ISBN 978-82-326-3384-5 (printed version)
ISBN 978-82-326-3385-2 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Kasper Trolle Borup

Doctoral theses at NTNU, 2018:297

NTNU

Doctoral Thesis

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

Doctoral theses at NTNU, 2018:297

Kasper Trolle Borup

Air Data Estimation for Small Unmanned Aircraft

NTNU
Norwegian University of
Science and Technology

Kasper Trolle Borup

Air Data Estimation for Small Unmanned Aircraft

Thesis for the degree of Philosophiae Doctor

Trondheim, 2018:297

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 978-82-326-3384-5 (printed version)
ISBN 978-82-326-3385-2 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2018:297

MILJØMERKET 241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

To Ragnhild

Summary

The use of small fixed-wing unmanned aerial vehicles has shown an explosive growth in recent years and this thesis is motivated by the need for low-cost, accurate air data estimators for these aircraft. The air data contains information that is directly related to the performance of the unmanned aerial vehicle and is therefore valuable for operational control. However, the available market systems are often large, heavy, and expensive and alternatives without these limitations could potentially allow for new unmanned aircraft possibilities and applications.

This thesis briefly accounts for the influence of air data on fixed-wing unmanned aerial vehicle flight and continues to present three main contributions in air data estimation. The main results are:

A nonlinear model-based wind velocity observer for unmanned aerial vehicles: The observer exploits an aerodynamic model of the aircraft and an airspeed sensor together with a standard sensor suite consisting of a GNSS receiver, an inertial measurement unit, a Pitot-static probe, and a heading reference. The observer is shown to provide exponential stability and convergent estimates of both wind velocity and relative velocity from which estimates of the air data can be computed. The observer is verified through simulation using a realistic wind signal.

A machine learning approach for estimating air data for small fixed-wing unmanned aerial vehicles using distributed pressure sensors: The air data estimation method consists of combining machine learning algorithms with an array of low-cost pressure sensors embedded in the surface of the unmanned aircraft. Two machine learning algorithms based on artificial neural networks and linear regression are implemented, tested, and assessed using data collected from wind tunnel experiments and a flight test and the results are compared to a benchmark flight test. Training the machine learning algorithms using wind tunnel data was found to introduce several potential error sources that need to be addressed in order to provide accurate estimation on the benchmark flight test, whereas training the algorithms using flight data provides lower estimation RMSE values. The performance of the neural network structures has been found to slightly outperform the linear regression algorithms in estimation accuracy. Lastly, results from using different sensor configurations and a pseudo Reynolds number are presented in an effort to evaluate the influence of sensor number and placement on the accuracy of the method.

Kalman filters for air data bias correction for a fixed-wing unmanned aerial vehicle: Two distinct Kalman filter approaches for correcting air data systems with low-frequency drift or a constant error bias for fixed-wing unmanned aerial vehicles are presented. The estimators use a GNSS receiver, an IMU, and a heading reference, combined with an air data system that is assumed to provide measurements with an unknown additive slowly time-varying bias. Neither estimator is dependent on the aircraft model. The estimators are, without the effect of noise, proven to have globally exponentially stable equilibrium points of the error dynamics if provided with persistence-of-excitation in the angular velocity and angular acceleration of the unmanned aircraft. The estimators are verified through simulation and using experimental flight data. The results from the experimental flight data are obtained using the machine learning approach described in the preceding paragraph. The flight results indicate that a certain amount of excitation is needed in order to have converging bias estimates during turbulent wind conditions.

Contents

List of figures

List of tables

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU), Trondheim.
The work presented has been conducted at the Autonomous Marine Operations and Systems (AMOS) centre of excellence at the Department of Engineering Cybernetics at NTNU. My project supervisor has been Professor Thor Inge Fossen and co-supervisor Professor Tor Arne Johansen, both from the Department of Engineering Cybernetics. The work was supported by the Norwegian Research Council (grant no. 221666 and 223254) at AMOS and FORNY (project no. 90189003) at NTNU.

Acknowledgements

Although my name stands alone on the front page, this thesis represents anything but a single persons work. I have been blessed with Professor Thor Inge Fossen and Professor Tor Arne Johansen as my supervisors. Thor's enthusiasm for theory and research has been highly contagious and he is a master of both. Tor Arne is ridiculously hard-working, frustratingly never ever wrong, and he has spent a great deal of time combing through my writings for weird errors. He also deserves an extra thanks for taking a chance in starting a company with Kim and me. A great thanks to Lars and Pål, the department pilots, for providing me with experimental flights. And many thanks to Sigurd for providing the means to accurately log data during these flights. Thanks to Ol'ga Hubová and Boris Rohal'-Ilkiv at the Slovak University of Technology in Bratislava (STU) for their help, hospitality, and wind tunnel time. Thanks to the skilled and helpful Glenn and Terje from the workshop in the basement. And thanks to Roberto Galeazzi for inspiring me to pursue a PhD.

Thanks to Richard for being an awesome sport in helping me with aerodynamics. Thanks to Andreas Wenz for good topic discussions and for collaborating on collecting data. Thanks to Artur and Richard for being a part of the UBIQ adventure. A thanks to Bård for joining in on the last paper and for making me the second-most inappropriate guy in the room. Thanks to the administrative personnel at ITK - you have simply been awesome and extremely helpful, and the first part of my stay would have been a lot worse without the aid of Tove. Thanks to my friends and UBIQ co-founders at TTO, Kristin and Fredrik. A special thought also goes out to Torbjørn who is dearly missed.

Thanks to Mauro and Frederik for being great office mates and thanks to all of my other lovely colleagues at both AMOS and ITK. Thanks to Mikkel for being a great training partner, and for constantly referring to me as his great fitness inspiration, haha! Thanks to Signe for being an excellent house mate and for always laughing with me at stuff we probably shouldn't laugh at. Thanks to Jakob for being a good friend and for the many long conversations about how we would hopefully survive the PhD. Thanks to the friends at home in Denmark, among them Rejseholdet, Det Rustne Blink, and Chicken Dinners. Thanks to Mathias, Per, and Niels-Jakob for being my weird and wonderful friends for as long as I can remember. A thank you to Johan and Asger for visiting me everywhere I go. You are, in the undying words of Johan's favorite poet Tina Turner, simply the best. A special thanks to Kim, my partner in crime, for always having my back and for letting me piggy-back his PhD project into a hopefully soon-to-be ridiculously successful company.

I would also like to thank my family for always being there for me and for doing a great job at not forgetting me. That means the world to me.

Lastly, I would like to thank my Ragnhild for turning what could have been the worst year, into the best year. You are truly something special and I am very grateful to have you in my life.

Abbreviations and Nomenclature

Abbreviations

AOA	Angle Of Attack
BVLOS	Beyond Visual Line Of Sight
CFD	Computational Fluid Dynamics
ECEF	Earth-Centered-Earth-Fixed
ECI	Earth-Centered-Inertial
EKF	Extended Kalman Filter
FADS	Flush Air Data Sensing
GES	Global Exponential Stability
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
LR	Linear Regression
ML	Machine Learning
NED	North-East-Down
NN	Neural Network
PE	Persistence of Excitation
PVA	Position Velocity Attitude
SSA	SideSlip Angle
UAV	Unmanned Aerial Vehicle
UCC	Uniform Complete Controllable
UCO	Uniform Complete Observable

Greek Symbols

α	angle of attack
β	sideslip angle
μ_∞	freestream viscocity [Pa·s]
ω	angular velocity [rad/s^2]
ρ_∞	density of air [kg/m^3]

Roman Symbols

a_∞	freestream speed of sound [m/s]
C	aerodynamic coefficients
C_D	drag coefficient
C_L	lift coefficient
c	cord length [m]
F_D	drag force [N]
F_L	lift force [N]
F_R	resultant force [N]
F_T	thrust force [N]
F_W	weight force [N]
f	specific force [m/s^2]
g	gravity vector [m/s^2]
J	inertia tensor
M_∞	Mach number
m	mass [kg]
p	position [m]
v_r	relative velocity [m/s]
Re	Reynolds number
R_b^n	rotation matrix from BODY to NED frame
S	planform area of the wing [m^2]
V_a	airspeed [m/s]
v_g	velocity over ground [m/s]
v_w	wind velocity [m/s]

Superscripts

b	BODY coordinate frame
n	NED coordinate frame

Chapter 1

Introduction

1.1 Motivation

The recent years have shown an explosive growth in the use of small unmanned aerial vehicles (UAVs) [9, 18]. This is a direct result of the technological advances in fields such as batteries, MEMS[1]-based sensors, and computer processors, and these advances have driven down the weight, size, and price of the components needed to build small unmanned aircraft. This has in turn allowed for producing smaller UAVs with relatively low price tags, and the use of UAVs have opened up outside of high budget military use. The primary advantages of using unmanned aircraft is arguably the low cost compared to manned flight as well as the reduction in risk to personnel. The application areas of UAVs are many and diverse, and new innovative approaches are taken in solving previously impossible tasks or greatly reducing the cost of existing tasks. A few of the UAV applications include:

Disaster management: In case of large scale natural disasters such as earthquakes or forest fires, UAV technology enables a fast response when reaction time is critical. UAVs can survey an area and provide communication network relays for disaster survivors. Furthermore, UAVs are capable of accessing hard-to-reach areas and can fly into dangerous zones with thermal cameras to provide situational awareness without the concern for the human operators [26, 64].

Sea ice monitoring: One of the main challenges of oil explorations in the artic is the abundance of sea ice. If not mitigated, the sea ice can result in damaging loads on installations. By utilizing UAVs to map the surrounding areas, it is possible to help alleviate this threat by providing an overview. The collected data can then be used for qualified decisions on whether to break up the ice with a ship [28].

Search and rescue missions: By employing UAVs, it is possible to quickly provide an eye in the sky in locating a lost person using visual spectrum and thermal cameras, e.g. from a capsized ship or an avalanche [81, 85]. Multiple UAVs flying in formation allows for searching large areas in a short amount of time [79].

[1] Micro-electromechanical systems [10, 84].

An illustration of the search and rescue UAV application is depicted in Figure 1.1.

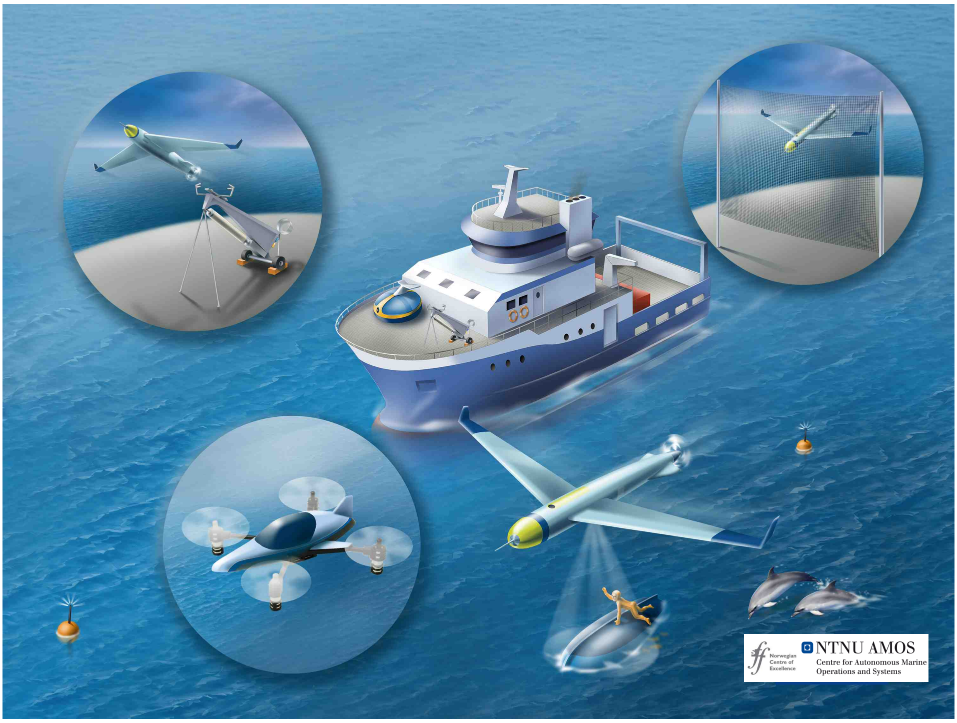

Norwegian Centre of Excellence
NTNU AMOS
Centre for Autonomous Marine Operations and Systems

Figure 1.1: Using UAVs to locate a lost person with both launch and retrieval of UAV on board ship. Copyright: Bjarne Stenberg/NTNU.

Other UAV applications include wireless connectivity coverage, precision airdrop of packages, inspection, surveillance, power generation, mapping, oil spill detection, etc. Unmanned aircraft are commonly divided into fixed-wing or rotory-wing and the preceding examples of applications make use of both types. However, for this thesis and the results presented, the focus is on small fixed-wing UAVs[2].

Knowledge of the wind is very important in fixed-wing UAV control and operation. The relative velocity of the UAV with respect to the wind contains information from which the angle of attack (AOA), sideslip angle (SSA), and airspeed are directly computable. The AOA, SSA, and airspeed variables are commonly referred to as the air data[3] and their values during flight is directly related to the perfor-

[2] There is no one standard of UAV size classification, but here *small* refers to UAVs with a wing span of less than three meters.

[3] Often the Mach number and Reynolds number are also included, but in this thesis we shall concern ourselves primarily with the airspeed, AOA, and SSA, since the range of airspeeds operated in by smaller aircraft makes the effects of the Mach number and Reynolds number approximately constant [6].

mance and safety conditions of the unmanned aircraft.

The light weight of the small UAVs make the vehicles susceptible to strong winds, and having accurate estimates of the air data, enables control of the aircraft that mitigates this threat. In addition, small unmanned aircraft operate in low airspeeds where the influence of the windspeed on the ground speed is not negligible. This is particularly essential for beyond visual line of sight (BVLOS) operations where the wind cannot be reliably assessed by the operating pilot. The effects of wind therefore has to be factored into the maximally obtainable range of the UAV as well as in the path planning [42].

