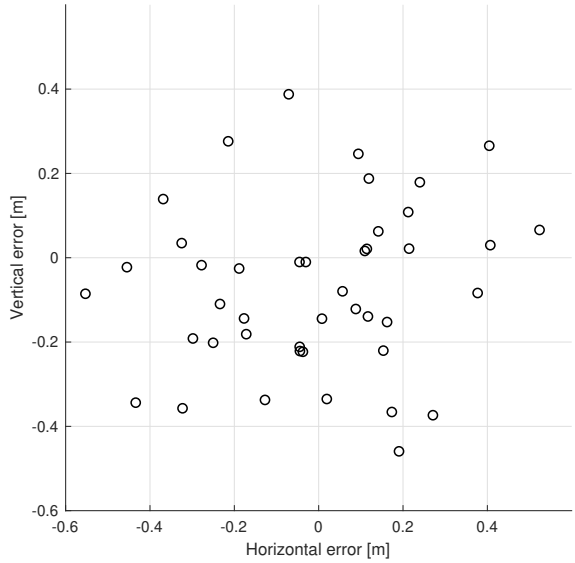


Figure 9.12: The position of the UAV when impacting the net, for all attempts, as seen into the net from the front.

Table 9.1: Net impact performance of system with stationary net

	Vertical	Horizontal
Mean	-0.07m	-0.01m
RMS	0.21m	0.25m
Std. dev	0.20m	0.25m

The results shown in this section were obtained before the aforementioned bug in the ArduPilot TECS-controller [1] was fixed, and before the integral effect in eqs. (9.6) and (9.14) were implemented. Prior to this fix, the vertical error was close to 1 m, so naturally the results with the moving net should only be considered preliminary.

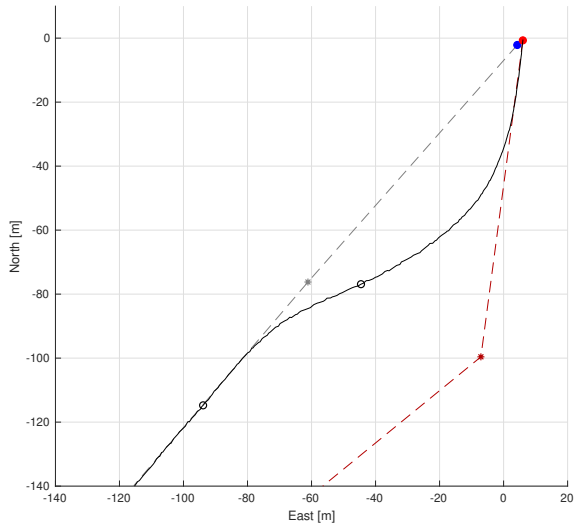
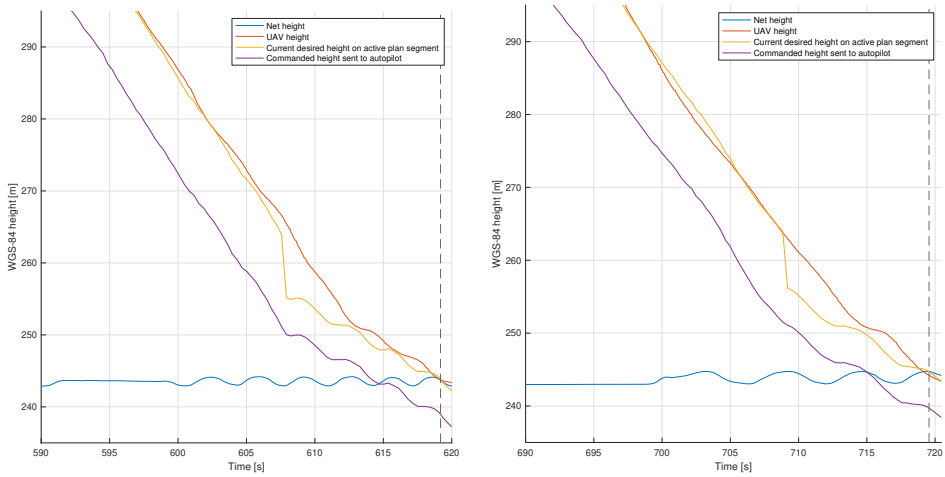


Figure 9.13: A landing where the net was rotated by close to 35 degrees in the approximately 4 second long period when the UAV was between the black circles. The plan up to this point is shown in gray, while the dynamic plan after the net movement is shown as the red dotted line. The blue dot is the initial net position and the red is the final net position, and the final part of the plan point straight into the net.



(a) The impact was approximately 0.05m above net center

(b) The impact was approximately 0.5m below net center

Figure 9.14: Vertical antenna movement simulating waves. The commanded height sent to the autopilot is the height of the lookahead point $\boldsymbol{p}^n_{\text{look}}$ which is lower since the point lies further down the descend profile. The vertical dashed lines indicate the time of impact with the net.

9.4 Conclusion

This chapter presented a modular system for automatic recovery of fixed-wing UAVs in a moving arrest system, based on non-intrusive additions to an autopilot with very general assumptions on its interface. This is achieved by line-of-sight guidance, which sends an augmented desired position to the autopilot, to ensure line following along the virtual runway that guides the UAV into the arrest system. The translation and rotation of this line is determined by the pose of the arrest system, determined through two GNSS receivers, where one is configured as an RTK base station for high precision, that estimates the position, heading and roll angle. The autopilot in the UAV is also equipped with a GNSS receiver that receives corrections from the arrest-system base, ensuring the accuracy of the baseline between the UAV and arrest system, and that the arrest system and UAV are in the same reference system. The system was shown to accurately land a flying-wing UAV 0.07 ± 0.20 meter below and 0.01 ± 0.25 meter to the right of the center of a net, during 43 trials, and demonstrated promising results with a manually moved net.

Chapter 10

Lateral guidance using the coordinated turn

As seen in chapters 8 and 9, guidance is an important part of many UAV applications, as it dictates where the UAV will fly. This chapter focus on performance of guidance in wind, by comparing a line-of-sight-based approach, that relies on chapter 7, to the state-of-the-art L_1 guidance controller[1] [144].

The chapter is based on the publication

- [64] K. Gryte, T. A. Johansen, and T. I. Fossen, "Coordinated-turn based path following for fixed-wing unmanned aircraft", *Journal of Guidance, Control, and Dynamics*, 2020, Submitted

The presented guidance law has also has been successfully applied in

- [121] S. G. Mathisen, F. S. Leira, H. H. Helgesen, K. Gryte, and T. A. Johansen, "Autonomous ballistic airdrop of objects from a small fixed-wing unmanned aerial vehicle", *Autonomous Robots*, Jan. 2020. DOI: 10.1007/s10514-020-09902-3

Section 10.2 introduces the equations of motion, and the LOS and L_1 guidance laws. This is further expanded in section 10.3, by relating the methods to path following for fixed-wing aircraft, where also the main results are presented, see section 10.3.1. The different guidance laws are validated in section 10.4, by comparison to the L_1 guidance law [143, 144], a variation of nonlinear guidance that is currently the default lateral guidance for the ArduPlane autopilot, through a simulation study, and by experimental verification. Finally, the results are discussed in section 10.5, and conclusions are drawn in section 10.6.

[1] L_1 refers to the line segment from the vehicle position to a reference point, and is not to be confused with L_1 adaptive control [135]

10.1 Introduction

The literature contains many different approaches to the path following problem, see e.g. the survey papers [181, 145] that describe and compare approaches based on proportional-integral-derivative control, nonlinear guidance, vector field guidance [136, 137], pure pursuit and line-of-sight (LOS). Some other guidance laws, primarily used in missile guidance, are Lamberts guidance, differential game guidance, proportional navigation and optimal control [203, 198, 171].

LOS guidance originates from the missile community, where the missile is guided, for example by a laser beam, along the straight line from the position of the tracking unit to the target. This is in contrast to pure pursuit guidance, that follows the straight line from the missile would follow the straight line between *itself* and the target. Under LOS guidance, the equilibrium at zero cross-track error is shown to have uniform semiglobal exponential stability [51].

The coordinated-turn relation, sometimes referred to as the bank-to-turn relation, is a common simplification of the lateral dynamic of fixed-wing aircraft, often used in guidance [14]. The relation assumes that there is no acceleration along the body y-axis during the turn, and that the centripetal acceleration is produced by the horizontal component of the lift. This ultimately relates roll or bank angle to heading or course angle. In the literature, it is presented in many different ways, with varying formulations and degree of simplification, see e.g. [180, 155, 148, 14]. Due to the relatively slow airspeed of small UAVs, it is particularly important to consider effects of environmental forces, primarily wind [163, 161, 162]. This is often neglected, either implicitly by performing heading guidance instead of course guidance, or explicitly by neglecting the difference between course and heading. For UAVs with hovering capabilities, course might not always be defined, but for fixed-wing UAVs it is well defined, under the assumption of a positive forward speed. Formulating the guidance law in course, rather than heading, reduce the need for integral effect, as course is invariant to wind, while integral effect is still needed to overcome sensor misalignments and model errors. Another way to simplify the lateral dynamics is through the skid-to-turn relation, where the aircraft is yawed to produce a sideslip angle. As a result, the propulsion force and the additional aerodynamic forces resulting from the increased projected area of the fuselage into the wind, cause a change in heading and course. However, the increased drag caused by the sideslip, and the relatively small force that can be created in this manner, makes this approach less popular. One exception is applications with downward-facing sensors, as addressed in [128], where the skid-to-turn maneuver cause the sensor to be directed more favorably than in a coordinated turn.

This work therefore seek to express the LOS guidance law, through the coordinated turn equation, using course. To achieve this, and to be clear on what assumptions are made, the coordinated turn equation is derived under rather general conditions, clearly stating the assumptions. The main contributions, in addition to the resulting variations of the guidance law, are its accompanying stability proof, along with a

comparative study with the state-of-the-art. An experimental flight test verification is also presented.

10.2 Definitions

10.2.1 Equations of motion

The motion of a vehicle in the horizontal north-east frame can be described by the two sets of differential equation

$$\begin{aligned} \dot{x} &= V_g \cos(\chi) &&= V_a \cos(\psi) + W_x \\ \dot{y} &= V_g \sin(\chi) &&= V_a \sin(\psi) + W_y, \end{aligned} \tag{10.1}$$

where x, y is the position along the north and east axes, respectively, and where W_x and W_y are wind speeds in the north and east direction. As noted by [137], representing eq. (10.1) in course angle and ground speed, independent of wind velocity, significantly improves wind-disturbance rejection, and is thus used in the following. Furthermore, obtaining a good estimate of the course comes for free, given the high accuracy and precision of available GNSS receivers with Doppler velocity measurements, in contrast to accurate heading measurements that depend on magnetometers, and careful calibration procedures.

When considering an arbitrary horizontal path, $[x_p(\varpi) y_p(\varpi)]^\intercal$, parameterized by the path variable ϖ, the vehicle's cross-track error y_e is defined as [51]

$$\overbrace{\begin{bmatrix} 0 \\ y_e \end{bmatrix}}^{\boldsymbol{x}_e^p} = \overbrace{\begin{bmatrix} \cos(\chi_p(\varpi)) & \sin(\chi_p(\varpi)) \\ -\sin(\chi_p(\varpi)) & \cos(\chi_p(\varpi)) \end{bmatrix}}^{\boldsymbol{R}_{pn}} \overbrace{\begin{bmatrix} x - x_p(\varpi) \\ y - y_p(\varpi) \end{bmatrix}}^{\boldsymbol{x}_e^n}, \tag{10.2}$$

where $\chi_p(\varpi) = \text{atan2}(y_p'(\varpi), x_p'(\varpi))$ is the course angle of the path, and where the along-track error $x_e = 0$ follows by design, see fig. 10.1. This can also be seen as a rotation of the error vector $\boldsymbol{x}_e$ in NED to the ground velocity frame, which is tangential to the path. For straight-line paths between two waypoints, $\boldsymbol{x}_k^n = [x_k, y_k, z_k]^\intercal$ and $\boldsymbol{x}_{k+1}^n = [x_{k+1}, y_{k+1}, z_{k+1}]^\intercal$, $\chi_p(\varpi) = \text{atan2}(y_p'(\varpi), x_p'(\varpi)) = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k)$. Differentiating eq. (10.2), and inserting eq. (10.1), as given in [51], the cross-track velocity can be put on the form

$$\dot{y}_e = V_g \sin\big(\chi - \chi_p(\varpi)\big). \tag{10.3}$$

In the following, the dependency on the path variable ϖ is assumed implicitly, for the sake of brevity.

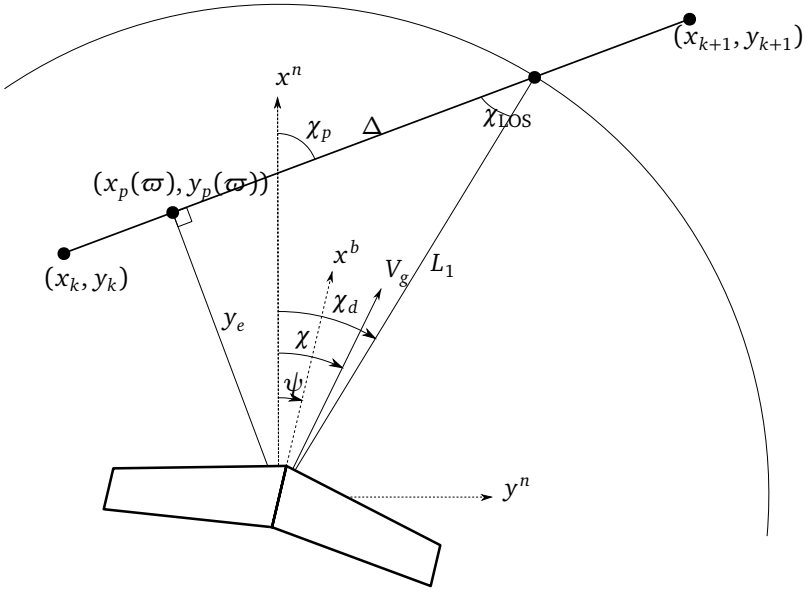


Figure 10.1: LOS guidance geometry

10.2.2 Guidance laws

10.2.2.1 Line-of-sight guidance

Line-of-sight (LOS) guidance mimics an experienced navigator, by aiming to intercept the desired path a positive, time-varying lookahead distance Δ ahead of the current position, see fig. 10.1. By assuming perfect tracking of the course angle, i.e. $\chi = \chi_d$, [51] shows that the LOS guidance law

$$\chi_d = \chi_{\text{LOS}} + \chi_p = \text{atan}\left(\frac{-y_e}{\Delta(t)}\right) + \chi_p, \tag{10.4}$$

renders the equilibrium point $y_e = 0$ of the cross-track error dynamics eq. (10.3) uniformly semiglobally exponential stable (USGES), through the Lyapunov function candidate $V = \frac{1}{2}y_e^2$, if $0 < \Delta_{\min} < \Delta(t) < \Delta_{\max}, 0 < V_{g,\min} < V_g < V_{g,\max}$. In the following the time dependency of Δ is assumed implicitly.

10.2.2.2 L_1 guidance

The L_1 guidance controller developed in [143], is considered state-of-the-art within course control for UAVs. It is implemented and extensively used in the popular ArduPlane autopilot [9], and is based on finding a commanded lateral acceleration to bring the vehicle closer to the desired path. The fundamental realization of the

guidance law, is that when moving in a direction $\vec{V}_g$, tangentially along a circular arc of radius R that ends up in a desired point on the desired path, a distance L_1 from the vehicle, the radius R can be expressed as

$$R = \frac{L_1}{2\sin\tilde{\chi}} \tag{10.5}$$

where $\tilde{\chi}$ is the angle between the ground speed vector and the L_1 vector. Then, the lateral acceleration required to stay on the circular arc, to approach the desired path, a distance L_1 in front of the vehicle, can be shown to be [2]

$$a_{s_{\text{cmd}}} = 2\frac{V_g^2}{L_1}\sin(\tilde{\chi}). \tag{10.6}$$

The angle $\tilde{\chi}$ can be found from the definition of the cross-product [33]

$$\sin(\tilde{\chi}) = \frac{\vec{V}_g \times \vec{L}_1}{|V_g||L_1|} \tag{10.7}$$

as seen in fig. 10.1. This expression is not used in the Ardupilot implementation, where the course error is found from $\tilde{\chi} = \text{asin}(\frac{y_e}{L_1}) + \text{atan2}(v_{p,y}, v_{p,x})$, where $v_{p,x}$ and $v_{p,y}$ represent the ground velocity vector decomposed in along- and cross-track components. [144] shows that the equilibrium $\tilde{\chi} = 0$ is asymptotically stable, for autonomous systems, assuming constant L_1 distance and path curvature.

A clear distinction between LOS and L_1 is that while the LOS lookahead distance Δ is some distance ahead of the vehicle, along the path, the L_1 distance is simply a radius around the vehicle, and the reference point is the intersection of this circle and the desired path. This resembles enclosure-based LOS [50], where translations between lookahead-based and enclosure-based LOS are made through

$$\Delta = \sqrt{L_1^2 - y_e^2}. \tag{10.8}$$

From eq. (10.8) it is apparent, as stated by [33], that L_1 guidance is not defined for $y_e > L_1$. See appendix A for notational differences between L_1 presented in [144] and the notation used here.

10.3 Fixed-wing aircraft guidance

Due to the underactuated nature of fixed-wing aircraft, the most common way to achieve lateral path following is by commanding a desired roll angle, ϕ_d. Thus, the guidance laws presented in section 10.2.2, must be extended to compute a desired roll angle to be applicable for fixed-wing aircraft. For L_1 guidance, this is not

[2] This instantaneous circle is merely a conceptual circle used in the derivation, see [144, Fig. 1], and should not be confused with the circle of acceptance in fig. 10.1.

explicitly addressed by [143, 144], but [33] attributes this to a simplified version of the coordinated turn equation, leading to

$$\phi_d = \text{atan}\left(\frac{a_{s_{\text{cmd}}}}{g}\right), \tag{10.9}$$

where g is the acceleration due to gravity, such that

$$\phi_d = \text{atan}\left(2\frac{V_g^2}{gL_1}\sin(\tilde{\chi})\right). \tag{10.10}$$

In the implementation in ArduPlane an additional $\cos\theta$-term is multiplied with eq. (10.10), and the factor 2 in eqs. (10.6) and (10.10) is replaced by a navigational constant K_{L_1}.

10.3.1 Coordinated-turn inspired LOS

The LOS guidance law eq. (10.4) guarantees $y_e \to 0$ if $\tilde{\chi} \to 0$, thus it is necessary to find a lateral controller such that $\chi \to \chi_d$. Using the coordinated turn relation, which relates roll angle and turn rate, the lateral dynamics can be approximated. Under the assumptions in chapter 7, two equivalent expressions for the coordinated turn can be formulated as

$$\dot{\chi} = \frac{g\cos(\chi-\psi)}{V_g\cos\gamma\cos\theta}(\tan\phi + \sin\theta\tan(\chi-\psi)) \tag{10.11}$$

$$\phi = \text{atan}\left(\left(\frac{V_g\cos\gamma\cos\theta}{g\cos(\chi-\psi)}\dot{\chi} - \tan(\theta)\tan(\chi-\psi)\right)\right), \tag{10.12}$$

where ϕ, θ, ψ represent the aircraft attitude in the Euler angles, roll, pitch, yaw, respectively, and γ is the path angle flown by the aircraft. An intuitive explanation for the $\sin\theta\tan(\chi-\psi)$-term in eq. (10.11) can be obtained from fig. 2.4 by assuming the scenario where the wind speed V_w is orthogonal to the airspeed V_a, which is directed along the body x-axis if one also assumes that $\beta = 0$. Then

$$\tan(\chi-\psi) = \frac{V_w}{V_a}, \tag{10.13}$$

and it is seen from eq. (10.11) that for the same roll angle, a stronger wind blowing to the right, across the path of the aircraft, will naturally increase the course rate. Equivalently, a stronger wind blowing to the right calls for a decreased or more negative roll angle to maintain a constant course rate.

To show that $\chi \to \chi_d$, the Lyapunov function candidate $V_2 = \frac{1}{2}\tilde{\chi}^2$, with $\tilde{\chi} = \chi_d - \chi$, can be considered. Its time derivative can be shown to be

$$\dot{V}_2 = \tilde{\chi}\dot{\tilde{\chi}} = \tilde{\chi}\left(-\frac{\Delta}{\Delta^2+y_e^2}\dot{y}_e + \dot{\chi}_p - \frac{g\cos(\chi-\psi)}{V_g\cos\gamma\cos\theta}(\tan\phi + \sin\theta\tan(\chi-\psi))\right) \tag{10.14}$$

where the desired course rate

$$\dot{\chi}_d = -\frac{\Delta}{\Delta^2 + y_e^2}\dot{y}_e + \dot{\chi}_p, \tag{10.15}$$

is found from time differentiation of eq. (10.4). By assuming that the time constant of the roll is small, and that the roll controller is well tuned, it is assumed that $\phi = \phi_d$. Choosing

$$\phi_d = \text{atan}\left(\frac{V_g \cos\gamma \cos\theta}{g\cos(\chi - \psi)}\left(f(\tilde{\chi}) - \frac{\Delta}{\Delta^2 + y_e^2}\dot{y}_e + \dot{\chi}_p\right) - \sin\theta\tan(\chi - \psi)\right), \tag{10.16}$$

where $f(\tilde{\chi})$ is an odd function, renders the equilibrium point $\tilde{\chi} = 0$ UGAS by [93, theorem 4.9], since $\dot{V}_2 = -f(\tilde{\chi})\tilde{\chi} < 0$, and since $f(\tilde{\chi})\tilde{\chi} > 0$. However, further restricting $f(\tilde{\chi})$ to be any polynomial of odd powers with positive coefficients renders the equilibrium point GES, since $\dot{V}_2 = -f(\tilde{\chi})\tilde{\chi} < -K\tilde{\chi}^2$ for some constant $K > 0$[3].

10.3.2 Linear analysis

Seeking an intuitive way to adjust the tuning parameters Δ and K_1 of guidance law eq. (10.16), in terms of damping factor and bandwidth of the system, linear analysis is applied, like in [143]. This is achieved by approximating the cross-track error dynamics as a mass-spring-damper, starting from the cross-track acceleration $\ddot{y}_e$. The linear analysis is shown for eq. (10.16) with $f(\tilde{\chi}) = K_1\tilde{\chi}$, but it is equivalent for higher order polynomials as well, since the linear term dominates near the linearization point. Time differentiation of eq. (10.3) gives

$$\ddot{y}_e = \dot{V}_g \sin\big(\chi - \chi_p\big) + V_g \cos\big(\chi - \chi_p\big)\big(\dot{\chi} - \dot{\chi}_p\big) \tag{10.17}$$

$$\ddot{y}_e = V_g \cos\left(\text{asin}\left(\frac{\dot{y}_e}{V_g}\right)\right)\left(K_1\tilde{\chi} - \frac{\Delta}{\Delta^2 + y_e^2}\dot{y}_e\right) \tag{10.18}$$

$$= -\sqrt{{V_g}^2 - \dot{y}_e^2}\left(K_1\,\text{asin}\left(\frac{\dot{y}_e}{V_g}\right) + K_1\,\text{atan}\left(\frac{y_e}{\Delta}\right) + \frac{\Delta}{\Delta + {y_e}^2}\dot{y}_e\right) \tag{10.19}$$

where it is assumed that $\dot{V}_g = 0$ and that V_g is horizontal, such that $\chi = \chi_p - \text{asin}\left(\frac{\dot{y}_e}{V_g}\right)$. Linearizing this around $\boldsymbol{x} = [y_e, \dot{y}_e]^\intercal = \boldsymbol{x}^* = [0, 0]^\intercal$, gives the linear system

$$\dot{\bar{\boldsymbol{x}}} = \left[\begin{array}{cc} 0 & 1 \\ \frac{\partial \ddot{y}_e}{\partial y_e} & \frac{\partial \ddot{y}_e}{\partial \dot{y}_e} \end{array}\right]\Bigg|_{\boldsymbol{x}=\boldsymbol{x}^*} \bar{\boldsymbol{x}} \tag{10.20}$$

[3] As shown by [18], mechanical systems with rotational degrees of motion cannot be globally stabilized by continuous feedback due to the topological obstruction imposed by SO(3). Hence, the UGAS and GES properties are based on the assumption that $\chi \in \mathscr{R}$ and not $[-\pi, \pi)$. However, if χ is mapped to $[-\pi, \pi)$ in the implementation of the guidance law, the stability proof still holds.

where

$$\left.\frac{\partial \ddot{y}_e}{\partial y_e}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = \left.-\frac{\sqrt{V_g^2-\dot{y}_e^2}\Delta\left(\Delta^2 K_1 + K_1 {y_e}^2 - 2\, y_e\, \dot{y}_e\right)}{\left(\Delta^2+y_e^2\right)^2}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = -\frac{V_g K_1}{\Delta} \tag{10.21}$$

$$\begin{aligned}\left.\frac{\partial \ddot{y}_e}{\partial \dot{y}_e}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} &= -K_1 + \frac{K_1 \dot{y}_e\left(\operatorname{asin}\left(\frac{\dot{y}_e}{V_g}\right)+\operatorname{atan}\left(\frac{y_e}{\Delta}\right)\right)}{\sqrt{V_g^2-\dot{y}_e^2}} - \left.\frac{\Delta\left({V_g}^2-2\,\dot{y}_e^2\right)}{\sqrt{V_g^2-\dot{y}_e^2}\left(\Delta^2+{y_e}^2\right)}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} \\ &= \frac{-\Delta K_1 - V_g}{\Delta}.\end{aligned} \tag{10.22}$$

The resulting linear approximation

$$\ddot{\bar{y}}_e + V_g\left(\frac{K_1}{V_g}+\frac{1}{\Delta}\right)\dot{\bar{y}}_e + \frac{V_g K_1}{\Delta}\bar{y}_e = 0, \tag{10.23}$$

where $\bar{y}_e$ is the linear approximation of y_e around $y_e = 0$, combined with the general form of the mass-spring-damper

$$\ddot{\bar{y}}_e + 2\zeta\omega_0\dot{\bar{y}}_e + \omega_0^2\bar{y}_e = 0, \tag{10.24}$$

gives the tuning guidelines $\Delta = \frac{V_g K_1}{\omega_0^2}, K_1 = \omega_0\left(\zeta \pm \sqrt{\zeta^2-1}\right)$, by direct comparison. From these relations it is clear that $\zeta \geq 1$ to avoid complex values for Δ and K_1.

10.3.3 L_1 guidance similarity

To see that L_1 guidance can be considered a special case of eq. (10.16), a key observation is that another possible choice for $f(\tilde{\chi})$ in eq. (10.16) is $K_1(\sin(\tilde{\chi})+\epsilon\tilde{\chi})$, with $0 < \epsilon << 1$ chosen arbitrarily small[4] , resulting in the control law

$$\phi_d = \operatorname{atan}\left(\frac{V_g\cos\gamma\cos\theta}{g\cos(\chi-\psi)}\left(K_1(\sin(\tilde{\chi})+\epsilon\tilde{\chi}) - \frac{\Delta}{\Delta^2+y_e^2}\dot{y}_e + \dot{\chi}_p\right) - \sin\theta\tan(\chi-\psi)\right), \tag{10.25}$$

which renders the equilibrium $\tilde{\chi} = 0$ asymptotically stable in the region $\tilde{\chi} \in D = [-\pi, \pi]$, which can be considered global for all practical purposes, as an error angle, such as $\tilde{\chi}$, can be projected into D. It is noted Adapting this to enclosure-based LOS by applying eq. (10.8), while assuming $\gamma = \theta = \dot{\chi}_p = \epsilon = 0$ and $\chi = \psi$, and ignoring $\frac{\Delta}{\Delta^2+y_e^2}\dot{y}_e$, results in L_1 guidance, as introduced in section 10.2.2.2

$$\phi_d = \operatorname{atan}\left(\frac{V_g}{g}K_1\sin(\tilde{\chi})\right), \tag{10.26}$$

by choosing $K_1 = \frac{2V_g}{L_1}$. Choosing the control law eq. (10.25) instead of eq. (10.16), give identical results as the linerization analysis in section 10.3.2.

[4] ϵ is introduced only to render $\dot{V}_2(\tilde{\chi} = \pm\pi) < 0$, to include $\tilde{\chi} = \pi$ in D.

10.3.4 Stability of the interconnected system

Section 10.3.1 analyzed the GES course error dynamics eqs. (10.11) and (10.15) under the control law eq. (10.16), and referred to the LOS law eq. (10.4) by [51] which renders the cross-track error dynamics eq. (10.3) USGES. What remains to be shown is the uniform semiglobal exponential stability of the interconnected system eq. (10.4) with eq. (10.16). Following [107, 51], which rely on the stability analysis of cascaded nonlinear systems from [142, 108, 109], it remains to be showed that the cascade

$$\begin{aligned} \Sigma_1 &: \dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(t, \boldsymbol{x}_1) + \boldsymbol{g}(t, \boldsymbol{x})\boldsymbol{x}_2 \\ \Sigma_2 &: \dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(t, \boldsymbol{x}_2) \end{aligned} \tag{10.27}$$

where $\boldsymbol{x}_1 \in R^n, \boldsymbol{x}_2 \in R^m, \boldsymbol{x} \triangleq [\boldsymbol{x}_1, \boldsymbol{x}_2]^\intercal$, where $\boldsymbol{f}_1(t, \boldsymbol{x}_1)$ is continuously differentiable, and where $\boldsymbol{f}_2(t, \boldsymbol{x}_2), \boldsymbol{g}(t, \boldsymbol{x})$ are continuous and locally Lipschitz, fulfill the assumptions 3-5 in [109]. The presented guidance law eq. (10.4) with eq. (10.16) can be represented in cascade form as

$$\begin{aligned} \Sigma_1 &: \dot{y}_e = \boldsymbol{f}_1(t, y_e) + \boldsymbol{g}(t, \boldsymbol{x})\tilde{\chi} \\ \Sigma_2 &: \dot{\tilde{\chi}} = \boldsymbol{f}_2(t, \tilde{\chi}) \end{aligned} \tag{10.28}$$

with $\boldsymbol{x} \triangleq [y_e, \tilde{\chi}]^\intercal$ Inserting eq. (10.4) into eq. (10.3) to find $\boldsymbol{f}_1(t, y_e), \boldsymbol{g}(t, \boldsymbol{x}), \boldsymbol{f}(t, \tilde{\chi})$, gives

$$\dot{y}_e = V_g \sin\left(\chi - \chi_p\right) \tag{10.29}$$

$$= V_g \sin\left(\chi_{\text{LOS}} - \tilde{\chi}\right) \tag{10.30}$$

$$= V_g \sin\left(\chi_{\text{LOS}}\right)\cos\left(\tilde{\chi}\right) - V_g \cos\left(\chi_{\text{LOS}}\right)\sin\left(\tilde{\chi}\right) \tag{10.31}$$

$$= \frac{-V_g y_e}{\sqrt{\Delta^2 + y_e^2}}\cos\left(\tilde{\chi}\right) - \frac{V_g \Delta}{\sqrt{\Delta^2 + y_e^2}}\sin\left(\tilde{\chi}\right) \tag{10.32}$$

$$= \frac{-V_g y_e}{\sqrt{\Delta^2 + y_e^2}} + \frac{V_g}{\sqrt{\Delta^2 + y_e^2}}\left(y_e\left(1 - \cos\left(\tilde{\chi}\right)\right) + \Delta\sin\left(\tilde{\chi}\right)\right) \tag{10.33}$$

and further insertion of eq. (10.16) into eq. (10.11) by assuming $\phi = \phi_d$

$$\dot{\tilde{\chi}} = -\frac{\Delta}{\Delta^2 + y_e^2}\dot{y}_e + \dot{\chi}_p - \frac{g\cos\left(\chi - \psi\right)}{V_g \cos\gamma\cos\theta}\left(\tan\phi + \sin\theta\tan\left(\chi - \psi\right)\right) \tag{10.34}$$

$$= -f\left(\tilde{\chi}\right)\tilde{\chi}, \tag{10.35}$$

results in

$$\boldsymbol{f}_1(t, y_e) = \frac{-V_g y_e}{\sqrt{\Delta^2 + y_e^2}} \tag{10.36}$$

$$\boldsymbol{g}(t, \boldsymbol{x}) = \frac{V_g}{\sqrt{\Delta^2 + y_e^2}}\left(\frac{1 - \cos\left(\tilde{\chi}\right)}{\tilde{\chi}} y_e + \Delta\frac{\sin\left(\tilde{\chi}\right)}{\tilde{\chi}}\right) \tag{10.37}$$

$$\boldsymbol{f}_2(t, \tilde{\chi}) = -f\left(\tilde{\chi}\right)\tilde{\chi}, \tag{10.38}$$

which are all continuously differentiable, assuming $\Delta > 0$, which also implies that $\boldsymbol{f}_2(t, \boldsymbol{x}_2), \boldsymbol{g}(t, \boldsymbol{x})$ are continuous and locally Lipschitz. Having defined the functions in eq. (10.28), the stability of the interconnected system can be proven.

Theorem 10.1. *The interconnected system eq.* (10.28) *with eqs.* (10.36) *to* (10.38) *has a uniformly semiglobal exponentially stable equilibrium point at* $\boldsymbol{x} = \boldsymbol{0}$ *if the desired roll and course are given by eq.* (10.16) *and eq.* (10.4)*, respectively.*

Proof. This follows from satisfying [109, Assumptions 3-5].

Assumption 3

From [51], re-cited in section 10.2.2.1, it is known that the equilibrium $y_e = 0$ is USGES when the course angle is perfectly tracked, under the Lyapunov function candidate $V(y_e) = \frac{1}{2} y_e^2$. It can then be shown that

$$\left\| \frac{\partial V(t, y_e)}{\partial y_e} \right\| \|y_e\| = \|y_e\| \|y_e\| \leq c_1 V(t, y_e) \quad \forall c_1 \geq 2, \|y_e\| > 0, \tag{10.39}$$

and furthermore that

$$\left\| \frac{\partial V(t, y_e)}{\partial y_e} \right\| = \|y_e\| \leq c_2 \quad \forall \|y_e\| \leq \eta = c_2, \tag{10.40}$$

such that the conditions for the assumption are fulfilled.

Assumption 4

The interconnection term $\boldsymbol{g}(t, \boldsymbol{x})$ can be shown to be bounded

$$\|\boldsymbol{g}(t, \boldsymbol{x})\| = \left\| \frac{V_g}{\sqrt{\Delta^2 + y_e^2}} \left(\frac{1 - \cos(\tilde{\chi})}{\tilde{\chi}} y_e + \Delta \frac{\sin(\tilde{\chi})}{\tilde{\chi}} \right) \right\| \tag{10.41}$$

$$\leq \left\| \frac{V_g}{\sqrt{\Delta^2 + y_e^2}} \left(\frac{1 - \cos(\tilde{\chi})}{\tilde{\chi}} y_e \right) \right\| + \left\| \frac{V_g}{\sqrt{\Delta^2 + y_e^2}} \left(\Delta \frac{\sin(\tilde{\chi})}{\tilde{\chi}} \right) \right\| \tag{10.42}$$

$$\leq \frac{V_g}{\Delta} \left\| \frac{1 - \cos(\tilde{\chi})}{\tilde{\chi}} y_e \right\| + V_g \left\| \frac{\sin(\tilde{\chi})}{\tilde{\chi}} \right\| \tag{10.43}$$

$$\leq \frac{V_g}{\Delta} \left\| \frac{1 - \cos(\tilde{\chi})}{\tilde{\chi}} \right\| \|y_e\| + V_g \left\| \frac{\sin(\tilde{\chi})}{\tilde{\chi}} \right\| \tag{10.44}$$

$$\leq \frac{V_g}{\Delta} \frac{1 - \cos(\|\tilde{\chi}\|)}{\|\tilde{\chi}\|} \|y_e\| + V_g \left(\frac{\sin(\|\tilde{\chi}\|)}{\|\tilde{\chi}\|} + 0.5 \right), \tag{10.45}$$

by the triangle inequality [93]. By defining $\theta_1(\|\tilde{\chi}\|) = \frac{V_g}{\Delta} \frac{1 - \cos(\|\tilde{\chi}\|)}{\|\tilde{\chi}\|}$ and $\theta_2(\|\tilde{\chi}\|) = V_g \left(\frac{\sin(\|\tilde{\chi}\|)}{\|\tilde{\chi}\|} + 0.5 \right)$ the assumption is satisfied, as $\theta_1(\|\tilde{\chi}\|)$ and $\theta_2(\|\tilde{\chi}\|)$ are continuous.

Assumption 5

The existence of a class $\mathcal{K}$-function $\alpha(\cdot)$ such that for all $t_0 \geq 0$, the trajectories of the system satisfy

$$\int_0^\infty \|\tilde{\chi}(t; t_0, \tilde{\chi}(t_0))\| \, dt \leq \alpha(\|\tilde{\chi}(t_0)\|) \tag{10.46}$$

is guaranteed by the GES property of Σ_2. Since $\|\tilde{\chi}(t)\| \leq \lambda \|\tilde{\chi}(0)\| e^{-2(t-t_0)}$, chosing $\alpha(\|\boldsymbol{x}_2(t_0)\|) = \lambda \|\tilde{\chi}(t_0)\|$ satisfies the assumption.

As all three assumptions are satisfied, while Σ_1 is GES and Σ_2 is USGES, the interconnected system has a USGES equilibrium in $\boldsymbol{x} = \boldsymbol{0}$. □

10.4 Validation

To assess the performance of the presented guidance controller framework, it is implemented in the ArduPlane autopilot software, to simplify the comparison with its state-of-the-art guidance controller. The Ardupilot L1 controller, hereafter referred to as APL1, is based on the L_1 guidance controller by Park, Deyst, and How, introduced in section 10.2.2.2. It should be noted that the implementation of APL1 is not completely equivalent to [144]. The main differences are:

- APL1 adds an integral term on χ_{LOS}. This is set to zero, to make the comparison fair.
- APL1 explicitly calculates the desired roll angle, hereby compensating for the pitch of the aircraft, as mentioned in section 10.3.
- APL1 implements various changes to comply with practical aspects, such as logic to prevent indecision when the aircraft is moving away from a new target and tries to turn around, and handle the corner cases of being in front of A or after B, when flying the line segment AB. All these changes have also been applied to the presented controller.

Depending on the choice of $f(\tilde{\chi})$ in eq. (10.16), a myriad of different controllers can be made. Also, it is of interest to investigate the effect of the $\sin(\theta)\tan(\chi - \psi)$-term in eq. (10.16), as this is not present in APL1. The following controllers are compared:

APL1 The standard Ardupilot implementation of [144], as of ArduPlane 3.9.9.

APL1 damp=0.75 identidal to APL1, but with a damping factor of 0.75 as opposed to 1.0.

AP-like eq. (10.25).

LOS1 line-of-sight based: eq. (10.16), with $f(\chi) = K_1 \tilde{\chi}$.

LOS2 identical to LOS1, but neglecting the $\sin\theta \tan(\chi - \psi)$-term.

LOS3 eq. (10.16), with $f(\chi) = K_1 \tilde{\chi} + K_3 \tilde{\chi}^3$, with $K_3 = 1$.

Table 10.1: ArduPlane parameters in the simulation

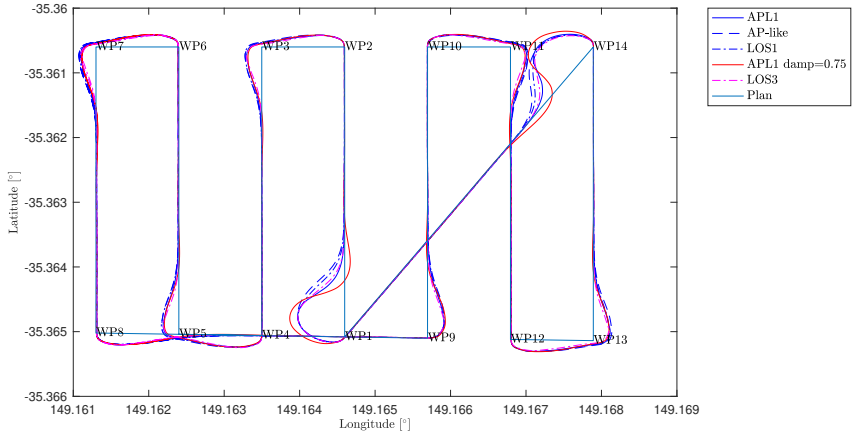
WP_RADIUS	10
NAVL1_PERIOD	8
NAVL1_DAMPING	1

In all the controllers $\dot{\chi}_p$ is zero, as the path derivative is not available in Ardupilot, and since the paths the controllers are tested on are piece-wise linear. For a fair comparison, all the controllers are tested with the same tuning, in terms of damping factor and bandwidth. Since the presented guidance laws only are valid for $\zeta \geq 1$, the damping factor is set to 1.0 for all controllers (except for the **APL1 damp=0.75** version).

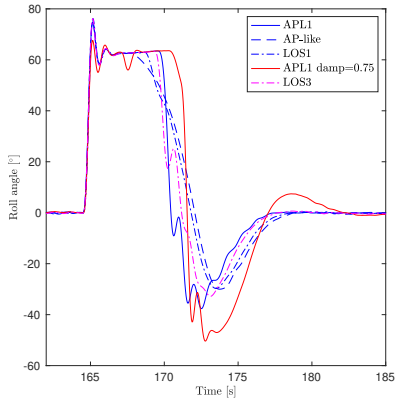
10.4.1 Simulations without wind

The controllers are compared using the standard Ardupilot software-in-the-loop (SITL) framework, that relies on the Rascal 110 UAV [83, 10] model in the JSBSim simulator [16], a 2.8 m wingspan, 5.9 kg UAV with a cruise speed of about 22 m/s. In addition to the extra parameter that have been added to select the different controller types, some Ardupilot parameters, seen in table 10.1 have been changed to ensure a fair comparison and to highlight the differences in the controllers. The main difference is that the controllers are tuned to be faster, by setting a low *NAVL1_PERIOD*, and forced to start the turn close to the waypoint, by setting *WP_RADIUS* small. This will force aggressive maneuvers, intended to emphasize the differences. Further, the saturation of χ_{LOS} to 45° in the ArduPlane implementation has been removed, while the limit on $\tilde{\chi}$ has been extended to ± 90°, such that the behavior of the controllers in these extremities can be assessed. It should be noted that these values and modifications are not necessarily optimal or safe for an experimental verification. The path used in the simulation study is one of the default paths provided by Ardupilot; the CMAC-grid, a lawnmower-pattern over the CMAC runway.

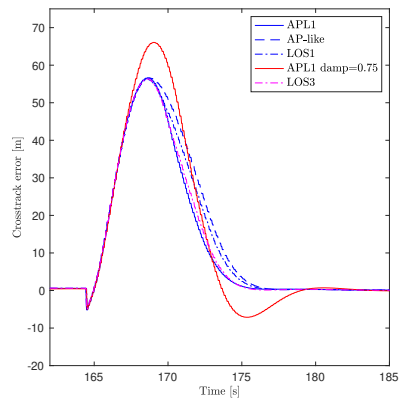
The results are plotted in fig. 10.2. From fig. 10.2a it is apparent that the turns at WP1 and WP14 are of most interest. In both turns the version of APL1 with reduced damping moves closer to the path faster, but clearly overshoots, as expected. From fig. 10.2c it is clear that APL1 and LOS3 have similar performance when considering convergence and transient behaviour. LOS1 and AP-like initially behave similarly to the other controllers, but converge slower, the AP-like controller being the slowest. The roll response, fig. 10.2b, overshoot the limitation on the desired roll of 65° for all the controllers, albeit only slightly for APL1. This is caused by the fast increase in the desired roll, and shows that the assumption $\phi = \phi_d$ is not always true. One way to improve the overshoot could be to ensure a smooth path that considers the kinematic and dynamic capabilities of the aircraft, avoiding the steps in the piece-wise linear path used in these simulations. However, this would also lead to slower response.