Larger fixed-wing aircraft are often equipped with air data parameter sensors, such as angle of attack vanes and multi-hole probes[4], but for smaller UAVs there are often strict limitations on size, weight, power consumption, and price. It is therefore highly desirable to have an air data estimation system only utilizing measurements that are obtainable for a UAV through a combination of standard sensor suite measurements and/or extra sensors that have low cost and can easily be integrated in the UAV. This integration of sensors into the airframe of the UAV should have a minimal negative influence on the aerodynamic capabilities of the aircraft and should have a negligible weight and size to not reduce the payload possibilities. The focus of this thesis is therefore to provide and evaluate new methods to estimate the air data given this framework.

1.2 The Influence of Air Data on Flight

This section aims to explain the influence of air data on aircraft flight and why these parameters are important to measure and estimate. In order to understand the influence of air data on aircraft, it is necessary to quickly introduce a few concepts. In aerodynamic literature, the wing is often considered as fixed with air flowing past it, but here it shall be considered from the perspective of the relative velocity of a wing moving through the air. The AOA α is defined as the angle between the chord line of the aircraft airfoil and the relative velocity of the UAV with respect to the wind vector ${v_r}^b$, projected onto the plane spanned by the airfoil, as shown in Figure 1.2.

The SSA β is similarly defined as the angle between the relative velocity projected onto plane spanned by the wings, and the plane dividing the aircraft perpendicular to the wings as shown in Figure 1.3. The airspeed V_a is defined as the magnitude of the relative velocity vector.

From the geometric definitions, the relation from the relative velocity vector

[4] Sometimes in literature also referred to as Prandtl probes or air data booms.

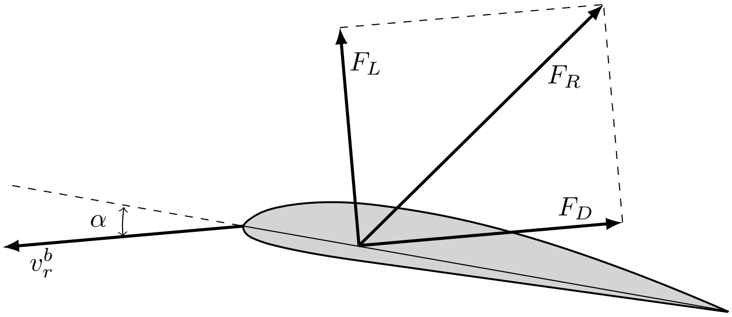


Figure 1.2: An airfoil with the angle of attack definition and the resultant aerodynamic force along with its corresponding components, the lift and drag force.

$v_r^b = [u_r, v_r, w_r]^\top$ to the airspeed, AOA, and SSA, can be found as [6]

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{1.1}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{1.2}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \tag{1.3}$$

The aerodynamic forces shown in Figure 1.2 are a result of the pressure and shear stress distribution over the aircraft body [5]. The resultant force F_R can be decomposed into the the lift and drag components. The lift force F_L is defined as the component of F_R perpendicular to the v_r^b and the drag force F_D is defined as the component of F_R parallel to v_r^b. For a given shape, at zero degree sideslip angle, and a given angle of attack in a steady airflow[5], the lift and drag forces are

$$F_L = \frac{1}{2}\rho_\infty S V_a^2 C_L(\mathrm{Re}, M_\infty, \alpha) \tag{1.4}$$

$$F_D = \frac{1}{2}\rho_\infty S V_a^2 C_D(\mathrm{Re}, M_\infty, \alpha) \tag{1.5}$$

where ρ_∞ is the freestream air density, Re is the freestream Reynolds number, M_∞ is the freestream Mach number, S is the planform area of the wing, and C_L and C_D are the dimensionless lift and drag coefficients [5]. The dimensionless parameters Reynolds number and Mach number are called similarity parameters and are defined as

$$\mathrm{Re} = \frac{\rho_\infty V_a c}{\mu_\infty}$$

$$M_\infty = \frac{V_a}{a_\infty}$$

[5] Such as experienced during a wind tunnel experiment or in an atmosphere with low turbulence.

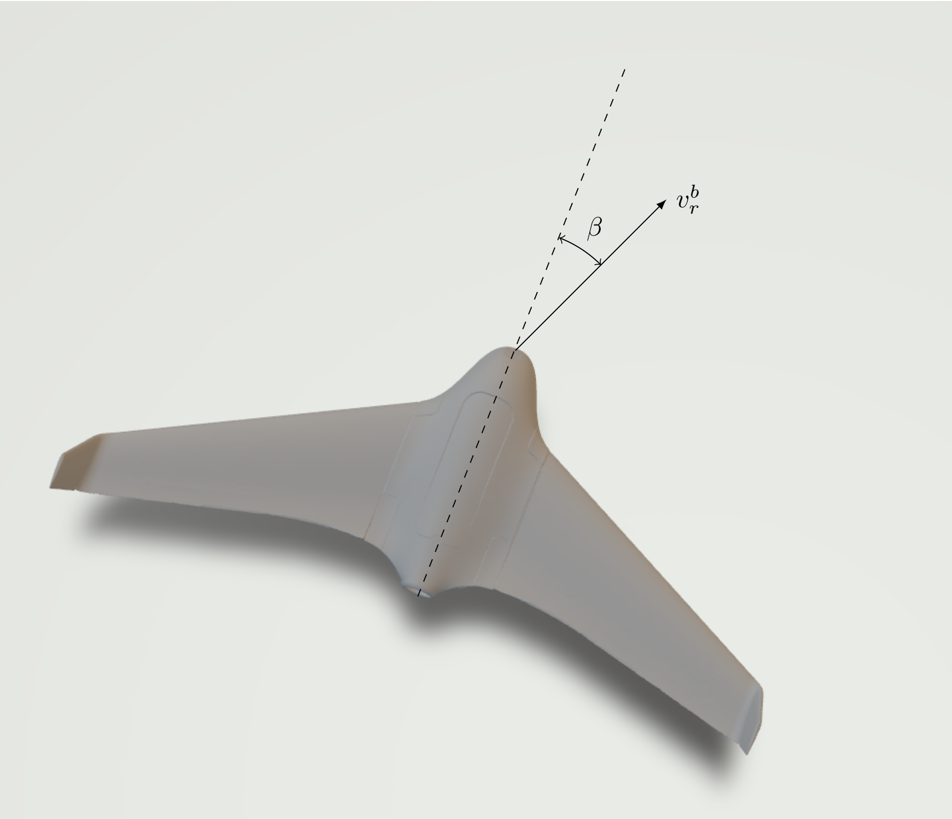


Figure 1.3: A top down view of a Skywalker X8 Flying Wing UAV 3D model with the SSA definition.

where c is the cord length[6], μ_∞ is the freestream viscocity coefficient, and a_∞ is the freestream speed of sound. At a given Reynolds and Mach number, the lift and drag coefficients are simply functions of the AOA α, which highlights the importance of the AOA on flight. Furthermore, for the range of airspeeds operated in by smaller aircraft, the effects of the Mach number and Reynolds number are approximately constant [6]. A generic example of how the lift and drag coefficients versus the AOA may look for an airfoil is shown in Figure 1.4.

Notice the linear increase in lift coefficient with α until a certain point where the slope starts to decrease and C_L reaches its peak value $C_{L,\text{max}}$. Under normal flow conditions, the flow follows the curvature of the wing and this is referred to as attached flow. At a certain AOA, the flow on the top side of the wing cannot overcome the local pressure gradient and starts to separate from the surface. This detached flow leads to large scale vortices which causes a significant increase in drag and a drop in the lift force produce by the wing. This phenomenon is referred

[6] In this expression for Re, the cord length c is used as a reference length, but it is possible to exchange this by different reference length.

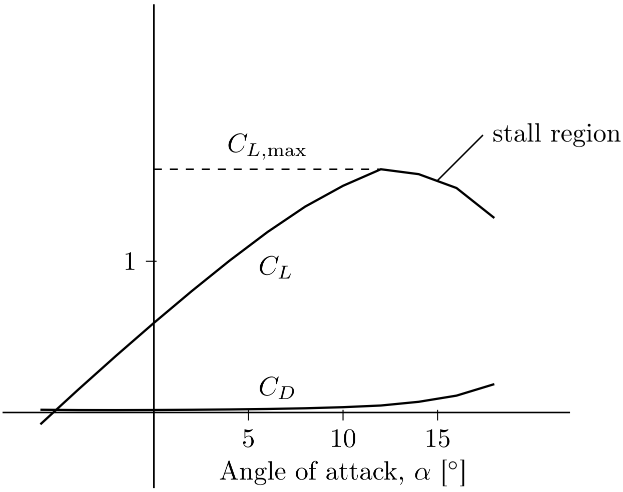


Figure 1.4: The lift and drag coefficients versus the angle of attack for an NREL S826 airfoil in incompressible flow with Re = 2×10^5. The graph is based on a fixed amount of data points and it should be noted that a finer resolution would produce a smoother curve. Data for the graph has been obtained through numerical CFD simulations and is provided by Richard Hann.

to as stall and due to the drastic loss of lift in this region, aircraft can be difficult to recover once they enter stall, e.g. spin that results from stall on one wing. Having an estimate of the AOA can therefore warn the operator when stall conditions are impending, or be exploited in an autopilot and is therefore essential to help recover the aircraft from an otherwise unrecoverable situation or intentionally operate under turbulent conditions. In addition, having estimates of the airspeed and AOA can open up for more complex maneuvers such as high precision deep-stall landings, see Mathisen et al. [63].

For the aircraft to sustain steady and level flight[7], the following two relations hold

$$
\begin{aligned}
F_L &= F_W \\
F_T &= F_D
\end{aligned}
$$

where F_W is the weight and F_T is the thrust force generated by the propulsive mechanism. That is, the lift force of the aircraft must equal the weight and the thrust force must equal the drag force. For fixed values of Re, M_∞, F_W, ρ_∞, and S in a steady flight state, the lift coefficient is

$$C_L = \frac{2F_L}{\rho_\infty S V_a^2} = \frac{2F_W}{\rho_\infty S V_a^2} \tag{1.6}$$

Consequently, under steady and level flight, each value of V_a will correspond to a specific α value and vice versa. A lower airspeed will result in a higher AOA

[7] The wind velocity is assumed to have zero vertical component to simplify this example.

and a higher airspeed will result in a lower AOA. For small UAVs, the range and endurance is usually of high importance and estimates of airspeed will allow controlling the UAV to maximize either of those parameters. Hovenburg et al. [43] presented an analysis of mission performance trade-offs for small unmanned aircraft.

The weather phenomenon icing is a major threat to unmanned aviation. A combination of sub-zero temperatures and high humidity can lead to atmospheric icing conditions where supercooled water droplets impinge and freeze on exposed surfaces of an aircraft. As ice builds on the wings, the aerodynamic capabilities of the UAV deteriorate (for a treatment of aircraft icing, the reader is directed to Gent et al. [32]). This deterioration is manifested in the lift and drag coefficients and if not mitigated, can result in an aircraft not capable of sustaining flight. Using the ANSYS computational fluid dynamics (CFD) tool FENSAP-ICE, Hann et al. [37] investigated the influence of different realistic ice accretion shapes on an NREL S826 airfoil with low airspeed typical for a small UAV. The ice shapes considered in order of severity is *glaze*, *rime*, and *mixed*. Figure 1.5 compares the lift and drag coefficients under the influence of the different ice shapes compared to a clean airfoil baseline. The results show that under the presence of ice, the stall region is reached at a lower AOA, the value of the maximum lift coefficient is reduced, and larger drag penalties are imposed on lower AOA values.

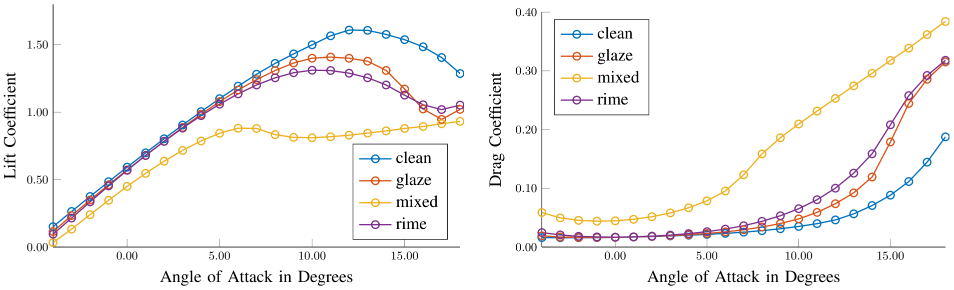


Figure 1.5: FENSAP-ICE simulation results for the lift and drag coefficient for the NREL S826 airfoil clean and under different icing shapes for Re $= 2{\times}10^5$. The plots are from [37].

Sørensen et al. [82] proposed a structural analysis icing detection scheme. The method states residuals where both lift and drag coefficients as well as AOA and airspeed values are assumed known. By applying a statistical change detection, simulation showed that it is possible to detect the structural changes of a UAV due to icing. Estimates of icing can either be be used to warn the operator or exploited in an autonomous icing protection systems, such as the one proposed by Sørensen et al. [83]. In addition, Rotondo et al. [78] presented an actuator fault diagnosis method that assumes air data estimates and is capable of detecting actuator icing faults.

The influence of wind is not only interesting with respect to the UAV. Consider the application of precision airdrop, where an unmanned aircraft is expected to deliver an object to a precise position by releasing the object from a given altitude [7]. One approach is to assume a known constant wind velocity and model the object using gravity and drag forces in order to calculate the desired air-release point [62]. Another approach to object delivery as well as object retrieval, is to use a cable-body system and having the UAV engage in a circular towing maneuver that allows the cable endbody position to remain relative constant[8]. While this is fairly straightforward in the absence of wind, it becomes a challenging problem when the UAV is subjected to wind. Merz and Johansen [65] proposed a path adjustment approach using optimal control methods to ensure a stable position of the cable endbody when flying in a known constant uniform wind field.