(a) 2D position plot



(b) Roll angle, at WP1



(c) Cross-track error, at WP1

Figure 10.2: Simulations

10.4.2 Simulations with wind

With a non-zero, steady wind speed, $\chi \neq \psi$, so the $\cos(\chi - \psi)$ term in eq. (10.16) comes into play. For this reason, it is of interest to compare the effect of these terms, as they are often neglected. The results from a similar simulation scenario as in section 10.4.1, but with steady wind of 15 m/s from the south, are plotted in fig. 10.3. The wind direction was chosen to make the response in the turns extreme. It is clear that the controllers have qualitatively similar performance for the majority of the time, but there are some notable differences. From considering the turn at WP14 in fig. 10.3a it is clear that LOS2 reaches the next line segment closer to the start of the segment than LOS1, while LOS1 reaches the line faster. LOS1 and LOS2 have

Table 10.2: Lateral ArduPilot tuning for the Skywalker X8

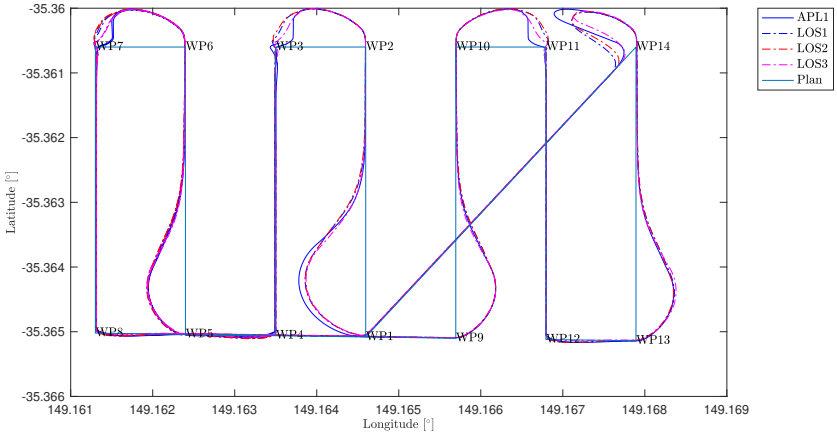
NAVL1_PERIOD	15	RLL2SRV_P	2
NAVL1_DAMPING	1.0	RLL2SRV_D	0.07
NAVL1_XTRACK_I	0	RLL2SRV_I	0.15
NAVL1_LIM_BANK	0	RLL2SRV_RMAX	60
WP_RADIUS	50	RLL2SRV_IMAX	3000
RLL2SRV_TCONST	0.2	RLL2SRV_FF	0

very similar performance for turn 1. For the cross-track error in WP1, see fig. 10.3c, it is also clear that LOS1, LOS3 and APL1 reach the next linear segment of the path almost at the same time. However, APL1 clearly has the largest maximum cross-track error, about 7 m or 10 % more than the other controllers, and also overshoots slightly after reaching the line. LOS1 and LOS2 have similar transients, but LOS2 converges slightly slower to the next linear segment. The corresponding roll responses of the aircraft are plotted in fig. 10.3b, in which the aforementioned overshoot is still present.

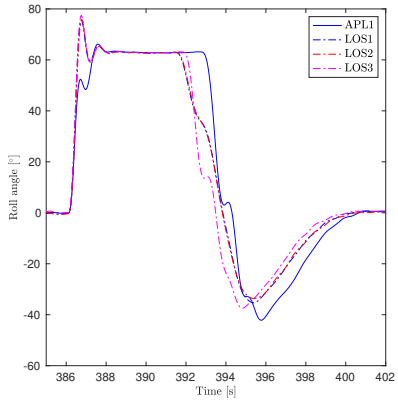
It should be noted that in this experiment the removal of the saturation of χ_{LOS} is apparent; using standard ArduPilot would have caused the aircraft to approach the path at an angle of 45° after WP14, which seems like a more kinematically tractable approach, safer for physical experiments, despite the later convergence to the line segment.

10.4.3 Experimental verification

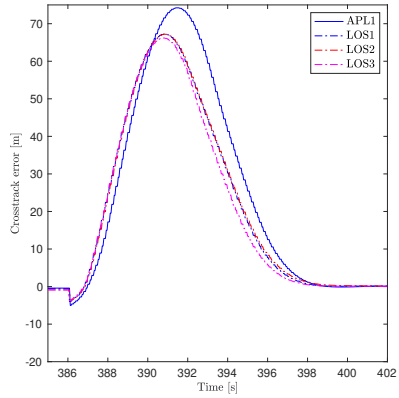
To show that the controllers work in a real-world scenario, a physical experiment was conducted at the Udduvoll airfield near Trondheim, Norway, using a Skywalker X8 fixed-wing UAV, as presented in chapter 4. The comparison from the simulations were not repeated in the experimental verification, as the potentially subtle differences in the results would be highly affected by the stochastic nature of the wind. Instead, only the AP-like controller, whose relevant tuning parameters are found in table 10.2, was selected for verification. The resulting path is shown in fig. 10.4a, whereas fig. 10.4b and fig. 10.4c show the corresponding roll and cross-track error, respectively. It should be noted that the experiments were made in winds of about 10 m/s from the south-south-east, which is considerable for an aircraft with a cruise speed of 18 m/s. Despite the challenging conditions, it is clear that the presented controller is able to track the desired path.



(a) 2D position plot

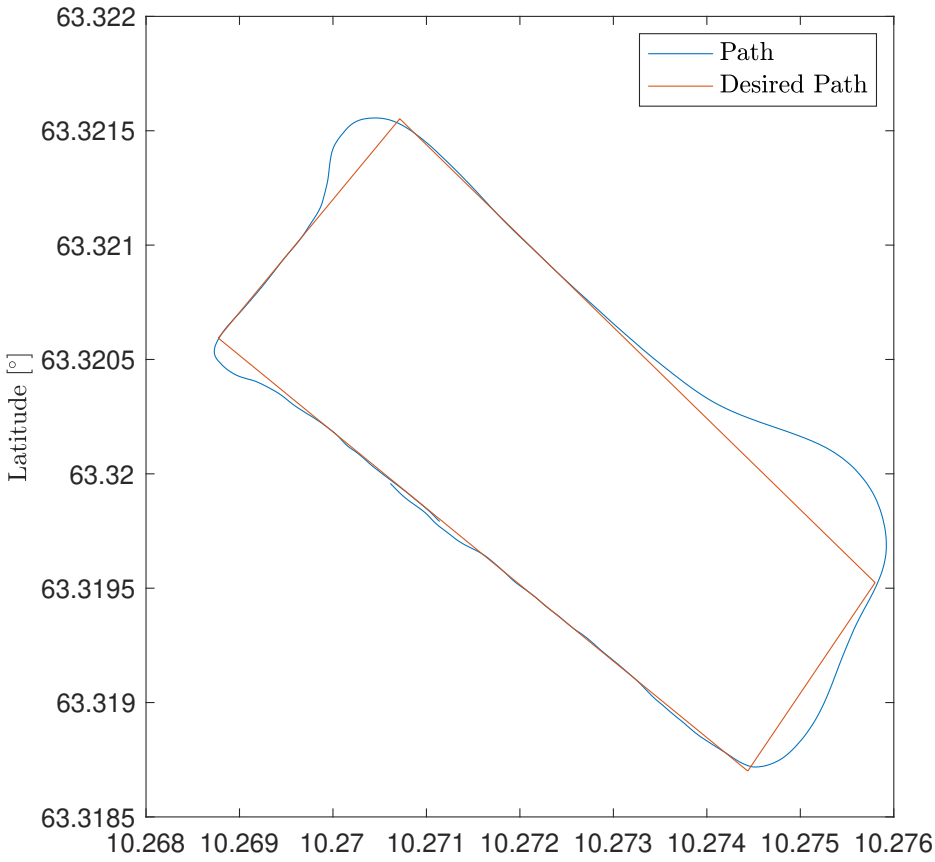


(b) Roll angle, at WP1

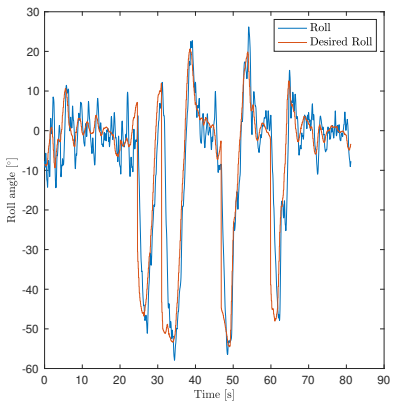


(c) Cross-track error, at WP1

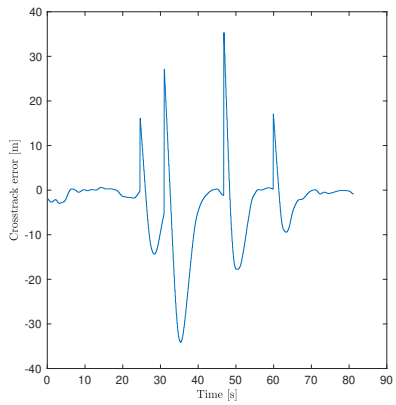
Figure 10.3: Simulations with wind



(a) 2D position plot



(b) Roll angle



(c) Cross-track error

Figure 10.4: Experimental verification

10.5 Discussion

When considering the path following performance of the evaluated controllers, it seems like the presented LOS-based controllers have a more damped approach to the path, which can be attributed to the additional damping term $-\frac{\Delta}{\Delta^2+y_e^2}\dot{y}_e$ in eq. (10.16). An implication of the increased damping in the LOS-based controllers can also be seen from the tuning guidelines of section 10.3.2, where it is clear that $\zeta \geq 1$ to avoid complex values for K_1 and Δ. This restriction can be seen as a disadvantage with the presented guidance laws.

One of the main difference between the presented guidance law and the L_1 guidance law, is the consideration of steady wind effects, resulting in a dependency on more measurements, such as the heading ψ through the term $\cos(\chi - \psi)$, making the presented guidance law less robust to poor measurements. This is particularly relevant for heading, which is more difficult to estimate with good accuracy in a standard UAV payload, using a magnetometer, compared to estimating course χ, due to magnetic anomalies and electromagnetic interference (EMI). This is manifested in eqs. (10.34) and (10.38), where a small error in any of the measurements will lead to errors in the inversion of the kinematic model, rendering the course error dynamics eq. (10.38) dependent on the cross-track error, thus violating an assumption of the stability proof.

Other possible future extension would be to include integral effect, to overcome sensor misalignments and model mismatches. Inclusion of integral effect leads to an interesting comparison of a course-error based integral, like in the ArduPlane implementation of APL1, or an integral based on cross-track error, as in [106]. Another improvement that will improve performance is to include the path derivative feed-forward $\dot{\chi}_p$ in the simulations, which is arguably more intuitive than the feed-forward presented in [144] that resembles a translational offset from a linear nominal path.

10.6 Conclusion

The path following problem for fixed-wing UAVs is studied in a line-of-sight framework, which is known to have uniform global exponential stability. This is achieved through the use of the coordinated-turn relation, formulated in course angles, which has been derived to emphasize what assumptions are made. The analysis of the interconnected system consisting of the line-of-sight guidance and the coordinated-turn based controller show that the equilibrium $[\tilde{\chi}, y_e]^\intercal = \mathbf{0}$ is uniformly semiglobally exponentially stable, through a Lyapunov-based analysis of the cascade. The resulting controllers show similarities to the state-of-the-art L_1 guidance law, in both structure and performance in a simulation study. But there are some differences, that have been discussed. Most notably is an additional damping term, and the improved performance in steady wind, where the maximum cross-track error of the presented controllers is about 10 % lower.

Chapter 11

Deep stall landing

This chapter concerns the use of constrained non-linear model predictive control for high-precision deep stall landing of a fixed-wing UAV, through simulations in six degrees of freedom, where the UAV is controlled to obtain an end velocity that is considerably lower than the cruise velocity of the UAV. The controller handles a variation of side wind and time varying turbulence, but is sensitive to wind gusts. The chapter is based on the publication

- [123] 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 SciTech*, 2016

11.1 Introduction

To recover a small UAV in a small space without a runway, belly landing on a soft surface or an arrest system like a recovery net can be used. It is desirable to ease the impact by minimizing the speed at which the UAV meets the landing target. One method for doing this is to land the UAV in a deep stall. In this case, the UAVs angle of attack needs to go beyond the stall angle, where the drag coefficient of the UAV increases and the lift coefficient decreases. This makes the UAV loose altitude while at the same time losing speed in the horizontal direction. Since control authority is reduced, it is difficult to control a UAV in a deep stall, and side winds or turbulence might cause the UAV to spin or lose control. It is also important to be able to accurately guide the UAV at the same time as it is being decelerated, to be able to accurately hit the landing target.

In [170], a study of the flight characteristics of a manned deep stall is presented. Using the Schweizer SGS 1-36 Sprite sailplane, the pilot smoothly entered a deep stall and let the angle of attack stabilize between 30° and 70°. After falling down rapidly, at a speed of around 20 m/s, the pilot lowered the angle of attack to around

$25° - 30°$, resulting in an unstable flight, before increasing the speed and returning to a safe flight. It was registered that while in the deep-stall phase, though flying stable, a lateral-directional oscillation could be noticed. This was explained as the pilot's lack of lateral-directional control when the angles of attack went beyond 60°. A thorough flight characteristic with both predicted and measured data is appended in the article. Also the article presented in [195] describes experiences achieved during piloted flight in deep stall in a simulator. Their conclusions focus on how a deep stall can be avoided, as it normally causes the airplane to loose altitude and it may take long time to regain control over the plane.

Experimental results for a deep-stall landing in a UAV can be found in [183]. The article presents an analysis with longitudinal dynamics of a deep-stall landing, including remarks on how the lift and drag coefficients appear in a deep stall. It describes how the speed is decreased in the deep stall as a consequence of the drag and lift coefficients in this angle of attack, and that a high touch down speed might limit this landing method to small UAVs. In both the simulations and the experiments, an unspecified plane with a horizontal tail plane is used. The article explains that "the only mechanism to get the plane into deep stall is quickly tilt of the horizontal tail plane ". The article concludes that deep-stall landing can decrease the speed while maintaining a deep path angle, which was confirmed through trim analysis, flight simulations and flight data from a flight test. In [31], the post stall landing manoeuvre with a tail plane UAV is modeled and simulated. The longitudinal route of the UAV is divided into an initial relatively steep conventional landing approach where the altitude is reduced, before an extended flare phase causes the UAV to gain altitude, and thus potential energy, reducing the speed, before it stalls to further reduce the speed and then land in post-stall. The article investigates the effect of deflecting the elevator over time periods of various lengths, and graphs are shown with the simulated trajectory and the velocity of the UAV for the different cases.

In [131], a post-stall landing inspired by bird perching is described. A model of a simple UAV is formulated and linear quadratic regulator (LQR) trees were used to implement the controller. Non-linear model predictive control (NMPC) was also considered as a candidate, but LQR was found to be more efficient and have better theoretical performance guarantees . The theory was supported by experimental flight test results. In [152], an analysis of a safe deep-stall landing is performed. The article only considers longitudinal dynamics, and the model uses a delta wing vehicle by AeroSpy Sence & Avoid Technology. Simulations and experimental deep-stall landing manoeuvres are performed and discussed in the article, and a hybrid system that is able to safely abort a deep-stall landing under hazardous conditions is presented.

In this chapter, an NMPC is employed on the 6-DOF dynamics of a fixed-wing UAV, to guide it in a deep stall while at the same time landing in a given location. The model for the fixed-wing UAV is described in [14, p. 156] and consists of north-east-down

positions relative to the inertial frame, body velocity, quaternions for representation of attitude, and angular velocity in the body frame. The UAV used in this chapter is controlled by rudder, elevator, aileron, and throttle. The nonlinear problem (NLP) of the NMPC consists of an objective function, which should be minimized over a time horizon of N discrete time intervals, and constraints.

This work is a continuation of [122], where the NMPC guided deep-stall landing of a longitudinal model of a fixed-wing UAV was studied. In this chapter, the lateral dynamics are also included, considering the side wind influence on the UAV. Through simulations done in Python with the open-source software package CasADi [8], it is investigated how the landing precision and minimal speed is influenced by wind gusts and crosswinds.

11.2 Model

The UAV model in this chapter is the 6-DOF dynamics of a fixed-wing UAV in quaternion representation[14, 50]. The specific UAV model used is an Aerosonde UAV[14].

$$\begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{p_d} \end{bmatrix} = R_{nb}(\boldsymbol{q}) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{11.1}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{M} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \tag{11.2}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2} \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix}, \tag{11.3}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{I_y} m \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix}, \tag{11.4}$$

The variables p_n, p_e, p_d are the inertial north, east and down positions of the UAV. M is the mass of the vehicle, I_y is the moment of inertia about the body frame y-axis, while Γ_i are functions of the moments and products of inertia, see [14].

The forces F_x, F_y, F_z and moments l, m, n that act upon the UAV are the aerodynamic, the gravitational, and the propulsion forces and moments. This is seen in eq. (11.5) and section 11.2. The aerodynamic force is highly dependent on the UAVs relative speed, making changes in the wind important to the model.

$$\mathbf{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \mathbf{F}_g + \mathbf{F}_a + \mathbf{F}_p, \tag{11.5}$$

$$\mathbf{F_g} = R_{bn}(\boldsymbol{q}) \begin{bmatrix} 0 \\ 0 \\ Mg \end{bmatrix}, \tag{11.6}$$

$$\mathbf{F_a} = \frac{1}{2}\rho V_a S \begin{bmatrix} C_X(\alpha) + C_{X_q}(\alpha)\frac{c}{2V_a}q + C_{X_{\delta_e}}(\alpha)\delta_e \\ C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\ C_Z(\alpha) + C_{Z_q}(\alpha)\frac{c}{2V_a}q + C_{Z_{\delta_e}}(\alpha)\delta_e \end{bmatrix}, \tag{11.7}$$

$$\mathbf{F_p} = \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} \begin{bmatrix} (k_{\text{motor}}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix}, \tag{11.8}$$

$$\mathbf{m} = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \begin{bmatrix} -k_{T_p}(k_\omega \delta_t)^2 \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\rho S \begin{bmatrix} b[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r] \\ c[C_m(\alpha) + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e] \\ b[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r] \end{bmatrix}. \tag{11.9}$$

where g is the acceleration of gravity, S_{prop} is the area swept by the propeller and C_{prop} is the aerodynamic propeller coefficient. k_{motor} is the motor efficiency coefficient, k_{T_p} is the propeller torque coefficient and k_ω is the propeller speed coefficient.

All constants for the UAV model and the specific values on the ones used in this chapter can be found in [14, p. 276]. The relationship between body-axis coefficients, C_{X_*} and C_{Z_*}, and stability-axis coefficients C_{D_*} and C_{L_*} is given by

$$\begin{bmatrix} C_{X_*} \\ C_{Z_*} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -C_{D_*} \\ -C_{L_*} \end{bmatrix}. \tag{11.10}$$

In this chapter, wind is modeled as a sum of a steady component in the inertial frame and a gust component in the body frame. While the steady wind components w_{n_s}, w_{e_s} and w_{d_s} are modeled as constant, the gust component is modeled as white noise

$w_i(s)$ that is shaped according to the Dryden wind model[124]:

$$u_{w_g}(s) = \quad H_u(s)w_1(s) = \frac{\sigma_u\sqrt{\frac{2L_u}{\pi V_a}}}{\frac{L_u}{V_a}s+1}w_1(s) \tag{11.11}$$

$$v_{w_g}(s) = \quad H_v(s)w_2(s) = \sigma_v\sqrt{\frac{2L_v}{\pi V_a}}\frac{\frac{\sqrt{3}L_v}{V_a}s+1}{\left(\frac{L_v}{V_a}s+1\right)^2}w_2(s) \tag{11.12}$$

$$w_{w_g}(s) = \quad H_w(s)w_3(s) = \sigma_w\sqrt{\frac{2L_w}{\pi V_a}}\frac{\frac{\sqrt{3}L_w}{V_a}s+1}{\left(\frac{L_w}{V_a}s+1\right)^2}w_3(s) \tag{11.13}$$

Here, the turbulence intensities and turbulence scale lengths are given by

$$\sigma_w = 0.328W_{20} \tag{11.14}$$

$$\sigma_v = \frac{\sigma_w}{(0.177+0.00270\cdot h)^{0.4}} \tag{11.15}$$

$$\sigma_u = \frac{\sigma_w}{(0.177+0.00270\cdot h)^{0.4}} \tag{11.16}$$

$$L_u = \frac{3.28h}{(0.177+0.00270\cdot h)^{1.2}} \tag{11.17}$$

$$L_v = \frac{3.28h}{(0.177+0.00270\cdot h)^{1.2}} \tag{11.18}$$

$$L_w = 3.28h \tag{11.19}$$

where W_{20} is the steady wind speed at 20 feet, and where metric units have been used. All this is in agreement with [124], however to avoid singularities in the simulation as $h \to 0$ the height parameter h in the Dryden model is set equal to 50 meters.

The non-linear lift and drag coefficients $C_D(\alpha)$ and $C_L(\alpha)$, and the pitch moment $C_m(\alpha)$, are given by one 6^{th} and two 12^{th} degree polynomial equations. The UAV model in use in this chapter is the one of an Aerosonde, but a NACA 4415 wing profile with an aspect ratio of 15.24 has been chosen to represent its lift, drag and pitch moment coefficients. Although the wing profile of an Aerosonde differs from a NACA 4415 profile, it is still used as it is more realistic than the aerodynamic models given in [14], since the data is given for angles of attack above the stall angle. The lift, drag and pitch moment coefficient characteristics as functions of the angle of attack α used are the ones from Figures A.5 (a), (b) and (c) in [141]. They are only valid when $-10° \leq \alpha \leq 110°$, and are results from wind tunnel tests on airfoils. The relationships between the angle of attack, α, and the polynomial functions for the drag, lift and pitch moment coefficients are shown in fig. 11.1. In addition to these angle of attack dependent coefficients, there is an effect on the drag force, lift force,

and pitching moment caused by the pitch rate q and the elevator deflection δ_e. As can be seen in fig. 11.1, the UAV has a stall angle at around 18° ($C_L = 1.42$) and a secondary lift peak around 50°. The drag coefficient is at its maximum around 90°. The pitch moment coefficient will be zero depending on the elevator deflection angle, but is shown in the figure with zero elevator deflection angle.

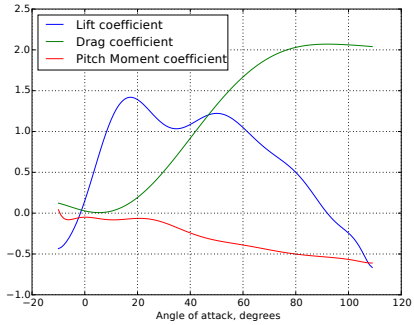


Figure 11.1: $C_L(\alpha)$, $C_D(\alpha)$, and $C_m(\alpha)$

11.3 Theory

11.3.1 Deep Stall Landing Concepts

When the UAV's angle of attack increases, the boundary layer of the air flow eventually separates from the airfoil of the wing and creates a turbulent wake behind the wing. This causes a massive reduction in the lift on the wing and an increase of drag. The angle of attack when this occurs, is called the *stall angle*. It can be recognized in a lift coefficient characteristic as the angle of attack that gives the maximum lift [91, 148, 7, 17, 14, 1]. The term *deep stall* refers to angles of attack higher than the stall angle, from which it might be difficult to recover [195]. [148] defines the deep stall as the condition where the angle of attack gives a stable trim point beyond the stall angle. There are many slightly different definitions of the trim point, as the condition where the sum of moments is zero [29], or where there is zero moment *and* zero resultant forces around the center of gravity [148]. In this chapter, the definition by [29] is used. It argues that as an aerial vehicle normally is naturally lateral-directionally stable because of symmetry, longitudinal trim is usually enough to find. The conditions for stable longitudinal trim are thus:

$$C_m = 0, \quad \frac{dC_m}{d\alpha} \leq 0, \tag{11.20}$$

The stability condition ensures that the aircraft will return to the trim position after a minor disturbance [148]. The pitch moment, given by section 11.2, contains one

component given by the elevator deflection, one by the angle of attack, and one by the pitch rate. In trim, the pitch rate will be zero as a consequence of no moment around the body y-axis of the UAV. With zero elevator deflection, the graph for the pitch moment is as shown in fig. 11.1. When the elevator deflection is non-zero, the graph will be vertically shifted, which means that with a positive elevator deflection component of the moment, the trim condition where the pitch moment is zero will be moved to a higher angle of attack. With a non-linear pitch moment coefficient, there can be multiple trim points. When the angle of attack is higher than the stall angle, and in addition the gradient of the moment coefficient is negative, this gives a stable trim point for a deep stall.

11.3.2 Model Predictive Control

Instead of using linear controllers like a PID controller or an LQR, a non-linear model predictive controller (NMPC) is used to control the UAV in these simulations. This is chosen both because it gives an optimal solution, and because the non-linearities and constraints in the model demands a more flexible controller than linear controllers. NMPC is a control method that tries to optimize an objective function constrained by the system's dynamics, as well as physical and operational constraints. This is done by optimizing a sequence of control actions over a time horizon based on the predicted state of the system, and then applying the first control action to the system. Then the optimization is repeated, using the measurement of the response of the system as an initial state for the optimization. The prediction horizon is shifted one time step further, and the process is warm started, using the time shifted optimal values from the previous optimization as an initial guess for the next optimization. The 6 degrees of freedom (6-DOF) dynamics of a UAV operating at high angle of attack is highly non-linear, which benefits from a non-linear MPC to control it. While a linear MPC defines an optimization problem that can be solved with convex quadratic programming, the NMPC is usually solved by transforming the control problem into a non-linear program (NLP) and then optimize it. Although an NMPC is a time and resource consuming control method, it gives advantages as it handles non-linearities and constraints well [102].

The NMPC problem contains an objective function, which should be minimized, and constraints, which define the area wherein the minimum should be found. Equation (11.21) show the objective function which is a function of the state and control trajectories $\mathbf{x}$ and $\mathbf{u}$. Equations (11.22) and (11.23) shows the constraint for the initial state of the system, where the prediction horizon starts, and the differential equations representing the model. Equations (11.24) and (11.25) show the constraints on the control variables and on the states.

$$\min_{\mathbf{x}(\cdot),\mathbf{u}(\cdot)} \int_{t_0}^{t_0+T} F(\mathbf{x}(t),\mathbf{u}(t))\mathrm{d}t + E(\mathbf{x}(T)) \tag{11.21}$$

subject to

$$\dot{\mathbf{x}}(t_0) = \mathbf{x}_0, \tag{11.22}$$

$$\mathbf{x}(t) - f(\mathbf{x}(t),\mathbf{u}(t)) = 0, \tag{11.23}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max}, \tag{11.24}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}(t) \leq \mathbf{x}_{\max}, \forall t \in [t_0, t_0+T]. \tag{11.25}$$

In the implementation, NMPC uses direct multiple shooting to discretize the model [39]. This way of creating a non-linear program first divides the control horizon into N discrete control intervals, and then solves one optimization problem for each control interval simultaneously.

11.4 Problem Definition

The aim of the research described in this chapter is to develop an algorithm to deep-stall land a small, fixed wing UAV in a specific location. To do this, a double objective is formulated: The UAV should track the pre-decided landing target precisely along a given flight path angle and course angle, and land there in a deep stall. This is done in two steps, as described in algorithm 1 and shown in figs. 11.2a and 11.2b: When the landing is initiated, the UAV follows the landing target by tracking its lateral line of sight (LOS) angle, but keeping the same altitude. When the UAV's longitudinal LOS angle to the landing target is smaller (or greater in magnitude, since the landing path angle is negative) than a given limit, γ_0, it starts descending with a path angle that follows the longitudinal LOS angle. While descending, the UAV also tracks the desired deep-stall angle of attack.

Algorithm 1 Landing procedure

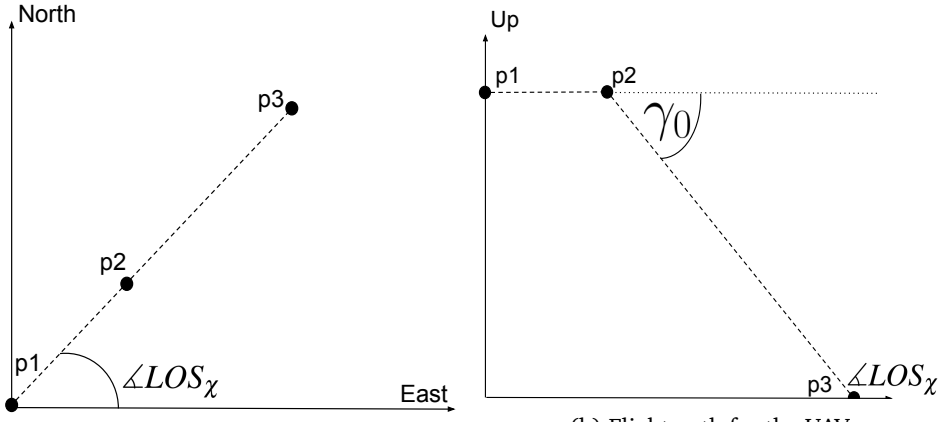
Require:
  while $\measuredangle LOS_\gamma > \gamma_0$ **do**
    $\gamma^* = 0$
    $\chi^* = \measuredangle LOS_\chi$
  end while
  while $\measuredangle LOS_\gamma \leq \gamma_0$ **do**
    $\gamma^* = \measuredangle LOS_\gamma$
    $\chi^* = \measuredangle LOS_\chi$
    $\alpha^* = \measuredangle DeepStall$
  end while

The LOS angles from algorithm 1 are given by:

$$\measuredangle LOS_\gamma = -\arctan \frac{-p_d - \text{targ}_d}{\sqrt{(p_n - \text{targ}_n)^2 + (p_e - \text{targ}_e)^2}} \tag{11.26}$$

$$\measuredangle LOS_\chi = atan2(\text{targ}_e - p_e, \text{targ}_n - p_n) \tag{11.27}$$

where $atan2$ is a function that calculates an angle between the two arguments in their correct quadrants.



(a) Horizontal course for the UAV

(b) Flight path for the UAV

Figure 11.2: Proposed trajectories for the UAV

To descend in deep stall to the landing target, the difference between the path angle and the longitudinal LOS angle is minimized while the course angle is still kept at the lateral LOS angle to the landing target. To keep a deep stall, the pitch moment is set to track a zero reference at the same time as the speed is minimized. The reason why the zero pitch moment is not enough, is because in a non-linear pitch moment coefficient, there might be several zero points. We want to track the one that has the highest angle of attack, both because we want to find the deep-stall trim angle of attack, which is higher than the stall angle and therefore has higher drag and lower speed, and because we want the gradient of the pitch moment coefficient to be negative, according to the theory about the deep stall. In addition to achieving zero pitch moment, the roll and yaw moment are minimized to prevent oscillations, and the change in control actions is minimized. The optimization problem can be defined as eqs. (11.28) and (11.30), with penalty parameters from eq. (11.31).

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \sum_{k=0}^{N-1} &(\chi_k - \chi^*)^2 Q_1^2 + (\gamma_k - \gamma^*)^2 Q_2^2 \\
&+ Q_3^2(u_k^2 + v_k^2 + w_k^2) + Q_4^2 m_k^2 + Q_5^2 n_k^2 + Q_6^2 l_k^2 \\
&+ (\mathbf{u}_k - \mathbf{u}_{k-1})^\intercal R^\intercal R(\mathbf{u}_k - \mathbf{u}_{k-1}) \\
&+ (\chi_N - \chi^*)^2 Q_7^2 + (\gamma_N - \gamma^*)^2 Q_8^2 \\
&+ Q_9^2(u_N^2 + v_N^2 + w_N^2) + Q_{10}^2 m_N^2 + Q_{11}^2 n_N^2 + Q_{12}^2 l_N^2
\end{aligned} \tag{11.28}$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) \tag{11.29}$$

$$\text{given} \quad \mathbf{x}_0 \tag{11.30}$$

$$\begin{aligned}
Q_1 = Q_7 = 2000, Q_2 = Q_8 = 2300, Q_3 = Q_9 = 2 \\
Q_4 = Q_{10} = 1800, Q_5 = Q_6 = Q_{11} = Q_{12} = 300 \\
R = \begin{bmatrix} 25 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 35 & 0 \\ 0 & 0 & 0 & 22 \end{bmatrix}
\end{aligned} \tag{11.31}$$

Both the states of the UAV model and the control variables are included as optimization variables in the NLP. Constraints are placed on the state variables u, v, w to stay within a maximum speed, on α since the lift, drag, and pitch moment curves are only valid in this area, and on the control inputs, since they have physical limitations.

$$\begin{aligned}
-25\,\text{m/s} \leq u, v, w \leq 25\,\text{m/s} \\
-10^\circ \leq \alpha \leq 110^\circ \\
-30^\circ \leq \delta_a, \delta_e, \delta_r \leq 30^\circ \\
0 \leq \delta_t \leq 1
\end{aligned} \tag{11.32}$$

The transition between the path angles in these two path legs is smoothed by a cubic spline. To make the UAV keep a path angle approximately equal to γ_0, the spline has to be initiated before the UAV reaches that LOS angle.

11.5 Simulation

The simulations are performed in Python, using the open-source software package Casadi [8] for numerical optimization. Each optimization uses a prediction horizon of

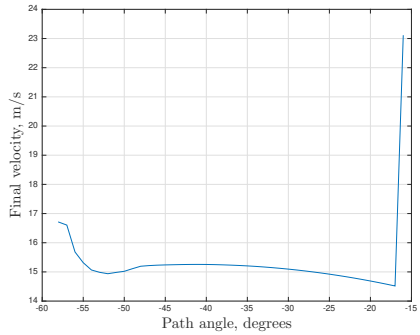


Figure 11.3: Speed and path angle dependency

$T = 3.0$ seconds, divided into $N = 30$ equidistant control intervals. The optimization variables and initial states are given by

$$\mathbf{X} = [\mathbf{x_0}\ \alpha_0\ \beta_0\ \gamma_0\ \chi_0\ \mathbf{u_0}\ \mathbf{x_1}\ \alpha_1\ \beta_1\ \gamma_1\ \chi_1\ \mathbf{u_1}\ \cdot\ \mathbf{x_N}\ \alpha_N\ \beta_N\ \gamma_N\ \chi_N] \tag{11.33}$$

$$\mathbf{x}_0 = [0,\ 0,\ -120,\ 25,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^\intercal \tag{11.34}$$

To confine the time it takes to solve each optimization problem, the maximum number of iterations for each optimization is set to 600. When the optimization is completed, $\mathbf{u_0}$ is the first occurrence of the control actions that are needed to produce an optimal series of states. This optimal control action is used to control the UAV, and the output state will be the start state of the next optimization.

As seen by eq. (2.20), the wind is composed by a steady component and a gust component, which follows the Dryden turbulence model [124]. The steady component is constant, and is assumed known to the model predictive controller since it can be estimated from GNSS and airspeed measurements [87]. However, the gust component is changed in each time step. To simulate time-delayed estimates of the wind gust, the last time step's gust measurement is used in the controller. The gust therefore introduces a partly unknown time varying disturbance, testing the robustness of the controller. This means that the simulations will never be perfect with respect to the wind, but lag one time step (100 ms).

What we want to investigate is how well the performance of the constrained NMPC for automatic deep-stall landing is. The performance can be divided into what the velocity is at the time of landing, what the deviation is from the desired landing target and what the attitude is at the time of landing.

11.5.1 Effect of varying path angle

First, it is investigated if some path angles in the descending phase gives a lower end velocity than others. To focus the analysis on the effect of different path angles, statistical deviations are avoided by assuming zero wind. The outcome is shown in fig. 11.3. When the path angle is above $-17°$, the angle of attack is below the stall angle and the UAV is not in deep stall. The speed is therefore higher than when the path angle is $-17°$ or below. When the path angle is steeper than 45°, oscillations arise and a stable deep-stall is not possible to achieve. Between these path angles, the speed is barely changing.

11.5.2 Simulations with varying gust strength

The gust is difficult to predict, as it does not follow any deterministic model. Because of this, its effect on the control can be more unpredictable than the effect of the steady wind, which can be taken into account in the optimization model. Therefore, this simulation setup studies the effect of varying wind conditions. Due to the stochastic property of the wind, this is done through Monte Carlo simulations with 101 trials, and considering the statistical mean and standard deviation. The target position is set to

$$\text{target} = \begin{bmatrix} \text{targ}_n \\ \text{targ}_e \\ \text{targ}_d \end{bmatrix} = \begin{bmatrix} 280 \\ 150 \\ 0 \end{bmatrix}, \tag{11.35}$$

which gives a desired course angle χ^* that is deliberately set different from zero. To compare the influence from the wind on varying path angles, the simulation considers $\gamma_0 \in \{-20°, -30°, -40°\}$. For simplicity, the steady wind component in the east and down directions are set to zero, while w_{n_s} is simulated with the different values 0 m/s, -2 m/s, -4 m/s, -7 m/s and -14 m/s. This yields a steady wind from the north direction with a magnitude from calm to a moderate gale wind. The gust components are all initialized to zero. As is seen from eqs. (11.14) and (11.16), a large steady wind will also give a large variance in the gust component. Both these factors contribute to making the landing increasingly more difficult with increasing wind conditions.

The end velocity and the final attitude of the UAV, as well as geometric deviation from the target location have been recorded from the simulations. The results are shown in table 11.1. The results are given as the mean ± the standard deviation, with the maximum absolute value included in parenthesis to better illustrate the outliers. This is based on 101 simulations, where the number of successful trials are indicated in the last column. Infeasible results from the optimization have been removed to better show the performance in the cases where it works.

The results show that the success rate is highly depending on the wind speed; the extreme case with $w_{n_s} = -14$ m/s only succeeds in 12 % of the trials, with unacceptably large deviations in all trials.

It is also seen that the steepest approach with $\gamma_0 = -40°$ is more susceptible to wind disturbances, as it has a lower success rate. An explanation to this is that as the path angles approach $-40°$, the vertical component of the velocity increase while the horizontal component decrease. At the same time $theta \approx 0$, so the body z-axis is parallel to the NED down-axis. Seeing this in connection with the definition of the body axis gust, eqs. (11.14) to (11.19), this indicates that the gust in the horizontal plane has a larger variance than the vertical component. The larger horizontal gust variance, combined with the decreased horizontal velocity component, makes the path of the UAV less predictable and more difficult to control.

With the above exceptions, it can be generally stated that the final landing speed is more dependent on the wind speed than the flight path angle; the landing speeds for the same wind speed has a maximal standard deviation of 1.5, whereas keeping the path angles constant and looking at how the different wind speeds affect the landing speed result in a standard deviation of 5.1. From table 11.1 it is also clear that the standard deviation in the final speeds increase with the magnitude of the wind speed. This is natural as more steady wind also means a larger variance in the gust wind, which directly affects the final speed.

Finally, the results also show that the final attitude is within reasonable values. The roll angle φ is close to zero, with the exception of a few outliers. Naturally the pitch angle θ is larger when the path angle is a smaller negative number, to keep the same α. ψ decreases with the wind speed, since the wind is coming from a different angle than the course angle. This forces the UAV to increase the sideslip to achieve the same course angle, thus the heading has to decrease.

The plots of the state- and control trajectories for one of the simulations with $\gamma_0 = -30°$ and $w_{n_s} = -4\,\text{m/s}$ can be seen in figs. 11.4 and 11.5. Similarly the gust component of the wind is plotted in fig. 11.6. The angle of attack holds a value of about 42°, which is close to the second peak of the lift curve in fig. 11.1, which gives high values for both the lift and drag foces. This can also be seen from fig. 11.4d. From the plots of the desired path angle in fig. 11.4b it is seen that the controller switches from flying straight to descending almost instantaneously, since the angle of the LOS vector is close to $-30°$ at the start of the simulation. Once the limit is crossed, the desired path angle is adjusted so that the target will be reached. However, due to the delay in the dynamics of the UAV, this has to enforce a slightly larger path angle.

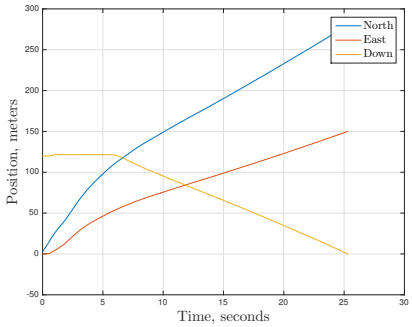
11.5.3 Simulations with varying side wind strength

Although the landing should ideally be against the wind, it is unrealistic to hope to achieve no disturbing side wind at all. Therefore, the effect side wind has on the performance of the algorithm is investigated. For simplicity, the target location is set to $[200, 0, 0]^\intercal$, enabling the adjustment of side wind to be an adjustment of w_{e_s}. Here, γ_0 was set to $-40°$, as this was showed to be more sensitive to wind in section 11.5.2.

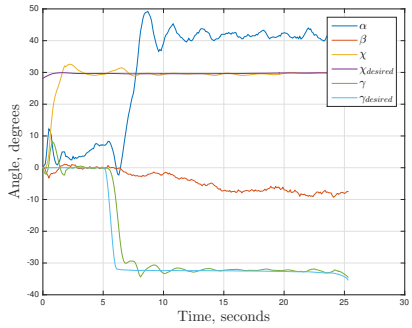
Again a Monte Carlo simulation with 100 trials is performed so that the effects of the varying gust component are averaged out in the results, as showed in table 11.2 using the same notation as in table 11.1. The simulations are only performed for $w_{e_s} \leq 0$, since the deterministic response of the model is symmetric with respect to the steady east wind component. As is expected the final landing speed decreases, while the heading angle increases for higher magnitude winds. This indicates that the UAV will not hit nose-first, forcing one of the wings will take some of the impact from the net.

11.5.4 Simulations with limited wind knowledge

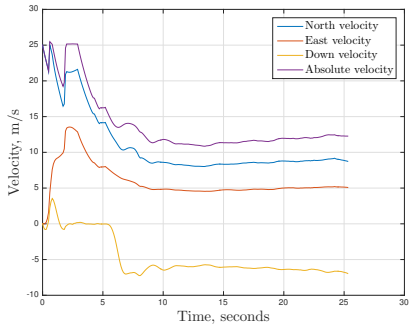
One of the limitations with the previously presented landing scheme is that it assumes perfect knowledge of the wind, up to the previous time step. Despite recent advances within wind estimation , this might be a too optimistic assumption. Therefore a simulation study with a more limited knowledge of the wind is performed. In this case the NMPC only knows the steady wind component, while the gust is assumed unknown. To limit the analysis, this study only looks at the case when $\gamma_0 = -30°$ and $w_{n_s} = -4\,\text{m/s}$. The limited knowledge of the wind makes it even more important for the UAV to not deviate from the path, as a sudden change in wind might make the return to the path infeasible. Since the variance of the Dryden gust wind is larger in x- and y-direction than in the z-direction, Q_1 was increased to 3500 in this simulation. Other than this the setup is the same as in section 11.5.2, to allow for easy comparison of the results. Figure 11.7 shows a histogram with unit bin widths of the deviations to the landing target. Out of a total 100 landing attempts, 33 of the landings are within 2 meters from the target, while 15 are 17 meters or more from the target. In addition to the 65 successful landings showed in the plot, 35 landing attempts were found to be infeasible. Compared to the results in table 11.1, the average landing speed is slightly higher in this case, with a much larger standard deviation; 14.79 ± 3.59 m/s. However if we look at the landing speed for the 33 attempts that land within 2 meters from the target, the average is 12.06 m/s with a standard deviation of 1.00.