The preceding section attempted to outline the importance of the relative velocity of the UAV with respect to the wind from which the air data can be computed. The lift and drag force produced by the aircraft are functions of the airspeed and lift and drag coefficients, and the lift and drag coefficients are functions of the AOA. Having estimates of the air data can provide both a higher level of autonomy, efficiency and robustness in operating the UAV and can potentially open up for new innovative approaches in the way that UAVs are controlled as well as the applications they are used for.

1.3 Background

This section contains a short survey on existing technology and previous results in the field of air data sensors and estimation. The contributions in this thesis can be divided into state estimation approaches and a method where machine learning (ML) is used in combination with an array of distributed pressure sensors. The survey will therefore account for results related to both approaches.

At the time of writing, the author is only aware of one company - the Aeroprobe Corporation - producing air data systems that are within the requirements of weight and space of the Skywalker X8 Flying Wing UAV that has been used as the experimental platform throughout the thesis (see Appendix A). Several other companies exist that produce similar solutions for aircraft that are slightly larger. The commercially available solutions all revolve around a pneumatic multi-hole probe protruding from the UAV that is connected to a pressure scanner inside the UAV through rubber tubes. The Aeroprobe Corporation solution, the Micro Air Data System [2], uses a 5-port air data probe connected to a pressure scanner. The pressure scanner contains a microcomputer that runs an algorithm to calculate air data parameters from the direct pressure measurements from the ports on the probe (shown in Fig. 1.6). The pressure scanner and the air data probes combined with the necessary wind tunnel tests needed to calibrate the system, results in a significant price compared to the Skywalker X8 UAV including the autopilot system. The expensive components and the need for expensive calibration therefore

[8] This interesting maneuver was pioneered by missionary Nate Saint who used it to deliver gifts to remote settlements in South America in the 1950s [40].

limits the use of the air data system to high-cost operations.

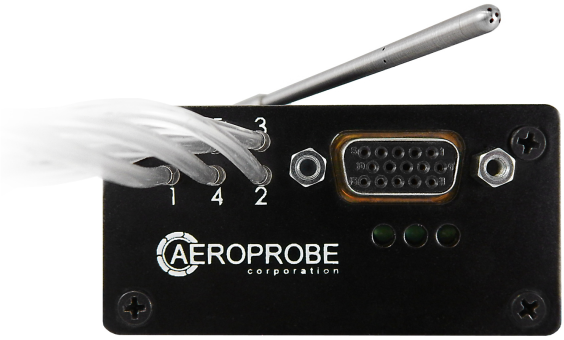


Figure 1.6: The Micro Air Data Computer measurement system. Copyright: Aeroprobe.

Several papers have been published on air data state estimation using a UAV standard sensor suite that consists of an inertial measurement unit (IMU), a global navigation satellite system (GNSS) receiver, and a differential pressure Pitot-static probe that provides a measurement of the relative velocity in the forward direction. Often, the attitude of the aircraft is also assumed to be either measured or estimated which may require a magnetometer in addition to the IMU. It is worth noting once again that there exists a mapping from relative velocity to air data, and also from air data to relative velocity[9]. In addition, if assuming knowledge of the attitude and velocity over ground of the UAV, it is possible to compute the relative velocity from the wind velocity and vice versa.

Long and Song [57] used sensor fusion in a modular architecture where both an aerodynamic and kinematic model was used to estimate the AOA and airspeed without using an airspeed sensor. A Newton–Raphson solver on an aerodynamic model combined with an extended Kalman filter (EKF, Brown [15]) was used by Ramprasadh and Arya [72] to obtain AOA and SSA estimates. Lie and Gebre-Egziabher [56] proposed a cascaded EKF structure and an aircraft dynamics model for estimating the air data without an airspeed measurement. Cho et al. [21] proposed an EKF method that assumes a scaled measurement of the airspeed combined with an aerodynamic model of the system to estimate the AOA, the SSA, and the airspeed sensor scaling factor. Wenz et al. [87] used a simplified aerodynamic model for the lift force combined with the Dryden wind model (as described in [29]) in an EKF structure. Wenz and Johansen [86] built on this result with a moving horizon estimator and improved the accuracy. A common denominator for the preceding methods is the use of an aerodynamic model of the forces on the UAV. Aerodynamic forces are complex and a potential disadvantage of this approach is that model errors due to simplifications or parameter inaccuracies may

[9] This assumes measuring or estimating in all three dimensions. The mathematical relation between the relative velocity of the UAV and air data will be stated in Chapter 2.

result in estimation errors.

An aerodynamic model-free air data state estimator has been proposed by Cho et al. [22] using a scaled airspeed sensor measurement to estimate wind speed, wind direction and airspeed sensor scaling factor. Johansen et al. [47] proposed an aerodynamic model-free kinematic approach to estimating the wind velocity and a Pitot-static probe scaling factor for small UAVs. The system was linear and Johansen et al. was able to show global exponential stability (GES) of the estimation error dynamic equilibrium points under persistence of excitation (PE) of the aircraft angular rates. Rhudy et al. [74] assumed to have an air data system providing air data measurements and used these to estimate the attitude of the aircraft and the wind velocity. An EKF structure with a thrust force model was used by Hansen and Blanke [39] for providing additional airspeeds measurements used in detecting sensor failure of the Pitot-static probe. Rhudy et al. [73] presented a nonlinear Kalman filter method to estimate airspeed that uses wind vanes that measure the AOA and SSA, but is free of an aerodynamic model of the aircraft and does not use an airspeed measurement, thus also allowing it to be used for Pitot-static probe icing detection. For this approach, Rhudy et al. implemented the method as both an EKF and as an unscented Kalman filter (see [48]) and found similar results in a comparison.

Using an array of pressure sensors in estimating the air data parameters for fixed-wing aircraft is not an entirely new approach. An extensive amount of research in flush air data sensing (FADS) systems originates from NASA in response to the problems associated with protruding Pitot probes. The FADS systems utilize pneumatic pressure orifices that are flush with the surface, placed in a symmetric and circular pattern on the nose of an aircraft combined with a aerodynamic pressure model. Larson et al. [53] demonstrated a FADS system in wind tunnels for subsonic airspeeds. Larson et al. extended this to transonic airspeeds [54], and Whitmore et al. [88] demonstrated the system in-flight. Using the pneumatic FADS system in combination with NNs to estimate freestream static and dynamic pressure was proposed and demonstrated by Rohloff et al. [77]. Rohloff et al. [76] proposed an air data sensing system, where NNs were used in combination with an aerodynamic model of the nose of the aircraft to estimate the air data estimates. Quindlen and Langelaan [71] presented a nose FADS system that used NNs to estimate air data for a soaring UAV. The system was trained using wind tunnel data and tested in flight without a ground truth sensor to provide verification. Furthermore, the size of the pneumatic system necessitated removing the electric motor, thereby requiring a less optimal launching procedure. Instead of pneumatic pressure sensors, Callegari et al. [17], proposed to combine a maximum likelihood estimator with redundant strip pressure sensors applied to the wings of a UAV to estimate airspeed and AOA. This was demonstrated in simulation, but only with a low level of noise and for small values of AOA. Samy et al. [80] developed a FADS system for a mini air vehicle where a matrix of pressure orifices was placed on the leading edge of the wing and a neural network was used to estimate the air data parameters. The system was tested in a wind tunnel and the pneumatic system designed by Samy et al. was connected to pressure transducers placed outside of the aircraft and the system in its presented configuration was therefore not usable for flight. Laurence et al. [55] proposed a distributed pressure sensor approach for estimating relative velocity for a small UAV. A CFD simulation software was used to investigate optimal sensor placement and both a nonlinear least-squares and a NN method were tested to estimate the AOA and SSA parameters in a wind tunnel experiment. However, the test setup by Laurence et al. was only tested for a single airspeed and the pressure sensors were not embedded in the aircraft, but instead located outside the tunnel connected by pneumatic tubes to the pressure points on the UAV.

Other approaches include Kumon et al. [51], where the influences of aerodynamic forces on a light-weight UAV is exploited to determine the wind velocity and direction. The wind velocity was assumed to be constant and with a zero vertical component. The system listed by Kumon et al. is underdetermined and they solve the problem by using an iterative optimization method over data spanning a period of time. Rodriguez et al. [75] presented a method for estimating wind velocity for a small UAV by using optical flow. Langelaan et al. [52] proposed a direct computation of the wind velocity, rate of change of wind velocity, and wind velocity spatial gradient based on GNSS velocity and acceleration measurements using linearized expressions for the aerodynamic forces and moments. Paces et al. [69] proposed a twin differential probe setup for estimating the AOA and SSA that could be integrated into an existing air data system, and twelve different probes was compared and tested. Martos and Rogers [61] showed that a low-cost off-the-shelf AOA-system utilizing a differential pressure probe together with airspeed measurements, makes the system invariant to ambient pressure changes if the measurements are normalized using the dynamic pressure[10].

1.4 Contributions of the Thesis

The thesis is organized into five chapters. Each chapter accounts for a different contribution in the field of estimating air data for small fixed-wing UAVs.

Chapter 2

This chapter centers on a short preliminary treatment of the required navigation basics and then continues into the design of a nonlinear model-based wind velocity observer. The contribution is the nonlinear wind velocity observer, which provide exponential stability and convergent estimates of wind velocity and relative velocity. The observer utilizes a standard UAV sensor suite combined with a model of the aerodynamic forces on the aircraft, and an airspeed sensor. By exploiting an aerodynamic model of the forces on the UAV and relating the forces to the observer states, the observer has the advantage of not having PE requirements, i.e. there are no requirements on maneuvers or flight trajectories in order to have converging estimates. A potential disadvantage is that model errors may result in

[10] The quantity defined as $q_\infty := 1/2\rho V_a^2$.

estimation errors. Another contribution of the paper is the compact representation of the small aircraft model of Beard and McLane [6] using the matrix-vector representation of Fossen [30]. For the proof, the aerodynamic forces were divided into a stabilizing linear term and a vector of the remaining nonlinear aerodynamic forces with physical properties such as energy dissipation, which can be exploited when constructing the Lyapunov function for observer error dynamics. Finally, the nonlinear observer is validated through simulation using a small fixed-wing UAV exposed to wind.

Chapter 3

This chapter contains an ML solution to estimating air data. The contribution of the chapter is the ML solution, which consists of combining non-intrusive low-cost MEMS-based pressure sensors embedded in the surface of the UAV with a linear regression (LR) or (NN) modeling approach. A strength of the presented solution lies in the flexibility of the sensor placement, since there are few geometric constraints. Depending on the UAV, this potentially allows equipping a UAV with an air data estimation system where other solutions are not viable. An example are aircraft with nose propellers, which denies a nose FADS system. Instead the pressure sensors can be embedded on the wings or fuselage of the aircraft, thereby still allowing air data parameter estimation. The wide range of possible layouts of the solution can be chosen to accommodate the geometry of the UAV it is designed for. The presented solution is not pneumatic, i.e. does not require tubing to pressure scanners that can be sensitive to mechanical stress. Furthermore, the approach removes the need for mounting a protruding probe that is exposed and susceptible to damage during landing. Another contribution consists of the validation obtained from extensive testing and comparing the results obtained from different ML models and sensor configurations in both wind tunnel and flight experiments.

Chapter 4

The contribution of this chapter is two distinct Kalman filter approaches for correcting an air data system or air data estimator with low-frequency drift or a constant error bias. These errors could be due to sensor performance degradation, structural changes on the UAV or sensors, or operating outside of the ambient conditions for which the system was calibrated for. Both presented estimators use only a standard sensor suite consisting of a GNSS receiver, an IMU, and a heading reference, combined with an air data system that is assumed to provide biased relative velocity measurements. Provided with persistence-of-excitation (PE) of the angular velocity of the UAV, the two linear time-varying systems are proven to be uniform completely observable (UCO) and uniform completely controllable (UCC), implying GES for the deterministic case and boundedness in the mean square sense for the stochastic case. The estimators are verified through simulation and flight experiments.

Chapter 5

This chapter summarizes findings in methods of air data estimation. It will also be accompanied by a discussion on future work and how the different methods could be tied together and further improved.

1.5 Publications

The research conducted during the project has resulted in the following publications. The patent application referenced below contains seven claims based on the concepts presented in Chapters 2 - 4 as well as the Future Work section from Chapter 5. These seven claims will potentially be taken out of the patent to function as the basis for a separate patent application.

Publications included in this thesis:

[13] K. T. Borup, T. I. Fossen, and T. A. Johansen. A Nonlinear Model-Based Wind Velocity Observer for Unmanned Aerial Vehicles. In *Non-linear Control Systems - 10th NOLCOS 2016*, 2016

[12] K. T. Borup, T. I. Fossen, and T. A. Johansen. A Machine Learning Approach for Estimating Air Data Parameters for Small Fixed-Wing UAVs Using Distributed Pressure Sensors. Submitted to IEEE Transactions on Aerospace and Electronic Systems. A preprint is available here

[14] K. T. Borup, B. N. Stovner, T. I. Fossen, and T. A. Johansen. Kalman Filters for Air Data System Bias Correction for a Fixed-Wing UAV. Submitted to IEEE Transactions on Control Systems Technology. A preprint is available here

- United Kingdom Patent Application No. 1614339.8, Detection of Icing on Wings, GJE Ref: HWH00066GB, Norwegian University of Science and Technology

Publications not included in the this thesis:

[11] K. T. Borup, T. I. Fossen, J. Braga, and J. Borge de Sousa. Nonlinear observer for depth-aided INS: Experimental evaluation using an AUV. In *2014 22nd Mediterranean Conference of Control and Automation (MED)*, 2014

[31] R. Galeazzi, K. T. Borup, N. K. Niemann, H. Poulsen, and F. Caponetti. Adaptive Backstepping Control of Lightweight Tower Wind Turbine. In *American Control Conference (ACC)*, 2015

Chapter 2

Nonlinear Model-Based Wind Velocity Observer

In this chapter, a nonlinear model-based wind velocity observer for small UAVs is presented. The observer exploits a simple model of the aerodynamic forces on the aircraft in combination with a standard sensor suite and an airspeed sensor. An advantage of the observer is that it has no requirements on PE, i.e. no requirements on the maneuvers and flight envelope of the UAV. Before the nonlinear observer is presented, it is convenient to have a short introduction to the navigation basics and the notation used.

2.1 Notation and Navigation Preliminaries

For a vector or matrix X, $X^\top$ denotes its transpose. The operator $\|\cdot\|$ denotes the Euclidean norm. For a vector $x \in \mathbb{R}^3$, $S(x)$ denotes the skew-symmetric matrix

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

The $n \times n$ identity matrix is denoted by I_n and the $m \times n$ zero element matrix by $0_{m\times n}$.

In navigation, different sensor systems measure variables in different coordinate frames, and in order to solve the navigation problem, it is necessary to have the relationship between the different coordinate frames properly modeled. For the observers presented in this thesis, it is sufficient to consider only two coordinate frames, namely the body-fixed (BODY) frame and the North-East-Down (NED) frame. The BODY coordinate frame is fixed to the aircraft at its center of gravity and moves and rotates with the vehicle. Figure 2.1 shows the definition of the BODY frame for a Skywalker X8 UAV. The angular velocity $\omega^b = [p, q, r]^\top$ and the relative velocity $v_r^b = [u_r, v_r, w_r]^\top$ are both defined with respect to the BODY frame.

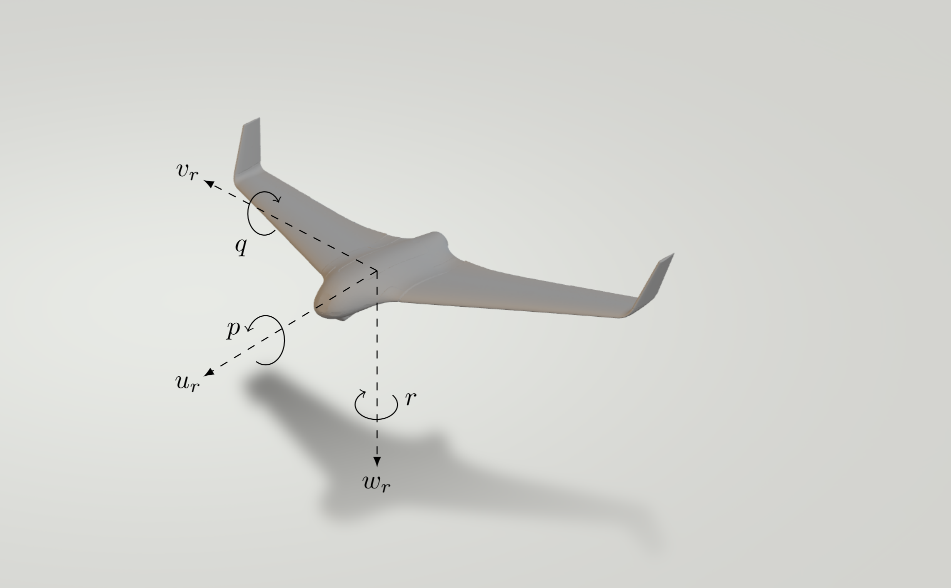


Figure 2.1: A BODY coordinate frame illustrated on a Skywalker X8 UAV.

The NED frame is the coordinate frame we refer to in our everyday lives. It is defined as a tangent plane on the surface of the Earth at a point of interest[1] and is convenient, since it provides an easy interpretation of position, velocity, and attitude relative to north, east and down directions. However, the NED frame does not account for the curvature of the Earth and should therefore not be used in navigation over larger areas. Since small UAVs often operate over a limited area, it shall suffice to limit ourselves to the BODY and NED frames for the scope of this thesis. Other coordinate frames commonly used in navigation are the Earth-Centered-Intertial (ECI) frame and the Earth-Centered-Earth-Fixed (ECEF) frame. For a more in-depth treatment on navigation and coordinate frames, the reader is directed to Farrell [27], Fossen [30] and Groves [34].

Vectors decomposed in the BODY and NED coordinate frames are denoted by the superscripts b and n, respectively. Consequently, the relation between a vector x^b decomposed in the BODY frame and its corresponding decomposition in the NED frame x^n, is governed by $x^n = R_b^n x^b$, where R_b^n is the rotation matrix from BODY to NED. The rotation matrix belongs to the special orthogonal group of order 3, $R_b^n \in \mathrm{SO}(3)$, such that $R \in \mathbb{R}^{3\times 3}$, R is orthogonal, and $\det(R) = 1$. Consequently, we have the corresponding rotation matrix from NED to BODY as $R_n^b = (R_b^n)^\top$. A common parameterization of the rotation matrix is through Euler angles. Euler angles are a set of three angles used to describe the orientation of a rigid body with respect to a given coordinate frame and have the advantage of allowing intuitive interpretation. The Euler angles can be defined in numerous ways, but it is common in guidance, navigation, and control applications ([30]) to

[1] For aircraft navigation, an example of this could be the runway or UAV launch position.

define them by the principal rotations about the z, y, and x axes from BODY to NED frame as specified by the angles:

$$\Phi_b^n = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

Using the Euler angle parameterization, the rotation matrix is given as

$$R_b^n = R(\Phi_b^n) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix}$$

where the notation $\mathrm{c}x = \cos(x)$ and $\mathrm{s}x = \sin(x)$ has been used for notational brevity. Another possible orientation representation is quaternions. Although less intuitive than Euler angles, quaternions are more computational efficient and do not allow for the possibility of a gimbal lock, i.e. if the the pitch angle is ±90 degrees, the aircraft is either pointed directly up or down and and the roll and yaw angles are undefined. Since the scope of this thesis is not attitude estimation, Euler angles and the rotation matrix shall be used for simplicity. On the topic of attitude representation, see [19, 25, 59].

The estimation of position, velocity over ground, and attitude (PVA) is the basis of navigation[2]. The kinematic navigation model is given by

$$\dot{p}^n = v_g^n \tag{2.1}$$

$$\dot{v}_g^n = f^n + g^n \tag{2.2}$$

$$\dot{R}_b^n = R_b^n S(\omega^b) \tag{2.3}$$

where p^n is the position decomposed in the NED frame, ${v_g}^n$ is the velocity over ground decomposed in the NED frame, and f^n is the specific force. The gravity vector decomposed in the NED frame is denoted by g^n, and ω^b is the angular velocity of the BODY frame relative to the NED frame, decomposed in the BODY frame. Although air data estimation is not a navigation problem, the navigation equations will play a role in this chapter and in chapter 4. A good overview of nonlinear attitude estimation is presented in Crassidis et al. [24] and the multiplicative EKF presented by Markley et al. [58] is becoming widely used. Grip et al. [33] proposed a globally exponentially stable nonlinear attitude and gyro bias observer requiring two or more pairs of vector measurements. Hosen et al. [41] build on these results by exploiting velocity over ground estimates obtained using computer vision-based optical flow. Lastly, Hansen et al. [38] presented an nonlinear attitude observer and a translational motion observer that uses tightly coupled integration of intertial measurements with global satellites measurements.

[2] Often velocity over ground is referred to as linear velocity or simply velocity, but given the importance of both wind velocity and relative velocity in this thesis, the explicit naming of velocity over ground shall be kept.

2.2 Problem Formulation

The velocity over ground of a UAV can be expressed as the sum of the relative velocity and the wind velocity according to

$$v_g^n = R_b^n v_r^b + R_b^n v_w^b \tag{2.4}$$

where $v_g^n = [v_n, v_e, v_d]^\top$ is the velocity over ground of the UAV decomposed in the NED frame, ${v_r}^b = [u_r, v_r, w_r]^\top$ is the relative velocity of the UAV with respect to the wind decomposed in the BODY frame and ${v_w}^b = [u_w, v_w, w_w]^\top$ is the wind velocity decomposed in the BODY frame. From the relative velocity, the airspeed, AOA and SSA are recognized as

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{2.5}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{2.6}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \tag{2.7}$$

And the inverse relation is

$$u_r = V_a\cos(\alpha)\cos(\beta) \tag{2.8}$$

$$v_r = V_a\sin(\beta) \tag{2.9}$$

$$w_r = V_a\sin(\alpha)\cos(\beta) \tag{2.10}$$

The goal pursued in this chapter is to estimate both the wind velocity v_w^b and the relative velocity v_r^b.

2.3 UAV Rigid-Body Kinetics

In order to obtain a detailed model of the UAV the rigid-body kinematics will be considered. By application of *Euler's first and second axioms* the rigid-body kinetics for the translational and rotational dynamics of a rigid body is (Fossen [30])

$$m(\dot{v}_g^b + S(\omega^b)v_g^b) = F^b \tag{2.11}$$

$$J\dot{\omega}^b - S(J\omega^b)\omega^b = M^b \tag{2.12}$$

where m is the mass of the vehicle, $J \in \mathbb{R}^{3\times 3}$ is the symmetric inertia tensor and F^b and M^b are the forces and moments on the vehicle. In Beard and McLain [6], it is shown that a small aircraft can be modeled by (2.11) and (2.12), where

$$J = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix} \tag{2.13}$$

is a matrix of products and moments of inertia, and the aircraft forces and moments can be approximated by

$$
\begin{aligned}
\begin{bmatrix} F^b \\ M^b \end{bmatrix} =& \frac{1}{2}\rho V_a^2 S \begin{bmatrix} -C_D(\alpha)\cos(\alpha) + C_L(\alpha)\sin(\alpha) \\ C_{Y_0} + C_{Y_\beta}\beta \\ -C_D(\alpha)\sin(\alpha) - C_L(\alpha)\cos(\alpha) \\ b\left(C_{l_0} + C_{l_\beta}\beta + C_{l_{\delta a}}\delta_a + C_{l_{\delta r}}\delta_r\right) \\ c\left(C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta e}}\delta_e\right) \\ b\left(C_{n_0} + C_{n_\beta}\beta + C_{n_{\delta a}}\delta_a + C_{n_{\delta r}}\delta_r\right) \end{bmatrix} \\
&+ \frac{1}{2}\rho V_a^2 S \begin{bmatrix} C_{X_q}(\alpha)\frac{c}{2V_a}q + C_{X_{\delta_e}}(\alpha)\delta_e \\ C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\ C_{Z_q}(\alpha)\frac{c}{2V_a}q + C_{Z_{\delta_e}}(\alpha)\delta_e \\ b\left(C_{l_p}p + C_{l_r}r\right)\frac{b}{2V_a} \\ c\left(C_{m_q}q\right)\frac{b}{2V_a} \\ b\left(C_{n_p}p + C_{n_r}r\right)\frac{b}{2V_a} \end{bmatrix} \\
&+ \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}}\left((k_{\text{motor}}\delta_t)^2 - V_a^2\right) \\ 0 \\ 0 \\ -k_{T_p}(k_\Omega\delta_t)^2 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{2.14}
$$

where ρ_∞ is the density of air, and g is the gravity constant. The aerodynamic lift and drag coefficients, $C_L(\alpha)$ and $C_D(\alpha)$, and the aerodynamic force coefficients are nonlinear functions of AOA:

$$
\begin{aligned}
C_{X_q}(\alpha) &\triangleq -C_{D_q}\cos(\alpha) + C_{L_q}\sin(\alpha) \\
C_{X_{\delta_e}}(\alpha) &\triangleq -C_{D_{\delta_e}}\cos(\alpha) + C_{L_{\delta_e}}\sin(\alpha) \\
C_{Z_q}(\alpha) &\triangleq -C_{D_q}\sin(\alpha) - C_{L_q}\cos(\alpha) \\
C_{Z_{\delta_e}}(\alpha) &\triangleq -C_{D_{\delta_e}}\sin(\alpha) - C_{L_{\delta_e}}\cos(\alpha)
\end{aligned}
$$

where $C_{Y_0}, C_{Y_\beta}, C_{l_0}, C_{l_\beta}, C_{l_{\delta_a}}, C_{n_0}, C_{n_\beta}, C_{n_{\delta_a}}, C_{n_{\delta r}}, C_{Y_p}, C_{Y_r}, C_{Y_{\delta_a}}, C_{l_p}, C_{l_r}, C_{m_q}$, C_{n_p}, C_{n_r}, and C_{prop} are constant aerodynamic coefficients[3]. $\delta = [\delta_a, \delta_e, \delta_r, \delta_t]^\top$ are the control signals of the aileron deflection, elevator deflection, rudder deflection and throttle deflection. The area of the wing is given by S, the propeller area is S_{prop}, b is the wing span, and c is the mean aerodynamic chord of the wing. k_{motor} is the efficiency of the motor and k_{T_p} and k_Ω are constants that relate the throttle deflection δ_t to the moment opposite the propeller rotation.

[3] these coefficients are, depending on the influence they model, usually referred to as either stability derivatives or control derivatives, and are found as the partial derivatives in a Taylor series approximation.