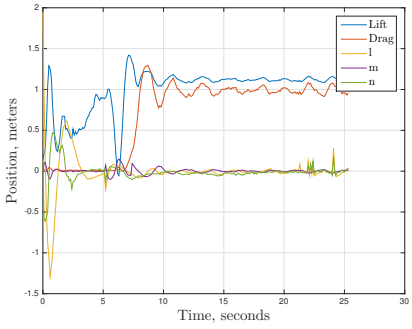
(a) Position and attitude in 3 dimensions



(b) Angles

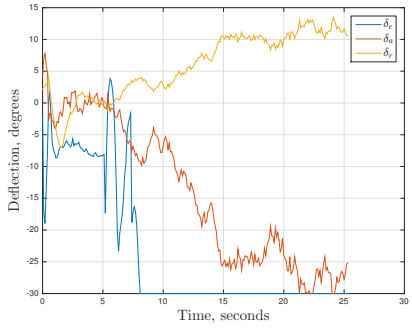


(c) Inertial velocity

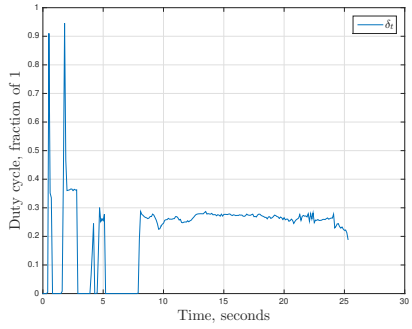


(d) Drag and lift coefficients and roll, pitch and yaw moments

Figure 11.4: State trajectory from simulation with $\gamma_0 = -30°$ and $w_{n_s} = -4\,\mathrm{m/s}$



(a) Control surface deflection in degrees



(b) Throttle control in used share of duty cycle

Figure 11.5: Control trajectory from simulation with $\gamma_0 = -30°$ and $w_{n_s} = -4\,\mathrm{m/s}$

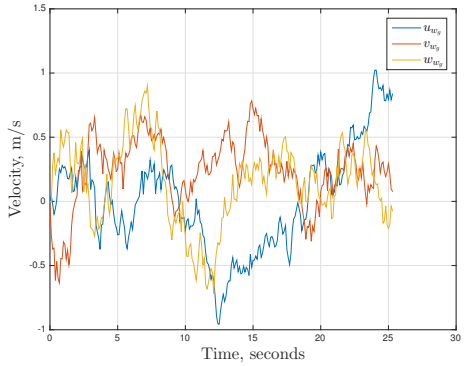


Figure 11.6: Dryden turbulence from simulation with $\gamma_0 = -30°$ and $w_{n_s} = -4\,\mathrm{m/s}$

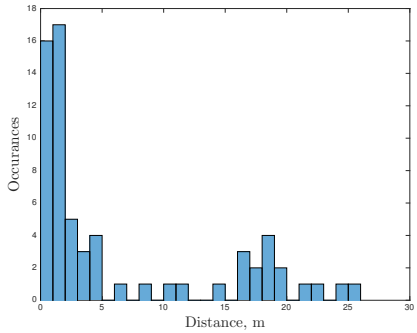


Figure 11.7: Histogram showing deviation from landing target when no knowledge of the gust wind is assumed

Table 11.1: Simulations of different wind conditions and path angles

w_{n_s}	γ_c	Final speed [m/s]	Target error [m]	φ_f[°]	θ_f[°]	ψ_f[°]	Successful trials
0	-20	14.69±0.00 (14.69)	0.68±0.00 (0.68)	-2.72±0.00 (2.72)	18.21±0.00 (18.21)	35.21±0.00 (35.21)	101
	-30	15.23±0.00 (15.23)	0.91±0.00 (0.91)	1.91±0.00 (1.91)	8.73±0.00 (8.73)	30.23±0.00 (30.23)	101
	-40	15.45±0.00 (15.45)	0.48±0.00 (0.48)	2.02±0.00 (2.02)	-0.71±0.00 (0.71)	27.79±0.00 (27.79)	101
-2	-20	12.99±0.34 (14.08)	0.34±0.20 (0.98)	-1.35±2.78 (5.54)	20.49±3.79 (31.26)	29.87±3.60 (35.25)	101
	-30	13.46±0.27 (14.16)	0.37±0.21 (0.82)	0.59±3.14 (4.78)	11.99±1.53 (15.89)	26.46±4.30 (32.72)	101
	-40	13.85±0.30 (14.60)	0.46±0.27 (0.95)	-0.74±2.71 (4.70)	2.05±2.14 (5.99)	28.05±3.95 (32.17)	66
-4	-20	11.05±0.58 (12.29)	0.35±0.22 (0.81)	0.54±3.70 (8.76)	24.03±2.45 (29.56)	22.63±3.88 (28.15)	101
	-30	11.76±0.57 (13.08)	0.35±0.19 (0.82)	0.39±3.63 (7.59)	16.03±2.09 (22.62)	22.41±3.66 (29.13)	101
	-40	12.27±0.55 (13.53)	0.38±0.24 (0.82)	-0.30±3.86 (7.65)	6.57±2.26 (13.52)	21.61±3.93 (27.32)	32
-7	-20	8.19±1.17 (11.35)	0.41±0.22 (1.02)	0.08±4.10 (10.95)	27.98±5.40 (43.33)	15.61±3.84 (25.76)	101
	-30	9.56±1.36 (15.80)	0.36±0.36 (2.82)	1.04±5.09 (23.27)	13.65±12.47 (33.85)	16.88±4.83 (30.02)	100
	-40	10.09±2.24 (20.43)	0.32±0.40 (2.72)	-1.69±6.56 (5.49)	10.16±12.89 (23.43)	14.97±5.33 (36.23)	46
-14	-20	1.68±0.61 (2.80)	178.83±33.79 (220.88)	2.02±0.77 (3.07)	31.39±8.53 (41.74)	-1.36±5.10 (6.07)	6
	-30	4.17±4.07 (14.01)	71.33±38.89 (113.10)	-1.56±3.74 (2.62)	22.96±12.21 (36.66)	6.92±12.38 (30.08)	9
	-40	4.43±3.59 (15.17)	10.46±18.45 (64.46)	-0.73±5.39 (10.79)	21.31±16.67 (41.39)	6.89±9.40 (38.12)	25

Table 11.2: Simulation of different side winds

w_{e_s}	Final speed [m/s]	Target error [m]	φ_f[°]	θ_f[°]	ψ_f[°]	Successful trials
0	15.54±0.00 (15.54)	0.72±0.00 (0.72)	0.00±0.00 (0.00)	-4.63±0.00 (4.63)	0.00±0.00 (0.00)	100
-1	15.53±0.14 (15.93)	0.51±0.26 (1.13)	-1.61±2.23 (2.54)	-5.09±0.51 (3.85)	4.12±2.20 (8.87)	100
-2	15.38±0.30 (16.10)	0.56±0.32 (1.79)	-0.67±2.08 (2.77)	-4.81±1.52 (2.73)	6.63±3.13 (15.01)	100
-3	15.22±0.46 (16.58)	0.58±0.26 (1.28)	-2.07±2.82 (6.32)	-4.37±1.52 (0.97)	11.48±3.59 (19.90)	100
-4	14.83±0.56 (16.59)	0.56±0.22 (1.15)	-2.50±3.00 (7.82)	-3.07±2.09 (4.32)	17.50±3.22 (25.45)	98
-5	14.39±0.72 (16.54)	0.61±0.24 (1.34)	-2.90±3.40 (5.75)	2.10±3.28 (9.15)	21.58±3.69 (28.52)	100
-6	13.70±0.99 (16.18)	0.56±0.26 (1.54)	-2.45±4.06 (10.12)	4.02±3.96 (16.29)	28.49±4.59 (39.09)	100
-7	13.45±1.10 (16.43)	0.62±0.35 (1.94)	-3.23±5.12 (14.07)	4.69±4.52 (19.09)	33.97±6.24 (55.18)	100
-8	12.60±1.23 (15.73)	0.54±0.29 (1.76)	-3.02±4.73 (8.46)	6.37±5.19 (19.14)	40.36±6.07 (57.11)	99
-9	11.60±1.63 (14.73)	0.57±0.33 (1.62)	-2.68±5.58 (9.76)	9.67±5.87 (26.64)	45.34±7.85 (62.78)	100
-10	10.56±2.35 (16.10)	0.56±0.36 (1.79)	-2.84±6.46 (16.97)	12.84±8.89 (46.85)	51.53±10.44 (84.58)	99
-11	10.03±2.50 (16.82)	0.51±0.39 (2.13)	-2.42±6.19 (16.59)	14.58±8.69 (36.36)	55.12±11.38 (84.53)	99

11.6 Conclusion

The algorithm presented in this chapter is able to guide a fixed-wing UAV in a precise deep-stall landing. The UAV is controlled with an NMPC, which first guides the UAV to a waypoint whose line-of-sight angle to the landing target equals a given angle, with a course angle directed towards the landing target. From that point, the NMPC combines tracking of the necessary path and course angles to the landing target, with keeping the UAV in a deep stall. The deep stall is maintained by keeping the pitch moment at zero at the same time as the speed is minimized. This leads to a trim point where the pitch moment is zero, and the pitch moment gradient with respect to the angle of attack is negative. When the UAV is deep-stall landing, the drag is increased and the speed is reduced. A small simulation study was performed to analyze which path angles gave the lowest landing speeds, concluding that $-43 \leq \gamma_0 \leq -17$ will land the plane in a deep stall. A larger Monte Carlo-style simulation with 101 trials was performed to consider the effects of the stochastic wind gust. This showed that the controller was able to land the UAV in steady winds of up to 7 m/s. However the number of successful trials depend largely on the wind magnitude. Also the results indicate that a steeper flight path is more susceptible to wind disturbances. Finally, the controllers dependence on knowing recent values of the gust wind component was challenged. This showed that the majority of the feasible paths land within 2 meters from the target. However it also showed that more research has to be performed in order to recover the UAV from paths that are infeasible or destined to miss the target.

Part III

Navigation in GNSS-denied environments

Chapter 12

Introduction

The discontinuation of the selective availability pseudorandom errors in global position system (GPS) in year 2000 laid the foundation for a boost in the use of global navigation satellite systems (GNSS) for many applications, such as navigation of manned and unmanned vehicles. Through the standard positioning services and differential correction services, GNSS has the advantage of providing high-accuracy positioning, navigation and timing (PNT) signals, with global coverage, through lightweight receivers at a low cost to the user. GNSS has proved particularly useful when integrated with an inertial measurement unit (IMU) with its complementary features such as high precision, albeit with a bias, and high sampling frequency, making it possible to obtain high accuracy, high precision position estimates at a high rate. However, when designing safety-critical systems, it is important to keep in mind the weaknesses of GNSS. The low signal power makes GNSS highly susceptible to radio frequency interface (RFI), both natural RFI, such as ionospheric scintillations [199] and multipath, and intentional RFI, such as jamming [149] and spoofing [92, 165]. The worst-case scenario of RFI is exemplified by the 2011 spoofing incident where Iranian forces successfully captured a U.S. RQ-170 UAV [80]. More in-depth studies on the severity of GNSS RFI directed towards the public has also been published [24]. Another aspect is the fact that the complex GNSS satellite systems are controlled by international bodies, and is believed to be one of the first systems to become unavailable in the event of a large international conflict. The complexity of these systems was showcased in July 2019, when it took the European Global Navigation Satellite System Agency (GSA) a week to fix and restore the Galileo satellite system PNT service [89]. With this in mind, it is clear that GNSS should be considered as a nice-to-have feature, and that it, in itself, cannot be relied upon for navigation of safety critical systems.

The need for a reliable alternative for safety critical applications motivates research into alternative, GNSS-free navigation solutions, of which there are many. In addition to overcoming GNSS RFI, these systems mitigate the issue of hardware or software single-point of failure in GNSS user equipment. Visual odometry [48, 132], visual SLAM [43, 133] and terrain/map matching [67] all rely on tracking of multiple

features, are self-contained solutions that do not need any infrastructure, but are either limited to local navigation or require a global map. Another option, that requires local infrastructure, is ground based radar positioning. This will also require a radio link to send the tracking data to the UAV.

This part considers two other alternative positioning systems to GNSS. As GNSS rely on multilateration of pseudorange measurements, the instinctive solution to GNSS-free navigation is to rely on range measurements from other sources [40], such as ultra-wideband (UWB) [55, 112]. The main drawback is the need for local infrastructure, as three or more ground fixed antennas will be needed. UWB is considered in chapter 13, where the focus is increased robustness to GNSS-dropout in a landing scenario.

Another alternative is position measurements from a phased-array radio system (PARS), which is considered in chapter 14. PARS is primarily designed as a high-bandwith radio communication link, providing the user with e.g. telemetry data from a UAV, including a live video streams [77], as well as relaying information through a UAV to units that themselves are beyond-radio-line-of-sight from the base antenna. As opposed to omnidirectional alternatives, PARS transmits a directed, narrow beam that allows for high transmission power and long range. Information from this directed beam can be used to deduce the position of the radio, relative to the ground station [3], similar to standard radar tracking calculations [12], but with a trivial data association step. While PARS does depend on local infrastructure, only a single ground antenna is needed to provide a full 3D position measurement.

This part also considers two different approaches to inertial navigation. The UWB range measurements in chapter 13 are integrated with a inertial and magnetometer measurements, using a cascaded nonlinear observer, consisting of an attitude observer and a translational motion observer. Advantages with the nonlinear observer include guarantee stability properties, reduced need for linearization, and a computational cost of about 25 % when compared to a multiplicative extended Kalman filter [72].

The PARS measurements in chapter 14 aids the inertial navigation through a multiplicative extended Kalman filter. The advantage of the Kalman filter approach is the additional information that is available through the kinematic coupling of the uncertainty via the state covariance matrix, coupling the rotational and translational kinematics. This has enabled the estimation of heading without the use of a magnetometer.

Chapter 13

Ultra-wideband radio navigation

In this chapter, ultra-wideband radio is presented as an additional means of estimating position, based on range measurements to beacons in known locations, to make the navigation system more robust to GNSS dropout. The work is based on the paper

- [63] 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

This chapter considers a tightly-coupled integration of IMU and double-differenced GNSS range measurements for RTK applications using a nonlinear observer structure aided with single-differenced UWB range measurements, offering increased robustness when operating in GNSS challenged or denied areas. The nonlinear observer is based on [71], but the addition of UWB requires the stability proof to be revisited, as the state vector for the translational motion observer is augmented to include timing parameters for the UWB signals.

13.1 Introduction

In recent years nonlinear observers have been proposed to estimate position, linear velocity and attitude (PVA) of vehicles as an alternative to the widely used KF. The nonlinear observers have the advantage of proven (often global) stability conditions, smaller computational footprint and reduced need for linearization. In [114] and [113] a nonlinear complementary filter was proposed, which was expanded upon in [78] to an attitude and velocity observer dependent only on inertial, magnetometer and GNSS measurements. Furthermore, in [56] and [57], these results were

expanded to include the quaternion representation of attitude in local and global coordinate frames. For an extensive overview of attitude estimators see [30].

One way to increase the precision of the PVA estimates is to utilize a tightly coupled integration scheme, integrating the inertial measurements with global range measurements from the satellites to the receiver, instead of using the global position estimates from the receiver. The GNSS measurements are thereby in the range domain instead of the position domain, allowing aiding by pseudo-range or carrier-phase measurements. Tightly coupled integration have traditionally been done using a KF variant; [59] and [45], with nonlinear observers recently being proposed; [86], [71], [84] and [85]. Using carrier-phase measurements as aiding, introduces the integer ambiguities which are a constant range offset, given by a number of wavelengths. If the ambiguities can be resolved correctly the carrier-phase measurements will typically have centimeter accuracy.

The global range measurements are subjected to disturbances from the atmosphere due to signal path obstructions in the ionosphere and troposphere. These atmospheric disturbances can be somewhat predicted using local and solar weather forecasts. Another option is to augment the measurement configuration by including a base station at a known position. The base station supplies the vehicle (often called the 'rover') with range measurements obtained at the base station, to be subtracted from the rover measurements, thereby reducing common disturbances. If the rover is within 20 km[59], the atmospheric disturbances can be canceled in the differenced range measurements. In a differential GNSS configuration the integration uses the differenced range measurements to aid the inertial navigation. When resolving the integer ambiguities in real time, the configuration can be considered a Real-Time-Kinematic (RTK) configuration. The quality of an RTK position estimate is often divided into three categories: a) 'single', indicating same precision as a single-receiver configuration, b) 'float', where the integer ambiguities are considered real valued numbers giving a precision on decimeter level, or c) 'fixed', when the ambiguities have been resolved to integer values often with a precision on centimeter level. Maintaining a 'fixed' RTK position estimate is desirable when the accuracy requirements are high, such as during a fixed-wing UAV net landing[172].

A drawback of the RTK positioning is the sensitivity to agile maneuvers and obstructed GNSS signal path, which can lead to loss of fix quality. If the RTK positioning looses sight of a satellite in the constellation, the integer ambiguity have to be determined again when the satellite is reintroduced. In order to improve the resistance to precision deterioration when satellites are obstructed the sensor configuration can be augmented with use of further aiding sensors, e.g. cameras, air speed sensors, or ultra wideband transceivers. Ultra wideband (UWB) technology has typically been used for indoor navigation due to its short range. The setup consists of a UWB receiver measuring the distance to one (or several) UWB nodes acting as pseudo-satellites. The UWB cannot, in practice, substitute the GNSS measurements in outdoor environment due to the short range, however it can aid the inertial navigation in GNSS denied or challenged areas, e.g. the UWB nodes can be placed strategically around the landing area to aid during the final part of a

UAV flight. UWB technology have been used in other outdoor applications such as in [146] where inter-vehicle positioning was achieved using DGPS in combination with UWB, or in [140] where position and heading of smoke divers equipped with IMU and UWB was estimated. Other previous work include [58] where double-differenced GPS measurements are coupled with a single UWB range measurement in a tightly coupled manner, using an unscented Kalman filter and a two-frequency GPS receiver. Their goal is to accurately determine the relative position between two UAVs in formation flight, with increased robustness to GPS loss-of-fix. Further, [103] presents a summary of the theoretical foundation for, and challenges related to, UWB self-localization systems, such as the Cramer-Rao lower bound and clock synchronization issues. Another UWB application is suggested in [186], where UWB position estimates are used to emulate GNSS measurements.

13.2 Problem Setup

This section describes the properties of UWB and RTK-GNSS, as well as why this type of sensors are beneficial to combine.

13.2.1 Real-Time-Kinematics GNSS

The idea behind RTK is to utilize differenced carrier-phase and pseudo-range measurements instead of measurements obtained at the rover, for computation of position. The differenced measurements are determined by subtracting ranges obtained at a stationary base station from those obtained at a moving rover, thereby canceling some common error terms. The measurements, obtained from a constellation of m satellites, by the two receivers are pseudo-range, ρ, and carrier-phase, φ. Considering measurements from the ith satellite, the measurements are given as:

$$\rho_i^r = \psi_i^r + \beta^r + \epsilon_{\rho,i}^r, \tag{13.1}$$

$$\varphi_i^r = \psi_i^r + N_i^r \lambda + \beta^r + \epsilon_{\varphi,i}^r, \tag{13.2}$$

where $\psi_i^r = \|\boldsymbol{p}_r^e - \boldsymbol{p}_i^e\|_2$ is the geometric distance between the rover, r, and the satellite, i, at position $\boldsymbol{p}_i^e$. The receiver clock range bias is denoted as $\beta^r := c\Delta_c$, where Δ_c is the clock bias and c is the speed of light. The receiver clock range bias is assumed to be slowly time-varying, and to be the same for all satellites in the constellation. The integer ambiguity is denoted N_i^r, with λ being the wavelength. The atmospheric disturbances of the range measurements are denoted $\epsilon_{\rho,i}^r$ and $\epsilon_{\varphi,i}^r$, and consists of the systematic environmental errors due to signal path obstruction in the ionosphere and troposphere.

Introducing the base station measurements, expressed by substituting the superscript r to s, the range measurements can be differenced to cancel the common environmental terms. When the rover and base station are sufficiently close to each

other, i.e. the baseline between them is less than 20km, the environmental errors experienced by the two receivers are considered to be the same, i.e. $\epsilon^r_{\rho,i} = \epsilon^s_{\rho,i} = \epsilon_{\rho,i}$ and $\epsilon^r_{\varphi,i} = \epsilon^s_{\varphi,i} = \epsilon_{\varphi,i}$[59]. For the ith satellite, the environmental errors are canceled in the single-differenced measurements:

$$\Delta\rho_i = \Delta\psi_i + \Delta\beta, \tag{13.3a}$$
$$\Delta\varphi_i = \Delta\psi_i + \Delta N_i\lambda + \Delta\beta, \tag{13.3b}$$

where $\Delta\rho_i = \rho^r_i - \rho^s_i$, $\Delta\varphi_i = \varphi^r_i - \varphi^s_i$, $\Delta\beta = \beta^r - \beta^s$, $\Delta N_i = N^r_i - N^s_i$, and $\Delta\psi_i = \psi^r_i - \psi^s_i$ is the length of the geometric baseline between rover and base station. The integer ambiguity is a vector, $\Delta N_i \in \mathbb{R}^m$, comprised of the ambiguities of the available satellites; $\Delta N_i = [N_1; N_2; \ldots; N_m]$.

The clock error $\Delta\beta$ can be removed by double-differencing the measurement against a satellite h at the same epoch:

$$\nabla\Delta\rho_{ih} = \nabla\Delta\psi_{ih}, \tag{13.4a}$$
$$\nabla\Delta\varphi_{ih} = \nabla\Delta\psi_{ih} + \nabla\Delta N_{ih}\lambda, \tag{13.4b}$$

where $\nabla\Delta\rho_{ih} = \Delta\rho_h - \Delta\rho_i$, $\nabla\Delta\psi_{ih} = \Delta\psi_h - \Delta\psi_i$, $\nabla\Delta\phi_{ih} = \Delta\phi_h - \Delta\phi_i$ and $\nabla\Delta N_{ih} = \Delta N_h - \Delta N_i$. It is assumed that the raw measurements are time stamped simultaneously by the receivers, such that the corresponding measurements for the two satellites can be found at the two receivers. Double-differencing the raw measurements reduces the dimensions of the observer, since clock error estimates are no longer needed. In the context of RTK-GNSS it can be advantageous to reduce the noise by double-differencing, since the residual measurement error, according to [147], should be less than $\frac{1}{4}$ wavelength to solve the integer ambiguity.

13.2.2 Ultra Wideband (UWB)

Ultra wideband is a radio technology that transmits over a very wide range of frequencies, typically several GHz. It is characterized by moderate range (typically 200 m) due to power limitations. UWB transmitters can not only be used for communication, but also for range measurements with sample rates in tens of Hertz. The technology has increased in popularity since the United States Federal Communications Commission (FCC) allowed for unlicensed use of the 3.1-10.6 GHz spectrum in 2002 [111]. In particular, impulse-radio UWB (IR-UWB), has become popular for centimeter-level accuracy ranging applications.

Advantages with UWB include robustness to interference, resistance to multipath, low energy consumption[130], small footprint, as well as good time resolution allowing for centimeter level precision of range measurements[111]. Disadvantages include range limitations, as well as the need for additional equipment in the UAV.

UWB ranging can be either synchronous or asynchronous[111]. For synchronous ranging, the time-of-flight (ToF) between the nodes are measured by comparing

the timestamp of the received message with the current time in the node. For this reason, the ranging precision depends on accurate synchronization of the clocks of each node. This can be mitigated using asynchronous two-way ToF ranging, where one node transmits a message to a second node, which simply transmits the message back. Again the first node compares the timestamp with the current time. However, as mentioned by [103], the one-way communication required by the asynchronous approach offers more flexibility since the range accuracy and rate of communication is not affected by introducing additional rovers to the system.

One common way to model UWB range measurements, as given in e.g. [28], is to include a bias term and a zero-mean noise term:

$$\mu_j^r = l_j^r + \alpha^r + \alpha_j + w_j^r. \tag{13.5}$$

Here, $l_j^r = \|\boldsymbol{p}_r^e - \boldsymbol{p}_j^e\|$ is the geometric distance between the rover r and the UWB node j. The noise term, w_j^r, represents the zero-mean white noise. The bias associated with the rover, including clock bias, radio oscillation frequencies and variations in the speed of light[111], is summarized in α^r, while the equivalent bias terms associated with UWB node j are summarized in α_j. The experimental study in [130] shows a distance-dependence in the bias and in the noise standard deviation, which both are fit to a linear function using the least squares recursion. In the following, the bias and standard deviation will be considered constant, as in [140]. Furthermore, only line-of-sight (LOS) operation will be considered. See [140] and [28] for studies on LOS and non-line-of-sight (NLOS) range measurements. By using the pseudo-range measurements from UWB node j to the rover and base station, the clock-bias from the node is removed using single differencing:

$$\Delta\mu_j = \Delta l_j + \Delta\alpha, \tag{13.6}$$

where $\Delta l_j = l_j^r - l_j^s$, and $\Delta\alpha = \alpha^r - \alpha^s$.

13.2.3 Interconnection

The reason for choosing to integrate UWB and RTK-GNSS is that they have some complimentary properties. While multipath can be a considerable error source in GNSS, this is not as pronounced for UWB due to the higher frequencies in the signal. On the other hand GNSS provides global coverage, whereas UWB coverage is significantly limited by the range of the signal. Additionally both systems are vulnerable to errors, if it is the only position sensor installed in the rover. Therefore the measurements from the two sensors can be beneficially fused, allowing for global coverage, and redundancy in critical situations such as landing a UAV, or UAV inspection in areas with high GNSS multipath noise levels.

A conceptual setup of the rover, base station, GNSS-satellites and UWB nodes is shown in fig. 13.1. Two satellites, $\mathcal{S}_1$ and $\mathcal{S}_2$, are shown transmitting to the base station, s, and rover, r. Additionally, two UWB nodes, $\mathcal{N}_1$ and $\mathcal{N}_2$, are depicted. The

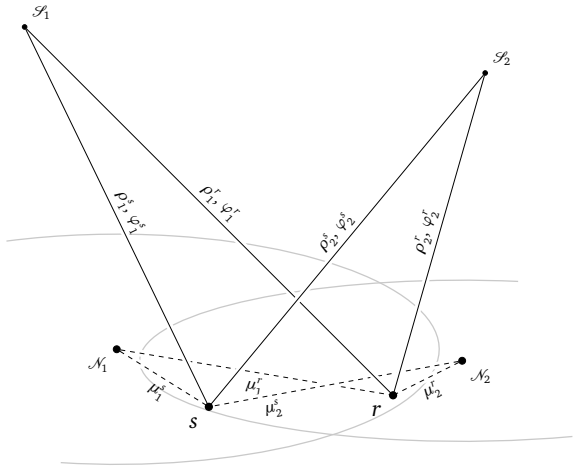


Figure 13.1: Conceptual setup of rover, base station, GNSS-satellites and UWB nodes

diagram has been simplified by including only two satellites and two UWB nodes. Both the rover and the base station are equipped with GNSS and UWB receiver, allowing for double-differencing of the GNSS and single-differencing UWB signals as explained in section 13.2.1 and section 13.2.2. In addition, the rover is equipped with an inertial measurement unit (IMU) and a magnetometer. In order to solve the range measurement equations for the three dimensional position of the rover, the time, and the bias, at least five GNSS satellites must be within LOS. Similarly, the setup needs five UWB nodes that are located close to the base station, around an area of special interest, typically a region that is critical for the mission. The relative position of the nodes to the base station is assumed constant and known.

13.3 Problem Definition

Inertial navigation using a nonlinear observer in a dual receiver configuration is considered, where a stationary base station supplies a moving rover with GNSS range measurements. The common atmospheric disturbances can be canceled provided that the rover is sufficiently close to the base station, by determining the double-differenced range measurements. The inertial navigation is further aided by UWB measurements from a set of nodes. The nonlinear observer is an extension of the double-differenced observer developed in [71], extended to include single-differenced UWB range measurements.

This chapter seeks to estimate the position, linear velocity and attitude (PVA) of a moving rover by use of tight integration of inertial measurements in the Body-frame aided by global GNSS and UWB range measurements. The GNSS data include pseudo-range and carrier-phase measurements, where the integer ambiguity are

initially considered as real valued and can later be fixed to integer value for increase precision. The UWB system will provide short ranged measurements with high precision to be used when the rover enters an area of special interest, which might be GNSS denied or challenged. Estimation of the rover PVA is of interest as well as an evaluation of the benefits of using UWB measurements for additional aiding.

The kinematic model is stated as:

$$\dot{\boldsymbol{p}}_r^e = \boldsymbol{v}_r^e, \tag{13.7a}$$

$$\dot{\boldsymbol{v}}_r^e = -2S\left(\boldsymbol{\omega}_{ie}^e\right)\boldsymbol{v}_r^e + \boldsymbol{f}^e + \boldsymbol{g}^e(\boldsymbol{p}_r^e), \tag{13.7b}$$

$$\dot{\boldsymbol{q}}_b^e = \frac{1}{2}\boldsymbol{q}_b^e \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ib}^b \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ie}^e \end{bmatrix} \otimes \boldsymbol{q}_b^e, \tag{13.7c}$$

$$\dot{\boldsymbol{b}}^b = 0, \tag{13.7d}$$

$$\dot{\boldsymbol{p}}_s^e = 0. \tag{13.7e}$$

Here $\boldsymbol{p}_r^e$ and $\boldsymbol{p}_s^e$ are the position of the rover and base station, while $\boldsymbol{q}_b^e$, $\boldsymbol{v}_r^e$ and $\boldsymbol{\omega}_{ib}^b$ are the attitude represented by a unit quaternion, the linear and angular velocity of the rover, respectively. The angular velocity of the Earth, $\boldsymbol{\omega}_{ie}^e$, and the position dependent gravitational vector-function, $\boldsymbol{g}^e$, are assumed known. The specific force experienced by the rover is denoted f^e, while $\boldsymbol{b}^b$ is the Body-frame gyro bias associated with the angular velocity measurements.

13.3.1 Measurement Assumptions

It is assumed that a sensor suite is mounted on the vehicle with inertial and global range measurement units. Moreover, it is assumed that the following measurements are available:

- Specific force as measured by an IMU, $\boldsymbol{f}_{IMU}^b = \boldsymbol{f}^b$.
- Angular velocity experiencing a bias, as measured by an IMU, $\boldsymbol{\omega}_{ib,IMU}^b = \boldsymbol{\omega}_{ib}^b + \boldsymbol{b}^b$.
- Magnetic field as measured by a magnetometer, $\boldsymbol{m}^b$.
- Pseudo-range as measured by a GNSS receiver from the ith satellite, ρ_i^r.
- Carrier-phase as measured by a GNSS receiver from the ith satellite, φ_i^r.
- UWB range as measured by a UWB receiver from the jth node, μ_j^r.

Furthermore, it is assumed that pseudo-range, carrier-phase and UWB measurements are available to the stationary base station, with at least $m \geq 5$ common satellites and $n \geq 4$ common UWB nodes.

13.4 Nonlinear Observer

The proposed nonlinear observer structure consisting of a nonlinear attitude observer, a translational motion observer (TMO) and a gain computation, is visualized in fig. 13.2. The block diagram also includes the required sensors; inertial, magnetometer, GNSS and UWB range sensors. Two GNSS receivers are necessary, one mounted on the rover and one on a stationary base station. The stationary receiver supplies the data for calculation of the satellite position, $\boldsymbol{p}_i^e$, in addition to the range measurements. The UWB nodes are positioned such that the receiver on the rover can determine single-differenced range measurements.

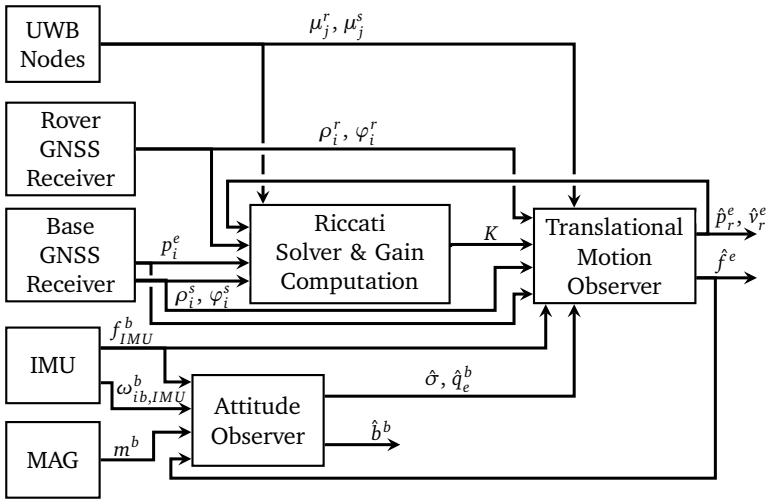


Figure 13.2: Block diagram of the proposed nonlinear observer structure

The proposed nonlinear observer is a modification to the loosely-coupled nonlinear observer initially presented in [57], which has been expanded in [86] to tightly-coupled systems using range and range-rate measurements, and further expanded in [71] where both a single-differenced and a double-differenced measurement structure was proposed.

The attitude observer is supplied with inertial measurements as well as a specific force estimate from the TMO. For the loosely-coupled system in [57] this was shown to be exponentially stable with a semiglobal region of attraction with respect to attitude observer initialization and local region of attraction with respect to translational motion observer initialization, due to the cascaded structure. An additional feedback, of rover position estimate, from the TMO to the gain computation is required to compute the injection terms necessary in the TMO.

The following sections will elaborate on the individual parts of the observer structure.

13.4.1 Attitude Observer

The attitude of the rover is determined as the rotation between Body- and ECEF-frame, using inertial measurements in Body-frame. The recent nonlinear attitude observer from [56], estimating the gyro bias in addition to the attitude as a unit quaternion, is restated as:

$$\dot{\hat{\boldsymbol{q}}}_b^e = \frac{1}{2}\hat{\boldsymbol{q}}_b^e \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ib}^b - \hat{\boldsymbol{b}}^b + \hat{\boldsymbol{\sigma}} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ie}^e \end{bmatrix} \otimes \hat{\boldsymbol{q}}_b^e, \tag{13.8a}$$

$$\dot{\hat{\boldsymbol{b}}}^b = \text{Proj}\left(-k_I \hat{\boldsymbol{\sigma}}, \|\hat{\boldsymbol{b}}^b\| \leq M_{\hat{b}}\right). \tag{13.8b}$$

Here, the Proj$(\cdot,\cdot)$-operator is a multi-dimensional saturation function that projects the gyro bias estimate to within a sphere with radius $M_{\hat{b}}$. The gain, k_I, is a positive constant, and $\hat{\boldsymbol{\sigma}}$ is an injection term determined from two vectors in the Body-frame, $\boldsymbol{v}_1^b$ and $\boldsymbol{v}_2^b$, with corresponding vectors in the ECEF-frame, $\boldsymbol{v}_1^e$ and $\boldsymbol{v}_2^e$:

$$\hat{\boldsymbol{\sigma}} = k_1 \boldsymbol{v}_1^b \times \boldsymbol{R}(\hat{q}_b^e)^\intercal \boldsymbol{v}_1^e + k_2 \boldsymbol{v}_2^b \times \boldsymbol{R}(\hat{q}_b^e)^\intercal \boldsymbol{v}_2^e, \tag{13.9}$$

where the tuning gains k_1 and k_2 satisfy $k_1 > k_p$ and $k_2 > k_p$ for some $k_p > 0$. The vectors in Body- and ECEF-frame can be chosen in various ways, and will here be considered as:

$$\boldsymbol{v}_1^b = \frac{\boldsymbol{f}_{IMU}^b}{\|\boldsymbol{f}_{IMU}^b\|}, \quad \boldsymbol{v}_1^b = \frac{\hat{\boldsymbol{f}}^e}{\|\hat{\boldsymbol{f}}^e\|}, \quad \boldsymbol{v}_2^b = \frac{\boldsymbol{m}^b}{\|\boldsymbol{m}^b\|} \times \boldsymbol{v}_1^b, \quad \boldsymbol{v}_2^e = \frac{\boldsymbol{m}^e}{\|\boldsymbol{m}^e\|} \times \boldsymbol{v}_1^e. \tag{13.10}$$

It is vital that the vectors $\boldsymbol{v}_1^e$ and $\boldsymbol{v}_2^e$ are not co-linear for the system to be observable. The magnetic field in ECEF-frame is assumed available, e.g. from a database. The estimated specific force, $\hat{\boldsymbol{f}}^e$, is fed back from the translational motion observer.

13.4.2 Translational Motion Observer

Once the estimated attitude $\hat{\boldsymbol{q}}_b^e$ and the injection term $\hat{\boldsymbol{\sigma}}$ are established, they can be used in estimating the translational motion of the UAV. Additionally, the translational motion observer relies on the specific force measurement $\boldsymbol{f}_{IMU}^b$, the GNSS pseudo-range measurements ρ_i^r and ρ_i^s from eq. (13.1), the GNSS carrier-phase measurements φ_i^r and φ_i^s from eq. (13.2), as well as the UWB pseudo-range measurements μ_j^r and μ_j^s from eq. (13.5). Similarly to [84] the measurement noise is omitted in the analysis of nominal stability, but will be included in the simulations in section 13.5.

By expanding the double-differenced observer from [71] with the UWB injection terms

$$\begin{aligned} e_{\mu,j} &= \Delta\mu_j - \Delta\hat{\mu}_j \\ &= \Delta\mu_j - \Delta\hat{l}_j^r - \Delta\hat{\alpha}_j^r, \end{aligned} \tag{13.11}$$

where $\Delta\hat{l}_j = \hat{l}_j^r - \hat{l}_j^s$, and $\Delta\hat{\alpha} = \hat{\alpha}^r - \hat{\alpha}^s$, the translational motion observer becomes:

$$\dot{\hat{\boldsymbol{p}}}_r^e = \hat{\boldsymbol{v}}_r^e + \sum_{i=1}^{m-1}\left(K_i^{p\rho} e_{\rho,mi} + K_i^{p\varphi} e_{\varphi,mi}\right) + \sum_{j=1}^{n}\left(K_j^{p\mu} e_{\mu,j}\right), \tag{13.12a}$$

$$\dot{\hat{\boldsymbol{v}}}_r^e = -2S(\boldsymbol{\omega}_{ie}^e)\hat{\boldsymbol{v}}_r^e + \hat{f}^e + g^e(\hat{\boldsymbol{p}}_r^e) + \sum_{i=1}^{m-1}\left(K_i^{v\rho} e_{\rho,mi} + K_i^{v\varphi} e_{\varphi,mi}\right) + \sum_{j=1}^{n}\left(K_j^{v\mu} e_{\mu,j}\right), \tag{13.12b}$$

$$\dot{\boldsymbol{\xi}} = -\boldsymbol{R}(\hat{\boldsymbol{q}}_b^e)S(\hat{\boldsymbol{\sigma}})\boldsymbol{f}_{IMU}^b + \sum_{i=1}^{m-1}\left(K_i^{\xi\rho} e_{\rho,mi} + K_i^{\xi\varphi} e_{\varphi,mi}\right) + \sum_{j=1}^{n}\left(K_j^{\xi\mu} e_{\mu,j}\right), \tag{13.12c}$$

$$\hat{\boldsymbol{f}}^e = \boldsymbol{R}(\hat{\boldsymbol{q}}_b^e)\boldsymbol{f}_{IMU}^b + \boldsymbol{\xi}, \tag{13.12d}$$

$$\dot{\hat{\boldsymbol{p}}}_s^e = \sum_{i=1}^{m-1}\left(K_i^{s\rho} e_{\rho,mi} + K_i^{s\varphi} e_{\varphi,mi}\right) + \sum_{j=1}^{n}\left(K_j^{s\mu} e_{\mu,j}\right), \tag{13.12e}$$

$$\Delta\dot{\hat{\alpha}} = \sum_{i=1}^{m}\left(K_i^{\alpha\rho} e_{\rho,mi} + K_i^{\alpha\varphi} e_{\varphi,mi}\right) + \sum_{j=1}^{n}\left(K_j^{\alpha\mu} e_{\mu,j}\right), \tag{13.12f}$$

$$\nabla\Delta\dot{\hat{N}} = \sum_{i=1}^{m-1}\left(K_i^{Np} e_{\rho,mi} + K_i^{N\varphi} e_{\phi,mi}\right) + \sum_{j=1}^{n}\left(K_j^{N\mu} e_{\mu,j}\right). \tag{13.12g}$$

where K_*^{**} are gains, and e_* are injection terms. The injection terms are defined as the difference between the true and the estimated measurements: $e_{\rho,mi} := \nabla\Delta\rho_{mi} - \nabla\Delta\hat{\rho}_{mi}$ and $e_{\varphi,mi} := \nabla\Delta\varphi_{mi} - \nabla\Delta\hat{\varphi}_{mi}$, in which the estimated double-differenced pseudo-range, carrier-phase and geometric range are given by

$$\begin{aligned}
\nabla\Delta\hat{\rho}_{mj} &= \nabla\Delta\hat{\psi}_{mj} \\
\nabla\Delta\hat{\varphi}_{mj} &= \nabla\Delta\hat{\psi}_{mj} + \nabla\Delta\hat{N}_{mj}\lambda \\
\nabla\Delta\hat{\psi}_{mj} &= \|\hat{\boldsymbol{p}}_r^e - \boldsymbol{p}_j^e\|_2 - \|\hat{\boldsymbol{p}}_r^e - \boldsymbol{p}_m^e\|_2 - \|\hat{\boldsymbol{p}}_s^e - \boldsymbol{p}_j^e\|_2 + \|\hat{\boldsymbol{p}}_s^e - \boldsymbol{p}_m^e\|_2.
\end{aligned}$$

To analyze the stability of the error state $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}} = \tilde{\boldsymbol{x}} = \left[\tilde{\boldsymbol{p}}_r, \tilde{\boldsymbol{v}}_r, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{p}}_s, \Delta\tilde{\alpha}, \nabla\Delta\tilde{N}\right]^\intercal$, with $\nabla\Delta\tilde{N}_{mi} = \nabla\Delta N_{mi} - \nabla\Delta\hat{N}_{mi}$, it is desirable to put the system on the closed loop form $\dot{\tilde{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\,\tilde{\boldsymbol{x}}$. Thus, it is necessary to express the injection terms by the error state $\tilde{\boldsymbol{x}}$:

$$e_{\mu,j} = \|\boldsymbol{p}_r^e - \boldsymbol{p}_j^e\|_2 - \|\boldsymbol{p}_s^e - \boldsymbol{p}_j^e\|_2 - \|\hat{\boldsymbol{p}}_r^e - \boldsymbol{p}_j^e\|_2 + \|\hat{\boldsymbol{p}}_s^e - \boldsymbol{p}_j^e\|_2 + \Delta\tilde{\alpha}. \tag{13.13}$$

Further, following [71], the norm $h(\boldsymbol{p}_r^e) = \|\boldsymbol{p}_r^e - \boldsymbol{p}_j^e\|_2$ is approximated by a second order Taylor approximation.

$$h(\boldsymbol{p}_r^e) = \|\hat{\boldsymbol{p}}_r^e - \boldsymbol{p}_j^e\|_2 + c_j^{r\intercal}\tilde{\boldsymbol{p}}_r^e + \text{h.o.t.}, \tag{13.14}$$

where $c_\dagger^* = \frac{\hat{\boldsymbol{p}}_*^e - \boldsymbol{p}_\dagger^e}{\|\hat{\boldsymbol{p}}_*^e - \boldsymbol{p}_\dagger^e\|_2}$, with $\dagger \in \{i, j\}$ and $* \in \{r, s\}$, is a general expression for the estimated line-of-sight vector between $*$ and $\dagger$. Through the same steps, a similar

approximation can be found for the base station: $h(\boldsymbol{p}_s^e) = \|\hat{\boldsymbol{p}}_s^e - \boldsymbol{p}_j^e\|_2 + c_j^s \tilde{\boldsymbol{p}}_s^e + \text{h.o.t.}$. The coefficients are combined in $\boldsymbol{C}_{\mu,j} = \begin{bmatrix} c_j^r, 0, 0, -c_j^s, 0, 1, 0 \end{bmatrix}$, $\boldsymbol{C}_{\rho,mi} = \begin{bmatrix} c_{mi}^r, 0, 0, -c_{mi}^s, 0, 0 \end{bmatrix}$, $\boldsymbol{C}_{\varphi,mi} = \begin{bmatrix} c_{mi}^r, 0, 0, -c_{mi}^s, 0, \lambda 1_{i,m-1} \end{bmatrix}$, with $1_{i,m-1} = [0, \ldots, 1, \ldots, 0]$ describing an $(m-1)$-dimensional zero-vector with a non-zero ith element. The double-differenced coefficients are given as[71]:

$$c_{mi}^r = \frac{\hat{p}_r^e - \boldsymbol{p}_i^e}{\|\hat{p}_r^e - \boldsymbol{p}_i^e\|_2} - \frac{\hat{p}_r^e - \boldsymbol{p}_m^e}{\|\hat{p}_r^e - \boldsymbol{p}_m^e\|_2}, \quad c_{mi}^s = -\frac{\hat{p}_s^e - \boldsymbol{p}_i^e}{\|\hat{p}_s^e - \boldsymbol{p}_i^e\|_2} + \frac{\hat{p}_s^e - \boldsymbol{p}_m^e}{\|\hat{p}_s^e - \boldsymbol{p}_m^e\|_2}.$$

The injection terms can then be expressed as

$$\begin{aligned} e_{\mu,j} &= c_j^r \tilde{\boldsymbol{p}}_r + c_j^s \tilde{\boldsymbol{p}}_s + \Delta\tilde{\alpha} \ + \text{h.o.t.} \\ &= \boldsymbol{C}_{\mu,j} \tilde{x} + \varepsilon_{l,j} + \varepsilon_{l_s,j}, \end{aligned} \tag{13.15}$$

$$e_{\rho,m,i} = \boldsymbol{C}_{\rho,m,i} \tilde{x} + \varepsilon_{\psi,i} + \varepsilon_{\psi_s,i}, \tag{13.16}$$

$$e_{\varphi,m,i} = \boldsymbol{C}_{\varphi,m,i} \tilde{x} + \varepsilon_{\psi,i} + \varepsilon_{\psi_s,i}, \tag{13.17}$$

where ε_* are the linearization errors from truncating the higher order terms in the Taylor expansion. By the same arguments that [86] applies for GNSS pseudo-range, the UWB linearization error is assumed. The linearization errors are bounded as:

$$\|\varepsilon_{l,j}\| \leq \frac{1}{\underline{l}} \|\tilde{\boldsymbol{p}}_r\|_2^2, \qquad \|\varepsilon_{l_s,j}\| \leq \frac{1}{\underline{l}_s} \|\tilde{\boldsymbol{p}}_r\|_2^2, \qquad \|\varepsilon_{\psi,i}\| \leq \frac{1}{\underline{\psi}} \|\tilde{\boldsymbol{p}}_r\|_2^2, \qquad \|\varepsilon_{\psi_s,i}\| \leq \frac{1}{\underline{\psi}_s} \|\tilde{\boldsymbol{p}}_r\|_2^2 \tag{13.18}$$

where $\underline{l}$ and $\underline{l}_s$ are positive, lower bounds on the distance between the UWB nodes and rover, and base station. Similarly $\underline{\psi}$ and $\underline{\psi}_s$ are positive, lower bounds on the geometric range between the satellites and rover, and base station.

Now the convergence of the estimates can be analyzed, by expressing the error dynamics as:

$$\dot{\tilde{\boldsymbol{x}}} = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\tilde{\boldsymbol{x}} + \boldsymbol{\delta}_1(t, \tilde{\boldsymbol{x}}) + \boldsymbol{\delta}_2(t, \tilde{\boldsymbol{\chi}}) + \boldsymbol{\delta}_3(t, \tilde{\boldsymbol{x}}), \tag{13.19}$$

where $\tilde{\boldsymbol{\chi}} = \begin{bmatrix} \boldsymbol{q}_v, b^b \end{bmatrix}^\intercal$, where $\boldsymbol{q}_v$ is the vector component of the quaternion. As no additional dynamics are introduced by the UWB measurements, this is similar to [71]. All the terms have been augmented to accommodate the inclusion of the UWB bias term $\Delta\alpha$ to the state-space. [57] shows that $\delta_1(t, \tilde{x}) = [0; \delta_{12}(t, \tilde{\boldsymbol{x}}); 0; 0; 0; 0]$, where $\delta_{12}(t, \tilde{\boldsymbol{x}}) = -S(\boldsymbol{\omega}_{ie}^e)\tilde{\boldsymbol{x}}_2 + \left(\boldsymbol{g}^e(\boldsymbol{p}_r^e) - \boldsymbol{g}^e(\boldsymbol{p}_r^e - \tilde{\boldsymbol{x}}_1)\right)$, and further that $\delta_2(t, \tilde{\boldsymbol{\chi}}) = \begin{bmatrix} 0; 0; 0; \tilde{d}; 0; 0 \end{bmatrix}$, with

$$\begin{aligned} \tilde{\boldsymbol{d}} = &\left(\boldsymbol{I} - \boldsymbol{R}(\tilde{\boldsymbol{q}}_b^e)^\intercal\right)\boldsymbol{R}(\boldsymbol{q}_b^e)\left(S(\boldsymbol{\omega}_{ib}^b)\boldsymbol{f}^b + \dot{\boldsymbol{f}}^b\right) \\ &- S(\boldsymbol{\omega}_{ie}^e)\left(\boldsymbol{I} - \boldsymbol{R}(\tilde{\boldsymbol{q}}_b^e)^\intercal\right)\boldsymbol{R}(\boldsymbol{q}_b^e)\boldsymbol{f}^b - \boldsymbol{R}(\tilde{\boldsymbol{q}}_b^e)^\intercal \boldsymbol{R}(\boldsymbol{q}_b^e) S(\boldsymbol{b}^b)\boldsymbol{f}^b. \end{aligned} \tag{13.20}$$

The most significant changes are that the matrices $\boldsymbol{K}$ and $\boldsymbol{C}$ have been expanded to include the gains and linearized coefficients associated with the UWB injection terms, and that $\delta_3(t, \tilde{\boldsymbol{x}}) = K\boldsymbol{\varepsilon}(t, \tilde{\boldsymbol{x}})$ includes the UWB linearization terms $\varepsilon_{\mu,j}$ from

eq. (13.15) in addition to the GNSS linearization terms $\varepsilon_{\rho,i}$ and $\varepsilon_{\varphi,i}$ from eq. (13.16) and eq. (13.17). The matrices are:

$$\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{I}_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{13.21a}$$

$$\boldsymbol{K} = \begin{bmatrix} K_1^{p\rho} & \cdots & K_{m-1}^{p\rho} & K_1^{p\varphi} & \cdots & K_{m-1}^{p\varphi} & K_1^{p\mu} & \cdots & K_n^{p\mu} \\ K_1^{\nu\rho} & \cdots & K_{m-1}^{\nu\rho} & K_1^{\nu\varphi} & \cdots & K_{m-1}^{\nu\varphi} & K_1^{\nu\mu} & \cdots & K_n^{\nu\mu} \\ K_1^{\zeta\rho} & \cdots & K_{m-1}^{\zeta\rho} & K_1^{\zeta\varphi} & \cdots & K_{m-1}^{\zeta\varphi} & K_1^{\zeta\mu} & \cdots & K_n^{\zeta\mu} \\ K_1^{s\rho} & \cdots & K_{m-1}^{s\rho} & K_1^{s\varphi} & \cdots & K_{m-1}^{s\varphi} & K_1^{s\mu} & \cdots & K_n^{s\mu} \\ K_1^{\alpha\rho} & \cdots & K_{m-1}^{\alpha\rho} & K_1^{\alpha\varphi} & \cdots & K_{m-1}^{\alpha\varphi} & K_1^{\alpha\mu} & \cdots & K_n^{\alpha\mu} \\ K_1^{N\rho} & \cdots & K_{m-1}^{N\rho} & K_1^{N\varphi} & \cdots & K_{m-1}^{N\varphi} & K_1^{N\mu} & \cdots & K_n^{N\mu} \end{bmatrix}, \tag{13.21b}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{\rho,m,1}; & \cdots & \boldsymbol{C}_{\rho,m,m-1}; & \boldsymbol{C}_{\varphi,m,1}; & \cdots & \boldsymbol{C}_{\varphi,m,m-1}; & \boldsymbol{C}_{\mu,1}; & \cdots & \boldsymbol{C}_{\mu,n} \end{bmatrix}. \tag{13.21c}$$

Here, n is the number of UWB-nodes, while m is the number of available satellites, which both may change with every epoch.

Due to the inclusion of the UWB injection terms, the proof of Proposition 1 in [86] is insignificantly modified. Firstly, eq. (13.19) is transformed by $\boldsymbol{\eta} = \boldsymbol{L}_\theta \tilde{\boldsymbol{x}}$ to assign a desirable time-scale structure to the dynamics. Here,

$$\boldsymbol{L}_\theta = \operatorname{diag}\left(\boldsymbol{I}_3, \frac{1}{\theta}\boldsymbol{I}_3, \frac{1}{\theta^2}\boldsymbol{I}_3, \frac{1}{\theta^3}\boldsymbol{I}_3, \frac{1}{\theta^4}1, \frac{1}{\theta^5}\boldsymbol{I}_n, \frac{1}{\theta^6}\boldsymbol{I}_m \right) \tag{13.22}$$

to include the additional states. By following [86], the derivative of the Lyapunov function candidate $U(\boldsymbol{\eta}, t) = \frac{1}{\theta}\boldsymbol{\eta}^\intercal \mathscr{P}^{-1}\boldsymbol{\eta}$ becomes:

$$\begin{aligned} \dot{U} \leq & -\gamma_1 \|\boldsymbol{\eta}\|_2^2 + \frac{2}{\theta}\|\boldsymbol{\eta}\|_2 \cdot \|\boldsymbol{C}^\intercal \mathscr{R}^{-1}\| \cdot \|\boldsymbol{E}_\theta\| \cdot \left(2\sum_{i=0}^{m-1}(\varepsilon_{\psi,i}^2 + \varepsilon_{\psi,i}^2) + \sum_{j=0}^{n}(\varepsilon_{l,j}^2 + \varepsilon_{l_s,j}^2) \right) \\ & + \frac{1}{\theta}\gamma_2\gamma_4\|\boldsymbol{\eta}\|_2^2 + \frac{1}{\theta^3}\gamma_3\gamma_4\|\boldsymbol{\eta}\|_2 \cdot \|\tilde{\boldsymbol{\chi}}\|_2, \end{aligned}$$

where $\boldsymbol{E}_\theta = \boldsymbol{C}L_\theta\boldsymbol{C}^+$, and where $\boldsymbol{C}^+$ is the Moore-Penrose right pseudo-inverse of $\boldsymbol{C}$. The constants γ_1, γ_2, γ_3 and γ_4 are positive constants, independent of θ. The rest of the proof follows [86], yielding exponentially stable error dynamics with a local region of attraction with respect to the initialization point. Here it is assumed that the UWB and GNSS ranges are upper and lower bounded, that their range measurements come from at least five satellites, where at least three of the LOS vectors are linearly independent, and that the position of their transponders are known. Further it is assumed that the specific force and its derivative are bounded, that $(\boldsymbol{A}, \mathscr{R}^{-\frac{1}{2}}\boldsymbol{C})$ is completely uniformly observable, that the initial conditions are nice, and that the observer gains are chosen appropriately, see Assumptions 1 and 3 − 8 in [86] for details.

13.4.3 Riccati Solver

The few gains in the attitude observer are directly tuned, however, for the translational motion observer the gains must be time-varying and the large number of gains require other methods.

One approach to determine the gains is the Riccati solver, also utilized by the KF, where the discrete-time time-varying Riccati equation is solved giving the gain and the covariance matrices, $\boldsymbol{K}$ and $\mathscr{P}$. The gain matrix is determined using the symmetric, positive definite matrices $\mathscr{Q}$ and $\mathscr{R}$, which can be interpreted as the covariance of the process and measurement noise. The matrices $\mathscr{P}$, $\mathscr{Q}$ and $\mathscr{R}$ can be stated similarly to when used in a KF, as the observer error dynamics is the same as for the KF thereby allowing the observer gain to stabilize and tune the error dynamics. The matrices are determined using an iterative process for each discrete time instance, k:

$$\mathscr{P}_{k|k-1} = \boldsymbol{A}_d \mathscr{P}_{k-1|k-1} \boldsymbol{A}_d^\intercal + \mathscr{Q}^\intercal, \tag{13.23a}$$

$$\boldsymbol{K}_k = \mathscr{P}_{k|k-1} \boldsymbol{C}^\intercal \left(\boldsymbol{C} \mathscr{P}_{k|k-1} \boldsymbol{C}^\intercal + \mathscr{R} \right)^{-1}, \tag{13.23b}$$

$$\mathscr{P}_{k|k} = (I - \boldsymbol{K}_k \boldsymbol{C}) \mathscr{P}_{k|k-1} (I - \boldsymbol{K}_k \boldsymbol{C})^\intercal + \boldsymbol{K}_k \mathscr{R} \boldsymbol{K}_k^\intercal. \tag{13.23c}$$

Here, $\boldsymbol{A}_d = e^{\boldsymbol{A}\Delta t_{IMU}}$ is the discretized A matrix, where Δt_{IMU} is the IMU sample rate. The Riccati solver can be implemented such that the covariance for the state estimate, $\mathscr{P}_{k|k-1}$, $\mathscr{P}_{k|k}$ and the gain matrix $\boldsymbol{K}_k$ are only updated when new range measurements arrive.

For the stability proof it is assumed that the gain matrix is found from the transformed error dynamics, [86]:

$$\boldsymbol{K} := \theta \boldsymbol{L}_\theta^{-1} \boldsymbol{K}_0 \boldsymbol{E}_\theta, \tag{13.24}$$

where $\theta \geq 1$ is a tunings parameter, while $\boldsymbol{K}_0 := \mathscr{P} \boldsymbol{C}^\intercal \mathscr{R}^{-1}$ where $\mathscr{P}$ satisfies the time-scaled Riccati equation.

An accurate initialization method was proposed in [86], ensuring stability when the gain matrix $\boldsymbol{K}$ is determined by solving the time-varying Riccati equation.

13.5 Simulations

This section offers a simulation study of a UAV operating in GNSS denied or challenged areas. The goal of the simulations is to evaluate the UWB as aiding sensor and determine the effect it contributes to the state estimation.

13.5.1 Implementation

To simulate sensor data from a UAV, the simulator developed in [14] is used with an Aerosonde UAV model. The measurements are generated from eqs. (13.1), (13.2) and (13.5), at 5 Hz for the GNSS and 10 Hz for the UWB. Here, ψ_i^r and l_j^r are calculated from the known satellite and UWB node positions, with λ set to 0.1903 m. Measurement noise is added to the range measurements, where the GNSS noise are first order Markov processes and the UWB noise is white noise. The Markov processes have a time constant of 60s, and is generated from white noise with standard deviation of 5 m. Additional white noise is added to the GNSS measurements to simulate receiver noise with standard deviations of 0.10 m and 0.001 m for the pseudo-range and carrier-phase measurements, respectively. The white noise on the UWB measurements are simulated with standard deviation of 0.026 m corresponding to the BeSpoon UM100 UWB module, whereas the IMU data is simulated with noise levels corresponding to the Analog Devices 16488 IMU; acceleration: 0.0015m/s^2, angular rate: 0.16°/s and magnetometer: 0.45mGauss.

The nonlinear observer is implemented in a corrector-predictor architecture [50, p. 300], where the corrector part consists of the aiding measurements and can be implemented at low rate, while the predictor part is implemented at IMU frequency. The observer runs at 400 Hz, which is the frequency of the IMU, while GNSS and UWB measurements arrive at their appropriate frequencies. When GNSS/UWB measurements are not received, the corresponding elements in the measurement matrix $\boldsymbol{C}$, in the gain matrix $\boldsymbol{K}$, and in the injection terms e are masked out.

For the attitude observer the parameters are chosen as $M_b = 0.0087, k_1 = 0.8, k_2 = 0.2, k_I = 0.004$. The gains in the translational motion observer are calculated using the discrete time-varying Riccati equation eq. (13.23). The covariance matrices can be initialized as diagonal matrices, where the diagonal elements correspond to the individual state or measurement variance: In this simulation, the covariance matrices are given by $\mathscr{R} = \text{blkdiag}(0.2I_{m-1}, 2 \cdot 10^{-6} I_{m-1}, 0.956 \cdot 10^{-4} I_n)$ and $\mathscr{Q} = \text{blkdiag}(10^{-18} I_3, 10^{-5} I_3, 2.5 \cdot 10^{-8} I_3, 10^{-18} I_3, 10 I_n, 10^{-7} I_{m-1})$.

13.5.2 GNSS Dropout during Flight

During an outage in GNSS coverage the inertial navigation estimates will drift from the true value. It is therefore of interest to see how the addition of UWB measurements might prevent the estimates from drifting in GNSS challenged areas. This scenario compares the position estimates with and without UWB measurements for a simulated fixed-wing UAV in a steady, banked clockwise turn. A simulated dropout of GNSS measurements is enforced in the time interval from 35s to 60s, as indicated by the solid vertical lines in figs. 13.3 to 13.6. Similarily the UAV is within range of the UWB nodes in the approximate interval of 30 to 62s, as indicated by the dashed vertical lines in the same figures.

In this simulation, five UWB node are placed in the vicinity of the flight path. Four of the nodes form a square of height 10 m and width 600 m, placed vertically and perpendicular to the flight path. The fifth node is placed perpendicular to the centre of the square, in center height 500 m behind the square. The range of the UWB nodes is set to 700 m.

Two simulations are included: a) using only GNSS as aiding, and b) using GNSS and UWB as aiding. The results of the simulations are shown in fig. 13.3 and fig. 13.4, depicting the position estimation error in NED-frame. The estimation error of the base station is shown in dashed lines, with the rover estimation errors shown in solid lines. The period of GNSS dropout is shown between the two vertical solid lines, whereas the period in which the UWB measurements are available is shown with vertical dashed lines.

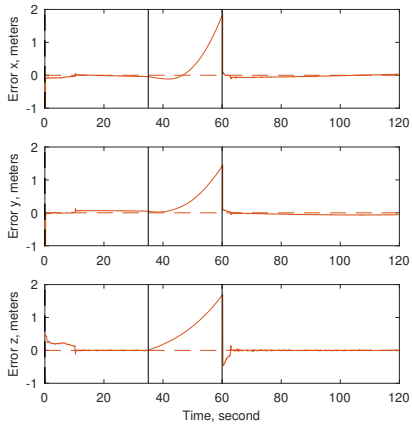


Figure 13.3: Position error using only GNSS aiding, GNSS dropout from 35*s* to 60*s*

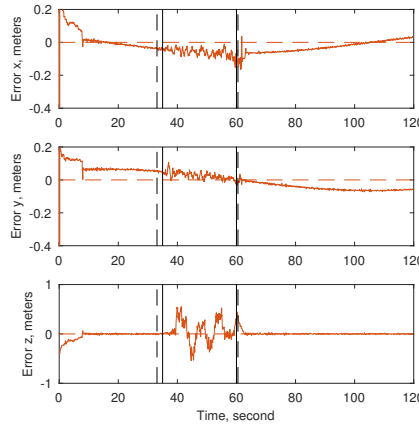


Figure 13.4: Position error using GNSS and UWB aiding, GNSS dropout from 35*s* to 60*s*

The estimation error in fig. 13.3 is seen to divert due to the missing aiding, whereas the additional use of UWB nodes in fig. 13.4 ensures a smaller diversion from the desired position (note the difference in scale on the vertical axes). For the case with UWB available the error of the rover position is less than 0.53 m away from the true position, whereas the case with only GNSS aiding experience rover position errors exceeding 2.91 m. In both cases, the position error is quickly reduced once the GNSS measurements are available again.

The errors in attitude estimation are shown in fig. 13.5 and fig. 13.6, and is expressed in Euler angles in the Body-frame. The qualitative behavior of the attitude error is not visibly affected by the GNSS dropout, neither with nor without UWB aiding.

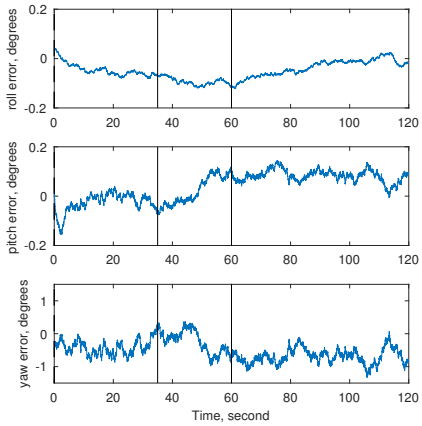


Figure 13.5: Attitude error using only GNSS aiding, GNSS dropout from 35*s* to 60*s*

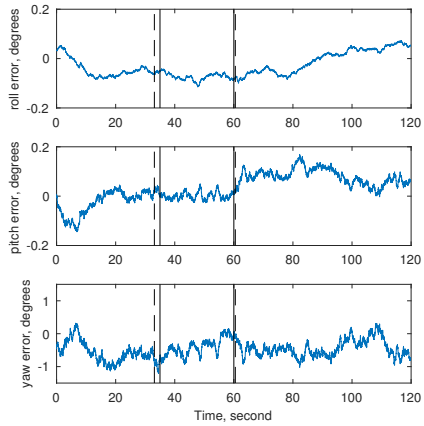


Figure 13.6: Attitude error using only GNSS and UWB aiding, GNSS dropout from 35*s* to 60*s*

13.5.3 Landing of UAV in GNSS Challenged Area

In order to test the performance of the proposed observer in GNSS challenged area, a landing scenario is investigated. The UAV starts with a relative altitude of 4 m to the desired landing zone. For this scenario the four first nodes form a square of size 5×5 m, which represent the corners of a net for the UAV to land in. This square is placed vertically and perpendicular to the flight path such that the landing target position of the UAV is in the center of the square. The fifth node is placed perpendicular to the centre of the square, 15 m behind it. The GNSS measurements are simulated with multipath errors when the UAV approaches landing. The multipath is simulated as additional Markov noise, which is generated from white noise with standard deviation of 1 m and 0.1 m for the pseudo-range and carrier-phase respectively. The Markov process has a time constant of 60s. Due to the increased GNSS noise, the R-elements corresponding to the GNSS carrier-phase and pseudo-range measurement were multiplied by a factor of 50, while all other tuning parameters remained the same.

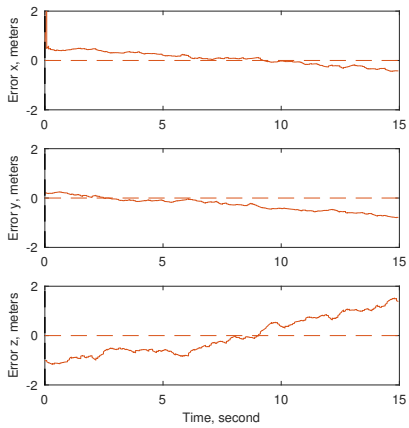


Figure 13.7: Position error using only GNSS aiding, simulated GNSS multipath

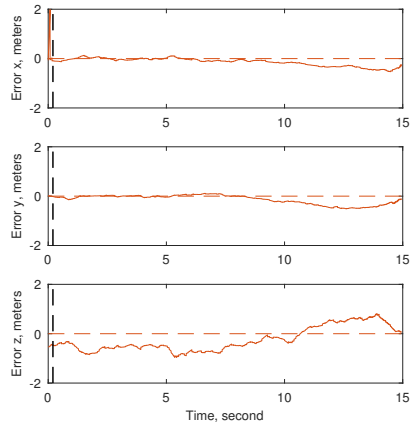


Figure 13.8: Position error using GNSS and UWB aiding, simulated GNSS multipath

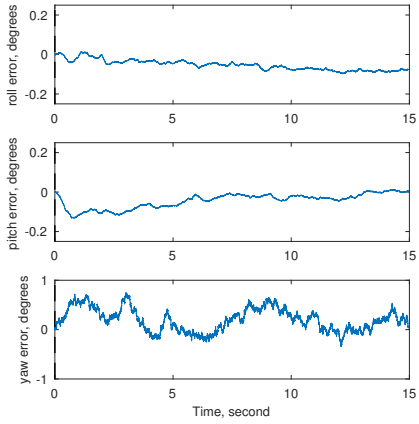


Figure 13.9: Attitude error using only GNSS aiding, simulated GNSS multipath

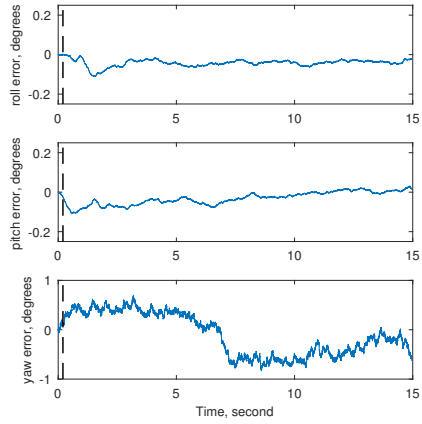


Figure 13.10: Attitude error using GNSS and UWB aiding, simulated GNSS multipath

The error in the estimated position without UWB aiding, shown in fig. 13.7, generally has a larger magnitude than for the UWB aided case in fig. 13.8. This is especially true for the final stage of the landing approach where the error in the estimated position is 0.32 m for the final position in the UWB aided case, whereas in the non-UWB case the norm is 1.60 m. A viable explanation for the improvements for the UWB aided case is the improved geometry configuration as the UAV gets closer to the landing target, since the angle between the UWB range measurements increase

Table 13.1: End position mean square errors (RMSE), landing scenario

	Rover Position (m)		
	x	y	z
GNSS	$4.31 \cdot 10^{-1}$	$5.34 \cdot 10^{-1}$	1.05
GNSS+UWB	$3.70 \cdot 10^{-1}$	$2.43 \cdot 10^{-1}$	$2.50 \cdot 10^{-1}$

Table 13.2: Position root mean square errors (RMSE), circle scenario.

	Rover Position(m)		
	x	y	z
GNSS	$5.03 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	$9.31 \cdot 10^{-3}$
GNSS+UWB	$5.02 \cdot 10^{-2}$	$4.66 \cdot 10^{-2}$	$9.05 \cdot 10^{-3}$

with decreasing distance to the net. The attitude errors for the two cases are of similar magnitude.

13.5.4 Monte Carlo Simulation

To compare the results with and without UWB, 100 simulations of the landing scenario in section 13.5.3 are run. For each set of simulated measurements, the observer is run once with and once without UWB measurements. Figure 13.11 shows a comparison of the root mean square error (RMSE) of the end position for the two cases, while table 13.1 shows the RMSE averaged over the 100 simulations. The inclusion of UWB measurements decreases the rover position RMSE of the end point by close to three orders of magnitude: the norm of the RMSE averaged over all the simulations is reduced from 1.37 m to 0.584 m. The reduction is particularly pronounced in the z-axis. However, the inclusion of UWB increases the norm of the RMSE for the final base station position from $2.40 \cdot 10^{-9}$ m to $1.75 \cdot 10^{-5}$ m^2, but they are both zero for practical considerations.

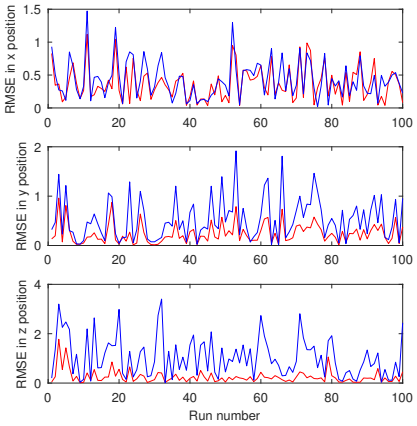


Figure 13.11: Rover position RMSE for the landing scenario, with (red) without (blue) UWB

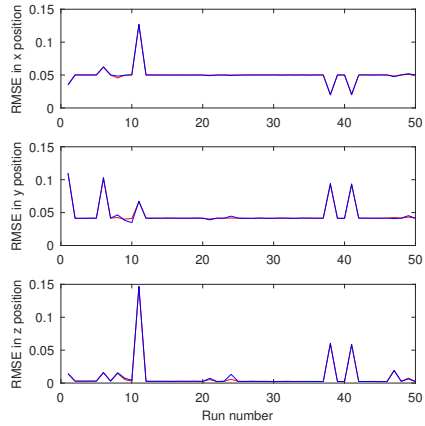


Figure 13.12: Rover position RMSE for the modified circle scenario, with (red) and without (blue) UWB

A similar Monte Carlo simulation is performed on a setup comparable to that of section 13.5.2, but in order to compare the accuracy of the system with and without UWB, the case is assumed more ideal: the UAV is not experiencing any GNSS dropout and the UWB transponders is assumed to have infinite range. The RMSE values, averaged over all the runs, for the base station position and the attitude and position of the rover are found in table 13.2. Due to the longer duration of this test, only 50 Monte Carlo simulations was run for this scenario. The UWB leads to a very slight reduction of the rover position RMSE: the norm is reduced from $7.20 \cdot 10^{-2}m$ to $7.19 \cdot 10^{-2}m$. The RMSE for each run is plotted in fig. 13.12, where the follows the blue almost exact and is thus hardly visible. This limited reduction shows that the position accuracy in the ideal case is not affected by the inclusion of the UWB nodes, since the position accuracy for RTK GNSS is already very good.

13.6 Conclusion

This chapter has presented an extension to a double differenced nonlinear translational motion observer with applications for UAVs. The extension enables tight integration of ultra wideband range measurements with INS and real-time kinematic GNSS. Simulations have showed that the UWB extension increases the operational window of UAVs in GNSS-denied regions, by keeping the position error well within 1 meter, despite loss of GNSS. Also when GNSS measurements are available, but are influenced by multipath, the UWB extension showed improved performance over the GNSS-only case, particularly in the z-axis. Through a Monte Carlo simulation with 100 runs, the root mean square error for the position estimate of a UAV in landing

was reduced by close to one order of magnitude when UWB measurements where included.

Chapter 14

Phased-array radio navigation

This chapter focus on GNSS-denied navigation using inertial navigation aided by a phased-array radio system, and are based on the work from the three papers

- [61] K. Gryte, T. H. Bryne, and T. A. Johansen, "Unmanned aircraft flight control aided by phased-array radio navigation", *Journal of Field Robotics*, 2020, Submitted
- [23] T. H. Bryne, K. Gryte, S. M. Albrektsen, and T. A. Johansen, "GNSS-free navigation of unmanned aerial vehicles based on phased-array radio systems", *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 1–15, 2020, Submitted
- [60] K. Gryte, T. H. Bryne, S. M. Albrektsen, and T. A. Johansen, "Field test results of GNSS-denied inertial navigation aided by phased-array radio systems for UAVs", in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 1398–1406. DOI: 10.1109/ICUAS.2019.8798057

The first part of the chapter gives an introduction to the principles behind phased-array radio systems, before the multiplicative extended Kalman filter is introduced, focusing on how the phased-array radio measurements can be used for navigation. The system is first evaluated on data recorded from two different UAV flights, before a real-time implementation is validated in a beyond-visual-line-of-sight flight in controlled airspace, over open waters, without the use of GNSS for positioning. To safely test the navigation solution a guidance, navigation and control (GNC) system was designed and implemented, where the UAV is controlled based on the estimates from the PARS navigation.

14.1 Introduction

Phased-array radio systems (PARS) have in recent years been tested and demonstrated as a possible alternative positioning system for small UAVs [5, 3, 6]. PARS's primary usage is as a high bandwith radio communication link. For UAVs, the usage of the link can involve providing the user with telemetry data, including a live video streams [77], as well as relaying information to units that are beyond radio line of sight from the base antenna. In addition, such systems utilizes electronic beamforming to avoid omnidirectional transmission, they also provide the user with information which can be used to deduce position of the radio relative the ground station [3]. The PARS-based position is derived from the raw radio system measurements in a similar manner as with standard radar tracking calculations, [12], but with a trivial data association step.

The PARS system utilized in [3] complements the aspect of GNSS standard position service (SPS) solutions lack of security, in addition to the aforementioned redundancy, with high-power radio transmission. The former is achieved through encryption of the radio communication, while the latter ensures a high signal-to-noise ratio, making the PARS PNT signals jamming resistant. PARS does, however, typically have lower position accuracy than GNSS solutions, which depend on the range from the base station. PARS is, as GNSS, dependent on line of sight (LOS), but to the base radio antenna rather than to satellites. PARS is also dependent on locally deployed infrastructure. Previous results achieved with a fixed-wing UAV [5, 3, 6] indicated a horizontal position accuracy ten times lower than what is achieved with the SPS of the Global Positioning System (GPS). These navigation results were obtained using a nonlinear observer, relying on measurements from the PARS, exogenous altitude and a magnetometer.

During UAV flights, knowledge of the aircraft attitude and velocity, in addition to its position, are also necessary since these quantities are part of the autopilot's control objective, ideally at a high rate. Considering this fact, and that the PARS position accuracy is lower than that of GNSS, motivates the use of a PARS-aided inertial navigation system (INS), being based on mechanization of inertial sensors such as accelerometers (ACC) and angular rate sensors (ARS). Usage of INS enables filtering of the PARS position signal while enabling estimation of the aircraft's velocity and attitude. In this chapter the INS aiding is realized using an indirect multiplicative extended Kalman filter (MEKF), with the unit quaternion [116, Ch. 2.7] as attitude representation. The reasons for using the MEKF compared to using nonlinear observers, as in [3, 6], are that with an EKF approach one is able to couple the angular and translational estimation error kinematics. Given sufficient excitation, this enables heading estimation without a dedicated heading sensor. Therefore, this work does not use a heading sensor, such as a magnetic compass or a magnetometer. Both of the mentioned sensors are considered to be unreliable yet they are commonly used to determine heading. The covariance estimates in the Kalman filter also laid the foundation for outlier rejection of the PARS measurements, in the event of signal blockage or severe multipath errors.

The presented solution can be seen as a GNSS-free GNC system opposite to that used for indoor and underground applications, e.g. [95, 129]: Such environments are usually cluttered and hence provides many features for the visual sensors, albeit putting high demands on control and collision avoidance. The scenario presented here is less challenging in terms of control, but the sea surface makes feature detection difficult using visual systems to aid the navigation system. To the best of our knowledge, this system is the first successful civilian GNSS-free UAV GNC system deployed in BVLOS flight in controlled airspace.

14.2 Phased-array radio systems

PARS is primarily designed for communication, where the narrow beam shape allows for high power transmission. In its simplest form, a uniform linear array of D antennas, separated by a distance d_{tx}, the transmission direction Ψ_{tx} can be controlled by adjusting the phase shift for each antenna [118]. The transmitted signal is delayed by an additional δ_t for each antenna, such that the phase shift from one antenna to the next is the same for all the antennas. From fig. 14.1a it is clear that

$$\sin \Psi_{\text{tx}} = \frac{\delta_t c}{d_{\text{tx}}}, \tag{14.1}$$

where $c = f\lambda$ is the signal speed, expressed in terms of frequency f and wavelength λ. Rearranging, and realizing that phase can be expressed in terms of time delay through $\phi = 2\pi f \delta_t$, results in

$$\Psi_{\text{tx}} = \arcsin\left(\frac{\phi\lambda}{2\pi d_{\text{tx}}}\right). \tag{14.2}$$

The inverse problem, estimating the direction Ψ_{rx} of a received signal based on its phase shift at the different antennas in the array, see fig. 14.1a, is known as the direction-of-arrival (DoA) problem [25, 69], which is based on the measurement model

$$\begin{gathered} \boldsymbol{X} = \boldsymbol{C}\boldsymbol{F} + \boldsymbol{W} \\ \begin{bmatrix} X_1 \\ \vdots \\ X_M \end{bmatrix} = \begin{bmatrix} a(\boldsymbol{\theta}_1) & \cdots & a(\boldsymbol{\theta}_D) \end{bmatrix} \begin{bmatrix} F_1 \\ \vdots \\ F_D \end{bmatrix} + \begin{bmatrix} W_1 \\ \vdots \\ W_M \end{bmatrix} \end{gathered} \tag{14.3}$$

The vector of measurements, $\boldsymbol{X}$, contains the signals measured at all the M antennas. They measure how the D signals in $\boldsymbol{F}$ are affected by the accompanying steering vector $a(\boldsymbol{\theta}_i)$ and the additive noise $\boldsymbol{W}$. Here, i is the index representing one of the D transmitting antenna arrays. The steering vector is a known function of the variable vector $\boldsymbol{\theta}_i$, to be identified. For the simplest case of a linear, equidistant array, the steering vector is only dependent on $\boldsymbol{\theta}_i = \left[\Psi_{\text{rx},i}\right]$, and can be expressed as

$$a(\boldsymbol{\theta}_i) = \left[1, e^{-j2\pi f_c \frac{d_{\text{rx}} \cos\left(\Psi_{\text{rx},i}\right)}{c}}, e^{-j2\pi f_c \frac{2d_{\text{rx}} \cos\left(\Psi_{\text{rx},i}\right)}{c}}, \cdots, e^{-j2\pi f_c \frac{(M-1)d_{\text{rx}} \cos\left(\Psi_{\text{rx},i}\right)}{c}}\right]. \tag{14.4}$$

The DoA problem for estimating Ψ_{rx} [100] can be solved by e.g. MUSIC [166], ESPRIT [157] and SAMV [2]. As $\boldsymbol{\theta}_{\text{sph}} = [\Psi_{\text{rx}}, \alpha_{\text{rx}}, \rho]$, i.e. the azimuth angle, elevation angle and geometric range, whose computation is inherent to the PARS communication solution, represent a full 3D position of the transmitting antenna in the frame of the receiving antenna, it can also be used as a position measurement for a UAV.

The geometric range, azimuth angle and elevation angle, given in the $\{r\}$-frame, are in the following assumed to originate from any DoA algorithm on the basis of eq. (14.3), and are assumed to be affected by additive noise, according to

$$y_\rho = \rho_u + \varepsilon_\rho, \tag{14.5}$$

$$y_\Psi = \Psi_u + \varepsilon_\Psi, \tag{14.6}$$

$$y_\alpha = \alpha_u + \varepsilon_\alpha, \tag{14.7}$$

with $\varepsilon_\bullet \sim \mathcal{N}(0, \sigma_\bullet^2)$. Here the subscript u symbolizes the UAV, while Ψ_u is shorthand notation for $\Psi_{\text{rx},1}$, assuming that a single UAV is the only transmitting source. The exogenous height measurement is denoted $-p^n_{nb,z,\text{exo}}$. Furthermore

$$\begin{aligned}\rho_u &= \|\boldsymbol{p}^r_{\text{PARS}}\|_2, \\ &= \sqrt{\left(p^r_{rb,x}\right)^2 + \left(p^r_{rb,y}\right)^2 + \left(p^r_{rb,z}\right)^2},\end{aligned} \tag{14.8}$$

$$\Psi_u = \text{atan2}\left(p^r_{rb,y}, p^r_{rb,x}\right), \tag{14.9}$$

$$\alpha_u = \arctan\left(\frac{-p^r_{rb,z}}{\bar{\rho}_u}\right) \tag{14.10}$$

where $\boldsymbol{p}^r_{\text{PARS}} = \begin{bmatrix} p^r_{rb,x} & p^r_{rb,y} & p^r_{rb,z} \end{bmatrix}$ is the true position of the UAV in the $\{r\}$-frame, see fig. 14.1.

Two slightly different ways to utilize the PARS measurements for navigation are presented in the following, based on transformations from spherical and cylinder coordinates, respectively, to Cartesian coordinates.

14.2.1 Spherical-coordinates-based PARS positioning

The first approach of deriving position from the PARS measurements is based on a spherical-to-Cartesian coordinate transformation.

An illustration of the of the geometric range ρ_u, elevation angle α_u and azimuth angle Ψ_u, relative to the $\{r\}$-frame, can be seen in fig. 14.1b. Moreover, the relationships of eqs. (14.8) to (14.10) are similar to those in [187, Ch. 13.6.2.2], used for radar tracking of aircraft, and can be derived from

$$\boldsymbol{p}^r_{\text{PARS},s} = \begin{bmatrix} p^r_{rb,x} \\ p^r_{rb,y} \\ p^r_{rb,z} \end{bmatrix} = \begin{bmatrix} \rho_u \cos(\Psi_u)\cos(\alpha_u) \\ \rho_u \sin(\Psi_u)\cos(\alpha_u) \\ -\rho_u \sin(\alpha_u) \end{bmatrix}, \tag{14.11}$$

which can be deduced from fig. 14.1b, where s symbolizes that eq. (14.11) is based on spherical coordinates. Based on the respective spherical coordinate measurements, y_ρ, y_Ψ, y_α, one can calculate the UAV position based on eq. (14.11). However, as pointed out in [12, Section 1.7.4], the nonlinear mapping of the azimuth and elevation angle measurement noise into Cartesian coordinates leads to a bias in the position calculation. This can be corrected for by replacing eq. (14.11) with

$$\bar{\boldsymbol{p}}^r_{\text{PARS},s} = \begin{bmatrix} b_\Psi^{-1} b_\alpha^{-1} y_\rho \cos(y_\Psi)\cos(y_\alpha) \\ b_\Psi^{-1} b_\alpha^{-1} y_\rho \sin(y_\Psi)\cos(y_\alpha) \\ -b_\alpha^{-1} y_\rho \sin(y_\alpha) \end{bmatrix}, \tag{14.12}$$

to derive the position.

Debiasing the measurements does not only improve accuracy, but also makes it more suitable for Kalman filtering, whose core assumption does not hold for colored noise. The bias estimates are given as $b_\Psi = \mathbb{E}[\cos(\varepsilon_\Psi)] = e^{-\sigma_\Psi^2/2}$ and $b_\alpha = \mathbb{E}[\cos(\varepsilon_\alpha)] = e^{-\sigma_\alpha^2/2}$, as given in [12, Section 1.7.4] and references therein. Based on eq. (14.12), the debiased PARS position measurement is given in the $\{n\}$ frame with

$$\boldsymbol{y}^n_{\text{PARS}} = \boldsymbol{R}_{nr}(\boldsymbol{\Theta}_{\text{PARS}})\bar{\boldsymbol{p}}^r_{\text{PARS},s}, \tag{14.13}$$

where $\boldsymbol{\Theta}_{\text{PARS}}$ represents rotation angles of $\{r\}$ about $\{n\}$. $\boldsymbol{\Theta}_{\text{PARS}}$ is obtained from pose calibration of the PARS ground antenna, [5].