2.4 Matrix-Vector Form Aircraft Model

The aircraft model of Beard and McLain list in equations 2.12 - 2.14 can be expressed in matrix-vector form according to (Fossen [30])

$$M_{RB}\nu + C_{RB}(\nu^b)\nu^b = \tau^b_{RB} \tag{2.15}$$

where $\nu^b = [(v^b_g)^\top, (\omega^b)^\top]^\top$ is the 6-DOF generalized velocity vector expressed in BODY frame. The rigid-body mass matrix M_{RB} and rigid-body Coriolis and centripetal matrix $C_{RB}(\nu)$ are given by

$$M_{RB} = \begin{bmatrix} mI_3 & 0_{3\times3} \\ 0_{3\times3} & J \end{bmatrix} \tag{2.16}$$

$$C_{RB}(\nu) = \begin{bmatrix} mS(\omega^b) & 0_{3\times3} \\ 0_{3\times3} & -S(J\omega^b) \end{bmatrix} \tag{2.17}$$

The generalized vector of external forces and moments decomposed in the BODY frame is $\tau^b_{RB} = [(F^b)^\top, (M^b)^\top]^\top$. The wind velocity is assumed irrotational and steady relative to the Earth. The generalized wind velocity of an irrotational flow decomposed in the BODY frame is

$$\nu^b_w = [\underbrace{u_w, v_w, w_v}_{v^b_w}, 0, 0, 0]^\top \tag{2.18}$$

and the BODY frame generalized relative velocity vector $\nu^b_r = \nu^b - \nu^b_w$. Since $v^n_w = R^n_b v^b_w$, then under assumption of steady wind ($\dot{v}^n_w = 0$), we get

$$\dot{v}^n_w = \dot{R}v^b_w + R\dot{v}^b_w = 0 \tag{2.19}$$

Consequently,

$$\dot{v}^b_w = S(\omega^b)v^b_w \tag{2.20}$$

This leads to the property

$$M_{RB}\dot{\nu}^b_w + C_{RB}(\nu^b_r)\nu^b_w = 0 \tag{2.21}$$

which is easily verified by expanding the matrices M_{RB} and C_{RB}. Hence,

$$M_{RB}\dot{\nu}^b + C_{RB}(\nu^b)\nu^b = M_{RB}[\dot{\nu}^b_r + \dot{\nu}^b_w] + C_{RB}(\nu^b_r)[\nu^b_r + \nu^b_w]$$

and finally, by inserting (2.21) we get

$$M_{RB}\dot{\nu}^b + C_{RB}(\nu^b)\nu^b = M_{RB}\dot{\nu}^b_r + C_{RB}(\nu^b_r)\nu^b_r \tag{2.22}$$

Notice that (2.22) is equivalent to Property 8.1 in Fossen [30]. The dynamics of the relative velocity ν^b_r can finally be expressed as

$$M_{RB}\dot{\nu}^b_r + C_{RB}(\nu^b_r)\nu^b_r = \tau^b_{RB} \tag{2.23}$$

With the aircraft model relating the aerodynamic forces to the generalized relative velocity vector, the nonlinear wind velocity observer can be designed.

2.5 Nonlinear Relative Velocity Observer

The following section presents the relative velocity observer.

2.5.1 Measurements

The proposed relative observer assumes that the UAV is equipped with a standard sensor suite consisting of an IMU, a heading reference such as a magnetometer, a GNSS receiver, and a Pitot-static probe, and in addition, an airspeed sensor. Furthermore, the UAV is also assumed to be equipped with an attitude estimator using the standard sensor suite measurements. The following measurements are needed to implement the observer:

- An attitude estimate from the attitude estimator, typically represented as Euler angles or unit quaternions, used to compute the rotation matrix R_b^n.
- Measurement of the airspeed V_a.
- A Pitot-static probe measuring the relative longitudinal velocity $u_{r,m} > 0$, which relates to the relative longitudinal velocity $u_r = \gamma u_{r,m}$ by an unknown positive scaling factor γ.
- An angular rate measurement ω^b from the IMU, which has been compensated for gyro drift and bias.
- The control deflections $\delta = [\delta_a, \delta_e, \delta_r, \delta_t]^\top$ from the autopilot.
- The UAV velocity over ground v_g^n measured by the GNSS.

It should be noticed that the velocity over ground signal is only needed, if the observer should also estimate the wind velocity. The cascaded structure of the model-based nonlinear relative velocity observer is shown in Figure 2.2.

Comment on the V_a measurement: Small UAVs are usually equipped with Pitot-static probes that measure the longitudinal component of the relative velocity u_r, but a direct measurement of the airspeed V_a is not usually available. It is however possible to implement the observer without the airspeed measurement by exchanging the V_a measurement with the estimate $\hat{V}_a$. The results obtained using this substitution are similar to the simulation results presented in this chapter and does not require the airspeed measurement. However, no stability analysis has been performed to support this version of the observer.

2.5.2 Assumptions

The assumptions needed to prove exponential stability are:

- The measurements R_b^n, V_a, ω^b and v_g^n are smooth bounded signals. Hence, these signals will be treated as time-varying known signals and not states in the observer. This implies that the observer error dynamics become non-autonomous.

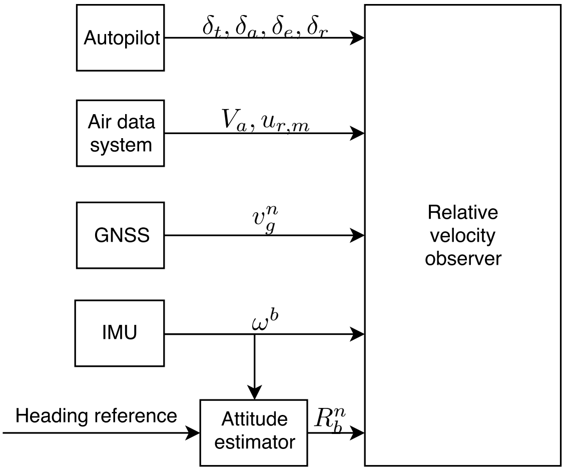


Figure 2.2: Block diagram showing the cascaded structure of the observer including the signals used in the wind velocity observer where the air data system constitutes a air data sensor and a Pitot-static probe.

- The wind velocity vector is slowly time-varying such that $\dot{v}_w^n = 0$.
- The sensor scaling factor γ is positive and slowly time-varying such that $\dot{\gamma} = 0$, and the relative longitudinal velocity u_r is positive.
- The relative velocity in the lateral direction v_r is small compared to the airspeed V_a and consequently the SSA β, is small.
- The lift and drag coefficient, $C_L(\alpha)$ and $C_D(\alpha)$, are linear in α, i.e. α is small.
- The aircraft and autopilot system is closed-loop stable (stable flight) and the flight envelope ensures the σ_1, as defined in the proof of Lemma 2.1, is positive.

2.5.3 Observer model

Throughout the following design and analysis the argument for time-varying signals have been omitted for notational simplicity, except when emphasis is deemed important. The translational motion components of (2.23) can be written as:

$$m\dot{v}_r^b + mS(\omega^b)v_r^b + m(R_b^n)^\top g^n = F_{\text{aero},1}(v_r^b, \omega^b, \delta) + F_{\text{aero},2}(V_a, \omega^b, \delta) \tag{2.24}$$

where $g^n = [0, 0, g]^\top$ is the gravity vector decomposed in the NED frame. The functions below depend on the state v_r^b and time-varying measurements represented by the arguments V_a, ω^b, and δ:

$$F_{\text{aero},1}(v_r^b,\omega^b,\delta) = \frac{1}{2}\rho S V_a^2 \begin{bmatrix} -C_D(\alpha)\cos(\alpha) + C_L(\alpha)\sin(\alpha) \\ C_{Y_0} + C_{Y_\beta}\beta \\ -C_D(\alpha)\sin(\alpha) - C_L(\alpha)\cos(\alpha) \end{bmatrix} + \frac{1}{2}\rho S V_a^2 \begin{bmatrix} C_{X_q}(\alpha)\frac{c}{2V_a}q + C_{X_{\delta_e}}(\alpha)\delta_e - \frac{1}{S}S_{\text{prop}}C_{\text{prop}} \\ 0 \\ C_{Z_q}(\alpha)\frac{c}{2V_a}q + C_{Z_{\delta_e}}(\alpha)\delta_e \end{bmatrix} \tag{2.25}$$

$$F_{\text{aero},2}(V_a,\omega^b,\delta) = \begin{bmatrix} \frac{1}{2}\rho S_{\text{prop}}C_{\text{prop}}\left(k_{\text{motor}}\delta_t\right)^2 \\ \frac{1}{2}\rho S V_a^2\left(C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\right) \\ 0 \end{bmatrix} \tag{2.26}$$

The reason for this separation of the aerodynamic forces will become apparent in the proof, but it is worth noticing that the airspeed is not an argument for $F_{\text{aero},1}(\cdot)$, since $V_a = \|v_r^b\|$. To simplify notation, the aerodynamic force terms will be denoted using t as replacement for measured signal arguments, i.e.

$$\begin{aligned} F_{\text{aero},1}(v_r^b,t) &:= F_{\text{aero},1}(v_r^b,\omega^b,\delta) \\ F_{\text{aero},2}(t) &:= F_{\text{aero},2}(V_a,\omega^b,\delta) \end{aligned}$$

2.5.4 Relative velocity observer design

Lemma 2.1 (Aerodynamic forces). *Under the assumptions stated in Section 2.5.2, the linear and nonlinear terms of $F_{aero,1}(v_r^b,t)$ can be expressed as the sum:*

$$F_{aero,1}(v_r^b,t) := -D(t)v_r^b - d(v_r^b,t) \tag{2.27}$$

Proof: Defining the longitudinal wind speed

$$V_{\text{lon}} \triangleq \sqrt{u_r^2 + w_r^2} \tag{2.28}$$

By application of trigonometric relations

$$\sin(\alpha) = \sin\left(\tan^{-1}\left(\frac{w_r}{u_r}\right)\right) = \frac{w_r}{V_{\text{lon}}} \tag{2.29}$$

$$\cos(\alpha) = \cos\left(\tan^{-1}\left(\frac{w_r}{u_r}\right)\right) = \frac{u_r}{V_{\text{lon}}} \tag{2.30}$$

The following expressions for the aerodynamic forces are based on linear theory and the assumption that $V_a \approx V_{\text{lon}}$. Hence,

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \approx \frac{v_r}{V_a} \tag{2.31}$$

The linear sway force in (2.25) for an aircraft that is symmetrical about the xz-plane ($C_{Y_0} = 0$) then becomes:

$$C_{Y_0} + C_{Y_\beta}\beta = C_{Y_\beta}\frac{v_r}{V_a} \tag{2.32}$$

Furthermore

$$\alpha = \sin^{-1}\left(\frac{w_r}{V_{\text{lon}}}\right) \approx \frac{w_r}{V_a} \tag{2.33}$$

$$C_L(\alpha) \approx C_{L_0} + C_{L_\alpha}\alpha = C_{L_0} + C_{L_\alpha}\frac{w_r}{V_a} \tag{2.34}$$

$$C_D(\alpha) \approx C_{D_0} + C_{D_\alpha}\alpha = C_{D_0} + C_{D_\alpha}\frac{w_r}{V_a} \tag{2.35}$$

and

$$\begin{aligned} C_{X_{\delta_e}}(\alpha)\delta_e &\triangleq (-C_{D_{\delta_e}}\cos(\alpha) + C_{L_{\delta_e}}\sin(\alpha))\delta_e && (2.36)\\ &\approx (-C_{D_{\delta_e}}\frac{u_r}{V_a} + C_{L_{\delta_e}}\frac{w_r}{V_a})\delta_e && (2.37) \end{aligned}$$

$$\begin{aligned} C_{Z_{\delta_e}}(\alpha)\delta_e &\triangleq (-C_{D_{\delta_e}}\sin(\alpha) - C_{L_{\delta_e}}\cos(\alpha))\delta_e && (2.38)\\ &\approx (-C_{D_{\delta_e}}\frac{w_r}{V_a} - C_{L_{\delta_e}}\frac{u_r}{V_a})\delta_e && (2.39) \end{aligned}$$

Hence, with abuse of notation we define a function $F_{\text{aero},1}(v_r^b, t)$, which depends on the state v_r^b and several time-varying measurements all denoted by the single argument t as:

$$\begin{aligned} F_{\text{aero},1}(v_r^b, t) :=& \frac{1}{2}\rho S V_a^2 \begin{bmatrix} -C_D(\alpha)\cos(\alpha) + C_L(\alpha)\sin(\alpha) \\ C_{Y_0} + C_{Y_\beta}\beta \\ -C_D(\alpha)\sin(\alpha) - C_L(\alpha)\cos(\alpha) \end{bmatrix} \\ &+ \frac{1}{2}\rho V_a^2 \begin{bmatrix} C_{X_q}(\alpha)\frac{c}{2V_a}q + C_{X_{\delta_e}}(\alpha)\delta_e - \frac{1}{S}S_{\text{prop}}C_{\text{prop}} \\ 0 \\ C_{Z_q}(\alpha)\frac{c}{2V_a}q + C_{Z_{\delta_e}}(\alpha)\delta_e \end{bmatrix} \\ =& \frac{1}{2}\rho V_a^2 S \begin{bmatrix} -\left(C_{D_0} + C_{D_\alpha}\frac{w_r}{V_a}\right)\frac{u_r}{V_a} + \left(C_{L_0} + C_{L_\alpha}\frac{w_r}{V_a}\right)\frac{w_r}{V_a} \\ C_{Y_\beta}\beta \\ -\left(C_{D_0} + C_{D_\alpha}\frac{w_r}{V_a}\right)\frac{w_r}{V_a} - \left(C_{L_0} + C_{L_\alpha}\frac{w_r}{V_a}\right)\frac{u_r}{V_a} \end{bmatrix} \\ &+ \frac{1}{2}\rho V_a^2 S \begin{bmatrix} \left(-C_{D_q}\frac{u_r}{V_a} + C_{L_q}\frac{w_r}{V_a}\right)\frac{c}{2V_a}q \\ 0 \\ \left(-C_{D_q}\frac{w_r}{V_a} - C_{L_q}\frac{u_r}{V_a}\right)\frac{c}{2V_a}q \end{bmatrix} \\ &+ \frac{1}{2}\rho V_a^2 S \begin{bmatrix} (-C_{D_{\delta_e}}\frac{u_r}{V_a} + C_{L_{\delta_e}}\frac{w_r}{V_a})\delta_e - \frac{1}{S}S_{\text{prop}}C_{\text{prop}} \\ 0 \\ \frac{c}{2V_a}q + (-C_{D_{\delta_e}}\frac{w_r}{V_a} - C_{L_{\delta_e}}\frac{u_r}{V_a})\delta_e \end{bmatrix} \end{aligned} \tag{2.40}$$