When calculating the aircraft position, information about the covariance of the measurement is often desirable, particularly when using the measurements in a Kalman filter. This entails mapping the covariance from spherical coordinates to Cartesian coordinates. The first step of this procedure is carried out by linearizing $\bar{\boldsymbol{p}}^r_{\text{PARS},s}$ w.r.t. to the measurement noise [12, Ch. 1.7], $\boldsymbol{\varepsilon}_s = \begin{bmatrix} \varepsilon_\rho; & \varepsilon_\Psi; & \varepsilon_\alpha \end{bmatrix}$ in order to calculate a mapping Jacobian matrix,

$$\boldsymbol{M}_s = \frac{\partial \bar{\boldsymbol{p}}^r_{\text{PARS},s}}{\partial \boldsymbol{\varepsilon}_s} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}, \tag{14.14}$$

where

$$\begin{aligned}
m_{11} &= \frac{\cos(y_\Psi)\cos(y_\alpha)}{b_\psi b_\alpha} & m_{12} &= \frac{-y_\rho \cos(y_\alpha)\sin(y_\Psi)}{b_\psi b_\alpha} \\
m_{13} &= \frac{-y_\rho \cos(y_\Psi)\sin(y_\alpha)}{b_\psi b_\alpha} & m_{21} &= \frac{\cos(y_\alpha)\sin(y_\Psi)}{b_\psi b_\alpha} \\
m_{22} &= \frac{y_\rho \cos(y_\Psi)\cos(y_\alpha)}{b_\psi b_\alpha} & m_{23} &= \frac{-y_\rho \sin(y_\Psi)\sin(y_\alpha)}{b_\psi b_\alpha} \\
m_{31} &= -\frac{\sin(y_\alpha)}{b_\alpha} & m_{32} &= 0 \\
m_{33} &= \frac{-y_\rho \cos(y_\alpha)}{b_\alpha}.
\end{aligned}$$

The second step is to take the covariance of the measurement noise $\boldsymbol{\varepsilon}_s$

$$\mathscr{R}_s[k] = \text{diag}\Big(\mathbb{E}[\varepsilon_\rho^2[k]], \mathbb{E}[\varepsilon_\Psi^2[k]], \mathbb{E}[\varepsilon_\alpha^2[k]]\Big), \tag{14.15}$$

at time index k, given in spherical coordinates, and transform it into the Cartesian r-frame through the similarity transform

$$\mathscr{R}^r_{\text{pars}}[k] = \boldsymbol{M}_s[k]\mathscr{R}_s[k]\boldsymbol{M}_s^\intercal[k]. \tag{14.16}$$

Finally the covariance can be transformed into the n-frame, through another similarity transformation, defined by the rotation matrix between the two frames,

$$\mathscr{R}_{\text{pars}}[k] = \boldsymbol{R}_{nr}\mathscr{R}^r_{\text{pars}}[k]\boldsymbol{R}_{rn}, \tag{14.17}$$

to arrive at the linearized measurement covariance matrix $\mathscr{R}_{\text{pars}}$, in the $\{n\}$ frame.

14.2.2 Cylindrical-coordinates-based PARS positioning

As shown in [3], the PARS measurements are susceptible to multipath reflections from the sea surface, during flights over water. As the PARS system used was optimized for communication, this caused the tracked signal to jump from one cluster of measurements to another, when the signal strength of a reflected signal became stronger than the original signal. The phenomenon is particularly visible in the elevation measurements, that from time to time would alternate between the correct elevation angle and a non-physical downward-directed angle. To mitigate this, [3] suggest to instead use the PARS measurements in combination with an exogenous altitude measurement,

$$y_{\text{alt}} = -p^n_{nb,z,\text{exo}} + \varepsilon_{\text{alt}}, \quad \varepsilon_{\text{alt}} \sim \mathcal{N}(0, \sigma^2_{\text{alt}}), \tag{14.18}$$

e.g. from a barometer, instead of the error prone elevation angle measurment, resulting in cylindrical-coordinates-based PARS positioning.

As the elevation angle is discarded, the geometric range measurement must be projected into the horizontal plane for the triplet range, azimuth angle and altitude to constitute a valid 3D position. The horizontal range $\bar{y}_\rho$ is a measurement of

$$\bar{\rho}_u = \sqrt{(p^r_{rb,x})^2 + (p^r_{rb,y})^2}, \tag{14.19}$$

derived as

$$\bar{y}_\rho = \sqrt{y_\rho^2 - y_{\text{alt}}^2} \tag{14.20}$$

An illustration of the of the geometric range ρ_u, horizontal range $\bar{\rho}_u$, elevation angle α_u and azimuth angle Ψ_u, relative to the $\{r\}$-frame, can be seen in fig. 14.1b.

In a similar manner as eq. (14.12), the cylindrical-coordinates-based PARS position measurement can be transformed to Cartesian coordinates, in the ground radio

antenna frame, and debiased using

$$\bar{\boldsymbol{p}}^{r}_{\text{PARS},c} = \begin{bmatrix} b^{-1}_{\rho/\text{alt}} b^{-1}_{\Psi} \bar{y}_{\rho} \cos(y_{\Psi}) \\ b^{-1}_{\rho/\text{alt}} b^{-1}_{\Psi} \bar{y}_{\rho} \sin(y_{\Psi}) \\ y_{\text{alt}} \end{bmatrix}, \tag{14.21}$$

with biases $b_{\rho/\text{alt}} \approx 1$, following from the Monte-Carlo simulation in appendix B.1, and $b_{\Psi} = \mathbb{E}[\cos(\varepsilon_{\Psi})] = e^{-\sigma^2_{\Psi}/2}$, to account for the nonlinear mapping from cylindrical to Cartesian coordinates. The debiased PARS position measurement can then be transformed into the $\{n\}$ frame using

$$\boldsymbol{y}^{n}_{\text{PARS},c} = \boldsymbol{R}_{nr}(\boldsymbol{\Theta}_{\text{PARS}}) \bar{\boldsymbol{p}}^{r}_{\text{PARS},c}. \tag{14.22}$$

Again similarly to to section 14.2.1, the measurement covariance matrix in cylindrical space $\mathscr{R}_c$ is mapped to Cartesian space through the similarity transform

$$\mathscr{R}_{\text{pars}}[k] = \boldsymbol{R}_{nr}(\boldsymbol{\Theta}_{\text{PARS}}) \boldsymbol{M}_c[k] \mathscr{R}_c[k] \boldsymbol{M}^{\intercal}_c[k] \boldsymbol{R}^{\intercal}_{nr}(\boldsymbol{\Theta}_{\text{PARS}}), \tag{14.23}$$

to obtain the linearized measurement covariance matrix $\mathscr{R}_{\text{pars}}$, in the $\{n\}$ frame, where

$$\mathscr{R}_c[k] = \text{diag}\Big(\mathbb{E}[\varepsilon^2_{\rho}[k]], \mathbb{E}[\varepsilon^2_{\Psi}[k]], \mathbb{E}[\varepsilon^2_{\text{alt}}[k]]\Big) \tag{14.24}$$

and where the mapping Jacobian matrix is found by linearizing $\bar{\boldsymbol{p}}^{r}_{\text{PARS},c}$ w.r.t. $\boldsymbol{\varepsilon}_c = \begin{bmatrix} \varepsilon_{\rho}; & \varepsilon_{\Psi}; & \varepsilon_{\text{alt}} \end{bmatrix}$ to obtain

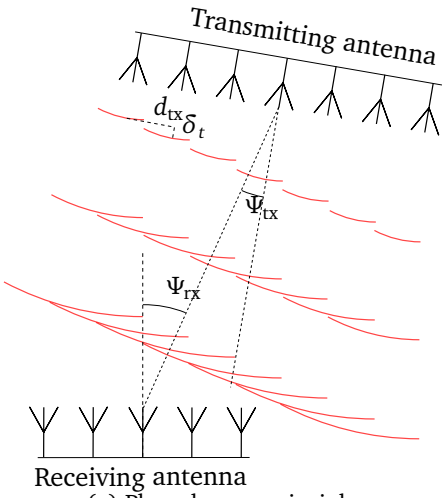
$$\boldsymbol{M}_c = \frac{\partial \bar{\boldsymbol{p}}^{r}_{\text{PARS},c}}{\partial \boldsymbol{\varepsilon}_c} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{14.25}$$

with

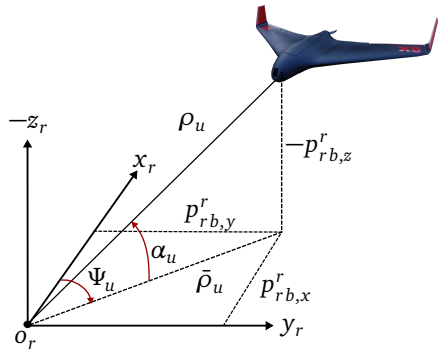
$$\begin{aligned} m_{11} &= \frac{\cos(y_{\Psi}) y_{\rho}}{\bar{y}_{\rho}} & m_{21} &= \frac{\sin(y_{\Psi}) y_{\rho}}{\bar{y}_{\rho}} \\ m_{12} &= -\sin(y_{\Psi}) \bar{y}_{\rho} & m_{22} &= \cos(y_{\Psi}) \bar{y}_{\rho} \\ m_{13} &= -\frac{\cos(y_{\Psi}) y_{\text{alt}}}{\bar{y}_{\rho}} & m_{23} &= -\frac{\sin(y_{\Psi}) y_{\text{alt}}}{\bar{y}_{\rho}}. \end{aligned}$$

14.2.3 PARS ground antenna calibration

Since the PARS only provides a position measurement relative to its ground station ($\{r\}$-frame), pre-flight calibration to obtain the full pose of the ground station antenna relative to the $\{n\}$-frame is crucial to obtain accurate absolute position



(a) Phased-array principle



(b) Range/azimuth/elevation (ρ_u / Ψ_u / α_u) of UAV relative ground radio. $\{r\}$ and $\{b\}$ denotes the radio and the BODY frame, respectively.

Figure 14.1: Phased-array radio navigation

estimates. The position of the PARS ground station antenna can be surveyed using a GNSS receiver, while a rough estimate of the heading of the antenna can be found from a compass. It can then be further adjusted until the the PARS-based position estimates and the GNSS-based position estimates from the autopilot align. In a scenario where the ground station experiences malicious GNSS attacks, a GNSS antenna with heavy shielding on its sides could be used, since most GNSS attacks come from land-based units and since attackers from above are easier to immobilize. For a completely GNSS-free solution, the ground station position can be found from a map, e.g. using a particle filter and terrain information or odometry [67], or by using feature matching [59, Chapter 11]. For obtaining the heading of the ground station antenna, there are also a variety of options [52], notably [154] where attitude is determined from a camera and a digital surface model with a standard deviation of 0.018° in heading.

14.3 The navigation system

When relying on only a single PARS antenna for position measurements, blocking of the signal path, resulting in measurement dropout, and the multipath discussed in section 14.2.2, resulting in outlier measurements, can be critical. This motivates the use of an INS aided by PARS, providing smoother position estimates, enabling detection and removal of outliers, and ability to navigate in the gaps between PARS measurements, if they are short. The INS also provides estimates of the velocity

and attitude of the UAV at high sampling rates, which is needed for feedback in the autopilot controllers.

The integrated INS of this chapter 14 use the NED frame as navigation frame. The simplification is motivated by the accuracy of the PARS position [5] which is typically ten times lower than that of standard GNSS solutions. This results in that the accuracy of the positioning system is much more significant than the assumed simplification of a Earth-fixed, non-rotating NED frame. In addition, the in-run ARS bias stability of MEMS IMUs is a at minimum one order of magnitude or higher than the Earth rate.

The MEKF, which is the key algorithm in the navigation system, consists of several parts, see fig. 14.2. The inertial navigation system, mimic the kinematic of the vehicle by integrating the inertial measurements, specific force and angular rate, using the strapdown equations eq. (2.15), running synchronously with the IMU measurements that are assumed to arrive at with precise timestamps. To account for the drift of the inertial sensors, the navigation system relies on an aiding position measurement, which in this case is PARS-based. For the presented INS, this aiding relies on indirect feedback from a multiplicative extended Kalman filter, which steer the INS state $\boldsymbol{x}_{\text{ins}} = \left(\boldsymbol{p}^n_{nb,\text{ins}}; \boldsymbol{v}^n_{nb,\text{ins}}; \boldsymbol{q}^n_{b,\text{ins}}; \boldsymbol{b}^b_{\text{acc,ins}}; \boldsymbol{b}^b_{\text{ars,ins}}\right)$ to the true state

$$\boldsymbol{x}_{\text{ins}} \to \boldsymbol{x} \tag{14.26}$$

when the error state goes to zero (or the identity quaternion), where the MEKF error state $\delta\boldsymbol{x} = (\delta\boldsymbol{p}; \delta\boldsymbol{v}; \delta\boldsymbol{a}; \delta\boldsymbol{b}_{\text{acc}}; \delta\boldsymbol{b}_{\text{ars}})$, and its model is derived from the true state, $\boldsymbol{x}$, based on

$$\boldsymbol{x} = \boldsymbol{x}_{\text{ins}} \oplus \delta\boldsymbol{x}, \tag{14.27}$$

where $\oplus$ represents the $+$ or the $\otimes$ operator, depending on whether the given INS error state is additive or multiplicative. The attitude error is represented using the four times Modified Rodrigues Parameters (MRP), eq. (2.14). As seen in fig. 14.2, the update from the aiding measurements may be run asynchronously.

14.3.1 INS

By mimicking the kinematics in section 2.2.3 the kinematic model

$$\dot{\boldsymbol{p}}^n_{nb,\text{ins}} = \boldsymbol{v}^n_{nb,\text{ins}} \tag{14.28}$$

$$\dot{\boldsymbol{v}}^n_{nb,\text{ins}} = \boldsymbol{R}_{nb}(\boldsymbol{q}^n_{b,\text{ins}})\hat{\boldsymbol{f}}^b_{nb} + \boldsymbol{g}^n_b \tag{14.29}$$

$$\dot{\boldsymbol{q}}^n_{b,\text{ins}} = \frac{1}{2}\boldsymbol{\Omega}\left(\hat{\boldsymbol{\omega}}^b_{nb}\right)\boldsymbol{q}^n_{b,\text{ins}}, \tag{14.30}$$

and the bias model

$$\dot{\boldsymbol{b}}^b_{\text{acc,ins}} = -\boldsymbol{T}^{-1}_{\text{acc}}\boldsymbol{b}^b_{\text{acc,ins}} \tag{14.31}$$

$$\dot{\boldsymbol{b}}^b_{\text{ars,ins}} = -\boldsymbol{T}^{-1}_{\text{ars}}\boldsymbol{b}^b_{\text{ars,ins}} \tag{14.32}$$

are obtained, where

$$\hat{\boldsymbol{f}}^b_{nb} := \boldsymbol{f}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{acc,ins}}, \quad \hat{\boldsymbol{\omega}}^b_{nb} := \boldsymbol{\omega}^b_{\text{ars}} - \boldsymbol{b}^b_{\text{ars,ins}}, \tag{14.33}$$

where the IMU measurements are modeled as

$$\boldsymbol{f}^b_{\text{imu}} = \boldsymbol{f}^b_{nb} + \boldsymbol{b}^b_{\text{acc}} + \boldsymbol{\varepsilon}^b_{acc}, \tag{14.34}$$

$$\boldsymbol{\omega}^b_{\text{imu}} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{\text{ars}} + \boldsymbol{\varepsilon}^b_{ars}, \tag{14.35}$$

and where $\boldsymbol{T}_{\text{acc}} = T_{\text{acc}} \cdot \boldsymbol{I}_3$ and $\boldsymbol{T}_{\text{ars}} = T_{\text{ars}} \cdot \boldsymbol{I}_3$ are the time constants for the Gauss-Markov model for the accelerometer and angular rate sensor biases, respectively. Being an error state Kalman filter, the state is integrated directly using eqs. (14.28) to (14.32), which also are important in the covariance update. With eq. (14.27), it is possible to arrive at a set of differential equations for the error state, see appendix B.1. These equations are then linearized to arrive at the linear, time-varying model

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\delta\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{\varepsilon}, \tag{14.36}$$

$$\boldsymbol{\varepsilon} = \left(\boldsymbol{\varepsilon}^b_{\text{acc}}; \boldsymbol{\varepsilon}^b_{\text{ars}}; \boldsymbol{\varepsilon}^b_{b,\text{acc}}; \boldsymbol{\varepsilon}^b_{b,\text{ars}}\right), \tag{14.37}$$

where the process noise form the spectral density matrix $\boldsymbol{\mathcal{Q}}(t)$ according to

$$\mathbb{E}[\boldsymbol{\varepsilon}(t)\boldsymbol{\varepsilon}^\intercal(t-\tau)] = \boldsymbol{\mathcal{Q}}(t)\delta(t-\tau). \tag{14.38}$$

The system matrix $\boldsymbol{A}(t)$ is then discretized using a first order approximation

$$\boldsymbol{A}_d[k] = \exp(\boldsymbol{A}(t)T_s) \approx \boldsymbol{I}_n + T_s\boldsymbol{A}(t) + \frac{T_s}{2}\boldsymbol{A}^2(t) + \ldots + \frac{T_s^m}{m!}\boldsymbol{A}^m, \tag{14.39}$$

while the process noise spectral density matrix is first mapped from the input space into the state space, through a similarity transform using $\boldsymbol{B}(t)$, before it is discretized to the covariance matrix $\boldsymbol{\mathcal{Q}}_d[k]$ according to

$$\boldsymbol{\mathcal{Q}}_d[k] = \int_0^{T_s} \boldsymbol{A}_d[k]\boldsymbol{B}(\tau)\boldsymbol{\mathcal{Q}}(\tau)\boldsymbol{B}^\intercal(\tau)\boldsymbol{A}_d^\intercal[k]\,\mathrm{d}\tau, \tag{14.40}$$

The discrete state transition matrix $\boldsymbol{A}_d$ and the process covariance matrix $\boldsymbol{\mathcal{Q}}_d$, are needed to propagate the state covariance

$$\boldsymbol{\mathcal{P}}^-[k+1] = \boldsymbol{A}_d[k]\boldsymbol{\mathcal{P}}^+[k]\boldsymbol{A}_d^\intercal[k] + \boldsymbol{\mathcal{Q}}_d[k]. \tag{14.41}$$

14.3.2 Aiding measurement

Pure integration of eqs. (14.28) to (14.32) leads to drifting state estimates, due to the bias and noise found in inertial sensors. The INS thus relies on aiding from other sensors to correct itself. Aiding can also be used to inject information, such as knowledge that the aircraft is are stationary before takeoff, to improve the INS convergence.

14.3.2.1 PARS and altitude

As a result of obtaining a Cartesian position measurement with the transforms eqs. (14.21) to (14.22), the measurement $\boldsymbol{y}$ is obtained from eq. (14.22) directly, where it is assumed that the exogenous altitude measurements are sampled at such a high frequency that the PARS and altitude measurements can be assumed to be simultaneous. The measurement matrix becomes

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}. \tag{14.42}$$

The covariance matrix $\boldsymbol{\mathcal{R}}$ is given in eq. (14.23), while the predicted measurement trivially becomes

$$\hat{\boldsymbol{y}} = \boldsymbol{p}^n_{nb,\text{ins}}. \tag{14.43}$$

14.3.2.2 Altitude

In the event of a single altitude measurement, without a PARS measurement, the measurement equation is given by eq. (14.18), while the measurement matrix becomes

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{1}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \tag{14.44}$$

where $\boldsymbol{1}_3 = \text{diag}(0, 0, 1)$. The covariance matrix

$$\boldsymbol{\mathcal{R}} = \sigma^2_{\text{alt}}, \tag{14.45}$$

is a single, tunable parameter, while the predicted measurement trivially becomes

$$\hat{y} = p^n_{nb,z}. \tag{14.46}$$

14.3.2.3 Pre-launch calibration

Accelerometer leveling is used to exploite information about the UAV being at rest, with the goal of initializing the roll and pitch angles. Equation (2.16) can then be simplified to

$$\boldsymbol{f}^b_{\text{imu}} \approx -\boldsymbol{R}^{\intercal}_{nb}\boldsymbol{g}^n_b + \boldsymbol{b}^b_{\text{acc}} + \boldsymbol{\varepsilon}_{\text{acc}}, \tag{14.47}$$

since it is know that the linear and angular velocities are zero. This result in

$$\boldsymbol{y}_{\text{acc}} = \boldsymbol{f}^b_{\text{imu}}, \tag{14.48}$$

$$\hat{\boldsymbol{y}}_{\text{acc}} = -\boldsymbol{R}^{\intercal}_{nb,\text{ins}}\boldsymbol{g}^n_b + \boldsymbol{b}^b_{\text{acc,ins}}, \tag{14.49}$$

$$\boldsymbol{C}_{\text{acc}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{S}\left(\boldsymbol{R}^{\intercal}_{nb,\text{ins}}\boldsymbol{g}^n_b\right) & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \end{bmatrix}, \tag{14.50}$$

The 3×3 covariance matrix

$$\boldsymbol{\mathcal{R}}_{\text{acc}} = \boldsymbol{\sigma}^2_{\text{acc}} \cdot I_3 \approx \boldsymbol{\mathcal{Q}}_{\text{acc}}/T_s, \tag{14.51}$$

is given by the covariance of the accelerometer measurements. Since it is also known that the UAV has zero angular velocity pre-launch, this can be utilized to estimate the ARS bias, since the ARS only measures the ARS bias and noise. Hence,

$$\boldsymbol{y}_{\text{ars}} = \boldsymbol{\omega}^{b}_{\text{imu}}, \tag{14.52}$$

$$\hat{\boldsymbol{y}}_{\text{ars}} = \boldsymbol{b}^{b}_{\text{ars,ins}}, \tag{14.53}$$

$$\boldsymbol{C}_{\text{ars}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \end{bmatrix}. \tag{14.54}$$

Again, the 3×3 covariance matrix

$$\mathscr{R}_{\text{ars}} = \boldsymbol{\sigma}^2_{\text{ars}} \cdot I_3 \approx \mathscr{Q}_{\text{ars}}/T_s, \tag{14.55}$$

is given by the covariance of the ARS measurements. Similarly, it is known that the UAV is standing still, and thus have no linear velocity, which can be exploited by the virtual measurement and

$$\boldsymbol{y}_{\text{vel}} = \boldsymbol{0}, \tag{14.56}$$

$$\hat{\boldsymbol{y}}_{\text{vel}} = \boldsymbol{v}^{n}_{nb,\text{ins}}, \tag{14.57}$$

$$\boldsymbol{C}_{\text{vel}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}. \tag{14.58}$$

Under the assumption of standing perfectly still, the covariance matrix can be set very small, which will give faster convergence. For heading initialization, one can utilize a magnetometer, $\boldsymbol{y}_{\text{mag}} = \boldsymbol{m}^{b}_{\text{nb}} + \boldsymbol{\varepsilon}_{\text{mag}}$, where

$$\boldsymbol{C}_{\text{mag}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{S}(\boldsymbol{R}^{\intercal}_{nb,\text{ins}}\boldsymbol{m}^{n}_{nb}) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}. \tag{14.59}$$

14.3.3 Outlier rejection

To avoid degradation of the PVA estimates from bad PARS measurements, as discussed in section 14.2, the reliability of the measurements is assessed by a test statistic, and possibly rejected in the event of an outlier.

Given the normalized residual [68, Section 7.6.1]

$$\epsilon = (\boldsymbol{C}\mathscr{P}^{-}\boldsymbol{C}^{\intercal} + \mathscr{R})^{-\frac{1}{2}}(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}), \tag{14.60}$$

in which the central limit theorem motivates the Gaussian approximation, the null hypothesis of the measurement being an inlier, is discarded if the test statistic

$$T(\boldsymbol{y}^{r}_{\text{pars}}) = (\boldsymbol{R}_{nr}\boldsymbol{y}^{r}_{\text{pars}} - \boldsymbol{C}\boldsymbol{x}_{\text{ins}})^{\intercal}(\boldsymbol{C}\mathscr{P}^{-}\boldsymbol{C}^{\intercal} + \mathscr{R})^{-1}(\boldsymbol{R}_{nr}\boldsymbol{y}^{r}_{\text{pars}} - \boldsymbol{C}\boldsymbol{x}_{\text{ins}}) \tag{14.61}$$

is above some limit χ^2_α, corresponding to a χ-squared distribution with a confidence interval of $1-\alpha$, which for the PARS measurements has three degrees of freedom.

14.3.4 Multiplicative extended Kalman filter (MEKF)

The measurement $\boldsymbol{y}$, prediction $\hat{\boldsymbol{y}}$, measurement matrix $\boldsymbol{C}$ and measurement covariance matrix $\mathscr{R}$, are formed from the available aiding measurement for the current time step using equations from section 14.3.2. For the given error state vector, the MEKF then becomes

$$\boldsymbol{K}[k] = \mathscr{P}^{-}[k]\boldsymbol{C}^{\intercal}[k]\Big(\boldsymbol{C}[k]\mathscr{P}^{-}[k]\boldsymbol{C}^{\intercal}[k] + \mathscr{R}[k]\Big)^{-1}, \tag{14.62}$$

$$\delta\hat{\boldsymbol{x}}[k] = \boldsymbol{K}[k](\boldsymbol{y}[k] - \hat{\boldsymbol{y}}[k])), \tag{14.63}$$

$$\mathscr{P}^{+}[k] = (\boldsymbol{I}_n - \boldsymbol{K}[k]\boldsymbol{C}[k])\,\mathscr{P}^{-}[k]\,(\boldsymbol{I}_n - \boldsymbol{K}[k]\boldsymbol{C}[k])^{\intercal} + \boldsymbol{K}[k]\mathscr{R}[k]\boldsymbol{K}^{\intercal}[k], \tag{14.64}$$

where $n = 15$ is the size of the error state space. The INS states with a linear relationship to the corresponding error states, are updated according to

$$\boldsymbol{p}^{n}_{nb,\text{ins}}[k] = \boldsymbol{p}^{n}_{nb,\text{ins}}[k] + \delta\hat{\boldsymbol{p}}[k], \tag{14.65}$$

$$\boldsymbol{v}^{n}_{nb,\text{ins}}[k] = \boldsymbol{v}^{n}_{nb,\text{ins}}[k] + \delta\hat{\boldsymbol{v}}[k], \tag{14.66}$$

$$\boldsymbol{b}^{b}_{\text{acc,ins}}[k] = \boldsymbol{b}^{b}_{\text{acc,ins}}[k] + \delta\hat{\boldsymbol{b}}_{\text{acc}}[k], \tag{14.67}$$

$$\boldsymbol{b}^{b}_{\text{ars,ins}}[k] = \boldsymbol{b}^{b}_{\text{ars,ins}}[k] + \delta\hat{\boldsymbol{b}}_{\text{ars}}[k], \tag{14.68}$$

after every iteration of the MEKF, while the attitude is corrected using

$$\delta\boldsymbol{q}(\delta\hat{\boldsymbol{a}}[k]) = \frac{1}{16 + \hat{\boldsymbol{a}}^{\intercal}[k]\hat{\boldsymbol{a}}[k]}\begin{bmatrix} 16 - \delta\hat{\boldsymbol{a}}^{\intercal}[k]\delta\hat{\boldsymbol{a}}[k] \\ 8\cdot\delta\hat{\boldsymbol{a}}[k] \end{bmatrix}, \tag{14.69}$$

$$\boldsymbol{q}^{n}_{b,\text{ins}}[k] = \boldsymbol{q}^{n}_{b,\text{ins}}[k] \otimes \delta\boldsymbol{q}(\delta\hat{\boldsymbol{a}}[k]). \tag{14.70}$$

similar to [115]. If the attitude error ever obtains a nonphysical value by exceeding ±180°, corresponding to $\|\delta\hat{\boldsymbol{a}}| > 4$, the attitude error must be replaced with its shadow set [116] before applying eq. (14.70). After the INS has been corrected for, the MEKF state and covariance is reset

$$\delta\hat{\boldsymbol{x}}[k] = \boldsymbol{0}_{n\times 1}, \tag{14.71}$$

$$\mathscr{P}^{+}[k] = \boldsymbol{G}(\delta\hat{\boldsymbol{u}}[k])\cdot\mathscr{P}^{+}[k]\cdot\boldsymbol{G}^{\intercal}(\delta\hat{\boldsymbol{u}}[k]), \tag{14.72}$$

where $\boldsymbol{G}$ depends on the attitude error. For the kinematics of the given parameterization, eq. (14.69), it is given as

$$\boldsymbol{G}(\delta\hat{\boldsymbol{u}}[k]) = \begin{bmatrix} \boldsymbol{I}_6 & \boldsymbol{0}_{6\times 3} & \boldsymbol{0}_{6\times 3} \\ \boldsymbol{0}_{3\times 6} & \delta\hat{q}_w\boldsymbol{I}_3 - \boldsymbol{S}\big(\delta\hat{\boldsymbol{q}}_v\big) & \boldsymbol{0}_{6\times 3} \\ \boldsymbol{0}_{6\times 6} & \boldsymbol{0}_{6\times 3} & \boldsymbol{I}_6 \end{bmatrix}. \tag{14.73}$$

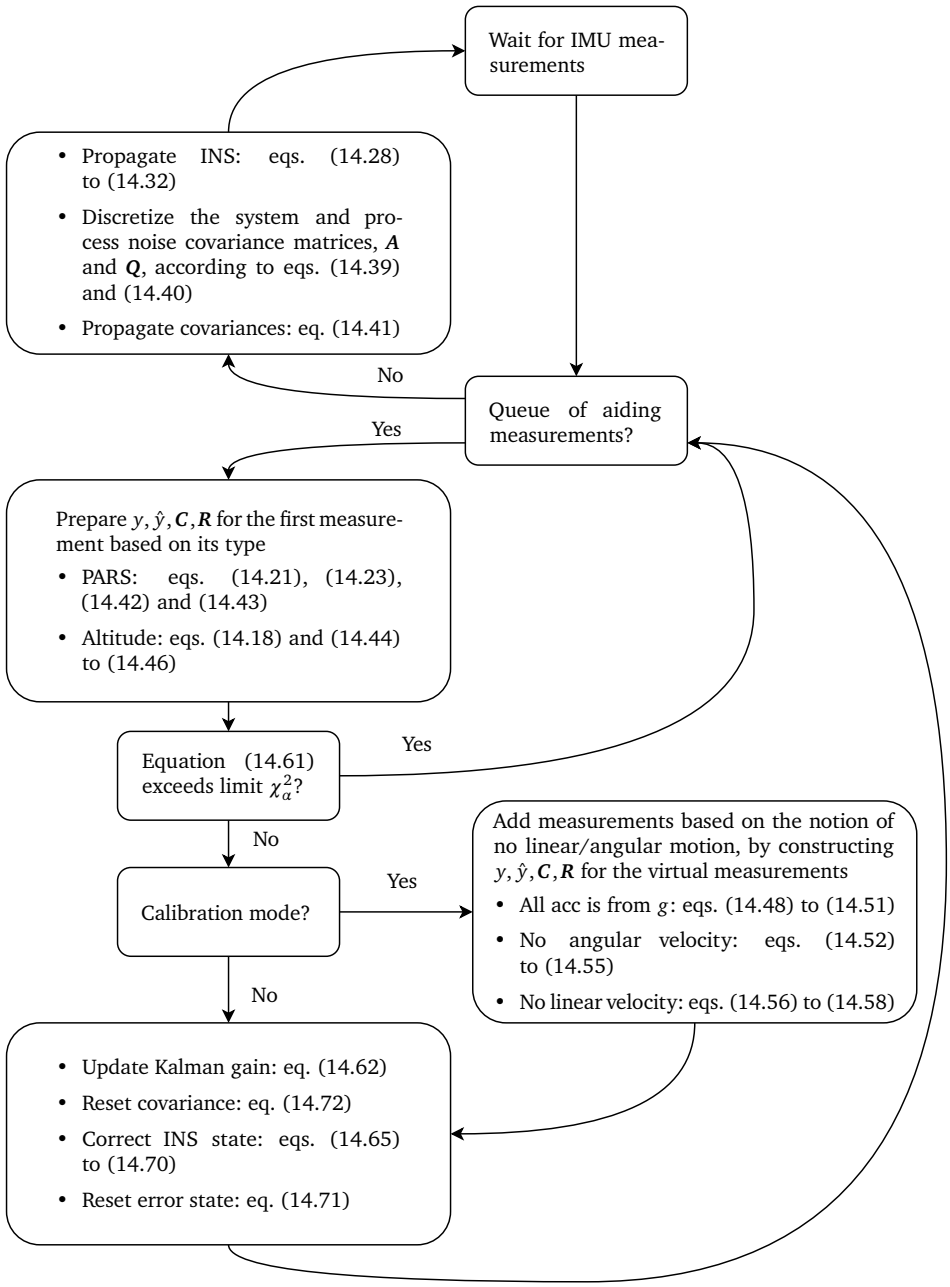


Figure 14.2: The different steps involved in the MEKF, and their accompanying equations

14.4 System architecture and safety

To be able to demonstrate the reliability of the presented navigation system in a BVLOS setting, a setup that ensures safe operation is required. BVLOS operations have strict requirements for risk management and airworthiness. This necessitates extended use of trusted and well-tested hardware and software solutions, allowing for few modifications. The ArduPlane autopilot software has been used extensively for BVLOS flights, both at our lab [204] and by the ArduPlane community, it is desirable to build on top of this existing solution instead of modifying it, risking to introduce new bugs in safety-critical mechanisms. For this reason, the PARS-based navigation and guidance system is implemented in the DUNE unified navigation environment [150] running on an onboard computer, a contained part of the payload. This is depicted in fig. 14.5, a simplified illustration of the different modules and how they are interconnected.

The ArduPlane subsystem runs independently of DUNE. Under normal operation, in any of the automatic modes in ArduPlane[1], the current desired waypoint is sent to the lateral L_1 guidance controller [144] and the longitudinal total-energy based *TECS* controller. These guidance controllers are based on the state estimates from the navigation system in ArduPlane, which relies on measurements from a GNSS-based sensor suite. The desired roll and pitch angles from the guidance controllers are passed on to the low-level roll and pitch controller, respectively, while the throttle command is sent to the electronic speed controller.

Contrary, the DUNE subsystem relies on ArduPlane as an interface to the UAV, both in terms of actuation and for some navigation sensor data. Similarly to the Ardupilot subsystem, the current desired waypoint is sent to two LOS-based guidance controllers, see section 14.4.2, that calculate the desired roll, pitch and throttle commands. These set points are then sent to the low-level ArduPlane controllers, through the Mavlink message *set attitude target* in GUIDED mode. Based on the quality of the GNSS-based state estimates from ArduPlane, the PARS-based state estimates from the presented navigation system, and input from the user, the navigation monitor selects which source is a viable candidate to provide the navgation state estimates to the rest of the DUNE subsystem, including the guidance controllers. In addition to the safety it provides, this switch simplifies the tuning process, as the guidance controllers can run based on estimates from both PARS- and GNSS-based navigation systems.

As the two navigation systems run in parallel, independent of each other, switching is seemless. This also implies that if any of the navigation systems should fail, it is possible to switch to the other system, which is assumed to be unaffected by the error that caused the first navigation system to fail. For these experiments, the system is focused on monitoring the state of the PARS-based estimates, but previous simulation studies [6] have shown the potential gains in robustness against GNSS

[1]The automatic modes in ArduPlane are AUTO,LOITER,CIRCLE,GUIDED,LAND and return-to-launch(RTL)

radio frequency interference, when the nominally precise GNSS-based estimates are validated against the robust, but imprecise PARS-based estimates.

It should be noted that when using the PARS-based estimates along with the guidance controllers in DUNE, the system still depends on the pitch and roll estimates from the GNSS-based ArduPlane navigation system, as they are used to find the pitch and roll control errors. However, the roll and pitch estimates rely primarily on IMU measurements, and will likely be very similar to the roll and pitch estimates from the PARS-based INS. This is obviously only a simplification made from practical considerations, as there is no simple, non-intrusive way to get the low-level ArduPlane controllers to use the PARS-based estimates with the existing ArduPlane interface.

14.4.1 Safety mechanisms

In addition to switching between navigation sources, the navigation monitor also monitors the difference between the position and attitude estimates for the two navigation systems, and dispatches an abort message if the differences are large. The following abort triggers are considered

- Sudden large jumps in either position or attitude
- Loss of contact with the PARS base antenna
- Manual trigger, where the operator looked for trends in the estimates, such as small and persistent offsets in roll or pitch, which could cause stall or severe loss of lift

The abort message is consumed by an abort plan dispatcher, which will initiate a plan of user-configurable waypoints that are either stationary or relative to the current GNSS-based UAV position. As this is an AUTO-mode plan, it switches to navigate using the ArduPlane controllers and GNSS-based state estimates.

In the extreme scenario that contact with the ground station is lost, e.g. as a result of poor navigation, ArduPlane will trigger a failsafe and enter a return-to-launch (RTL) mode. Being a non-GUIDED mode, this again causes the ArduPlane GNSS navigation and controllers to guide the UAV back to the lauch.

The mechanisms provide safe testing, by switching to GNSS when needed, while an operational system for use in GNSS-denied environments would be designed with opposite functionality, i.e. siwtch to PARS when GNSS is denied.

14.4.2 Guidance

Given the architecture described above, both lateral and longitudinal guidance controllers must be implemented in DUNE to calculate desired roll, pitch and throttle commands, based on the desired waypoints and on the current state of the UAV. For

lateral guidance, the LOS-based controller from chapter 10 was used. The longitudinal guidance law is also based on LOS path following [201]

$$\gamma_d = \gamma_p + \text{atan}\left(\frac{K_{ph} z_e + K_{ih} \int z_e \mathrm{d}\tau}{\Delta_h})\right) \tag{14.74}$$

where $K_\bullet$ are controller gains, and $\Delta_h = \sqrt{R_{\text{enc}}^2 - z_e^2}$ is the vertical lookahead distance, tuned from the enclosure radius R_{enc} [50]. The angle of the path in the vertical plane is denoted γ_p, while z_e is the vertical cross-track error.The desired pitch is then [73]

$$\theta_d = \gamma_d + \alpha_{\text{trim}} - K_{\gamma,p} \tilde{\gamma}, \tag{14.75}$$

where $\tilde{\gamma} = \gamma_p - \gamma$, α_{trim} is the angle of attack at trim condition, while $K_{\gamma,p} > -1$ is a controller gain.

The throttle command is a signal in the range 0 to 100 which is given by a PI control on the airspeed error $\tilde{V}_a = V_{a,d} - V_a$, with a height error feed forward term,

$$T = K_{V,p} \tilde{V}_a + K_{V,i} \int \tilde{V}_a \mathrm{d}\tau + K_{V,FF} z_e + T_{t,\text{trim}}, \tag{14.76}$$

where $K_\bullet$ are controller gains, and $T_{t,\text{trim}}$ is the trim throttle.

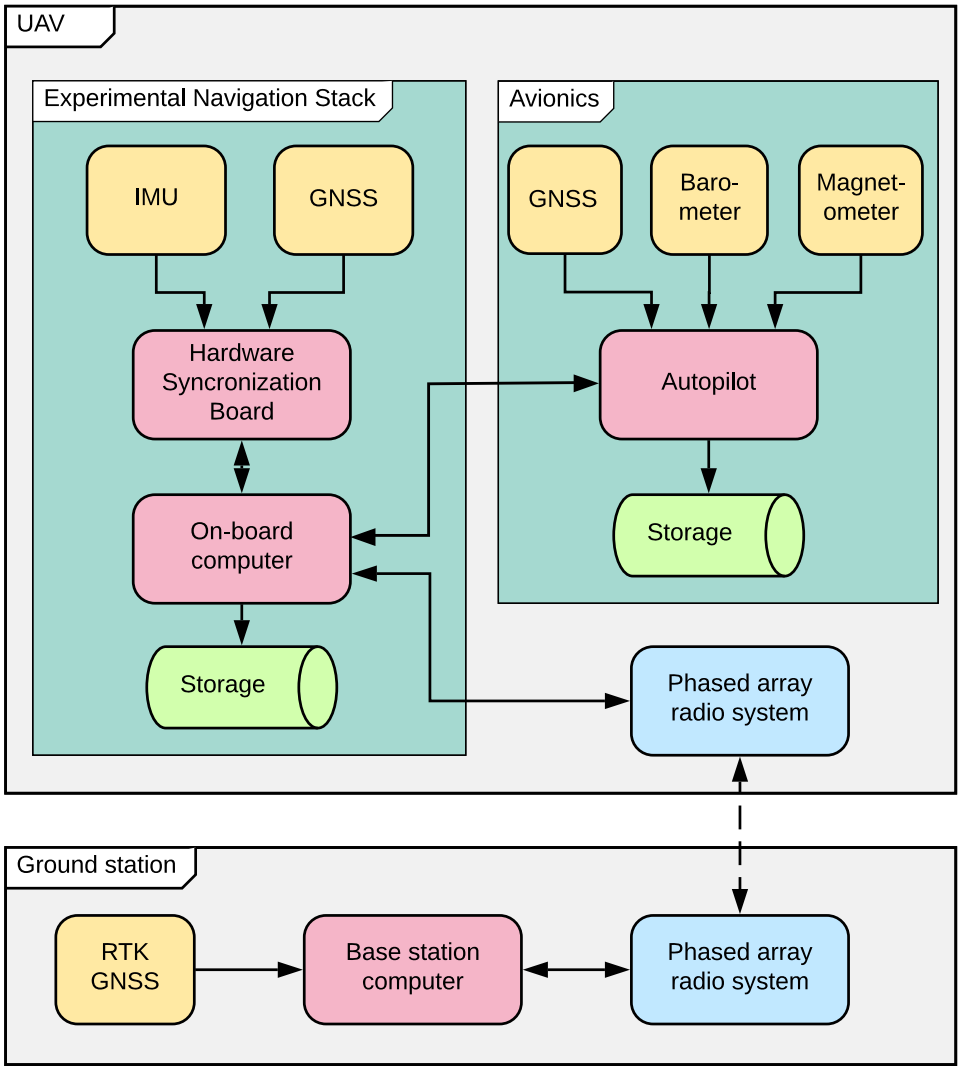


Figure 14.3: Overview of the PARS navigation stack and avionics. The UAV PARS position is calculated on the ground station, and is then transmitted to the UAV, over the PARS ethernet link. The barometer and magnetometer are internal to the autopilot, while the GPS is connected by UART. The autopilot connects to the onboard computer over a serial link. The IMU and second GNSS receiver are connected over serial links to the HW synchronization, which again is connected to the onboard computer over USB.

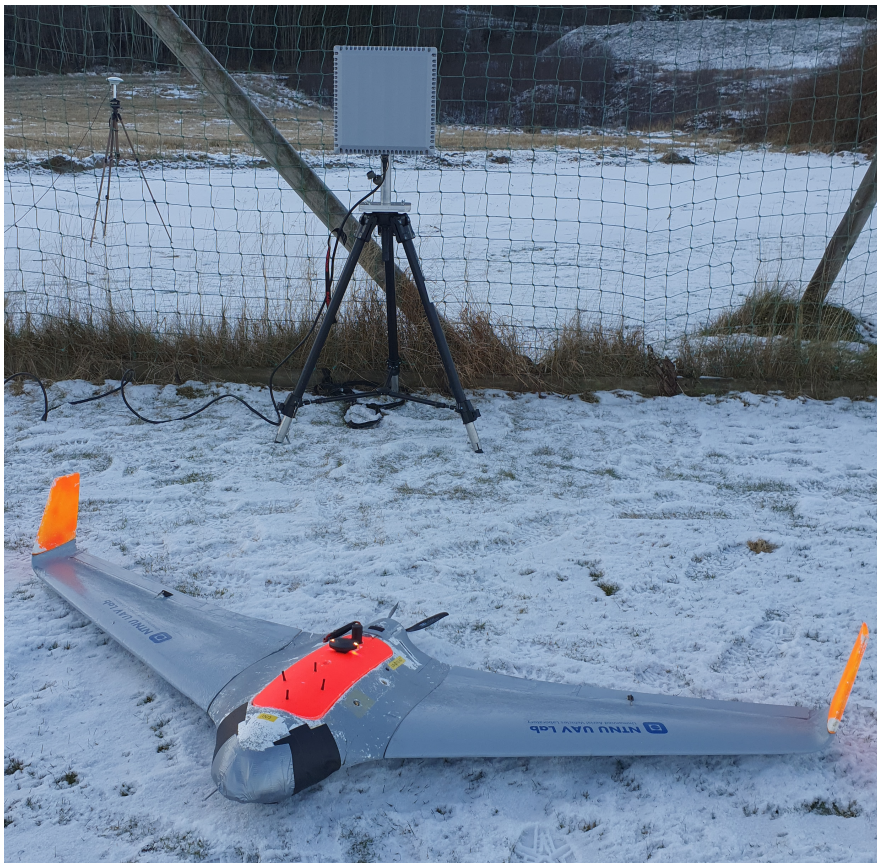


Figure 14.4: The four UAV antennas mounted in a diamond shape in the front part of the orange lid on the Skywalker X8, with the base antenna in the background

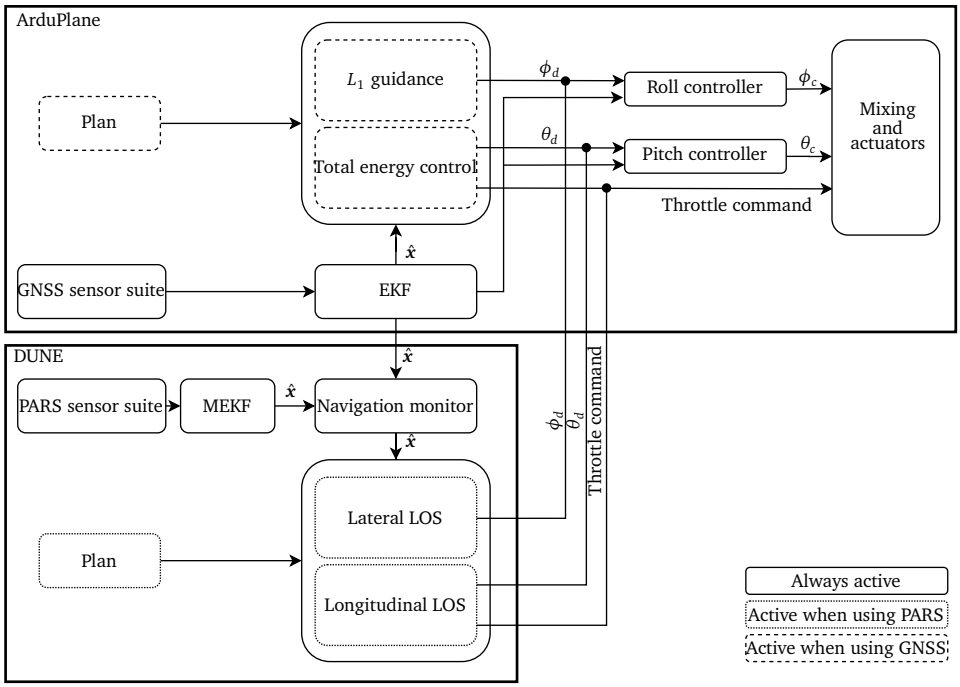


Figure 14.5: Block diagram of the interconnected system

14.5 Test Results

Through three different scenarios, this section seeks to demonstrate the performance of the presented navigation solution. First, its versatility and performance is demonstrated, by testing it on data from both a multirotor UAV in section 14.5.1.1, using the spherical-coordinates-based approach from section 14.2.1, and on a fixed-wing UAV in section 14.5.1.2, using cylindrical coordinates section 14.2.2, by comparison to RTK GNSS. Then its applicability to a real-world scenario is demonstrated, by using an online implementation of the navigation solution in feedback control of a fixed-wing UAV in a beyond-visual-line-of-sight scenario, where the results are compared to the GNSS-based position, velocity and attitude estimates of the autopilot.

The navigation hardware, based on the standard fixed-wing payload at the NTNU UAVlab [204, 5], is illustrated in fig. 14.3, and consists of

- Pixhawk autopilot, with accompanying magnetometer, barometer and airspeed sensor. For the fixed-wing flights, a Pixhawk 1 with ArduPlane was used, whereas the multirotor flight used a Pixhawk 3 Pro, running ArduCopter.
- Radionor CRE-189, a ground radio with 8×8 antenna elements, covering a $90°$ frustum both in elevation and in azimuth, with an root mean square error of $0.1°$ in each axis. It is pictured in fig. 14.4, along with the fixed-wing.
- Radionor CRE2-144-LW UAV-side PARS in the fixed-wing flights, while the multirotor was equipped with the rugged CRE2-144-M2-SMA version. This combination of radio modules provides a range of up to 114 km at a communication bandwidth of 0.5 Mbit/s. Both radios modules have a maximum user data throughput of 15 Mbit/s, and use AES-256 link encryption.
- Odroid XU4 onboard computer running Ubuntu Server 19.04
- SenTiBoard [4] sensor interface and hardware synchronization board
- GNSS receivers. Both payloads used to collect the data for the offline results used a Ublox m8t-neo receiver connected to the SenTiBoard, and an Ublox m8n connected to the Pixhawk[2]. For the online field experiments, two Ublox ZED-F9 GNSS receivers were used instead, sharing the same antenna, configured for 10 Hz GPS L1C/A and L2C, GLONASS G1 and G2, Galileo E1B/C and E5b, and Beidou B1l and B2l. One receiver is connected to the autopilot and another is connected to the SenTiBoard.
- Sensonor STIM300 tactical grade, MEMS-based IMU, providing specific force and angular rate measurements at 250 Hz

To avoid a single point of failure in the communication links in the fixed-wing BVLOS flights, the telemetry is sent over two independent links: an ethernet link provided by the PARS radio, and a serial link provided by a 3DR 433 MHz radio, which is the

[2] The m8n was only used by the internal estimator in the Pixhawk, meaning that the resulting attitude estimates, used as ground truth, is affected by this receiver.

standard radio for the Pixhawk. These links are then merged using Mavproxy on the ground station, and using the more lightweight Mavlink-router on the UAV side.

In all the flights, a first-order approximation of $\boldsymbol{A}_d[k]$ in eq. (14.39), was deemed sufficient when using eq. (14.40) to calculate $\boldsymbol{\mathcal{Q}}_d[k]$.

14.5.1 Offline results

The results presented in this section are based on data from two different campaigns, to illustrate that the presented navigation system is able to ensure operability of UAVs in GNSS denied conditions, independent of platform and mission type. In the first campaign, the system was deployed on a multirotor UAV, flying and hovering over land, whereas the second campaign was with a fixed-wing UAV flying mostly over water.

For both flights the bias time constants where chosen as $\boldsymbol{T}_\bullet = T_\bullet \cdot \boldsymbol{I}_3$, $T_\bullet = 3600\,\mathrm{s}$. In both campaigns the $\boldsymbol{\mathcal{Q}}$ matrix took the form of

$$\boldsymbol{\mathcal{Q}}(t) = \begin{bmatrix} \sigma^2_{\mathrm{acc}}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \sigma^2_{\mathrm{ars}}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \sigma^2_{\mathrm{b,acc}}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & q^2_{\mathrm{b,ars}}\boldsymbol{I}_3 \end{bmatrix}, \tag{14.77}$$

and where the diagonal elements were chosen as

$$\begin{aligned} \sigma_{\mathrm{acc}} &= 2.57 \times 10^{-2}\,\mathrm{m/s/\sqrt{s}}, \\ \sigma_{\mathrm{ars}} &= 9.59 \times 10^{-4}\,\mathrm{rad/\sqrt{s}}, \\ \sigma_{\mathrm{b,acc}} &= 2.55 \times 10^{-4}\,\mathrm{m/s^{5/2}}, \\ \sigma_{\mathrm{b,ars}} &= 6.29 \times 10^{-8}\,\mathrm{rad/s^{3/2}}. \end{aligned}$$

The tuning is based on the IMU (STIM300, rev D) datasheet values where the σ_{acc} and σ_{ars} values were scaled to be in the same order of magnitude as IMU measurements from previously recorded flights, while the bias in-run stability was increased by 10 per cent, both to ad hoc account for the fact that the flights were not conducted in a controlled environment.

The measurement covariance matrices had the structure of

$$\boldsymbol{\mathcal{R}}_{\mathrm{s}} = \begin{bmatrix} \sigma^2_\rho & 0 & 0 \\ 0 & \sigma^2_\Psi & 0 \\ 0 & 0 & \sigma^2_\alpha \end{bmatrix}, \quad \boldsymbol{\mathcal{R}}_{\mathrm{c}} = \begin{bmatrix} \sigma^2_\rho & 0 & 0 \\ 0 & \sigma^2_\Psi & 0 \\ 0 & 0 & \sigma^2_{\mathrm{alt}} \end{bmatrix}, \tag{14.78}$$

for the two setups presented in section 14.2.1 and section 14.2.2, respectively. In both campaigns, the range and azimuth covariance were set to $\sigma_\rho = 15\,\mathrm{m}$ and $\sigma_\Psi = 2^\circ$. For the multirotor data, $\sigma_\alpha = 2^\circ$, while during the fixed-wing flight, was $\sigma_{\mathrm{alt}} = 5\,\mathrm{m}$ chosen.

Figure 14.6: Nordic Unmanned Camflight BG-200 HL UAV

The outlier rejection threshold was chosen as $\chi^2_{\alpha=0.05} = 7.815$, which corresponds to a 95 % confidence interval in a three-degree-of-freedom χ-squared test.

14.5.1.1 Multirotor campaign

This section presents the estimates from the presented navigation system using spherical PARS measurements, as presented in section 14.2.1. This campaign was performed in the northern parts of Norway, using a *Nordic Unmanned Camflight BG-200 HL UAV* 140 cm-diameter octocopter, see fig. 14.6, flying and hovering with a range of about one hundred meters from the base antenna. Data from this campaign has been included to demonstrate the performance of the presented navigation system in short-range, relatively slow flight, including hovering, which is typical of multirotor operations.

The mean-error (ME), mean absolute-error (MAE), standard deviation (STD) and root mean square error (RMSE) statistics for position, velocity and attitude relative RTK GNSS position and ArduCopter velocity and attitude outputs are presented in table 14.1. The results are visualized in figs. 14.7 to 14.12. Comparing to a standalone RTK GNSS position estimate, in table 14.1, the norm of the position RMSE for the PARS-aided INS is comparable to that of a consumer grade GPS receiver, when operating at this range from the base radio. Looking at fig. 14.9, which includes the position measurement calculated directly from the PARS measurements, the utility of fusing the PARS and the IMU measurements become apparent. It is also clear that the presented MEKF is able to reject most of the outliers, and thus obtain smoother position estimates than without outlier rejection.

Regarding velocity, one can see in fig. 14.11 that the PARS-aided INS tracks the velocity of the autopilot well, except for a short time window around 450 seconds. This deviation probably stems from one or more measurements, that should have

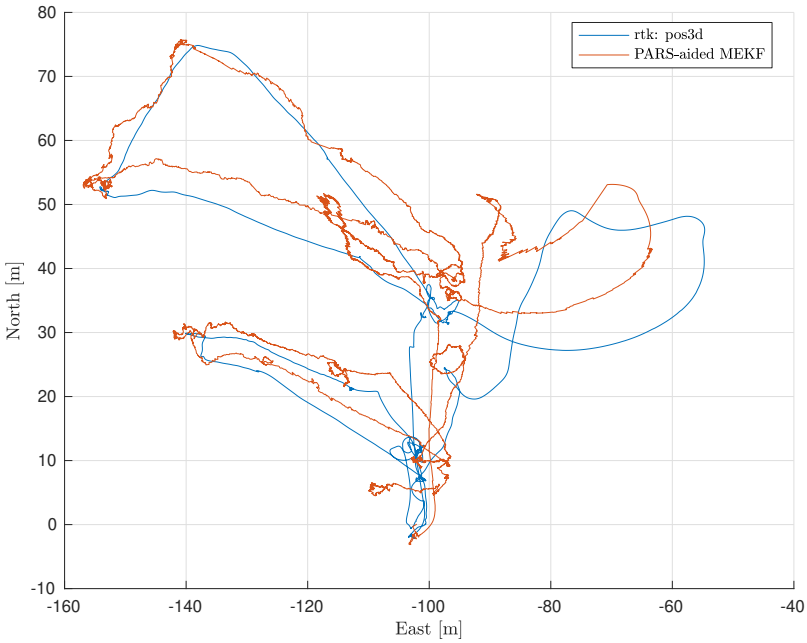


Figure 14.7: Multirotor campaign: 2D NED position with RTK GNSS reference

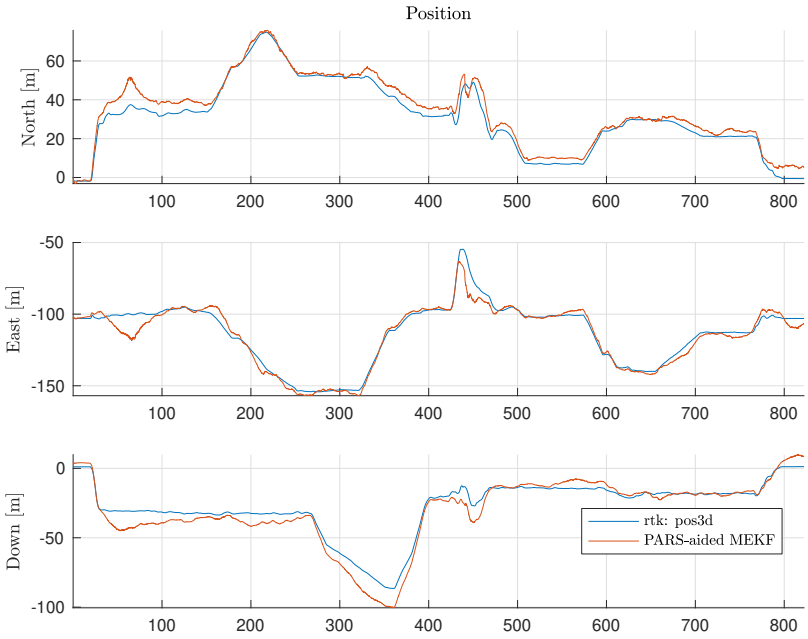


Figure 14.8: Multirotor campaign: NED position, compared to RTK GNSS reference

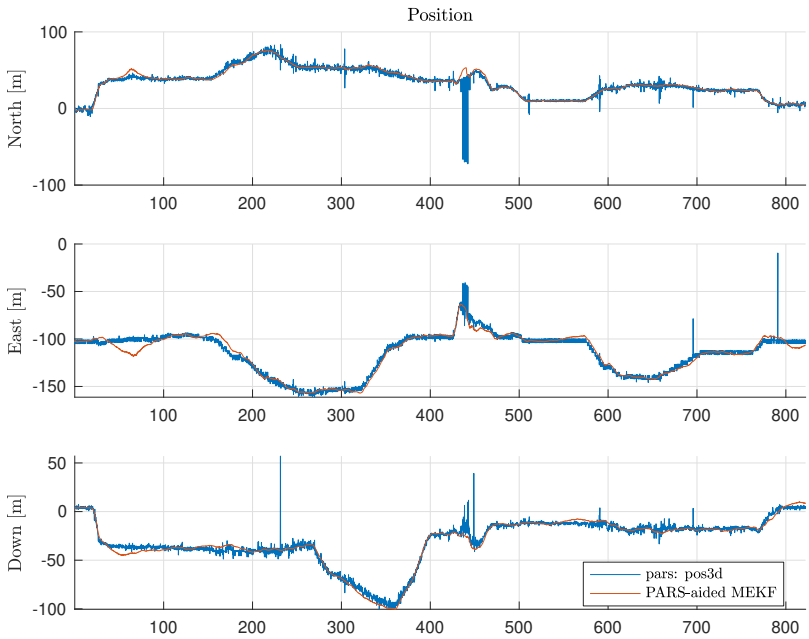


Figure 14.9: Multirotor campaign: NED position, compared to PARS reference

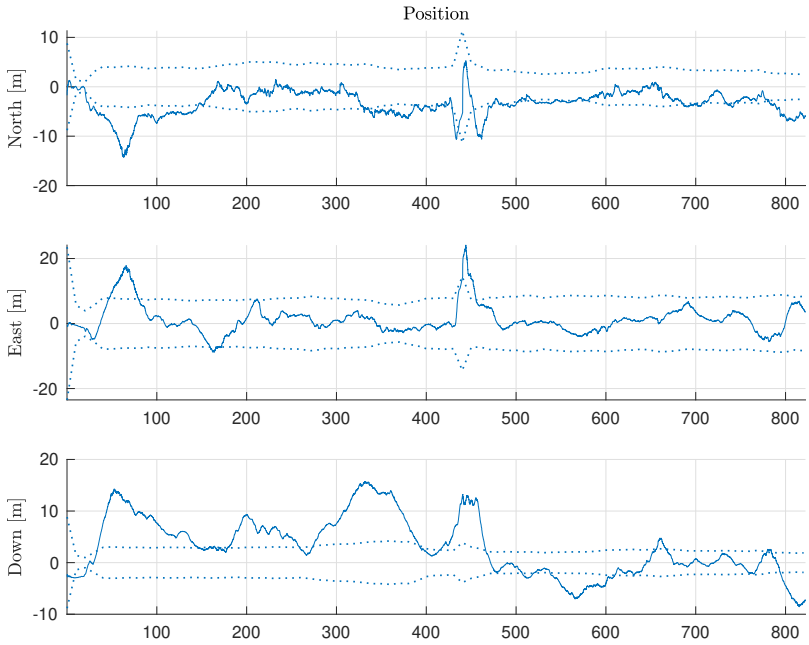


Figure 14.10: Multirotor campaign: Position error plot relative RTK GNSS

Table 14.1: Multirotor: Statistics PARS-aided INS relative RTK GNSS/autopilot

Position	**North** [m]	**East** [m]	**Down** [m]	**Norm** [m]
ME:	-3.25	1.25	3.14	4.69
MAE:	3.35	2.99	4.99	6.71
STD:	2.64	4.39	5.62	7.60
RMSE:	4.19	4.56	6.44	8.93

Velocity	**North** [m/s]	**East** [m/s]	**Down** [m/s]	**Norm** [m/s]
ME:	-0.14	0.05	0.09	0.17
MAE:	0.34	0.47	0.40	0.71
STD:	0.60	0.70	0.53	1.06
RMSE:	0.61	0.70	0.54	1.07

Attitude	**Roll** [°]	**Pitch** [°]	**Yaw** [°]	**Norm** [°]
ME:	0.50	-1.21	-6.39	6.53
MAE:	0.82	1.33	9.64	9.76
STD:	0.96	1.00	12.33	12.41
RMSE:	1.09	1.57	13.89	14.02

been rejected as outliers, were considered as inliers by the MEKF. When the actual inlier measurements again were used for corrections, the velocity recovered. Studying table 14.1 one can also see that the overall velocity estimates were accurate compared to the ArduCopter velocity.

Figure 14.12 illustrates the attitude estimation performance showing the attitude estimates obtained from the PARS-aided INS relative the attitude and heading reference (AHRS) obtained from ArduCopter. Considering that ArduCopter is based on low-cost sensors, and thus based on low accuracy components, it should not be considered a ground truth, especially as it relies on a magnetic compass which is vulnerable to magnetic disturbances. However, it is a well established navigation solution used in multirotor closed-loop flight, and is therefore considered an appropriate reference. Table 14.1 shows that the AHRS and the PARS-aided INS are reasonably similar, where the discrepancy of the PARS-INS heading estimate compared to the reference is larger than what is obtained with roll and pitch. The majority of the attitude error is the yaw error the first 50 s before gradually decreasing from then on. Moreover, the yaw angle is also the axis in which the AHRS accuracy is believed to be the worst, since the gravity vector measured by the accelerometer is independent of the yaw angle thus having to rely on magnetometers and excitation in an GNSS/INS framework. Furthermore, the accuracy of both the AHRS and the PARS-aided INS estimates are believed to improve in flights with more agile maneuvers, as indicated by the results to be presented next.

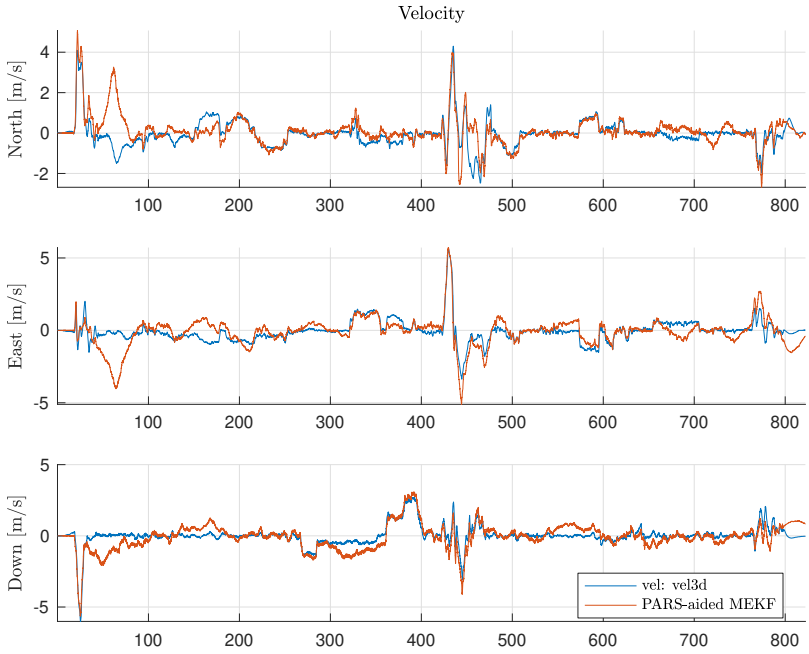


Figure 14.11: Fixed-wing campaign: NED velocity, compared to autopilot reference

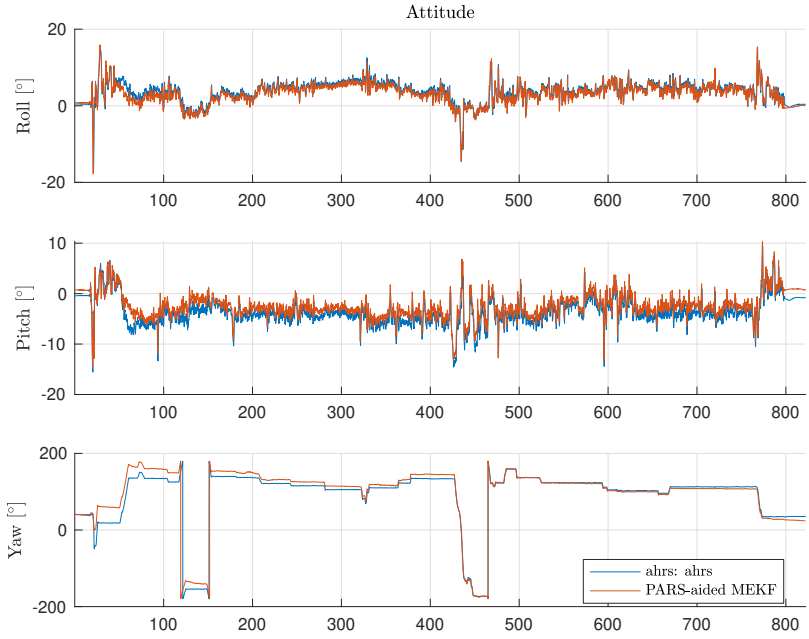


Figure 14.12: Multirotor campaign: Attitude, compared to autopilot reference

14.5.1.2 Fixed-wing campaign

This section presents the estimates from the presented navigation system using cylindrical PARS measurements, as presented in section 14.2.2. This campaign was performed in Central Norway, flying up to 5 km away from the base antenna, largely over water, using a Skywalker X8 fixed-wing UAV, as described in chapter 4, see fig. 14.13. Data from this campaign has been included to demonstrate the performance of the presented navigation system in medium-range, relatively high velocity flight, with a well-defined course angle, which is typical of fixed-wing operations.

The same estimation error statistics as in the multirotor campaign, relative RTK GNSS position and ArduPlane autopilot velocity and attitude outputs are presented in table 14.2. As fixed-wings guidance systems are highly dependent on estimated speed and course, statistics for these quantities also are included. More statistics comparing the PARS-aided INS to a RTK-aided INS is presented in table 14.3. There were added to illustrate some of the discrepancies between the PARS-aided INS and the ArduPlane data might be due to issues with the ArduPlane solution during this particular flight, probably due to a incorrect magnetometer and heading calibration. The results are visualized in figs. 14.14 to 14.20.

Comparing to the RTK GNSS position solution, the norm of the position RMSE of the PARS-aided INS relative the RTK GNSS in table 14.2, is higher than what is typically obtained with a consumer grade GNSS receiver. This is, however, expected as the PARS uncertainty increases with range from the base radio, which is clear from fig. 14.17 where the estimated 3σ of the EKF position error covariance increase with the range. In addition, the positioning performance is improved 14 per cent w.r.t. RMSE compared to our previous work [6], where a nonlinear observer framework was utilized as the sensor fusion algorithm. This solution also used an aiding heading measurement. When comparing fig. 14.14 to fig. 14.15, the utilitiy of PARS/INS integration is apparent. Furthermore, fig. 14.17 indicates that the EKF is fairly

Figure 14.13: The Skywalker X8 UAV

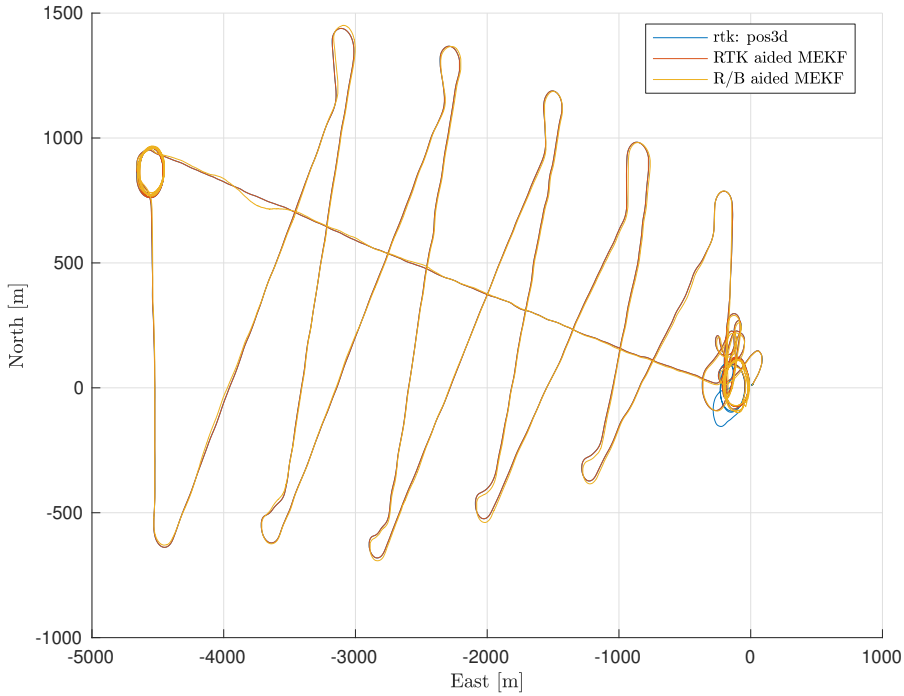


Figure 14.14: Fixed-wing campaign: 2D NED position with RTK GNSS reference

consistent, which shows that the position error relative RTK most of the time is within the 3σ of the EKF covariance. Moreover, looking a table 14.2 and fig. 14.18 one can see that the velocity estimates are accurate compared to the autopilot output, however, somewhat larger than what can be expected using a standalone GNSS solution. The RMSE performance of the horizontal velocity components are proportional to the position RMSE relative RTK.

Regarding attitude one can see that roll and pitch are accurate compared to the autopilot attitude cf. table 14.2. However, the PARS-aided INS has larger discrepancies compared to autopilot. A significant part of out this discrepancy might be due to erroneous magnetometer calibration. From table 14.3 this discrepancy is significantly smaller. The RMSE heading error relative the RTK-aided INS is approximately seven times better that compared to the autopilot data. Looking at the course statistics in table 14.2 and table 14.3 further motivates the claim of erroneous magnetic field measurement in the autopilot since the course estimate of the PARS-aided INS has three times lower RMSE when compared to the course of the RTK-aided INS than when compared to the autopilot course.

Table 14.2: Fixed-wing: Statistics PARS-aided INS relative RTK GNSS/autopilot

Position	**North** [m]	**East** [m]	**Down** [m]	**Norm** [m]
ME:	0.40	-4.34	3.08	5.34
MAE:	6.42	4.88	3.11	8.65
STD:	8.66	3.88	0.99	9.53
RMSE:	8.66	5.82	3.24	10.93

Velocity	**North** [m/s]	**East** [m/s]	**Down** [m/s]	**Norm** [m/s]
ME:	-0.02	-0.14	0.03	0.15
MAE:	0.56	0.38	0.15	0.70
STD:	0.76	0.58	0.24	0.98
RMSE:	0.76	0.59	0.24	1.00

Attitude	**Roll** [°]	**Pitch** [°]	**Yaw** [°]	**Norm** [°]
ME:	-0.20	0.40	2.34	2.38
MAE:	0.70	0.73	9.58	9.63
STD:	1.01	0.93	10.84	10.93
RMSE:	1.03	1.02	11.09	11.18

Speed & Course	$\boldsymbol{U}$ [m/s]	**SOG** [m/s]	**COG** [°]	
ME:	-0.07	-0.07	0.39	–
MAE:	0.55	0.54	2.42	–
STD:	0.76	0.76	10.74	–
RMS:	0.77	0.77	10.74	–

Table 14.3: Fixed-wing: Statistics PARS-aided INS relative RTK-aided INS

Position	**North** [m]	**East** [m]	**Down** [m]	**Norm** [m]
ME:	0.39	-4.32	3.09	5.33
MAE:	6.41	4.86	3.12	8.63
STD:	8.65	3.84	0.98	9.51
RMSE:	8.66	5.78	3.24	10.90

Velocity	**North** [m/s]	**East** [m/s]	**Down** [m/s]	**Norm** [m/s]
ME:	-0.00	-0.08	-0.01	0.08
MAE:	0.41	0.25	0.06	0.49
STD:	0.56	0.44	0.13	0.72
RMSE:	0.56	0.45	0.13	0.72

Attitude	**Roll** [°]	**Pitch** [°]	**Yaw** [°]	**Norm** [°]
ME:	0.03	0.29	-0.56	0.63
MAE:	0.14	0.31	1.03	1.09
STD:	0.22	0.22	1.46	1.49
RMSE:	0.22	0.36	1.56	1.62

Speed & Course	U [m/s]	**SOG** [m/s]	**COG** [°]	
ME:	-0.12	-0.12	0.06	–
MAE:	0.34	0.34	1.36	–
STD:	0.48	0.48	3.48	–
RMS:	0.50	0.49	3.48	–

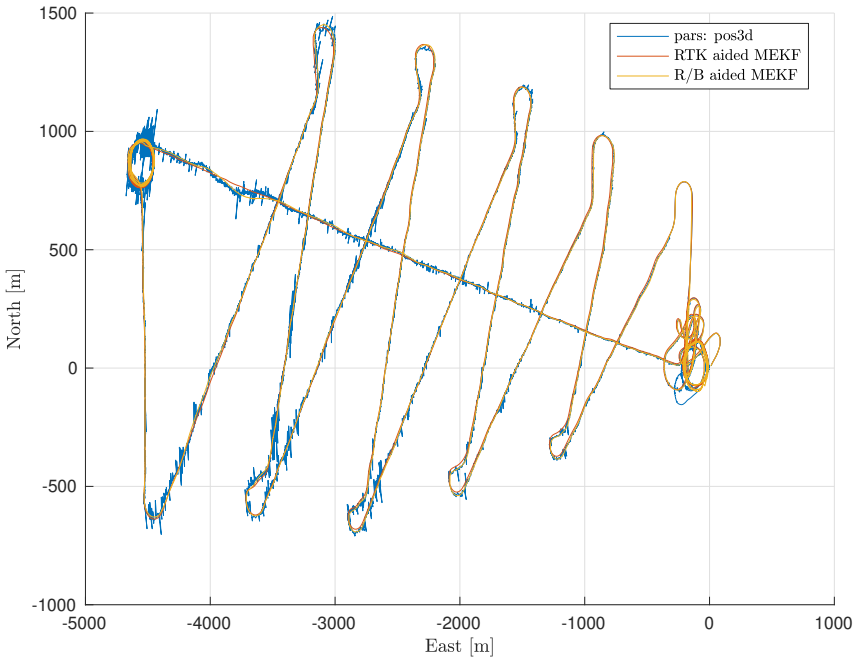


Figure 14.15: Fixed-wing campaign: 2D NED position with PARS reference

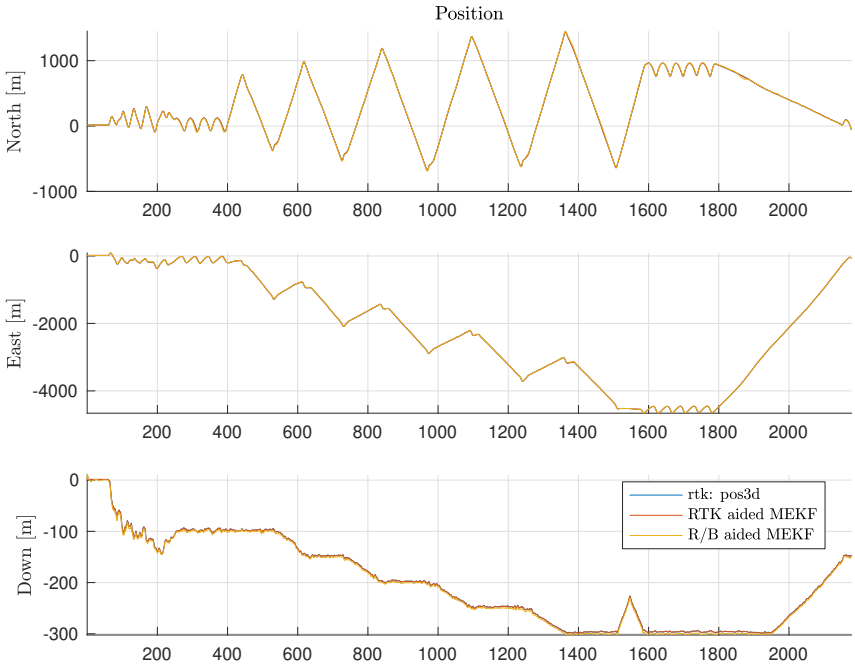


Figure 14.16: Fixed-wing campaign: NED position, compared to RTK GNSS reference

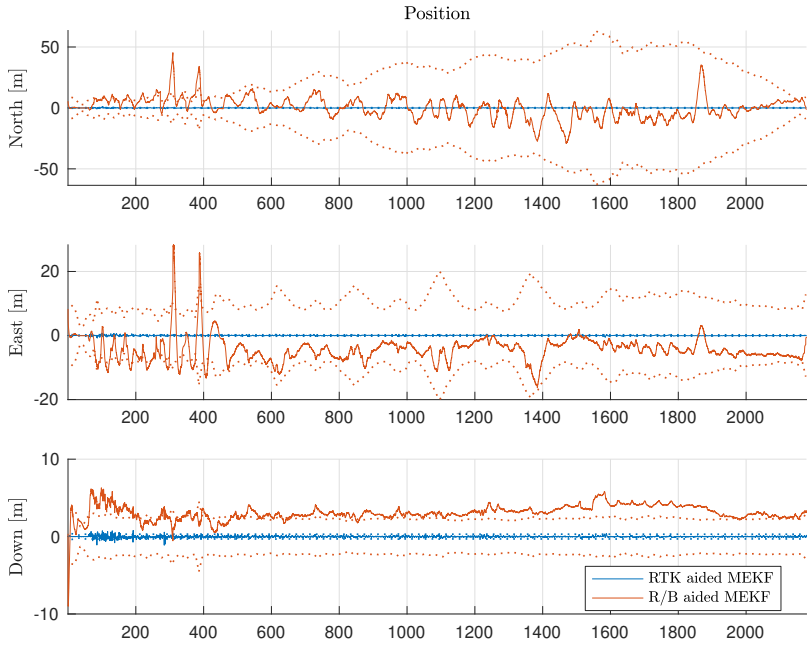


Figure 14.17: Fixed-wing campaign: Position error plot relative RTK GNSS

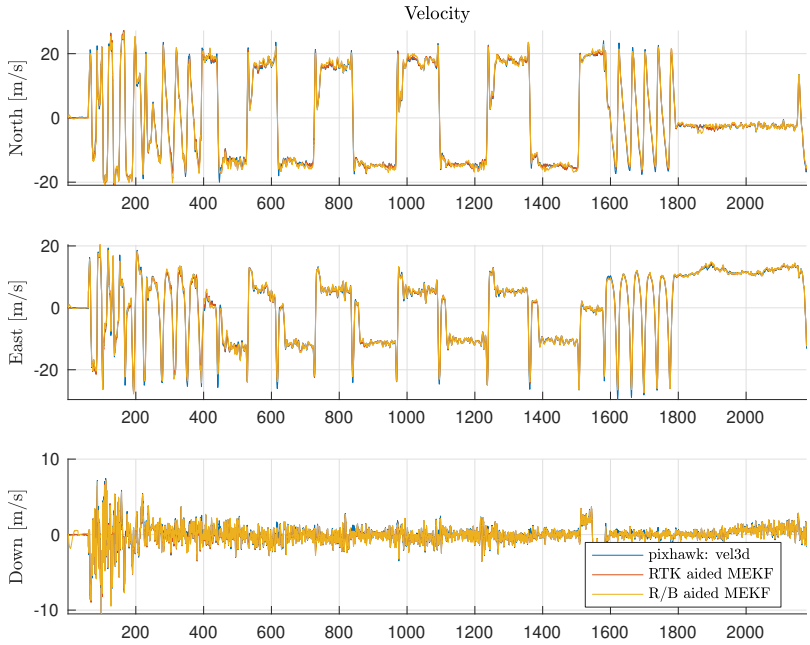


Figure 14.18: Fixed-wing campaign: NED velocity, compared to autopilot reference

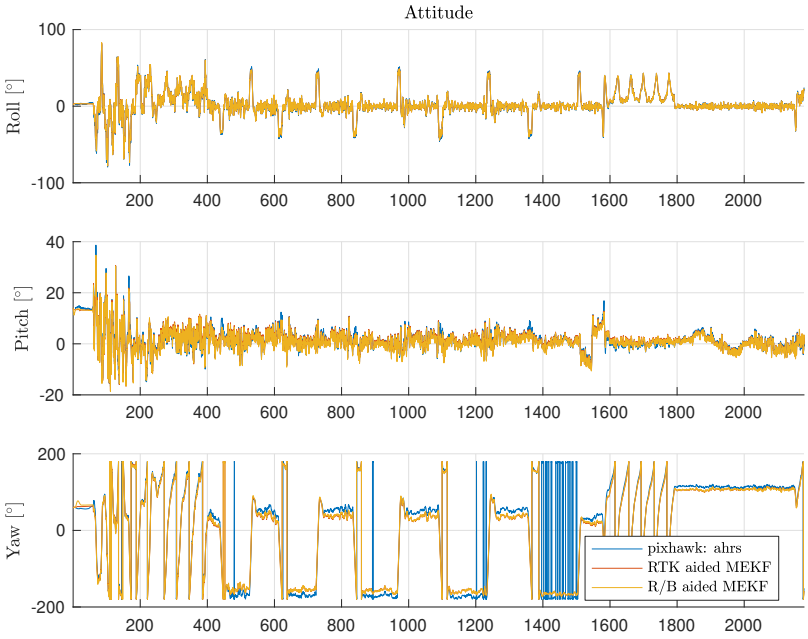


Figure 14.19: Fixed-wing campaign: Attitude, compared to autopilot reference

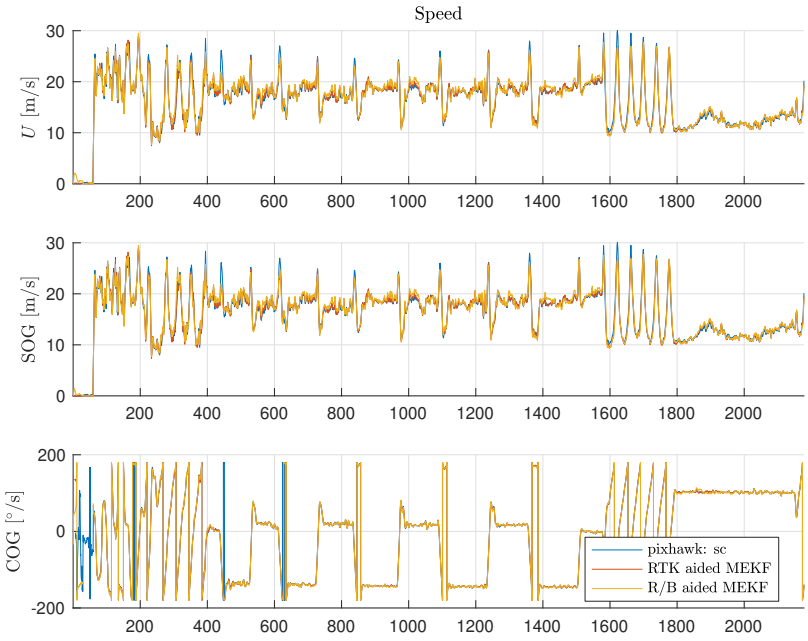


Figure 14.20: Speed and course plot. Total speed U, speed of ground (SOG) and course over ground (COG)

14.5.2 Field experiments

To test the GNC system, the INS, safety mechanisms and guidance controllers where implemented in DUNE, and loaded onto the onboard computer, which was placed in a Skywalker X8 fixed-wing UAV, see fig. 14.4. The PARS-based INS runs synchronously with the IMU measurements, while the aiding measurements are timestamped, added to a queue and processed sequentially as time allows. The time difference between the IMU measurements, as given by the SenTiBoard, determine the sample time for the INS. No compensation of time delays is made for the PARS measurement, as these are believed to be small compared to the measurement errors.

Values for the controller gains for the guidance controllers eqs. (10.16), (14.74) and (14.76) are given in table 14.5. The lateral guidance controller is moderately tuned, to not make abrupt changes in the control signals in the event of jumps in the estimates. Both guidance controllers also include rate-limiting filters for the output reference angles, that are activated for jumps larger than 10° in roll and 5° in pitch, to avoid abrupt changes.