Since the airspeed V_a will be positive for any values of v_r^b during flight, we rewrite V_a as

$$V_a = V_{a,min} + \Delta V_a \tag{2.41}$$

Combining (2.40) and (2.41) we obtain

$$\begin{aligned} F_{\text{aero},1}(v_r^b, t) = & -\underbrace{\frac{1}{2}\rho S \begin{bmatrix} \sigma_1 & 0 & -\sigma_2 \\ 0 & -C_{Y_\beta} & 0 \\ \sigma_2 & 0 & \sigma_1 \end{bmatrix}}_{D(t)} \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} \\ & \underbrace{-\frac{1}{2}\rho S \begin{bmatrix} C_{D_\alpha} u_r w_r - C_{L_\alpha} w_r^2 + \frac{1}{S} S_{\text{prop}} C_{\text{prop}} V_a^2 + \eta_1 \Delta V_a \\ 0 \\ C_{L_\alpha} u_r w_r + C_{D_\alpha} w_r^2 + \eta_2 \Delta V_a \end{bmatrix}}_{d(v_r^b, t)} \end{aligned} \tag{2.42}$$

where

$$\begin{aligned} \sigma_1 &\triangleq \left(C_{D_0} + C_{D_{\delta_e}} \delta_e\right) V_{a,min} + C_{D_q} c q(t)/2 \\ \sigma_2 &\triangleq \left(C_{L_0} + C_{L_{\delta_e}} \delta_e\right) V_{a,min} + C_{L_q} c q(t)/2 \\ \eta_1 &\triangleq \left(C_{D_0} + C_{D_{\delta_e}} \delta_e\right) u_r - \left(C_{L_0} + C_{L_{\delta_e}} \delta_e\right) w_r \\ \eta_2 &\triangleq \left(C_{L_0} + C_{L_{\delta_e}} \delta_e\right) u_r + \left(C_{D_0} + C_{D_{\delta_e}} \delta_e\right) w_r \end{aligned}$$

■

For a stable flight $D(t) > 0$ and $v_r^{b\top} d(v_r^b, t) \geq 0, \forall\, t \geq 0$. In addition, when designing the observer we assume that under stable flight the nonlinear aerodynamic terms satisfy:

$$P \left[\frac{\partial d(v_r^b, t)}{\partial v_r^b}\right] + \left[\frac{\partial d(v_r^b, t)}{\partial v_r^b}\right]^\top P \geq 0, \quad \forall\, v_r^b \in \mathbb{R}^3,\ t \geq 0 \tag{2.43}$$

where $P = P^\top > 0$. Then the following property ([1]) holds:

$$(x - y)^\top P(d(x, t) - d(y, t)) \geq 0, \quad \forall\, x, y \in \mathbb{R}^3,\ t \geq 0 \tag{2.44}$$

Note that $D(t)$ and $d(v_r^b, t)$ depends on time-varying measurements. Consequently, Condition (2.43) must be satisfied for all measured signals, which are assumed to be smooth and bounded.

Proposition 2.2 (Nonlinear model-based relative velocity observer): Under the assumptions given in Section 2.5.2 and inequality (2.43), the nonlinear observer:

$$\begin{aligned} m\dot{\hat{v}}_r^b = & - S(\omega^b)\hat{v}_r^b + F_{\text{aero},1}(\hat{v}_r^b, t) + F_{\text{aero},2}(t) \\ & - m(R_b^n)^\top g^n - K_u h \left(h^\top \hat{v}_r^b - \hat{\gamma} u_r^m(t)\right) \end{aligned} \tag{2.45}$$

$$\dot{\hat{\gamma}} = K_\gamma \left(h^\top \hat{v}_r^b - \hat{\gamma} u_r^m(t)\right) \tag{2.46}$$

$$\hat{v}_w^b = (R_b^n)^\top v^n - \hat{v}_r^b \tag{2.47}$$

renders the equilibrium point $(\tilde{v}_r^b; \tilde{\gamma}) = (0, 0)$ globally exponentially stable (GES) if the observer gains are chosen as $K_u > 0$ and $K_\gamma = K_u u_r^m$. The vector $h = [1\ 0\ 0]^\top$ is a selection vector, $u_r^m(t) > 0$ for all t is the Pitot-static probe relative velocity measurement in the longitudinal direction, and R_b^n are the attitude estimates, and ω^b and v_g^n are the IMU and GNSS measurements, respectively. Notice that the airspeed measurement V_a is used in the expression for $F_{\mathsf{aero},2}(t)$ given by (2.26), whereas the airspeed estimate $\hat{V}_a = \|\hat{v}_r^b\|$ is used in the expression for $F_{\mathsf{aero},1}(\hat{v}_r^b, t)$ given by (2.25).

Comment on local versus global stability: The stability result is in practice local since the observer is based on the linear expressions (2.32)–(2.35) for the aerodynamic forces. The linear drag and lift coefficients $C_L(\alpha)$ and $C_D(\alpha)$ cannot describe nonlinear maneuvers such as stall, spinning etc.

Proof: Since the wind velocity estimate $\hat{v}_w^b$ is algebraically related to $\hat{v}_r^b$ by (2.47), it is only necessary to prove that the estimated states $\hat{v}_r^b$ and $\hat{\gamma}$ converge to their true values. Consider the translational dynamics of the relative velocity and rewrite the correction term in terms of the error states (2.24) and (2.45) such that:

$$\begin{aligned} m\dot{\tilde{v}}_r^b =& F_{\mathrm{aero},1}(v_r^b, t) - F_{\mathrm{aero},1}(\hat{v}_r^b, t) \\ & + K_u h \left(h^\top \hat{v}_r^b - \hat{\gamma} u_r^m(t)\right) \\ =& -\left(D(t) v_r^b - D(t)\hat{v}_r^b\right) - \left(d(v_r^b, t) - d(\hat{v}_r^b, t)\right) \\ & + K_u h \left(h^\top \hat{v}_r^b - u_r^m(t)\gamma + u_r^m(t)\tilde{\gamma}\right) \\ =& - D(t)\tilde{v}_r^b - \left(d(v_r^b, t) - d(\hat{v}_r^b, t)\right) \\ & + K_u h \left(u_r^m(t)\tilde{\gamma} - h^\top \tilde{v}_r^b\right) \end{aligned} \tag{2.48}$$

$$\dot{\tilde{\gamma}} = -K_\gamma \left(u_r^m(t)\tilde{\gamma} - h^\top \tilde{v}_r^b\right) \tag{2.49}$$

Consider the Lyapunov function candidate:

$$V = \frac{1}{2} m \left(\tilde{v}_r^{b\top} \tilde{v}_r^b + \tilde{\gamma}^2\right) \tag{2.50}$$

Hence,

$$\begin{aligned} \dot{V} &= \tilde{v}_r^b \left(D(t)^\top + D(t)\right) \tilde{v}_r^b + \tilde{v}_r^{b\top} \left(d(v_r^b, t) - d(\hat{v}_r^b, t)\right) \\ &+ K_u \left(\tilde{v}_r^{b\top} h u_r^m(t)\tilde{\gamma} - \tilde{v}_r^{b\top} h h^\top \tilde{v}_r^b\right) \\ &- K_\gamma \left(u_r^m(t)\tilde{\gamma}^2 - \tilde{\gamma} h^\top \tilde{v}_r^b\right) \\ &\leq - \begin{bmatrix} \tilde{\gamma} & \tilde{v}_r^{b\top} \end{bmatrix} Q(t) \begin{bmatrix} \tilde{\gamma} \\ \tilde{v}_r^b \end{bmatrix} \end{aligned} \tag{2.51}$$

by using (2.44) and

$$Q(t) := \begin{bmatrix} K_\gamma u_r^m & -K_\gamma & 0 & 0 \\ -K_u u_r^m & \rho S \sigma_1 + K_u & 0 & 0 \\ 0 & 0 & -\rho S C_{Y_\beta} & 0 \\ 0 & 0 & 0 & \rho S \sigma_1 \end{bmatrix} \tag{2.52}$$

where the expression for $\sigma_1 = (C_{D_0} + C_{D_{\delta_e}}\delta_e(t))V_{a,\min} + C_{D_q}cq(t)/2$ was derived above in the proof of Lemma 2.1. To assess whether $Q(t)$ is positive definite we consider the leading principal minors of the symmetric matrix $Q(t) + Q^\top(t)$. If the gains are chosen such that $4K_\gamma u_r^m(\rho S\sigma_1 + K_u) - (K_u u_r^m + K_\gamma)^2 > 0$ and since $\sigma_1 > 0$ and $C_{Y_\beta} < 0$ then it follows that $\dot{V} < 0$ for $\tilde{v}_r^b \neq 0$ and $\tilde{\gamma} \neq 0$. Finally, by invoking Theorem 4.10 in Khalil [49] the conditions for GES are easily verified. ■

Corollary 2.3. *If the wind velocity observer (2.45)–(2.47) is in cascade with an attitude observer where the equilibrium point of the error dynamics $\tilde{R}_b^n = R_b^n - \hat{R}_b^n = 0$ is GES, then the nonlinear wind velocity observer (2.45)–(2.47) with R_b^n exchanged with the estimate $\hat{R}_b^n$ is GES.*

Proof: From theorem 4.14 in Khalil [49], we know there exists a Lyapunov function $V_a(\tilde{R}_b^n) : \mathbb{R}^{3\times 3} \mapsto \mathbb{R}$ that satisfies

$$\dot{V}_a \leq -c\|\tilde{R}_b^n\|^2 \tag{2.53}$$

for some $c > 0$. With an attitude estimate, instead of a measurement, (2.45) should be replaced with:

$$\begin{aligned} m\dot{\hat{v}}_r^b = &- S(\omega^b)\hat{v}_r^b + F_{\text{aero},1}(\hat{v}_r^b, t) + F_{\text{aero},2}(t) \\ &- m(\hat{R}_b^n)^\top g^n - K_u h\left(h^\top \hat{v}_r^b - \hat{\gamma} u_r^m(t)\right) \end{aligned} \tag{2.54}$$

and the error dynamics instead becomes

$$\begin{aligned} m\dot{\tilde{v}}_r^b = &- D(t)\tilde{v}_r^b - \left(d(v_r^b, t) - d(\hat{v}_r^b, t)\right) \\ &+ K_u h(u_r^m(t)\tilde{\gamma} - h^\top \tilde{v}_r^b) + m\tilde{v}_r^{b\top}(\tilde{R}_b^n)^\top g^n \end{aligned} \tag{2.55}$$

The time derivative of the Lyapunov function:

$$\begin{aligned} \dot{V} \leq &- \begin{bmatrix}\tilde{\gamma} & \tilde{v}_r^{b\top}\end{bmatrix} Q \begin{bmatrix}\tilde{\gamma} \\ \tilde{v}_r^b\end{bmatrix} + m\tilde{v}_r^{b\top}(\tilde{R}_b^n)^\top g^n \\ \leq &-\lambda_{\min}(Q)\|\tilde{\gamma}\|^2 - \lambda_{\min}(Q)\|\tilde{v}_r^{b\top}\|^2 + mg\|\tilde{v}_r^b\|\|\tilde{R}_b^n\| \end{aligned} \tag{2.56}$$

where the time argument of Q has been omitted for notational simplicity. Consider the augmented Lyapunov function

$$W(\tilde{\gamma}, \tilde{v}_r^b, \tilde{R}_b^n) = V(\tilde{\gamma}, \tilde{v}_r^b) + \kappa V_a(\tilde{R}_b^n) \tag{2.57}$$

where $\kappa > 0$. Hence,

$$\begin{aligned} \dot{W} \leq &- \lambda_{\min}(Q)\|\tilde{\gamma}\|^2 - \lambda_{\min}(Q)\|\tilde{v}_r^{b\top}\|^2 \\ &+ mg\|\tilde{v}_r^b\|\|\tilde{R}_b^n\| - \kappa c\|\tilde{R}_b^n\|^2 \\ \leq &- \begin{bmatrix}\|\tilde{v}_r^b\| & \|\tilde{R}_b^n\| & \|\tilde{\gamma}\|\end{bmatrix} H \begin{bmatrix}\|\tilde{v}_r^b\| & \|\tilde{R}_b^n\| & \|\tilde{\gamma}\|\end{bmatrix}^\top \end{aligned} \tag{2.58}$$

where

$$H = \begin{bmatrix} \lambda_{\min}(Q) & -\frac{1}{2}gm & 0 \\ -\frac{1}{2}gm & \kappa c & 0 \\ 0 & 0 & \lambda_{\min}(Q) \end{bmatrix} \tag{2.59}$$

Hence, H is positive definite since κ can always be chosen such that $\kappa > g^2m^2/(4\lambda_{\min}(Q)c)$. ■

2.6 Aerosonde UAV Stability Requirements

The stability requirements (Proposition 1) of the observer (2.45)–(2.47) can appear difficult to evaluate and a case study is therefore presented using the Aerosonde UAV (with the model as given by Beard and McLane [6]). For the Aerosonde UAV the requirement $\sigma_1 > 0$ reduces to (both $C_{D_{\delta_e}}$ and C_{D_q} are zero):

$$0.03V_{a,\min} > 0 \tag{2.60}$$

which is clearly satisfied. Since $C_{Y_\beta} = -0.98$ the matrix $D(t) > 0$ for all $t \geq 0$. The Condition (2.43) can be rewritten as (using the results from the proof of Lemma 2.1):

$$\left[\frac{\partial d(v_r^b, t)}{\partial v_r^b}\right] + \left[\frac{\partial d(v_r^b, t)}{\partial v_r^b}\right]^\top := \rho S \begin{bmatrix} \zeta_1 & 0 & \zeta_2 \\ 0 & 0 & 0 \\ \zeta_2 & 0 & \zeta_3 \end{bmatrix} \geq 0 \tag{2.61}$$

where

$$\begin{aligned}\zeta_1 =& 2C_{D_\alpha} w_r + 4\frac{1}{S} S_{\text{prop}} C_{\text{prop}} u_r + 2\eta_1 u_r / V_a \\ & + 2\left(C_{D_0} + C_{D_{\delta_e}} \delta_e\right) \Delta V_a \\ \zeta_2 =& C_{D_\alpha} u_r - C_{L_\alpha} w_r + 2\frac{1}{S} S_{\text{prop}} C_{\text{prop}} w_r \\ & + \left(\eta_1 w_r + \eta_2 u_r\right) / V_a \\ \zeta_3 =& 2C_{L_\alpha} u_r + 4C_{D_\alpha} w_r + 2\eta_2 w_r / V_a \\ & + 2\left(C_{D_0} + C_{D_{\delta_e}} \delta_e\right) \Delta V_a\end{aligned}$$

Hence, the inequality (2.43) is satisfied with $P = I_3$ for all $t \geq 0$ iff $\zeta_1 \geq 0$, $\zeta_3 \geq 0$, and $\zeta_1 \zeta_3 - \zeta_2^2 > 0$.