14.5.2.1 Dead-reckoning preliminary test

A series of tests were first performed to establish trust in the system, by testing the abort functionality, and ultimately by testing how the system behaved in the worst-case scenario of loosing contact between the aircraft and the ground station. To

Table 14.4: Abort trigger limits

Parameter	**Value**
Max 4xMRP norm error	0.8 rad
Max NE norm error	100 m
Max NED norm error	100 m
Max pitch error	10°
Max roll error	10°
Trigger Minimum Time	3 s

Table 14.5: Controller tuning

Parameter	**Value**	**Parameter**	**Value**
K_1	0.1	$K_{\gamma,p}$	0.2
Δ_t	6.0 s	$K_{V,p}$	5.0
Maximum Bank	35.0°	$K_{V,i}$	0.4
K_{ph}	0.006	$K_{V,FF}$	1.5
K_{ih}	0.05	α_{trim}	1°
R_{enc}	15.0 m	$T_{t,\text{trim}}$	65 −

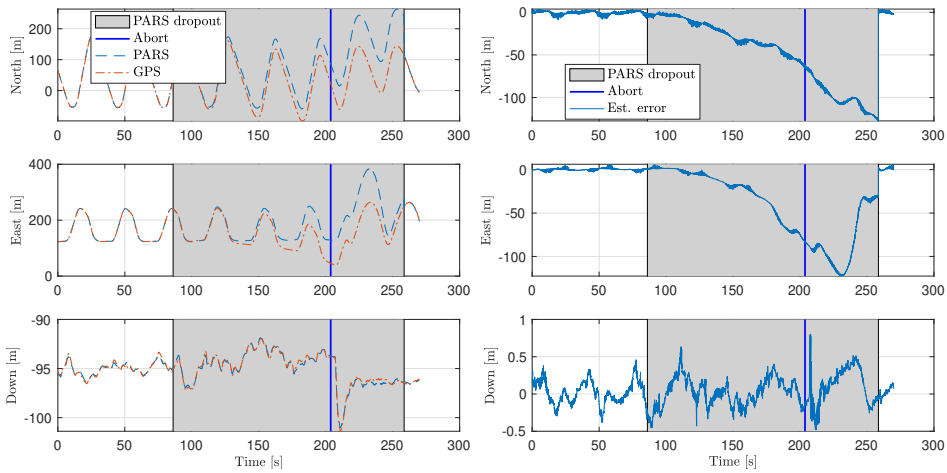
Table 14.6: INS tuning values. The listed covariances, spectral densities and time constants, are to be interpreted as the diagonal value in a 3 × 3 matrix, such that the values apply for both the x, y and z axis

Parameter	Value	Parameter	Value
σ_Ψ	2°	α	0.05
σ_{alt}	5 m	$\boldsymbol{\mathcal{Q}}_{b_{\text{acc}}}$	$4.150\,41 \times 10^{-6}\ \text{m}^2/\text{s}^3$
σ_ρ	15 m	$\boldsymbol{\mathcal{Q}}_{acc} = \boldsymbol{\sigma}^2_{\text{acc}}$	$6.587\,78 \times 10^{-4}\ \text{m}^2/\text{s}^5$
$\boldsymbol{P}_p[0]$	$100\,\text{m}^2$	$\boldsymbol{\mathcal{Q}}_{b_{\text{ars}}}$	$3.950\,05 \times 10^{-15}\ \text{rad}^2/\text{s}$
$\boldsymbol{P}_v[0]$	$4\,\text{m}^2/\text{s}^2$	$\boldsymbol{\mathcal{Q}}_{\text{ars}} = \boldsymbol{\sigma}^2_{\text{ars}}$	$9.214\,68 \times 10^{-7}\ \text{rad}^2/\text{s}^3$
$\boldsymbol{P}_{b_{\text{acc}}}[0]$	$1\,\text{m}^2/\text{s}^4$	$\boldsymbol{T}_{\text{acc}}$	3600 s
$\boldsymbol{P}_{b_{\text{ars}}}[0]$	$3.046\,17 \times 10^{-6}\ \text{rad}^2/\text{s}^2$	$\boldsymbol{T}_{\text{ars}}$	3600 s
$\boldsymbol{P}_a[0]$	$0.030\,461\,7\ \text{rad}^2$		

simulate this scenario, the program that calculates the spherical position from the PARS raw data was turned off, before being started again. This effectively disables the PARS measurements from reaching the INS, rendering it in a dead-reckoning situation, with only inertial and altitude measurements. The tests were performed at the Udduvoll airfield, near Trondheim, Norway, and consist of a simple 200 m by 100 m square.

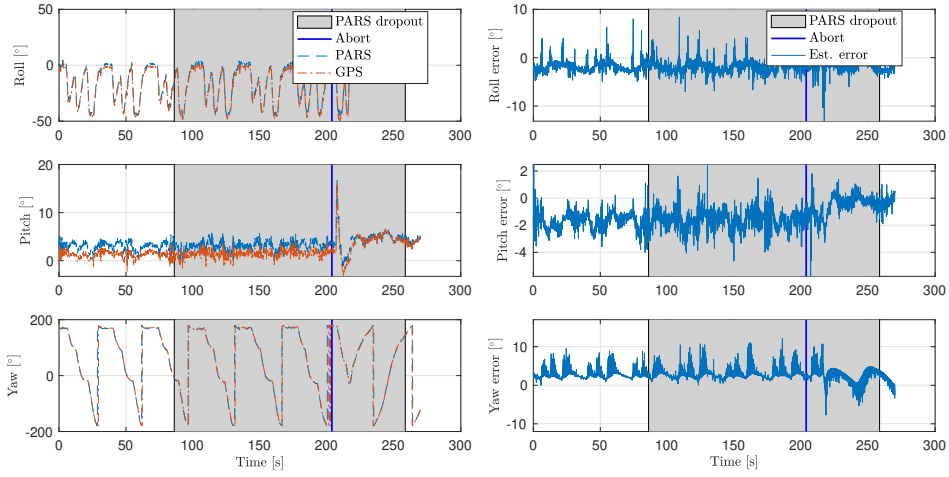
The results are plotted in fig. 14.23, where the shaded area indicate that the PARS measurements are unavailable, while the vertical line at about 200 s, 120 s after the loss of PARS measurements, indicate that the NED position error between the PARS-based and GNSS-based estimates are more than 100 m, which causes the abort plan to be triggered. As the abort plan differs from the previously flown plan, the response, particularly pitch, down position and down velocity, also changes. The PARS measurements are given in fig. 14.25 for reference.

As expected, figs. 14.21a and 14.24 shows that the horizontal position estimates of the INS drift when it is not aided by the PARS measurements. However, once the PARS measurements are available again, the position estimates quickly returns to the close vicinity of the GNSS-based estimates, about 260 s into the experiment. Notice also that in the first part of the PARS dropout in fig. 14.21a, the guidance controllers seemingly keep the UAV in its desired path, which also is seen from fig. 14.24 where the PARS-based position estimates follow the same pattern as before the dropout. However, the GNSS-based position estimate, which is considered the ground truth in this case, shows that the pattern flown by the UAV is actually drifting eastward. The attitude error in fig. 14.22b, particularly the pitch, becomes slightly more noisy in the dead-reckoning. This might be due to the strong kinematic coupling between the attitude parameter and the vertical position component. Considering the NE estimation error velocities of fig. 14.23a, they also have the expected drift under dead-reckoning. This is not so apparent in the derived speed and course over ground, in fig. 14.23b, but the accompanying course error undoubtedly grows. The characteristics of all the estimation errors clearly change after the abort is triggered, as the flown path differs.



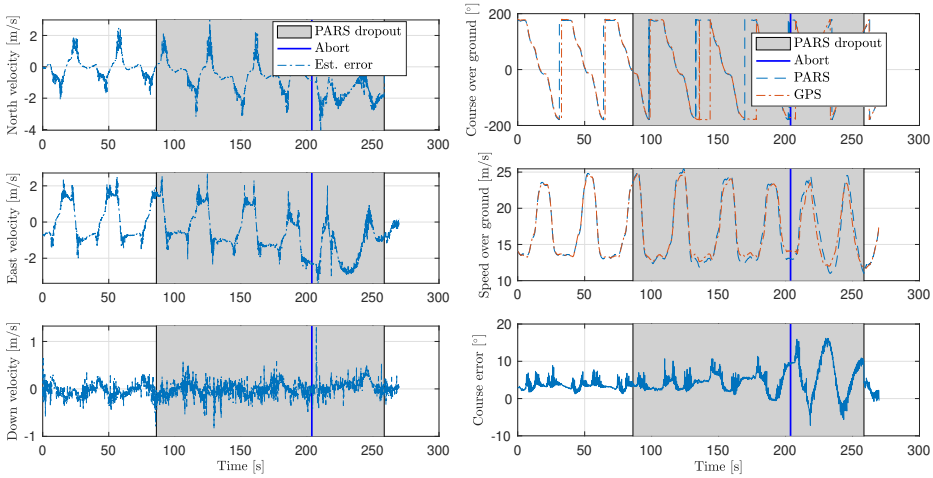
(a) NED position (b) NED position error

Figure 14.21: Position, dead-reckoning test



(a) Attitude (b) Attitude error

Figure 14.22: Attitude in Euler angles, dead-reckoning test



(a) NED velocity errors

(b) Speed and course over ground

Figure 14.23: Dead-reckoning test

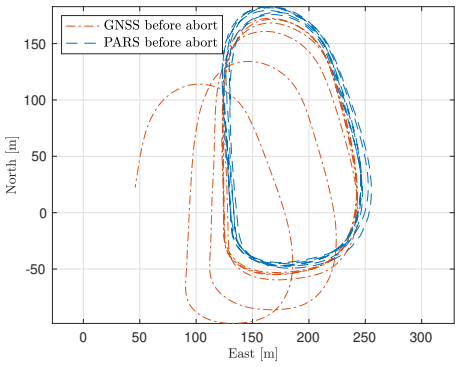


Figure 14.24: 2D NED position, dead-reckoning test. The data after the abort plan is started are not shown, for clarity.

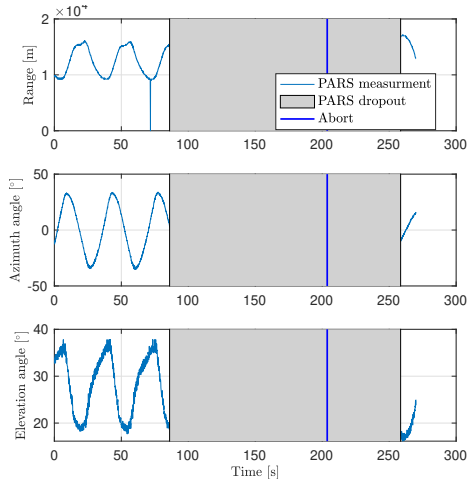


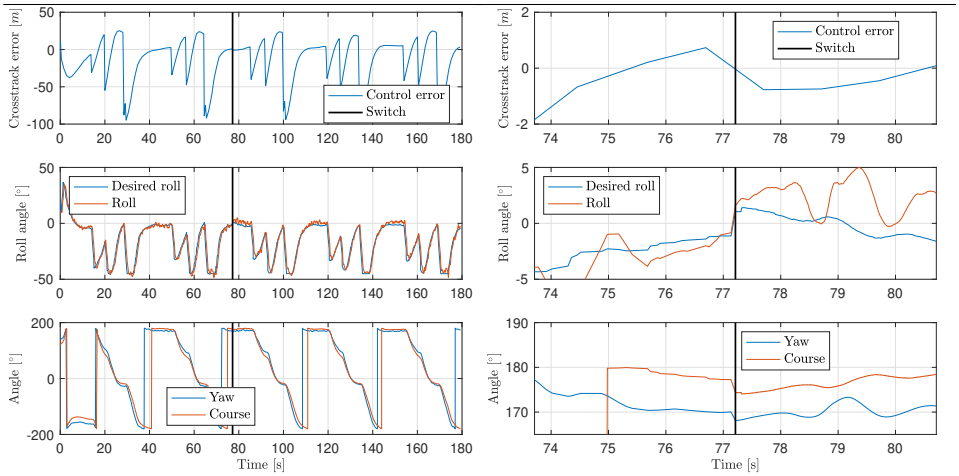
Figure 14.25: PARS measurements, dead-reckoning test

14.5.2.2 Navigation source switch preliminary test

To establish how the control performance is affected by the use of PARS-based navigation, compared to GNSS-based navigation, and during the switch between the two, another flight test was performed in the same location and with the same flight pattern as in section 14.5.2.1. After flying one lap using the GNSS-based estimates, the switch was made to the PARS-based navigation system, indicated by the solid, horizontal line in fig. 14.26, which also shows the cross-track error, roll angle, desired roll angle, yaw angle and course angle. The cross-track error is qualitatively similar before and after the switch. However, as the estimates from the two navigation systems differ, the jump of about 1 m in cross-track error and 2° in yaw and course, seen in fig. 14.26b, is inevitable. These jumps are propagated through the guidance system, leading to a minor jump of 2° in the desired roll angle. This jump is within the allowable limits for the UAV, and is therefore not smoothed. As there are no integral terms in the lateral guidance controller, the jumps will not cause integral windup. It is interesting to note that after the switch, the roll angle seems to have a slightly higher value compared to before the switch. This is likely caused by the misalignments between the two navigation systems, as seen in figs. 14.21b and 14.24, but could also be affected by the fact that the low-level ArduPlane roll controller compares the PARS-based desired roll angle with the roll angle from the GNSS-based navigation system.

The longitudinal response to the navigation system switch is plotted in fig. 14.27, and shows the same tendency as for the lateral response in that the pitch after the switch is offset from the pitch before the switch. There are integral terms in the longitudinal guidance controllers, that could potentially lead to integral windup in the switching, but this does not seem to be a problem in the data, as the difference between the height estimates of the two navigation systems is small.

When considering these results, it is important to keep in mind that the controller considers whatever navigation state it receives from the navigation monitor to be true. The cross-track error, and the derived output values, are functions of this navigation state, not the true position of the UAV. The uncertainty of the navigation system must also be considered to find the error between the true position of the UAV and the desired path, but this can be evaluated separately from the guidance controllers.



(a) Original view (b) Zoomed in on switching time

Figure 14.26: Lateral response from switching navigation source

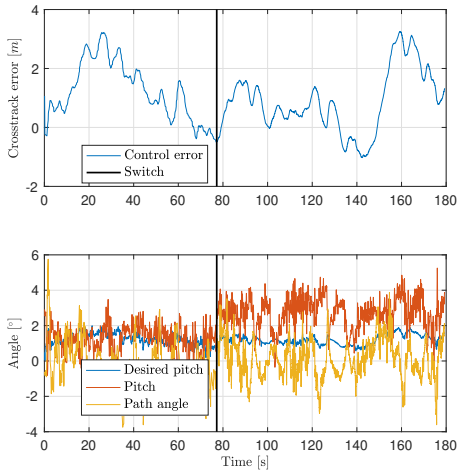


Figure 14.27: Longitudinal response from switching navigation source

14.5.2.3 BVLOS flight

To test the presented guidance and navigation system, a more extensive BVLOS flight in class D controlled airspace was performed at Raudstein, 45 km north-west of Trondheim, Norway. The flight consisted of a 20 km long track in a lawnmower pattern, see fig. 14.28, where the most distant waypoint was over 5 km away from the base antenna, located on the shore just east of the plot view.The flight starts near the origin, then moves along the lawnmower pattern, before returning back in a straight line. Qualitatively, the PARS-based position estimates follow the GNSS-based estimates well, albeit with error of up to 80 m in north and 40 m in east, see

Table 14.7: PARS-INS NED position error statistics (versus GNSS-INS ground truth)

	North [m]	**East** [m]	**Down** [m]	**Norm** [m]
ME	13.46	10.76	0.8643	19.95
MAE	15.35	11.05	0.8840	19.95
STD	16.38	7.601	0.6316	15.05
RMSE	21.20	13.17	1.071	24.99

fig. 14.33b, with mean error, mean absolute error, standard deviation and root mean square error given in table 14.7. Here, the norm column of e.g. the mean absolute error corresponds to the mean absolute error of the norm of the north/east/down components, as opposed to the norm of the mean absolute error of the north, east and down time series.

It is clear that the error in the down axis is small, for both position in fig. 14.33b and table 14.7, and velocity in fig. 14.29c and table 14.8, as the PARS-INS uses the exogenous altitude measurement. The altitude error undoubtedly grows with the geometric range, seen in fig. 14.33b, which likely is an effect of the simplification of representing the position and altitude measurements in NED, when in reality the altitude is given as height above the WGS-84 ellipsoid. The horizontal position estimation error also increase with range, as illustrated in fig. 14.33a.

The raw PARS geometric range, azimuth and elevation measurements are plotted in fig. 14.31, and show a handful of spikes in the measurements over the course of the flight. From the elevation angle it can also be seen when the PARS DoA algorithm jumps from one cluster of measurements to another which yields better communication, as discussed in section 14.2. This is particularly visible at about 750 s, when the elevation angle jumps from about 10° to −20°. However, the effect of these spikes are not seen in the position estimates in figs. 14.28 and 14.33b, as the measurements are rejected as outliers, and thus not considered by the INS.

Figure 14.29a and the accompanying error plot fig. 14.29b shows that there is a slow drift or convergence in the pitch estimate for one of the estimators. This drift is also manifested in the histogram fig. 14.32 of the Euler angle estimation errors, where the pitch is very skewed, while the roll and yaw estimation errors are closer to a bell curve. The drift could be caused by wrong initialization and slow convergence of the slowly varying ARS bias in the z-axis in the PARS-based navigation system, see fig. 14.30b, which affects the pitch estimate through

$$\dot{\theta} = \omega^b_{nb,y} \cos(\phi) - \omega^b_{nb,z} \sin(\phi), \tag{14.79}$$

and demonstrates that there is still some improvements to be made in the tuning. There is also some fluctuations in the accelerometer bias, particularly along the x-axis, seen in fig. 14.30a, which also indicate additional obtainable performance gains.

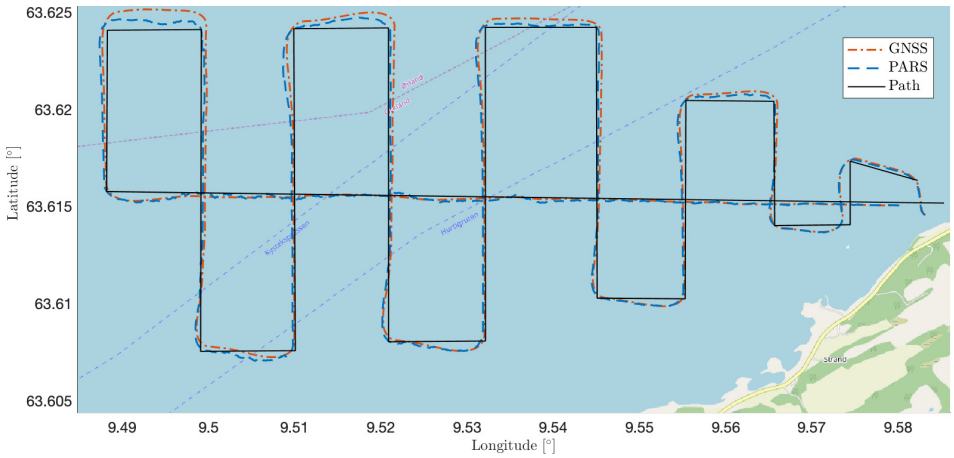


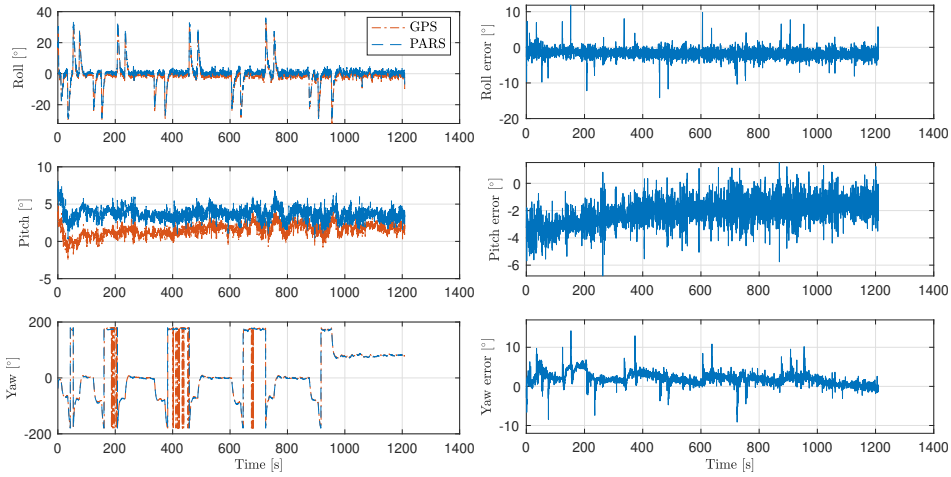
Figure 14.28: 2D position for the BVLOS experiment

Table 14.8: PARS-INS NED velocity error statistics (versus GNSS-INS ground truth)

	North [m/s]	**East** [m/s]	**Down** [m/s]	**Norm** [m/s]
ME	0.1080	-0.06524	-0.008223	0.7884
MAE	0.6648	0.2737	0.1212	0.7884
STD	0.8228	0.3777	0.1612	0.4899
RMSE	0.8298	0.3833	0.1614	0.9282

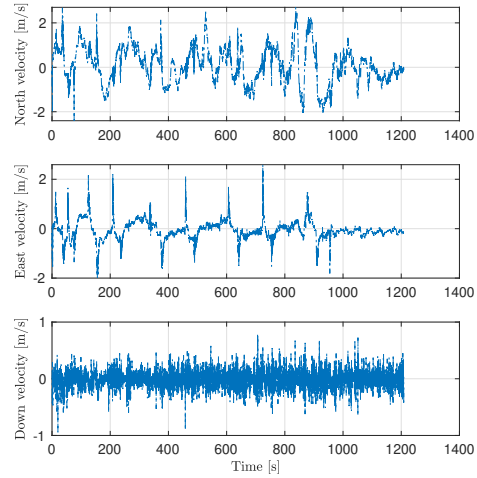
Table 14.9: PARS-INS attitude error statistics (versus GNSS-INS ground truth)

	Roll [°]	**Pitch** [°]	**Yaw** [°]	**Norm** [°]
ME	-1.675	-2.130	1.949	3.661
MAE	1.743	2.133	2.021	3.661
STD	0.9815	0.7837	1.429	1.165
RMSE	1.941	2.270	2.417	3.842

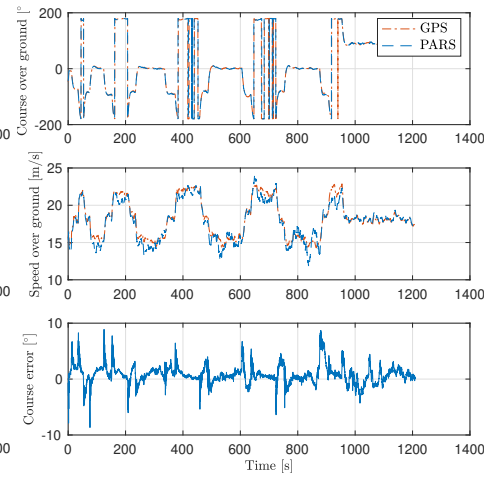


(a) Attitude

(b) Attitude error

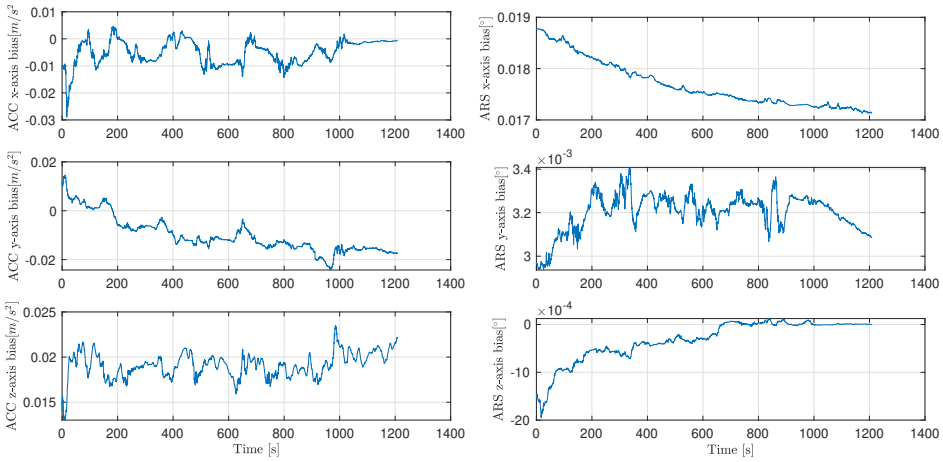


(c) NED velocity error



(d) Speed and course over ground

Figure 14.29: Beyond-visual-line-of-sight experiment



(a) Accelerometer bias estimate

(b) Angular rate bias estimate

Figure 14.30: Estimated IMU biases, BVLOS experiment

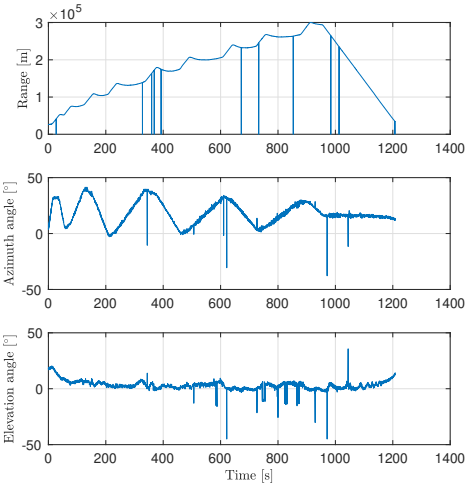


Figure 14.31: PARS measurements, BVLOS experiment

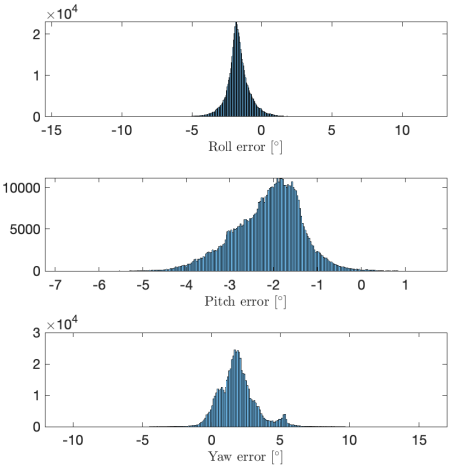
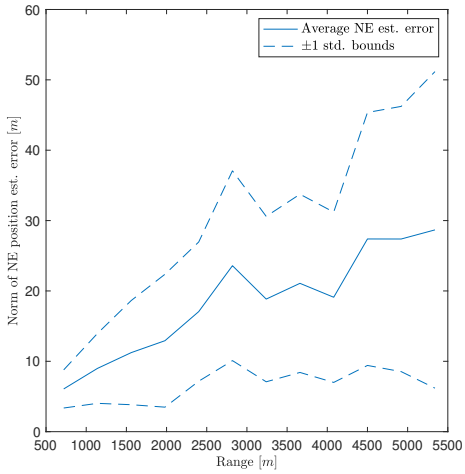
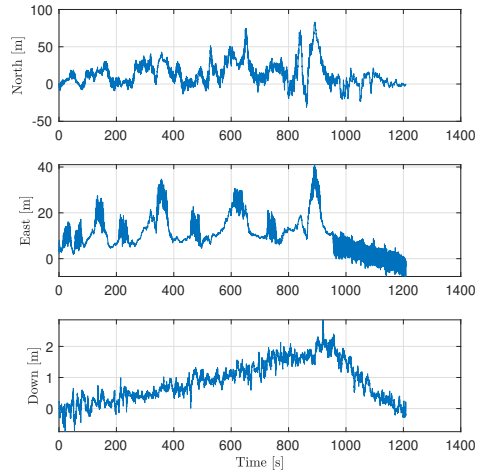


Figure 14.32: Histogram of attitude estimation error in Euler angles, BVLOS experiment



(a) Bin average horizontal position estimation error, with standard deviation bounds, versus geometric range, for 12 bins.



(b) NED position error

Figure 14.33: Position estimation error, BVLOS experiment

14.6 Discussion

The results show that the PARS-aided navigation system has merit for autopilot flight control, although the position errors are larger than what one can expect from GNSS-based systems. When considering this, it is important to note the potential misalignment of the ground station PARS antenna. The antenna is manually calibrated by leveling the roll and pitch angles of the antenna, before the azimuth angle is measured using a compass. As the compass is susceptible to magnetic anomalies, this is validated in a comparison of the GNSS-based and PARS-based position estimates while the UAV is stationary on the ground, where the azimuth angle is adjusted until the positions align. When flying at ranges of over 5 km, a misalignment of 1° in azimuth, which is an optimistic scenario given the above calibration routine, alone leads to horizontal position errors just short of 100 m. The non-zero-mean position error in fig. 14.33b also supports this argument. This motivates further research into an additional calibration mode, that utilize GNSS measurements when available/allowable, to estimate the misalignment bias of the ground station PARS antenna. The performance of the guidance controllers are marginally affected by the use of PARS-based navigation estimates. The only qualitative differences in the performance can be attributed to misalignments in the two navigation systems, and to the current interconnection of the GNC system, where the PARS-based roll and pitch references are compared to the GNSS-based roll and pitch estimates.

14.7 Conclusion

Through experimental verification, with both a multirotor and a fixed-wing UAV, it is shown that the PARS-aided INS provides robust position, velocity and attitude signal alternatives to GNSS-based solutions, for a wide range of UAV operations. The accuracy of the estimates, especially for position, are lower than what is achievable with GNSS-based navigation. The presented navigation system makes the UAV independent of GNSS, compass and magnetometers measurements, given that the pose of the PARS base radio antenna is known and the motion of the UAV provides sufficient excitation for full attitude and IMU bias estimation. The safety and accuracy of GNSS-free flight control system has also been demonstrated through field experiments involving beyond-visual-line-of-sight flight with a fixed-wing UAV in controlled airspace, over open water. Although the position errors from this test, 19.95 m in the norm, are larger than what would have been expected with a similar GNSS-based system, a portion can likely be attributed to the misalignment of the phased-array ground station antenna. To safely operate the UAV, a system consisting of a guidance controller module and several safety mechanisms, like switching between and monitoring the different navigation systems, was designed and implemented.

Part IV

Concluding Remarks

Chapter 15

Conclusion & Future Possibilities

This thesis has contributed to the increases level of autonomy for unmanned aerial vehicles (UAVs) by increasing the precision of control and the robustness of navigation.

The increased control performance is manifested in through precision recovery in an arrest system, precision deep-stall landing, and accurate guidance in the presence of wind. The presented, modular system for recovery in moving arrest systems, such as nets, was based on commercial-off-the-shelf components by building non-intrusively on available autopilots to retain their reliability. This was achieved by assuming a general interface, through which the autopilot receives position measurements, which in this case came from RTK-GNSS, and accepts position commands, who originate from a line-of-sight-based algorithm that allows path-following of a moving line, which represent a virtual runway into the arrest system. This was demonstrated through 43 recoveries in a stationary net, with a mean error of -0.07 ± 0.20 meter and -0.01 ± 0.25 meter in the vertical and horizontal axes, respectively. Preliminary experimental recoveries in a moving net are encouraging.

A more exotic landing strategy, deep stall, where the increased drag leads to reduced landing speeds, was investigated in a nonlinear-model-predictive-control framework, enabling firm constraints on the actuators and UAV dynamics. The controller was implemented using direct multiple-shooting in the CasADi framework. Extensive Monte-Carlo simulations showed that the algorithm was able to precisely deep-stall land the UAV near its desired landing location, even in winds of up to 7 m/s, with significant speed reduction compared to cruise.

Reductions in cross-track error, in strong winds, of 10 % compared to the state-of-the-art was achieved by deriving a line-of-sight-based guidance controller based on the coordinated-turn relation, which was derived in a general form. The controller was shown to have an uniformly semiglobally exponentially stable equilibrium point, through a Lyapunov-based analysis of the cascaded system.

Robustness of UAV navigation focused on how it could be made resilient to loss of GNSS. The first approach investigates the use of ultra-wideband radio beacons as a means to aid an inertial navigation system, in the event of GNSS-outage. A nonlinear observer, consisting of an interconnection of an attitude and a translational motion observer, was used to integrate tight integration of double-differenced GNSS measurements and single-differenced ultra-wideband range measurements with an inertial navigation system. Simulations showed that, trough the addition of ultra-wideband measurements, the observer was able to keep the positioning error well within 1 meter despite loss of GNSS, and showed improvement in position estimation, particularly in the z-axis, also when GNSS was available. This demonstrated that ultra-wideband-based navigation is particularly promising for local navigation, or to enhance other global solutions by increasing the robustness and precision when ultra-wideband is available.

This work has also demonstrated that a relatively new concept, phased-array radio systems (PARS), can also be used to aid an inertial navigation system, to become independent of GNSS. The PARS elevation angle, azimuth angle and range measurements were used in a multiplicative extended Kalman filter to provide estimates of position, velocity and attitude of the UAV, without the use of GNSS and magnetometer. The navigation solution was shown to be applicable to both fixed-wing and multirotor UAVs, with performance comparable to that of standard position service GPS. The system was also included into a complete guidance, navigation and control (GNC) system, to safely demonstrate its capabilities. Multiple flights without GNSS, the longest being a 20 km beyond-visual-line-of-sight flight in controlled airspace over open water, 5 km away from the base antenna, demonstrated that the system has merit.

Finally, through the identification of a complete aerodynamic, propulsion and inertial model for the Skywalker X8 flying-wing UAV, this thesis has laid the foundation for fault detection of e.g. icing, which can lead to more robust solutions, operations in a wider range of conditions with lower risk. The general derivation of the coordinated-turn equation gives insights into how the UAV is affected by wind, which has helped improve performance, through the presented guidance law, but it can also be used in path planning and to make more nuanced considerations regarding what operations are feasible under the given conditions.

Future improvements

Despite the successes listed above, UAV operations are still not fully autonomous. Some extensions of the above that would enhance this are

- A combination of the research on ultra-wideband-based navigation, PARS-based navigation and arrest system recovery, where the PARS navigates the UAV to the close vicinity of the landing site, where the ultra-wideband radio beacons are placed, to increase precision for the final stages of the automatic

landing. It is noted that field experiments using a similar UWB setup as in chapter 13 were performed in [26]. However, additional work is needed to get sufficient results.

- Adapt the deep-stall landing algorithm for implementation and testing in a physical UAV. This hinges on good models of UAVs that are capable of deep-stall landing. A possible first step would be to adapt the algorithm to control the presented X8 model in normal flight. Some progress towards an online implementation of the nonlinear model predictive controller was made in [120], but further research is needed.
- Deeper integration of PARS, where state is used in tracking and use multi hypothesis tracking, to become more robust against multipath. The online performance of the PARS navigation should also be improved by estimating the misalignment in the ground antenna by using GNSS when it can be trusted.
- Enhance the presented guidance law by adding integral effect, to overcome stationary errors from e.g. misalignment in the mounting of the inertial navigation system. Extending the guidance law to 3D could also provide interesting insights into the coupling between the lateral and longitudinal axes in wind.
- The automatic recovery in an arrest system should be tested on a ship. This was originally scheduled for March/April 2020, but was postponed due to COVID-19.

Appendices

Appendix **A**

Notational differences between chapter 10 and L_1 guidance

It should be noted that the variables $\eta = \eta_1 + \eta_2$ from [144] can be shown to be

$$\eta = \tilde{\chi} = \chi_d - \chi \tag{A.1}$$

$$\eta_1 = \chi_{\text{LOS}} \tag{A.2}$$

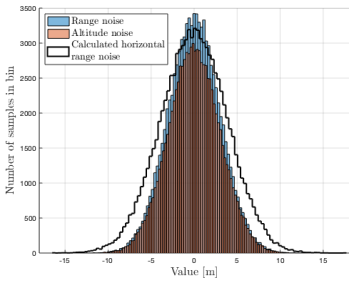
$$\eta_2 = \chi_p - \chi \tag{A.3}$$

Appendix **B**

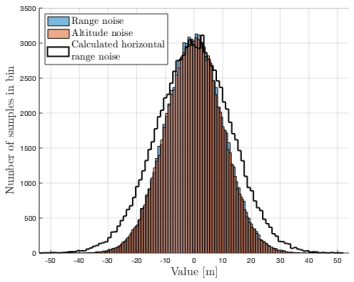
PARS derivations

B.1 Horizontal range bias calculation

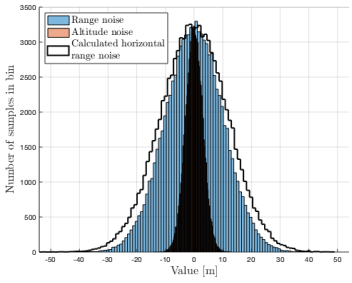
We performed Monte Carlo simulations to determine the distribution of the noise of the calculated horizontal range measurement in section 14.2.2. The results indicated that noise of the calculated horizontal range would be sufficiently normal distributed, illustrate in in fig. B.1. Therefore, $b_{\rho/\text{alt}} = 1$ in eq. (14.21) was set.



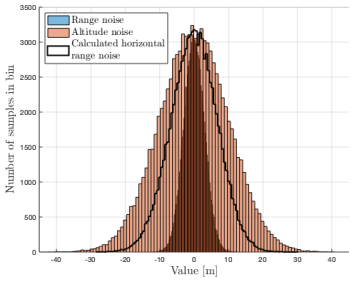
(a) Range noise: $\sigma = 3$ m. Altitude noise: $\sigma = 3$ m.



(b) Range noise: $\sigma = 10$ m. Altitude noise: $\sigma = 10$ m.



(c) Range noise: $\sigma = 10$ m. Altitude noise: $\sigma = 3$ m.



(d) Range noise: $\sigma = 3$ m. Altitude noise: $\sigma = 10$ m.

Figure B.1: Example distribution of the horizontal range noise calculations from simulations. True range value: 1000 m. True altitude value: 500 m

B.2 MEKF error-state kinematics

The kinematics of the error-state MEKF regarding the position error, ACC bias error and ARS bias error are trivial to derive. Therefore we chose to only prove the derivation for the velocity and attitude error kinematics below

B.2.1 Velocity error

The true velocity is given

$$\boldsymbol{v}^n_{nb} = \boldsymbol{v}^n_{nb,\text{ins}} + \delta\boldsymbol{v}. \tag{B.1}$$

Thus

$$\begin{aligned}\dot{\boldsymbol{v}}^n_{nb} &= \boldsymbol{R}_{nb}\left(\boldsymbol{f}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{acc,ins}} - \delta\boldsymbol{b}^b_{\text{acc}} - \boldsymbol{\varepsilon}^b_{\text{acc}}\right) + \boldsymbol{g}^n_b \\ &= \dot{\boldsymbol{v}}^n_{nb,\text{ins}} + \delta\dot{\boldsymbol{v}}\end{aligned} \tag{B.2}$$

using

$$\boldsymbol{f}^b_{\text{imu}} = \boldsymbol{f}^b_{nb} + \boldsymbol{b}^b_{\text{acc,ins}} + \delta\boldsymbol{b}^b_{\text{acc}} + \boldsymbol{\varepsilon}^b_{\text{acc}}. \tag{B.3}$$

and where

$$\dot{\boldsymbol{v}}^n_{nb,\text{ins}} = \boldsymbol{R}_{nb,\text{ins}}\left(\boldsymbol{f}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{acc,ins}}\right) + \boldsymbol{g}^n_b. \tag{B.4}$$

Moreover, using four times MRPs, $\delta\boldsymbol{a} = 4\delta\boldsymbol{a}_{\text{mrp}}$, we have that

$$\boldsymbol{R}_{nb} = \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{R}(\delta\boldsymbol{a}) \tag{B.5}$$

where $\boldsymbol{R}(\delta\boldsymbol{a})$ can be approximated to

$$\boldsymbol{R}(\delta\boldsymbol{a}) \approx \boldsymbol{I}_3 + 4\boldsymbol{S}(\delta\boldsymbol{a}_{\text{mrp}}) = \boldsymbol{I}_3 + \boldsymbol{S}(\delta\boldsymbol{a}), \tag{B.6}$$

as in [116, Ch. 3], since the error dynamics are slow and $\delta\boldsymbol{a}$ should be close to zero. Now be rearranging eq. (B.2) and inserting for eq. (B.4) and eq. (B.6) and using eq. (14.33) we get

$$\begin{aligned}
\delta\dot{\boldsymbol{v}} &= \dot{\boldsymbol{v}}^n_{nb} - \dot{\boldsymbol{v}}^n_{nb,\text{ins}} \\
&= \boldsymbol{R}_{nb,\text{ins}}\left(\boldsymbol{I}_3 + \boldsymbol{S}(\delta\boldsymbol{a})\right)\left(\hat{\boldsymbol{f}}^b_{nb} - \delta\boldsymbol{b}^b_{\text{acc}} - \boldsymbol{\varepsilon}^b_{\text{acc}}\right) \\
&+ \boldsymbol{g}^n_b - \boldsymbol{R}_{nb,\text{ins}}\hat{\boldsymbol{f}}^b_{nb} - \boldsymbol{g}^n_b \\
&= -\boldsymbol{R}_{nb,\text{ins}}\delta\boldsymbol{b}^b_{\text{acc}} - \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{\varepsilon}^b_{\text{acc}} + \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{S}(\delta\boldsymbol{a})\hat{\boldsymbol{f}}^b_{nb} \\
&\quad - \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{S}(\delta\boldsymbol{a})\delta\boldsymbol{b}^b_{\text{acc}} - \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{S}(\delta\boldsymbol{a})\boldsymbol{\varepsilon}^b_{\text{acc}}.
\end{aligned} \tag{B.7}$$

Then we exploit the property $\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = -\boldsymbol{S}(\boldsymbol{b})\boldsymbol{a}$ for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ and defining that $\delta\boldsymbol{f} := -\boldsymbol{b}^b_{\text{acc}} - \boldsymbol{\varepsilon}^b_{\text{acc}}$. Hence,

$$\delta\dot{\boldsymbol{v}} = \boldsymbol{R}_{nb,\text{ins}}\delta\boldsymbol{f} - \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{S}(\hat{\boldsymbol{f}}^b_{nb})\delta\boldsymbol{a} + \boldsymbol{R}_{nb,\text{ins}}\boldsymbol{S}(\delta\boldsymbol{a})\delta\boldsymbol{f}. \tag{B.8}$$

Moreover, since we linearize about $\delta\boldsymbol{a} = \delta\boldsymbol{b}^b_{\text{acc}} = \boldsymbol{0}_{3\times1}$ we achieve

$$\delta\dot{\boldsymbol{v}} \approx -\boldsymbol{R}_{nb,\text{ins}}\boldsymbol{S}(\hat{\boldsymbol{f}}^b_{nb})\delta\boldsymbol{a} - \boldsymbol{R}_{nb,\text{ins}}\left(\delta\boldsymbol{b}^b_{\text{acc}} + \boldsymbol{\varepsilon}^b_{\text{acc}}\right). \tag{B.9}$$

B.2.2 Attitude error

The true quaternion is given as

$$\boldsymbol{q}^n_b = \boldsymbol{q}^n_{b,\text{ins}} \otimes \delta\boldsymbol{q}. \tag{B.10}$$

Hence,

$$\Rightarrow \delta\boldsymbol{q} = \left(\boldsymbol{q}^n_{b,\text{ins}}\right)^* \boldsymbol{q}^n_b. \tag{B.11}$$

We are continuing with deriving the attitude error kinematics. From eq. (B.11) we have that

$$\delta\dot{\boldsymbol{q}} = \left(\dot{\boldsymbol{q}}^n_{b,\text{ins}}\right)^* \otimes \boldsymbol{q}^n_b + \left(\boldsymbol{q}^n_{b,\text{ins}}\right)^* \otimes \dot{\boldsymbol{q}}^n_b. \tag{B.12}$$

Moreover, by using that eq. (14.35) can be written like

$$\boldsymbol{\omega}^b_{\text{imu}} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{\text{ars,ins}} + \delta\boldsymbol{b}^b_{\text{ars}} + \boldsymbol{\varepsilon}^b_{\text{ars}}, \tag{B.13}$$

which results in

$$\begin{aligned}\boldsymbol{\omega}^b_{nb} &= \boldsymbol{\omega}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{ars,ins}} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{\varepsilon}^b_{\text{ars}}, \\ &= \hat{\boldsymbol{\omega}}^b_{nb} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{\varepsilon}^b_{\text{ars}},\end{aligned} \tag{B.14}$$

using eq. (14.33). By further applying sections 2.2.1, 2.2.3 and 14.3.1 we have that

$$\dot{\boldsymbol{q}}^n_b = \frac{1}{2}\boldsymbol{q}^n_b \otimes \begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{\varepsilon}^b_{\text{ars}} \end{bmatrix}, \tag{B.15}$$

$$\dot{\boldsymbol{q}}^n_{b,\text{ins}} = \frac{1}{2}\boldsymbol{q}^n_{b,\text{ins}} \otimes \begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} \end{bmatrix}. \tag{B.16}$$