For $\zeta_1 > 0$:

$$\begin{aligned}& 0.30w_r + 0.41u_r + 0.30\Delta V_a + 0.03u_r^2/V_a \\ & \qquad - (0.28 - 0.36\delta_e(t)) w_r V_a > 0\end{aligned} \tag{2.62}$$

This can be rewritten as

$$\begin{aligned}& 0.03u_r^2/V_a + 0.41u_r + 0.30\Delta V_a \\ & \qquad > 0.02w_r(1 - V_a) - 0.36\delta_e(t) w_r V_a\end{aligned} \tag{2.63}$$

Similarly for ζ_3 we have:

$$\begin{aligned}& 6.90u_r + 0.06\Delta V_a + 0.06w_r^2/V_a \\ & \qquad > -1.2w_r - (0.56 - 0.72\delta_e) u_r w_r / V_a\end{aligned} \tag{2.64}$$

Since δ_e and w_r are much smaller than u_r during a stable flight (2.63) and (2.64) holds during normal operation of the UAV. Throughout the simulations presented in Section 2.7, the values assumed by ζ_1 (with a $V_{a,\min} = 14$ [m/s]) fluctuates around 39.5 and never drops below 36.8. For ζ_3, the fluctuation is around 183 with a minimum of 177. The requirement $\zeta_1\zeta_3 - \zeta_2^2 > 0$ is difficult to analyze analytically but, the magnitude of ζ_2 never exceeds 20.2 and it is therefore evident that the requirement is fulfilled throughout the simulation. However, there exists combinations of u_r, w_r and δ_e, which can occur in other conditions that do not guarantee this condition holds. Because of the constraint (2.63) and the approximations the stability results are local for the Aerosonde UAV. Consequently, it is recommended to use the nonlinear observer only for flight envelopes accurately described by the aircraft model and stable flights.

2.7 Simulation Study

To assess the effectiveness of the proposed nonlinear wind observer, two different simulations have been conducted using Matlab Simulink. The simulations have been based on the complete model of the small aircraft system for the Aerosonde UAV presented in [6] including the autopilot module. The wind is modeled as a constant wind field with added turbulence. The turbulence is generated as white noise filtered through a Dryden model, an approach presented by [52] and also used by [6]. The Dryden transfer functions for the wind turbulence are given by

$$H_u(s) = \sigma_u \sqrt{\frac{2V_a}{L_u}} \frac{1}{s + V_a/L_u} \tag{2.65}$$

$$H_v(s) = \sigma_v \sqrt{\frac{3V_a}{L_v}} \frac{\left(s + V_a/(\sqrt{3}L_v)\right)}{(s + V_a/L_u)^2} \tag{2.66}$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_a}{L_w}} \frac{\left(s + V_a/(\sqrt{3}L_w)\right)}{(s + V_a/L_w)^2} \tag{2.67}$$

where σ_u, σ_v, σ_w and L_u, L_v, L_w are the turbulence intensities and spatial wavelengths along the vehicle frame axes. For the simulations the Dryden model has been implemented with a constant nominal airspeed $V_a = V_{a_0}$. The gust model used is for a low altitude, moderate turbulence gust with the parameters listed in Table 2.1. For assessing the simulation results, the focus will be on the estimates of the wind velocity and the air data. During simulations the observer measurements

Table 2.1: Dryden gust model parameters used in simulation.

altitude	50	m
L_u,L_v	200	m
L_w	50	m
σ_u,σ_v	2.12	m/s
σ_w	1.4	m/s
V_{a_0}	26	m/s

have been used with the noise from Appendix H and Chapter 7 in [6] on IMU measurements, Pitot static probe measurements and GNSS velocity measurements. For the GNSS velocity measurement noise a std. dev. of 0.05 m has been used. Since the noise on the estimate $\hat{\mathbf{v}}_w^b$ is proportional to the noise of the GNSS velocity measurement $\mathbf{v}^n$, the GNSS velocity measurement has been filtered through a simple observer. The observer gains are $K_\gamma = 4$ and $K_u = K_\gamma / u_r^m$.

2.7.1 Simulation study I

During the first simulation the aircraft autopilot control objectives are changed in steps. The aircraft control objectives start with an altitude of 50 m and and airspeed command of 26 m/s. After 30 seconds the course command control objective is increased by 10 degrees, where it stays for the next 30 seconds. At time 90 s, the altitude control objective is increased by 5 meters and once again decreased after 30 seconds. At time 150 s until time 180 s the speed control objective is increased from 26 m/s to 30 m/s. In the middle of the simulation after 100 seconds the airspeed measurement scaling factor γ steps from 1.0 to 1.1 to assess the scaling factor estimation capabilities of the observer. The wind velocity and wind velocity estimation is shown in Figure 2.3. The nonlinear wind observer is capable of providing accurate estimates for the wind velocity along the lateral axis of the aircraft. In the longitudinal and vertical directions the observer estimates appear to be slightly delayed.

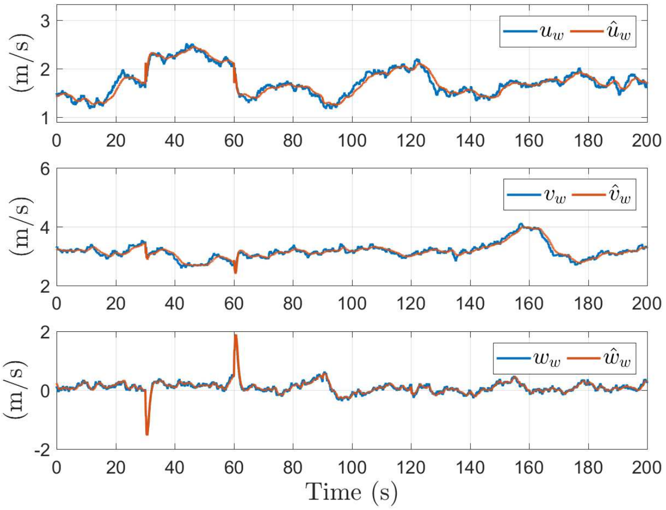


Figure 2.3: Three plots showing the true values of the wind along with their respective estimates from a simulation with a maneuvering UAV.

Figure 2.4 shows the air data ground truth and estimates values, calculated directly from the estimated relative velocity, along with the airspeed measurement scaling factor.

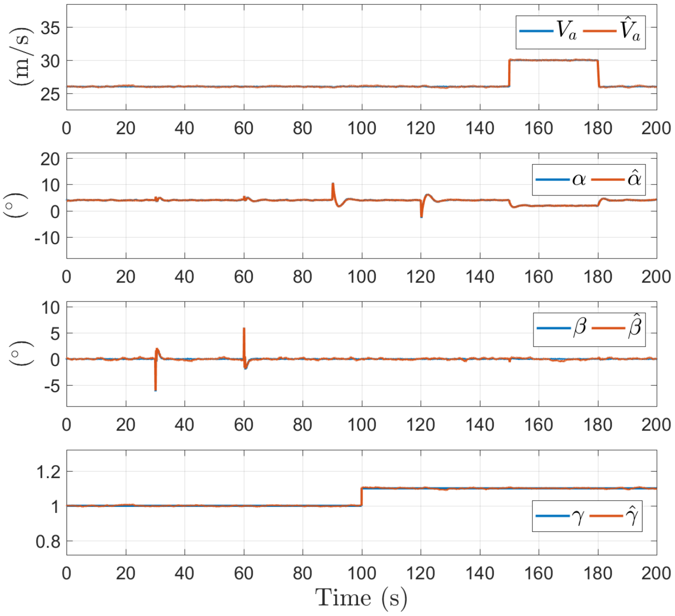


Figure 2.4: Three plots showing the AOA, SSA and airspeed measurement scaling factor true variables and their respective estimates from a simulation with a maneuvering UAV.

2.7.2 Simulation study II

To assess the nonlinear wind observer under actuator uncertainties, the second simulation includes a 10% mismatch between values of the aircraft parameters in the wind observer for the thrust force created by the propeller actuator and the true model values. This is emulated as

$$\mathbf{f}_{p,\text{model}} = 1.1 \cdot \mathbf{f}_{p\text{true}} \tag{2.68}$$

The simulation is conducted using the same autopilot control inputs as in Simulation I, but without the change in scaling factor. The simulation results are shown in Figures 2.5 and 2.6. The nonlinear wind observer displays an offset in the longitudinal wind velocity estimation, while the estimates in the lateral and vertical directions have only degraded mildly. The AOA estimate also appears to be affected by a small offset, whereas the SSA estimates displays similar performance to the former simulation.

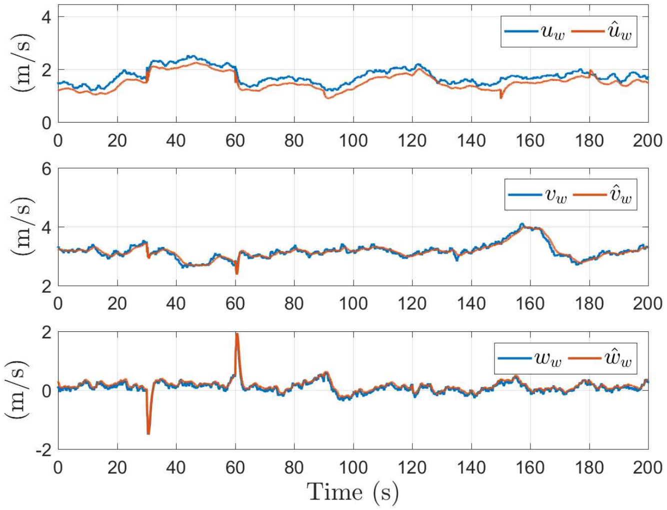


Figure 2.5: Three plots showing the true values of the wind velocity along with their respective estimates from a simulation with with a propulsion model uncertainty.

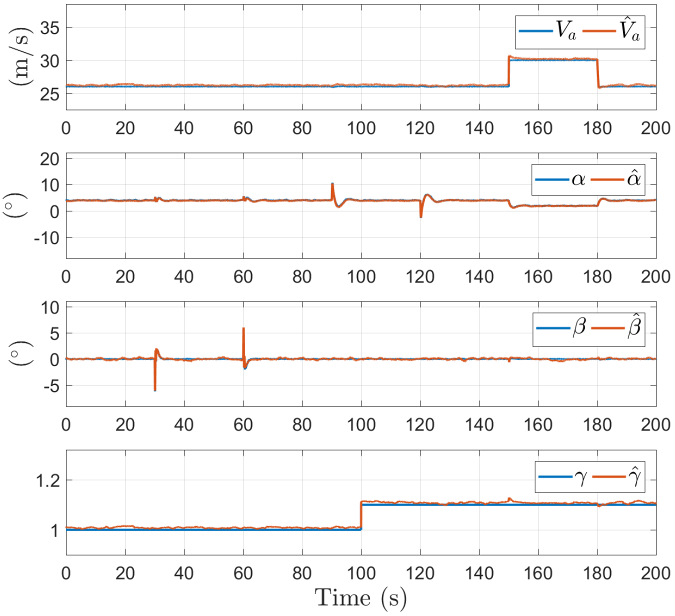


Figure 2.6: Three plots showing the AOA, SSA and airspeed measurement scaling factor true variables and their respective estimates from a simulation with a propulsion model uncertainty.

2.8 System Identification, Parameter Estimation and Modeling Accuracy

The observer was shown to work in simulation, however, the dynamics and behavior of the UAV during simulation is based on the same model as used by the observer. Successfully transferring the observer to flight requires that the model accurately captures the underlying physics. This depends on obtaining accurate estimates of the aerodynamic coefficients of the UAV model. The discipline of mathematically modeling the governing physics of a system based on observations of input and output is commonly referred to as system identification. For aircraft, this is done by numerically simulating the air flow, wind tunnel experiments, or flight experiments with predefined trajectories designed to excite relevant parameters. Given the highly nonlinear nature of aerodynamics combined with measurement noise, it can be difficult to obtain an accurate model for light-weight aircraft such as the Skywalker X8 that are highly susceptible to the influence of wind. Gryte et al. [35] presented a case study on establishing a six degrees-of-freedom aerodynamic model of the Skywalker X8. The aerodynamic forces are modeled as linear with the exception of the influence of AOA and SSA on drag, which is modeled as a second order polynomial, and the model structure is therefore different from the one used in this chapter. Gryte et al. did a comparison between model parameters found using wind tunnel experiments and the 3D panel software XFLR5. Even though consistencies were found across the models, there was also some discrepancies. For a treatment of system identification and parameter estimation for aircraft, see Klein and Morelli [50].

2.9 Chapter Summary

In this chapter a nonlinear wind observer for a UAV was presented. The wind observer combines a model of the aircraft with a GNSS receiver, an attitude reference, an airspeed sensor, and a Pitot-static probe. The nonlinear wind observer provides estimates of both the wind velocity and the relative velocity, from which the AOA and SSA are computable. The nonlinear wind observer developed does not have any requirements of PE of the aircraft. The nonlinear wind observer has been proven to be exponentially stable and is verified through simulation.

Chapter 3

A Machine Learning Approach to Estimating Air Data Using Distributed Pressure Sensors

This chapter presents a method for estimating the air data parameters for a small fixed-wing UAV using distributed low-cost MEMS-based pressure sensors embedded in the surface of the aircraft. The pressure measurements are used in combination with machine learning to estimate AOA, SSA, and airspeed. Two ML algorithms based on NNs and LR are implemented, tested, and assessed using data collected from wind tunnel experiments and a flight test and the results are compared to a benchmark flight test. The presented estimation method is build using low-cost, off-the-shelf components, and an advantage of the method is the flexibility of the sensor placement. Section 3.1 states the problem and the assumptions behind the method and Section 3.2 presents the two ML approaches to estimating the air data parameters from pressure measurements. Section 3.3 contains a description of the experimental setup and Section 3.4 presents the obtained results. The presented results are based on both wind tunnel tests and a flight test with the Skywalker X8 UAV.