Hence

$$\left(\dot{\boldsymbol{q}}^n_{b,\text{ins}}\right)^* = -\frac{1}{2}\begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} \end{bmatrix} \otimes \boldsymbol{q}^b_{n,\text{ins}} \tag{B.17}$$

using eq. (2.12) and eq. (2.11). Furthermore, it follows that

$$\begin{aligned}\delta\dot{\boldsymbol{q}} &= -\frac{1}{2}\begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} \end{bmatrix} \otimes \boldsymbol{q}^b_{n,\text{ins}} \otimes \boldsymbol{q}^n_b \\ &\quad + \frac{1}{2}\boldsymbol{q}^b_{n,\text{ins}} \otimes \boldsymbol{q}^n_b \otimes \begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{\varepsilon}^b_{\text{ars}} \end{bmatrix} \\ &= -\frac{1}{2}\begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} \end{bmatrix} \otimes \delta\boldsymbol{q} \\ &\quad + \frac{1}{2}\delta\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}^b_{nb} - \delta\boldsymbol{b}^b_{\text{ars}} - \boldsymbol{\varepsilon}^b_{\text{ars}} \end{bmatrix}.\end{aligned} \tag{B.18}$$

Moreover, by defining

$$\boldsymbol{\omega}^b_{nb} := \hat{\boldsymbol{\omega}}^b_{nb} + \delta\boldsymbol{\omega}, \tag{B.19}$$

we achieve

$$\begin{aligned}\delta\boldsymbol{\omega} &= \boldsymbol{\omega}^b_{nb} - \hat{\boldsymbol{\omega}}^b_{nb} \\ &= -\delta\boldsymbol{b}_{\text{ars}} - \boldsymbol{\varepsilon}^b_{\text{ars}}.\end{aligned} \tag{B.20}$$

By using this, eq. (B.18) can be written like

$$\begin{aligned}\delta\dot{\boldsymbol{q}} = \frac{1}{2}&\begin{bmatrix} 0 & -\left(\hat{\boldsymbol{\omega}}^b_{nb} + \delta\boldsymbol{\omega}\right)^\intercal \\ \left(\hat{\boldsymbol{\omega}}^b_{nb} + \delta\boldsymbol{\omega}\right) & -\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}^b_{nb} + \delta\boldsymbol{\omega}\right) \end{bmatrix}\delta\boldsymbol{q} \\ -\frac{1}{2}&\begin{bmatrix} 0 & -\left(\hat{\boldsymbol{\omega}}^b_{nb}\right)^\intercal \\ \left(\hat{\boldsymbol{\omega}}^b_{nb}\right) & \boldsymbol{S}\left(\hat{\boldsymbol{\omega}}^b_{nb}\right) \end{bmatrix}\delta\boldsymbol{q}\end{aligned} \tag{B.21}$$

by using relations eqs. (2.7) to (2.10). It follows that

$$\begin{aligned}\delta\dot{\boldsymbol{q}} &= \frac{1}{2}\begin{bmatrix} 0 & -\delta\boldsymbol{\omega}^{\intercal} \\ \delta\boldsymbol{\omega} & -\boldsymbol{S}(\delta\boldsymbol{\omega}) - 2\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right) \end{bmatrix}\delta\boldsymbol{q} \\ &= \frac{1}{2}\left(\boldsymbol{\Omega}(\delta\boldsymbol{\omega}) + \bar{\boldsymbol{\Gamma}}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right)\right)\delta\boldsymbol{q} \end{aligned} \tag{B.22}$$

where

$$\begin{aligned}\bar{\boldsymbol{\Gamma}}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right) &= \boldsymbol{\Omega}\left(\hat{\boldsymbol{\omega}}_{nb}^{b} + \delta\boldsymbol{\omega}\right) - \boldsymbol{\Gamma}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right) \\ &= \begin{bmatrix} 0 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & -2\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right) \end{bmatrix}. \end{aligned} \tag{B.23}$$

Furthermore, we can write eq. (B.22)

$$\begin{aligned}\begin{bmatrix} \delta\dot{q}_s \\ \delta\dot{\boldsymbol{q}}_v \end{bmatrix} &= \frac{1}{2}\begin{bmatrix} 0 & -\delta\boldsymbol{\omega}^{\intercal} \\ \delta\boldsymbol{\omega} & -\boldsymbol{S}(\delta\boldsymbol{\omega}) \end{bmatrix}\begin{bmatrix} \delta q_s \\ \delta\boldsymbol{q}_v \end{bmatrix} \\ &\quad + \frac{1}{2}\begin{bmatrix} 0 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & -2\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right) \end{bmatrix}\begin{bmatrix} \delta q_s \\ \delta\boldsymbol{q}_v \end{bmatrix}, \end{aligned} \tag{B.24}$$

which again result in

$$\delta\dot{q}_w = -\frac{1}{2}(\delta\boldsymbol{\omega})^{\intercal}\,\delta\boldsymbol{q}_v, \tag{B.25}$$

$$\delta\dot{\boldsymbol{q}}_v = \frac{1}{2}\left(\delta q_s\delta\boldsymbol{\omega} - \boldsymbol{S}(\delta\boldsymbol{\omega})\,\delta\boldsymbol{q}_v - 2\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right)\right). \tag{B.26}$$

Moreover, from eq. (2.14) we have that

$$\delta\boldsymbol{a} = 4\delta\boldsymbol{a}_{\text{mrp}} = 4\frac{\delta\boldsymbol{q}_v}{1+\delta q_w} \tag{B.27}$$

and

$$\delta\dot{\boldsymbol{a}}_{\text{mrp}} = \frac{\delta\dot{\boldsymbol{q}}_v}{1+\delta q_w} - \frac{\delta\dot{q}_w\delta\boldsymbol{q}_v}{(1+\delta q_w)^2}. \tag{B.28}$$

Then, by inserting for eqs. (B.25) to (B.26) one obtain

$$\begin{aligned}\delta\dot{\boldsymbol{a}}_{\text{mrp}} &= \frac{1}{4}\Bigg(-2\left(\boldsymbol{S}(\delta\boldsymbol{\omega}) + 4\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right)\right)\delta\boldsymbol{a}_{\text{mrp}} \\ &\quad + \left(1 - \delta\boldsymbol{a}_{\text{mrp}}^{\intercal}\delta\boldsymbol{a}_{\text{mrp}}\right)\delta\boldsymbol{\omega}\Bigg) \\ &\quad + \frac{1}{2}\left(\delta\boldsymbol{\omega}^{\intercal}\delta\boldsymbol{a}_{\text{mrp}}\right)\delta\boldsymbol{a}_{\text{mrp}}. \end{aligned} \tag{B.29}$$

See, [202] for details. By linearizing eq. (B.29) w.r.t. $\delta\boldsymbol{a}_{\text{mrp}}$ and $\delta\boldsymbol{\omega}$ and about $\delta\boldsymbol{a}_{\text{mrp}} = \delta\boldsymbol{b}_{\text{ars}}^{b} = \boldsymbol{0}_{3\times 1}$ we achieve

$$\begin{aligned}\delta\dot{\boldsymbol{a}} = 4\delta\dot{\boldsymbol{a}}_{\text{mrp}} &= -4\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right)\delta\boldsymbol{a}_{\text{mrp}} - \frac{4}{4}\delta\boldsymbol{b}_{\text{ars}}^{b} - \frac{4}{4}\boldsymbol{\varepsilon}_{b,\text{ars}}^{b} \\ &= -\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right)\delta\boldsymbol{a} - \delta\boldsymbol{b}_{\text{ars}}^{b} - \boldsymbol{\varepsilon}_{\text{ars}}^{b}. \end{aligned} \tag{B.30}$$

B.2.3 Linearized MEKF matrices

The resulting linearized $\boldsymbol{A}(t)$ and $\boldsymbol{B}(t)$ matrices from appendices B.2.1 to B.2.2 of the error model eq. (14.36) are given as

$$\boldsymbol{A}(t) = \begin{pmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{V}_a & \boldsymbol{V}_{\text{acc}} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{A}_a & \boldsymbol{0}_{3\times3} & \boldsymbol{A}_{\text{ars}} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{T}_{\text{acc}}^{-1} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{T}_{\text{ars}}^{-1} \end{pmatrix} \in \mathbb{R}^{15\times15}, \tag{B.31}$$

$$\boldsymbol{B}(t) = \begin{pmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ -\boldsymbol{R}_{nb}(\boldsymbol{q}_{b,\text{ins}}^{n}) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \end{pmatrix} \in \mathbb{R}^{15\times12}, \tag{B.32}$$

where

$$\boldsymbol{V}_a = -\boldsymbol{R}_{nb}(\boldsymbol{q}_{b,\text{ins}}^{n})\boldsymbol{S}\left(\hat{\boldsymbol{f}}_{nb}^{b}\right), \quad \boldsymbol{V}_{\text{acc}} = -\boldsymbol{R}_{nb}(\boldsymbol{q}_{b,\text{ins}}^{n}),$$
$$\boldsymbol{A}_a = -\boldsymbol{S}\left(\hat{\boldsymbol{\omega}}_{nb}^{b}\right), \qquad \boldsymbol{A}_{\text{ars}} = -\boldsymbol{I}_3.$$

Bibliography

[1] I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections*. Dover Publications Inc., 1959.

[2] 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.

[3] S. M. Albrektsen, T. H. Bryne, and T. A. Johansen, "Phased array radio system aided inertial navigation for unmanned aerial vehicles", in *Proc. of the IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2018, pp. 1–11. DOI: 10.1109/AERO.2018.8396433.

[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] S. M. Albrektsen, A. Sœgrov, and T. A. Johansen, "Navigation of uav using phased array radio", in *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, Linkoping, Sweden, Mar. 2017, pp. 138–143. DOI: 10.1109/RED-UAS.2017.8101657.

[6] S. M. Albrektsen, T. H. Bryne, and T. A. Johansen, "Robust and secure UAV navigation using GNSS, phased-array radiosystem and inertial sensor fusion", in *2nd IEEE Conference on Control Technology and Applications*, Copenhagen, Denmark, Aug. 2018, pp. 1338–1345. DOI: 10.1109/CCTA.2018.8511354.

[7] J. D. Anderson, *Fundamentals of Aerodynamics*, Second. McGraw-Hill, Inc., 1991.

[8] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control", *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019. DOI: 10.1007/s12532-018-0139-4.

[9] ArduPilot community, *Arduplane L1 guidance*, https://github.com/ArduPilot/ardupilot/pull/101, Accessed: 03.12.2019, 2013.

[10] ——, *Arduplane Rascal110 model*, https://github.com/ArduPilot/ardupilot/tree/6bf40b74b14d90af163bc7a4187b8387f734e457/Tools/autotest/aircraft/Rascal, Accessed: 03.12.2019, 2018.

[11] ArduPilot Development Team, *Ardupilot*, `http://ardupilot.org`, 2009–2018.

[12] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithm*. YBS Publishing, 2011.

[13] G. K. Batchelor, *An introduction to fluid dynamics*. Cambridge: Cambridge University Press, 1967.

[14] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft*. Princeton University Press, 2012.

[15] S. Benders, A. Wenz, and T. A. Johansen, "Adaptive path planning for unmanned aircraft using in-flight wind velocity estimation", in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2018, pp. 483–492.

[16] J. Berndt, "JSBSim: An open source flight dynamics model in C++", in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2004, p. 4923. DOI: 10.2514/6.2004-4923.

[17] J. J. Bertin and R. M. Cummings, *Aerodynamics for Engineers*, Sixth. Pearson Education Limited, 2014.

[18] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon", *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[19] J. Blazek, *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.

[20] K. T. Borup, T. I. Fossen, and T. A. Johansen, "A nonlinear model-based wind velocity observer for unmanned aerial vehicles", *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 276–283, 2016.

[21] C. E. Brennen, *Values of the added mass*, Accessed: 20.03.2020, 2006.

[22] A. H. Brodtkorb, U. D. Nielsen, and A. J. Sørensen, "Sea state estimation using vessel response in dynamic positioning", *Applied Ocean Research*, vol. 70, pp. 76–86, 2018.

[23] T. H. Bryne, K. Gryte, S. M. Albrektsen, and T. A. Johansen, "GNSS-free navigation of unmanned aerial vehicles based on phased-array radio systems", *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 1–15, 2020, Submitted.

[24] C4ADS, "Above us only stars – exposing GPS spoofing in Russia and Syria", C4ADS, Tech. Rep., 2019.

[25] P.-J. Chung, M. Viberg, and J. Yu, "Chapter 14 - doa estimation methods and algorithms", in *Academic Press Library in Signal Processing: Volume 3*, ser. Academic Press Library in Signal Processing, A. M. Zoubir, M. Viberg, R. Chellappa, and S. Theodoridis, Eds., vol. 3, Elsevier, 2014, pp. 599–650. DOI: 10.1016/B978-0-12-411597-2.00014-X.

[26] 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 *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–10. DOI: 10.1109/AERO.2018.8396534.

[27] E. M. Coates, A. Wenz, K. Gryte, and T. A. Johansen, "Propulsion system modeling for small fixed-wing UAVs", in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 748–757. DOI: 10.1109/ICUAS.2019.8798082.

[28] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization", *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 467–475, Feb. 2012. DOI: 10.1109/JSAC.2012.120227.

[29] M. V. Cook, *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability*. Butterworth-Heinemann, 1997.

[30] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods", *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[31] W. Crowther and K. Prassas, "Post stall landing for field retrieval of UAVs", in *14th Bristol International unmanned air vehicle systems conference*, 1999.

[32] K. Cunningham, D. Cox, D. Murri, and S. Riddick, "A piloted evaluation of damage accommodating flight control using a remotely piloted vehicle", in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6451.

[33] R. Curry, M. Lizarraga, B. Mairs, and G. H. Elkaim, "L2, an improved line of sight guidance law for UAVs", in *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 1–6.

[34] O. D. Dantsker, M. Vahora, S. Imtiaz, and M. Caccamo, "High fidelity moment of inertia testing of unmanned aircraft", in *2018 Applied Aerodynamics Conference*, 2018, p. 4219.

[35] DARcorporation, *Advanced aircraft analysis*, http://www.darcorp.com/Software/AAA/, Accessed 10.02.2018.

[36] A. Deperrois, *About XFLR5 calculations and experimental measurements*, Accessed 15.09.2014.

[37] ——, *XFLR5 analysis of foils and wings operating at low reynolds numbers*, http://sourceforge.net/projects/xflr5/files/Guidelines.pdf/download, 2013.

[38] P. S. Dias, S. L. Fraga, R. M. Gomes, G. M. Goncalves, F. L. Pereira, J. Pinto, and J. B. Sousa, "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.

[39] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Nonlinear Model Predictive Control - Towards New Challenging Applications", in *Nonlinear Model Predictive Control - Towards New Challenging Applications*, L. Magni, D. M. Raimondo, and F. A. wer, Eds., Springer Berlin Heidelberg, 2009, pp. 391–417. DOI: 10.1007/978-3-642-01094-1_32.

[40] J. Djugash, B. Hamner, and S. Roth, "Navigating with ranging radios: Five data sets with ground truth", *Journal of Field Robotics*, vol. 26, no. 9, pp. 689–695, 2009. DOI: 10.1002/rob.20311.

[41] M. Drela and H. Youngren, *XFOIL manual*, http://web.mit.edu/aeroutil_v1.0/xfoil_doc.txt, Accessed: 11.12.2014, 2001.

[42] 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.

[43] J. Engel, T. Schöps, and D. Cremers, "LSD–SLAM: Large-scale direct monocular SLAM", in *13th European Conference on Computer Vision ECCV*, Springer, Zurich, Switzerland, Sep. 2014, pp. 834–849.

[44] M. Faria, J. Pinto, F. Py, J. Fortuna, H. Dias, R. Martins, F. Leira, T. A. Johansen, J. Sousa, and K. Rajan, "Coordinating UAVs and AUVs for oceanographic field experiments: Challenges and lessons learned", in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 6606–6611.

[45] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill, 2008.

[46] A. S. Ferreira, J. Pinto, P. Dias, and J. B. de Sousa, "The LSTS software toolchain for persistent maritime operations applied through vehicular ad-hoc networks", in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, IEEE, 2017, pp. 609–616.

[47] Flight Level Engineering, *Surfaces aircraft design CFD software, web page*, http://www.flightlevelengineering.com/surfaces, Accessed 21.02.17.

[48] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry", in *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, Hong Kong, China, May 2014, pp. 15–22.

[49] J. Fortuna, F. Ferreira, R. Gomes, S. Ferreira, and J. Sousa, "Using low cost open source UAVs for marine wild life monitoring-field report", *IFAC Proceedings Volumes*, vol. 46, no. 30, pp. 291–295, 2013.

[50] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[51] 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.

[52] K. Gade, "The seven ways to find heading", *The Journal of Navigation*, vol. 69, no. 5, pp. 955–970, 2016.

[53] ——, "Inertial navigation – theory and applications", PhD thesis, Norwegian University of Science and Technology (NTNU), Jan. 2018. DOI: 11250/2491714.

[54] F. Gavilan, M. R. Arahal, and C. Ierardi, "Image deblurring in roll angle estimation for vision enhanced AAV control", *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 31–36, 2015.

[55] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "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.

[56] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, "Attitude estimation using biased gyro and vector measurements with time-varying reference vectors", *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1332–1338, 2012.

[57] ——, "Nonlinear observer for GNSS-aided inertial navigation with quaternion-based attitude estimation", in *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 272–279.

[58] J. N. Gross, Y. Gu, and B. Dewberry, "Tightly-coupled GPS/UWB-ranging for relative navigation during formation flight", in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, 2014.

[59] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.

[60] K. Gryte, T. H. Bryne, S. M. Albrektsen, and T. A. Johansen, "Field test results of GNSS-denied inertial navigation aided by phased-array radio systems for UAVs", in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2019, pp. 1398–1406. DOI: 10.1109/ICUAS.2019.8798057.

[61] K. Gryte, T. H. Bryne, and T. A. Johansen, "Unmanned aircraft flight control aided by phased-array radio navigation", *Journal of Field Robotics*, 2020, Submitted.

[62] 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 *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2018, pp. 826–835.

[63] 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.

[64] K. Gryte, T. A. Johansen, and T. I. Fossen, "Coordinated-turn based path following for fixed-wing unmanned aircraft", *Journal of Guidance, Control, and Dynamics*, 2020, Submitted.

[65] K. Gryte, M. L. Sollie, T. A. Johansen, and T. I. Fossen, "Control system architecture for automatic recovery of fixed-wing unmanned aerial vehicles in a moving arrest system", 2020, In progress.

[66] Y. Gui, P. Guo, H. Zhang, Z. Lei, X. Zhou, J. Du, and Q. Yu, "Airborne vision-based navigation method for UAV accuracy landing using infrared lamps", *Journal of Intelligent & Robotic Systems*, vol. 72, no. 2, pp. 197–218, Nov. 1, 2013. DOI: 10.1007/s10846-013-9819-5.

[67] F. Gustafsson, "Particle filter theory and practice with positioning applications", *IEEE Aerospace and Electronic Systems Magazine*, pp. 53–82, 2010.

[68] ——, *Statistical sensor fusion*, 2nd ed. Studentlitteratur, 2012.

[69] M. Haardt, M. Pesavento, F. Roemer, and M. N. E. Korso, "Chapter 15 - subspace methods and exploitation of special array structures", in *Academic Press Library in Signal Processing: Volume 3*, ser. Academic Press Library in Signal Processing, A. M. Zoubir, M. Viberg, R. Chellappa, and S. Theodoridis, Eds., vol. 3, Elsevier, 2014, pp. 651–717. DOI: 10.1016/B978-0-12-411597-2.00015-1.

[70] R. Hann, A. Wenz, K. Gryte, and T. A. Johansen, "Impact of atmospheric icing on uav aerodynamic performance", in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, Oct. 2017, pp. 66–71. DOI: 10.1109/RED-UAS.2017.8101645.

[71] J. M. Hansen, T. A. Johansen, and T. I. Fossen, "Tightly coupled integrated inertial and real-time-kinematic positioning approach using nonlinear observer", in *American Control Conference, Boston*, 2016, pp. 1–8.

[72] J. M. Hansen, T. I. Fossen, and T. A. Johansen, "Nonlinear observer design for gnss-aided inertial navigation systems with time-delayed gnss measurements", *Control Engineering Practice*, vol. 60, pp. 39–50, 2017. DOI: 10.1016/j.conengprac.2016.11.016.

[73] O. Harkegard and S. T. Glad, "A backstepping design for flight path angle control", in *Proceedings of the 39th IEEE Conference on Decision and Control, 2000.*, IEEE, vol. 4, 2000, pp. 3570–3575.

[74] 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)*, 2001, pp. 1146–1150.

[75] Hemisphere. (2019). Vector VR500 Smart Antenna. Accessed: 14.03.2020, [Online]. Available: https://www.hemispheregnss.com/wp-content/uploads/2019/02/hemispheregnss_vr500_userguide_875-0375-0_a4.pdf (visited on 03/14/2020).

[76] S. F. Hoerner, *Fluid-dynamic drag: practical information on aerodynamic drag and hydrodynamic resistance*. Hoerner Fluid Dynamics Midland Park, NJ, 1965.

[77] V. E. Hovstein, A. Sægrov, and T. A. Johansen, "Experiences with coastal and maritime UAS BLOS operation with phased-array antenna digital payload data link", in *Proc. International Conference Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, May 27, 2014, pp. 262–266. DOI: 10.1109/ICUAS.2014.6842264.

[78] M.-D. Hua, "Attitude estimation for accelerated vehicles using GPS/INS measurements", *Control Engineering Practice*, vol. 18, pp. 723–732, 2010.

[79] 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.

[80] G. Jaffe and T. Erdbrink, "Iran says it downed U.S. stealth drone; Pentagon acknowledges aircraft downing", *The Washington post*, Dec. 9, 2011.

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

[82] R. Jategaonkar, *Flight vehicle system identification: a time domain methodology*. Reston, VA, USA: AIAA, 2006, vol. 216.

[83] N. M. Jodeh, "Development of autonomous unmanned aerial vehicle research platform: Modeling, simulation, and flight testing", PhD thesis, Air force institute of technology, 2006.

[84] T. A. Johansen and T. I. Fossen, "Nonlinear observer for tightly coupled integration of pseudorange and inertial measurements", *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2199–2206, Nov. 2016. DOI: 10.1109/tcst.2016.2524564.

[85] T. A. Johansen, J. M. Hansen, and T. I. Fossen, "Nonlinear observer for tightly integrated inertial navigation aided by pseudo-range measurements", *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 1, Paper DS-15–1088, Oct. 2017, Paper DS-15-1088. DOI: 10.1115/1.4034496.

[86] T. A. Johansen and T. I. Fossen, "Nonlinear observer for inertial navigation aided by pseudo-range and range-rate measurements", in *Control Conference (ECC), 2015 European*, IEEE, 2015, pp. 1673–1680.

[87] 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, Jun. 2015, pp. 510–519. DOI: 10.1109/ICUAS.2015.7152330.

[88] E. D. Kaplan and C. J. Hegarty, *Understanding GPS/GNSS - Principles and Applications*. Artech House, 2017.

[89] P. Karasz, "Europe billed Galileo as its answer to GPS. It's been mostly down for days.", *The New York Times*, Jul. 16, 2019.

[90] J. Katz and A. Plotkin, *Low-speed aerodynamics*. Cambridge university press, 2001, vol. 13.

[91] A. C. Kermode, *Mechanics of Flight*. Pearson Education Limited, 2006.

[92] 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.

[93] H. K. Khalil, *Nonlinear systems*, ser. Pearson Education. Prentice Hall, 2002.

[94] 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.

[95] S. Khattak, C. Papachristos, and K. Alexis, "Keyframe-based thermal–inertial odometry", *Journal of Field Robotics*, 2019. DOI: 10.1002/rob.21932.

[96] H. J. Kim, M. Kim, H. Lim, C. Park, S. Yoon, D. Lee, H. Choi, G. Oh, J. Park, and Y. Kim, "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.

[97] 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.

[98] V. Klein and E. A. Morelli, *Aircraft system identification: theory and practice*. American Institute of Aeronautics and Astronautics Reston, Va, USA, 2006.

[99] 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.

[100] 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.

[101] J. Krøgenes, L. Brandrud, R. Hann, J. Bartl, T. Bracchi, and L. Sætran, "Aerodynamic performance of the NREL S826 airfoil in icing conditions", *Wind Energy Science Discussions*, vol. 2017, pp. 1–17, 2017. DOI: 10.5194/wes-2017-39.

[102] D. Kufoalor and T. Johansen, "Reconfigurable fault tolerant flight control based on nonlinear model predictive control", in *American Control Conference (ACC), 2013*, Jun. 2013, pp. 5128–5133. DOI: 10.1109/ACC.2013.6580635.

[103] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 3131–3137. DOI: 10.1109/IROS.2015.7353810.

[104] K. Lehmkühler, K. Wong, and D. Verstraete, "Methods for accurate measurements of small fixed wing UAV inertial properties", *The Aeronautical Journal*, vol. 120, no. 1233, pp. 1785–1811, 2016.

[105] F. S. Leira, T. A. Johansen, and T. I. Fossen, "A UAV ice tracking framework for autonomous sea ice management", in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 581–590.

[106] A. M. Lekkas and T. I. Fossen, "Integral LOS path following for curved paths based on a monotone cubic hermite spline parametrization", *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2287–2301, 2014.

[107] ——, "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.

[108] A. Loria, T. I. Fossen, and E. Panteley, "A separation principle for dynamic positioning of ships: Theoretical and experimental results", *IEEE Transactions on Control Systems Technology*, vol. 8, no. 2, pp. 332–343, 2000.

[109] A. Loría and E. Panteley, "Cascaded nonlinear time-varying systems: Analysis and design", in *Advanced topics in control systems theory*, Springer, 2005, pp. 23–64.

[110] 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.

[111] G. MacGougan, K. O'Keefe, and R. Klukas, "Tightly-coupled GPS/UWB integration", *Journal of Navigation*, vol. 63, no. 01, pp. 1–22, 2010.

[112] 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.

[113] R. Mahony, T. Hamel, J. Trumpf, and C. Lageman, "Nonlinear attitude observer on SO(3) for complementary and compatible measurements: A theoretical study", *IEEE Conference on Decision and Control*, pp. 6407–6412, 2009.

[114] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group", *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[115] F. L. Markley, "Attitude error representation for kalman filtering", *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 311–317, Mar. 2003. DOI: 10.2514/2.5048.

[116] 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.

[117] 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.

[118] T. Maruyama, K. Kihira, and H. Miyashita, "Phased arrays", in *Handbook of Antenna Technologies*, Z. N. Chen, D. Liu, H. Nakano, X. Qing, and T. Zwick, Eds. Singapore: Springer Singapore, 2016, pp. 1113–1162. DOI: 10.1007/978-981-4560-44-3_37.

[119] W. H. Mason, *Software for aerodynamics and aircraft design*, Accessed 21.02.2017.

[120] S. G. Mathisen, K. Gryte, S. Gros, and T. A. Johansen, "Precision deep stall landing of fixed-wing UAVs using nonlinear model predictive control", *Journal of intelligent and robotic systems*, 2020, Submitted.

[121] S. G. Mathisen, F. S. Leira, H. H. Helgesen, K. Gryte, and T. A. Johansen, "Autonomous ballistic airdrop of objects from a small fixed-wing unmanned aerial vehicle", *Autonomous Robots*, Jan. 2020. DOI: 10.1007/s10514-020-09902-3.

[122] S. H. Mathisen, T. I. Fossen, and T. A. Johansen, "Non-linear model predictive control for guidance of a fixed-wing UAV in precision deep stall landing", in *International Conference on Unmanned Aircraft Systems*, IEEE, Denver, 2015, pp. 356–365.

[123] 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 SciTech*, 2016.

[124] MathWorks, "Dryden wind turbulence model (continuous)", Following Military Specification MIL-F-8785C, Apr. 2015.

[125] A. D. Meadowcroft, S. Howroyd, K. Kendall, and M. Kendall, "Testing microtubular SOFCs in unmanned air vehicles (UAVs)", *ECS Transactions*, vol. 57, no. 1, pp. 451–457, 2013.

[126] 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*, Nov. 29, 2019. DOI: 10.1007/s12555-018-0489-8.

[127] M. P. Miller, "An accurate method of measuring the moments of inertia of airplanes", 1930.

[128] S. J. Mills, J. J. Ford, and L. Mejías, "Vision based control for fixed wing UAVs inspecting locally linear infrastructure using skid-to-turn maneuvers", *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 29–42, 2011.

[129] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico, K. Karydis, N. Atanasov, G. Loianno, D. Scaramuzza, K. Daniilidis, C. J. Taylor, and V. Kumar, "Fast, autonomous flight in GPS-denied and cluttered environments", *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2017. DOI: 10.1002/rob.21774.

[130] 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.

[131] J. L. Moore, R. Cory, and R. Tedrake, "Robust post-stall perching with a simple fixed-wing glider using LQR-trees.", *Bioinspiration & biomimetics*, vol. 9, no. 2, p. 025 013, Jun. 2014. DOI: 10.1088/1748-3182/9/2/025013.

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

[133] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB–SLAM: A versatile and accurate monocular SLAM system", *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[134] T. Muskardin, G. Balmer, L. Persson, S. Wlach, M. Laiacker, A. Ollero, and K. Kondak, "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, Dec. 1, 2017. DOI: 10.1007/s10846-017-0475-z.

[135] C. C. Naira Hovakimyan, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation (Advances in Design and Control)*. Society for Industrial & Applied Mathematics, 2010.

[136] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for small unmanned air vehicles", in *2006 American Control Conference*, IEEE, 2006, 7–pp.

[137] ——, "Vector field path following for miniature air vehicles", *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[138] L. T. Nguyen, M. E. Ogburn, W. P. Gilbert, K. S. Kibler, P. W. Brown, and P. L. Deal, "Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability.[f-16]", 1979.

[139] Norwegian Seafood council, "Norwegian seafood exports total nok 99 billion in 2018", *Press release*, Jan. 7, 2019.

[140] F. Olsson, J. Rantakokko, and J. Nygards, "Cooperative localization using a foot-mounted inertial navigation system and ultrawideband ranging", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2014, pp. 122–131.

[141] C. Ostowari and D. Naik, "Post-stall wind tunnel data for NACA 44XX series airfoil sections", Solar Energy Research Institute, Golden, Colorado, Tech. Rep. 4807, 1985.

[142] E. Panteley and A. Loria, "On global uniform asymptotic stability of nonlinear time-varying systems in cascade", *Systems & Control Letters*, vol. 33, no. 2, pp. 131–138, 1998.

[143] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking", in *AIAA guidance, navigation, and control conference and exhibit*, 2004, p. 4900.

[144] ——, "Performance and lyapunov stability of a nonlinear path following guidance method", *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.

[145] G. V. Pelizer, N. B. F. da Silva, and K. R. L. J. Branco, "Comparison of 3D path-following algorithms for unmanned aerial vehicles", in *international conference on unmanned aircraft systems (ICUAS),*, IEEE, 2017, pp. 498–505.

[146] M. G. Petovello, K. O'Keefe, B. Chan, S. Spiller, C. Pedrosa, P. Xie, and C. Basnayake, "Demonstration of inter-vehicle UWB ranging to augment DGPS for improved relative positioning", *Journal of Global Positioning Systems*, vol. 11, no. 1, pp. 11–21, 2012.

[147] M. Petovello, S. Feng, and W. Ochieng, "How do you trust centimeter level accuracy positioning?", *Inside GNSS Magazine*, Sep. 2014.

[148] W. F. Phillips, *Mechanics of flight*, ser. Engineering case studies online. Wiley, 2004.

[149] 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.

[150] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, "The LSTS toolchain for networked vehicle systems", in *OCEANS-Bergen, 2013 MTS/IEEE*, IEEE, 2013, pp. 1–9.

[151] G. Platanitis and S. Shkarayev, "Integration of an autopilot for a micro air vehicle", in *Infotech@Aerospace*, American Institute of Aeronautics and Astronautics (AIAA), Sep. 2005, p. 7066. DOI: 10.2514/6.2005-7066.

[152] W. Pointner, G. Kotsis, P. Langthaler, and M. Naderhirn, "Using formal methods to verify safe deep stall landing of a MAV", in *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, IEEE, 2011, pp. 1–10. DOI: 10.1109/DASC.2011.6096086.

[153] 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.

[154] 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)*, IEEE, 2018, pp. 306–313.

[155] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, ser. Airplane Flight Dynamics and Automatic Flight Controls. Roskam Aviation and Engineering Corporation, 1998.

[156] D. Rotondo, A. Cristofaro, K. Gryte, and T. A. Johansen, "LPV model reference control for fixed-wing UAVs", *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 559–11 564, 2017, 20th IFAC World Congress. DOI: 10.1016/j.ifacol.2017.08.1640.

[157] 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, Jul. 1989. DOI: 10.1109/29.32276.

[158] RUAG. (2020). Object position and tracking system (OPATS). Accessed: 14.03.2020, [Online]. Available: https://ruag.picturepark.com/Go/eemLIBDs/V/7623/1 (visited on 03/14/2020).

[159] R. S. Russel, *Non-linear F-16 simulation using simuling and Matlab*, Version 1.0, Jun. 2003.

[160] J. C. Ryan, A. L. Hubbard, J. E. Box, J. Todd, P. Christoffersen, J. R. Carr, T. O. Holt, and N. A. Snooke, "UAV photogrammetry and structure from motion to assess calving dynamics at store glacier, a large outlet draining the greenland ice sheet", 2015.

[161] R. Rysdyk, "Course and heading changes in significant wind", *Journal of guidance, control, and dynamics*, vol. 30, no. 4, pp. 1168–1171, 2007, Erratum-ibid. Rysdyk [162], Corrected in "Erratum on course and heading changes in significant wind", *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1311–1312, 2010.

[162] R. Rysdyk, "Erratum on course and heading changes in significant wind", *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1311–1312, 2010.

[163] R. Rysdyk, C. Lum, and J. Vagners, "Autonomous orbit coordination for two unmanned aerial vehicles", in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, p. 6362.

[164] 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, Apr. 1, 2014. DOI: 10.1007/s10846-013-9926-3.

[165] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, "A survey and analysis of the GNSS spoofing threat and countermeasures", *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1–31, May 2016. DOI: 10.1145/2897166.

[166] R. Schmidt, "Multiple emitter location and signal parameter estimation", *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986. DOI: 10.1109/TAP.1986.1143830.

[167] A. Shakoori, A. Betin, and D. Betin, "Comparison of three methods to determine the inertial properties of free-flying dynamically similar models", *J. Engineering Science and Technology*, 2016.

[168] Sierra Nevada Corporation. (2010). Dual-thread automatic takeoff and landing system (DT-ATLS). Accessed: 14.03.2020, (visited on 03/14/2020).

[169] ——, (2013). Unmanned aerial vehicle common automatic recovery system (UCARS) - version 2. Accessed: 14.03.2020, [Online]. Available: https://www.sncorp.com/media/1998/ucars-v2product-sheet.pdf (visited on 03/14/2020).

[170] A. G. Sim, "Flight Characteristics of a Manned, Low-Speed, Controlled Deep Stall Vehicle", *NASA Technical Memorandum*, 1984.

[171] G. M. Siouris, *Missile guidance and control systems*. Springer Science & Business Media, 2004.

[172] 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, 2015. DOI: 10.1109/MAES.2015.7119821.

[173] J. Smith, J. Su, C. Liu, and W.-H. Chen, "Disturbance observer based control with anti-windup applied to a small fixed wing UAV for disturbance rejection", *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2-4, pp. 329–346, 2017.

[174] S. C. Smith, "A computational and experimental study of nonlinear aspects of induced drag", 1996.

[175] J. Solà, *Quaternion kinematics for the error-state KF*, https://arxiv.org/abs/1711.02508v1, 2017.

[176] 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)*, IEEE, Jun. 2019, pp. 1425–1435. DOI: 10.1109/ICUAS.2019.8797746.

[177] K. L. Sørensen, "Autonomous icing protection solution for small unmanned aircraft: An icing detection, anti-icing and de-icing solution", PhD thesis, Norwegian University of Science and Technology (NTNU), 2016. DOI: 11250/2417471.

[178] M. Stachura and E. Frew, "Communication-aware information-gathering experiments with an unmanned aircraft system", *Journal of Field Robotics*, vol. 34, no. 4, pp. 736–756, 2017.

[179] R. F. Stengel, *Flight Dynamics*. Princeton University Press, 2004.

[180] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation*, 2nd. Hoboken, New Jersey: John Wiley & Sons, 2003.

[181] 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.

[182] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB", in *International symposium on GPS/GNSS*, International Convention Center Jeju Korea, 2009, pp. 4–6.

[183] H. Taniguchi, "Analysis of deepstall landing for UAV", in *International congress of the aeronautical sciences*, vol. 3, Anchorage, Alaska, 2008, pp. 1–6.

[184] The Boeing Company. (2020). Scaneagle product page. Accessed: 2020-02-30, [Online]. Available: https://www.boeing.com/defense/autonomous-systems/scaneagle/index.page.

[185] 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.

[186] J. Tiemann, F. Schweikowski, and C. Wietfeld, "Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct. 2015, pp. 1–7. DOI: 10.1109/IPIN.2015.7346960.

[187] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology*, 2nd. Institution of Electrical Engineers, American Institute of Aeronautics, and Astronautics, 2004.

[188] H. Tømmervik, S.-R. Karlsen, L. Nilsen, B. Johansen, R. Storvold, A. Zmarz, P. S. Beck, K. A. Høgda, S. Goetz, T. Park, *et al.*, "Use of unmanned aircraft systems (UAS) in a multi-scale vegetation index study of arctic plant communities in adventdalen on svalbard", 2014.

[189] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*. Wiley, 2011.

[190] UBIQ Aerospace. (2020). D·ICE UAV de-icing solution, [Online]. Available: https://www.ubiqaerospace.com (visited on 10/02/2020).

[191] A. Walker, "Hard real-time motion planning for autonomous vehicles", PhD thesis, Swinburne University, 2011.

[192] ——, (2013). Dubins-Curves GitHub repository. Accessed: 14.03.2020, [Online]. Available: https://github.com/AndrewWalker/Dubins-Curves (visited on 03/14/2020).

[193] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments", in *IEEE International Conference on Robotics and Automation*, IEEE, May 2012, pp. 957–964. DOI: 10.1109/ICRA.2012.6225147.

[194] A. Wenz and T. A. Johansen, "Moving horizon estimation of air data parameters for UAVs", *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2019. DOI: 10.1109/TAES.2019.2946677.

[195] M. D. White and G. E. Cooper, "Simulator studies of the deep stall", in *NASA Conference on Aircraft Operating Problems*, 1965.

[196] A. Winter, R. Hann, A. Wenz, K. Gryte, and T. A. Johansen, "Stability of a flying wing uav in icing conditions", *8th European conference for aeronautics and aerospace sciences (EUCASS)*, pp. 1–15, Jun. 2019. DOI: 10.13009/EUCASS2019-906.

[197] 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.

[198] R. Yanushevsky, *Guidance of unmanned aerial vehicles*. CRC press, 2011.

[199] 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.

[200] 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.

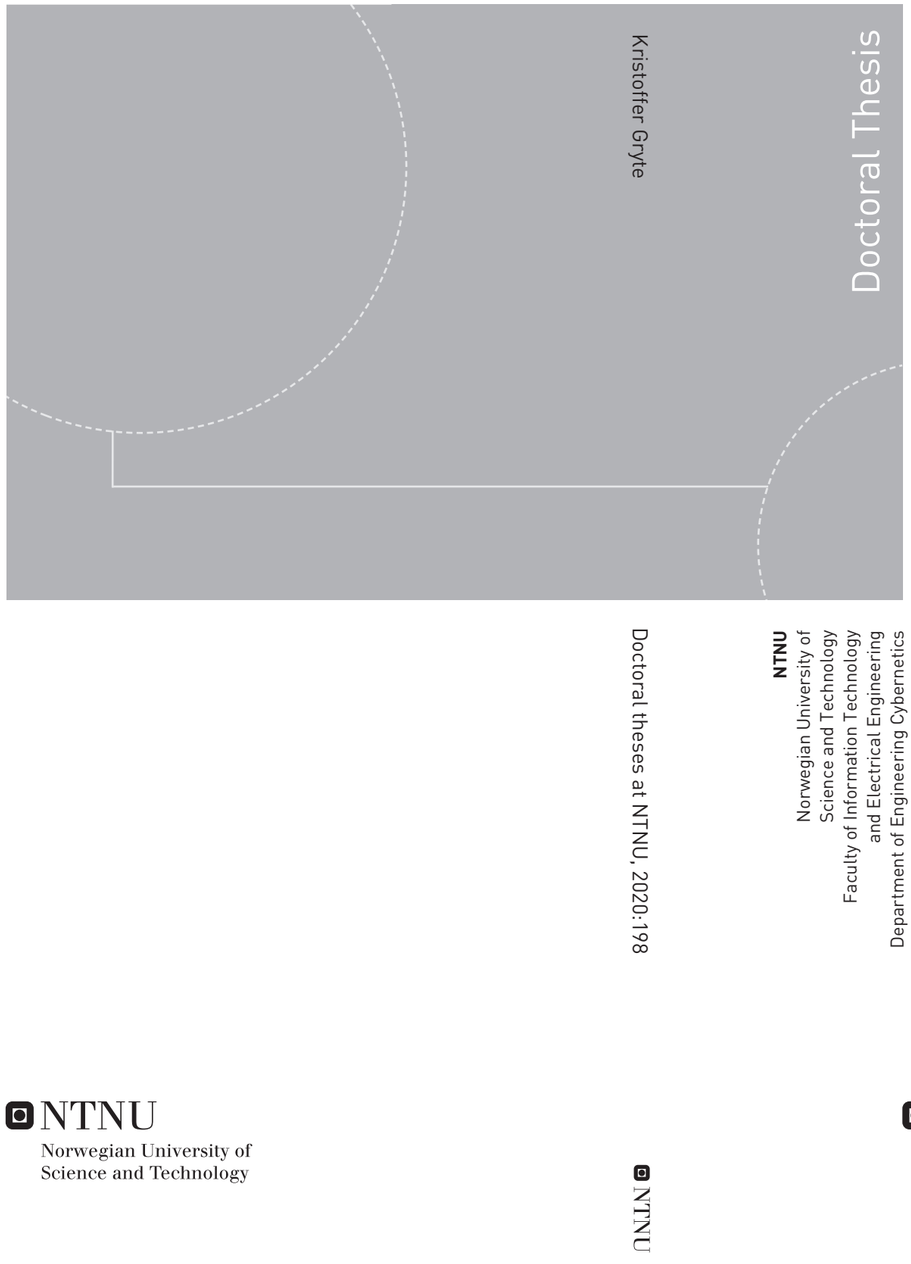
[201] D. I. You, Y. D. Jung, S. W. Cho, H. M. Shin, S. H. Lee, and D. H. Shim, "A guidance and control law design for precision automatic take-off and landing of fixed-wing UAVs", in *AIAA Guidance, Navigation, and Control Conference*, 2012, p. 4674.

[202] A. B. Younes, D. Mortari, J. D. Turner, and J. L. Junkins, "Attitude error kinematics", *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, 2014.

[203] P. Zarchan, *Tactical and strategic missile guidance*. American Institute of Aeronautics and Astronautics, Inc., 2012.

[204] A. Zolich, T. A. Johansen, K. Cisek, and K. Klausen, "Unmanned aerial system architecture for maritime missions. design & hardware description", in *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, IEEE, 2015, pp. 342–350.

ISBN 978-82-326-4744-6 (printed version)
ISBN 978-82-326-4745-3 (electronic version)
ISSN 1503-8181



Doctoral theses at NTNU, 2020:198

Kristoffer Gryte

Precision control of fixed-wing UAV and robust navigation in GNSS-denied environments

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

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

Doctoral theses at NTNU, 2020:198

NTNU