3.1 Main Principles

For a UAV the relative velocity can be expressed in the body coordinate frame as the difference of the ground velocity and the wind velocity:

$$v_r^b = v_g^b - v_w^b \tag{3.1}$$

where $v_r^b = [u_r, v_r, w_r]^\top$ is the relative velocity vector, $v_g^b = [u, v, w]^\top$ is the velocity over ground vector of the UAV, and $v_w^b = [u_w, v_w, w_w]^\top$ is the wind velocity vector. The goal is to use pressure sensor measurements to estimate the airspeed V_a, AOA

α, and SSA β defined as:

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{3.2}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{3.3}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \tag{3.4}$$

As described in Chapter 1, both the airspeed V_a and AOA α are directly related to the lift and drag forces and knowing these variables is valuable in controlling the aircraft. The relationship between the air data parameters and the pressure distribution across a cambered airfoil and a fuselage is not trivial to model. For an airfoil of a given shape at a given AOA, the resultant aerodynamic force, F_R, is dependent on five different parameters: The airspeed V_a, the freestream density ρ_∞, the viscosity of the fluid μ_∞, the size of the body by a reference length c, and the speed of sound a_∞[1]. By application of the Buckingham pi Theorem, dimensional analysis allows expressing the aerodynamic force in terms of a dimensionless force coefficient $C_R = F_R/\frac{1}{2}\rho_\infty V_a^2 c^2$, as a general function of only two variables, the freestream Reynolds number Re$= \rho_\infty V_a c/\mu_\infty$ and the Mach number $M = V_a/a_\infty$. For a treatment of dimensional analysis for an airfoil, the reader is directed to Anderson [5]. Furthermore, for Mach numbers below 0.3, the effects of compressibility are negligible and the flow can be considered incompressible. Since most small UAVs operate in Mach numbers well below 0.3, it is assumed that the use of Mach numbers for the dimensional analysis can be neglected. Extending dimensional analysis from two dimensional airfoil to a three dimensional aircraft requires the SSA. Hence, dimensional analysis allows for stating the force coefficient C_R for an aircraft of a given shape as a general function of only the Reynolds number, the AOA, and the SSA, where the expression for the Reynolds number also contains the airspeed.

The theoretical basis for the method presented in this chapter revolves around the relation from air data to force coefficient and pressure distribution. In general, the pressure distribution and flow over a given 3-dimensional object moving through a fluid is highly nonlinear and very complicated to model. Unless the object has certain favorable geometric properties, finding the pressure and flow around the object requires solving the Navier-Stokes equations numerically. In this chapter it is assumed that given a sufficient set of pressure measurements from a set of measurement points distributed on the surface of a UAV, it is possible to inversely map this spatially sampled pressure distribution back to the air data.

The pressure measurements obtained by the sensors have been used differentially with respect to a designated reference sensor, i.e. a selected sensor measurement has been subtracted from all the other sensor measurements. Biases and sensor noise will be addressed later, but for this discussion we assume perfect sensor

[1] Actually the compressibility of the fluid, but that value is representable by the speed of sound.

measurements. Instead of using the pressure measured directly, which is a function of altitude, temperature, and humidity, the differential pressure instead directly relates the measured pressure to the pressure distribution on the aircraft. The price is the reduction of the sensor measurements by one dimension, but should enable - given a proper training data set and a suitable ML structure - to make the ML method robust with respect to changes in ambient conditions. With the differential sensor approach, there is for a n-sensor setup in effect only $n-1$ differential pressure measurements available and throughout the rest of this chapter, they will be referred to using their sensor numbers as p_{s_1} to $p_{s_{n-1}}$.

Finally, it is assumed that the pressure differences stemming from hydrostatic differences between sensors are negligible. For example, if the UAV is engaged in a banked turn, one wing will be higher than the other. For small UAVs the pressure difference between the sensors on that wing relative to the sensors on the other wing will be relatively small and therefore not affect the results, i.e. the hydrostatic term in Bernoulli's equation can be neglected due to the low density of air and the small differences in height.

3.2 Machine Learning Modeling Approach

The two ML approaches used in this chapter are LR and NNs. The results obtained using both approaches are presented, compared, and discussed in the following sections. However, the concept used in this paper is not restricted to these ML algorithms and an algorithm such as support vector machines should for example be able to replace the mapping from pressure measurements to air data parameter estimates, although the quality of the results will depend on algorithm, training method, quality of training data, etc.

3.2.1 Linear Regression (LR)

LR has been called the "work horse" of ML [68]. Contrary to what the name implies, LR is not limited to modeling linear functions. The algorithm minimizes a least-squares error cost function to map a linear combination of input variables to an output, but these input variables can be chosen to be nonlinear functions of the basis input. Augmenting the basis input with nonlinear functions of the input is known as basis function expansion. The high complexity of the physics relating the air data to the pressure distribution over the aircraft does not allow for a first principles approach to choosing the basis function expansions. In this chapter, three generic basis function expansions have been tested, compared, and verified by experiments. Basis function expansions can potentially provide a higher accuracy to the linear regression method, but comes with the cost of a higher computational demand, both when computing the weights, but also during runtime. Another common consideration for model choice is the principle of parsimony, where the model ideally use the fewest possible parameters to adequately represent the input-output relations. The model choice is based on the fundamental trade-off between underfitting and overfitting the data and thereby balancing model bias with model variance [16]. For the LR approach presented, the basis input consists of the differential pressure measurements. Three basis function expansions will be considered, including two polynomial basis functions. Using low-level polynomial basis function expansions is a common modeling approach in LR [8, 68]. The basis input is the vector of differential pressure measurements:

$$[p_{s_1},\ p_{s_2}, \dots,\ p_{s_{n-1}}]$$

The first basis function expansion uses the first order cross terms between the differential pressure measurements. Using polynomial cross terms is a widely used in LR modeling (see [45], where the cross terms are referred to as interactions):

$$[p_{s_1}p_{s_2},\ p_{s_1}p_{s_3}, \dots,\ p_{s_2}p_{s_3},\ p_{s_2}p_{s_4}, \dots,\ p_{s_{n-2}}p_{s_{n-1}}]$$

where the p_{s_i} represents the i'th sensor differential pressure measurement. For a differential n sensor setup, this corresponds to an augmentation of k vector entries, where k is given by the binomial coefficient as:

$$k = \binom{n-1}{2} = \frac{(n-1)!}{2!(n-3)!} \tag{3.5}$$

The second expansion is a square of the input measurements:

$$[p_{s_1}^2,\ p_{s_2}^2, \dots,\ p_{s_{n-1}}^2]$$

The third and last expansion is the input measurements cubed:

$$[p_{s_1}^3,\ p_{s_2}^3, \dots,\ p_{s_{n-1}}^3]$$

There are obviously an infinite amount of possible basis function expansions and there are probably ones that could provide better results than the expansions chosen. However, the chosen basis function expansions are generic choices and should serve as a first approximation for testing the presented concept and could instead be augmented with more tailored functions, if available.

When presenting the LR results, the following naming abbreviations have been used to keep the results concise:

- **B**: Basis input vector.
- **X**: First order cross term expansion.
- **Q**: Quadratic expansion.
- **C**: Cubic expansion.

Following this convention, **BXC** will refer to a the basis input vector expanded by first order cross terms and the cubic terms, where **BQC** will refer to the basis input vector expanded by the quadratic and cubic terms, etc.

For the wind tunnel experiments, the data used for LR has been partitioned into a 85% training set and a 15% test set, and the data has been randomly divided into these two sets. The LR weights are found using the training data and the test set is used in providing a measure of the performance of the method. The influence of the training and test partition sizes were found to be negligible when kept in the ranges of 60/40% to 85/15% and the partition sizes of 85/15% was chosen to allow a large set of training data. The stochastic element from randomly dividing the data into training and test sets will result in different outcomes for every set of linear regression weights calculated. Therefore, whenever numerical results are presented, a mean of ten different training set results have been used to reduce the impact of randomness.

3.2.2 Artificial Neural Networks (NNs)

The structure of interconnected neurons in artificial NNs provide a generic method for approximating continuous functions from observational data. The NNs presented in this chapter have been designed using the MATLAB nftool toolbox. The NNs are trained using the Levenberg-Marquardt (LM) backpropagation algorithm (Marquardt [60]), since it is very efficient for networks consisting of a few hundred weights or less [36]. The LM algorithm minimizes a least-squares cost function by blending gradient descent with Gauss-Newton's algorithm, retaining the favorable stability properties of gradient descent and the speed of the Gauss-Newton algorithm. The LM algorithm uses the Gauss-Newton Jakobian-based method of estimating the Hessian, but with an added positively scaled identity matrix

$$H \approx J^\top J + \mu I \tag{3.6}$$

where H is the Hessian, J is the Jakobian, and μ is a positive scalar called the combination coefficient. The weights are updated using the update rule

$$w_{k+1} = w_k - H_k^{-1} J_k e_k \tag{3.7}$$

where k is the training iteration index, w are the weights, and e is the model output error vector. The combination coefficient can be viewed as a blending factor between gradient descend, if μ is large, and the Gauss-Newton algorithm, if μ is close to zero. The blending factor μ is reduced for each step that reduces the cost function output, and increased for each step that increases the cost function output, thereby allowing the LM algorithm the stability of gradient descent, but with faster convergence.

The NNs have been chosen with the hidden layer using the tan-sigmoid function and the output layer using a linear function [36]. The NNs have been designed with one or two hidden layers and a varying number of hidden neurons. It is known from NN approximation theory that with a sufficient number of hidden neurons, continuous functions can be approximated to arbitrary accuracy. The optimal structure depend on the training data and the chosen structures will be validated and compared through results. The three different air data parameters (airspeed, AOA, and SSA) will be estimated each by a separate NN and the NNs presented will therefore all have a single output neuron. Figure 3.1 shows the structure of a NN for estimating the airspeed from the measurements of $n-1$ pressure sensors using 10 neurons in a single hidden layer.

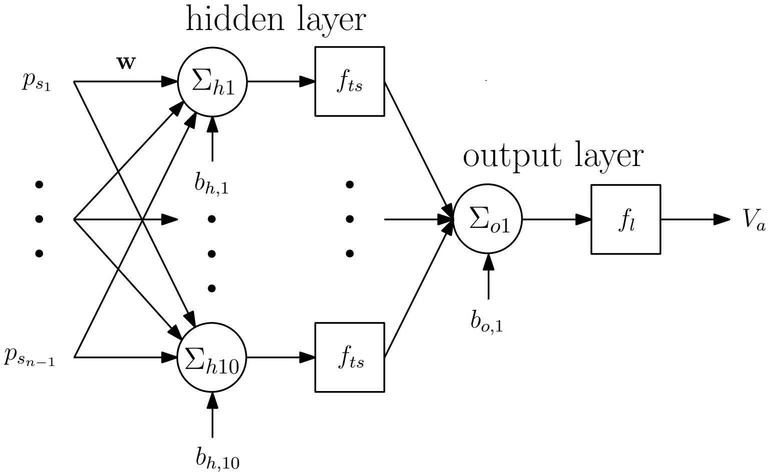


Figure 3.1: An illustration of a NN that provides estimates of the airspeed, V_a. The NN is here illustrated with $n-1$ differential pressure measurements as input, a single hidden layer with 10 tan-sigmoid function neurons f_{ts}, and a linear function output layer that outputs the airspeed estimate.

Since all the presented NNs are trained using the same algorithm and all have a single output neuron, the only variables will be the number of hidden layers and the number of neurons in each hidden layer. As with the linear regression results, a naming abbreviation has been employed to keep the results concise. $\mathbf{F}_k$ is used to denote the first hidden layer containing k neurons and $\mathbf{S}_l$ denotes the second hidden layer containing l neurons. Hence, $\mathbf{F}_8\mathbf{S}_0$ denotes a NN with a single hidden layer with 8 neurons and $\mathbf{F}_{12}\mathbf{S}_8$ denotes a NN with two hidden layers with respectively 12 and 8 neurons in first and second hidden layer.

For the wind tunnel experiments using NNs, the data has been partitioned into 70% training data, a 15% validation data, and a 15% test data. The data has randomly been divided into these three sets. Other possible data partition sizes, including 60/20/20% and 50/25/25%, were also tested, and the partition sizes were found to have a very small influence when kept in this range. Just as for the linear regression approach, dividing the data randomly results in varying results for each new NN trained and a mean of ten NN results are therefore provided whenever numerical results are provided and both training and test set results are provided.

3.3 Experimental Setup

The results in this chapter are obtained using a Skywalker X8 Flying Wing UAV as detailed in Appendix A. The presented method utilizes an array of low-cost pressure sensors strategically embedded in the surface of the UAV. The method is not dependent on a specific type of pressure sensor, but the accuracy of the chosen sensor will affect the results and so the sensor choice will be a trade-off between cost, accuracy, weight, size, power consumption, robustness, and ease-of-implementation. For the Skywalker X8 prototype implementation, the objective is a proof of concept, and the BMP280 sensor by Bosch was chosen with the breakout board designed by Adafruit. The Bosch BMP280 is a MEMS-based digital pressure and temperature sensor that offers a decent accuracy and a small footprint for a low cost, which makes it ideal for this application. The BMP280 has an absolute accuracy of ±100 Pa and a relative accuracy of ±12 Pa, a temperature range of -40 to 80 °C, and is in the highest resolution capable of logging at 26.3 Hz. The difference in absolute accuracy and relative accuracy stems from a slowly varying bias. A 24 hour test was conducted, where the bias for 16 sensors only varied minimally and it was therefore concluded that the BMP280 biases can be assumed as constant throughout a UAV flight and as a counter measure, these biases have been removed from the tests. This was done in a pre-flight calibration by finding the biases from data where there was no wind velocity on the Skywalker X8 as the difference between the mean measurement of a specific sensor and the mean measurement of all the sensors. This calibration method has been employed for all the data used in this chapter. Hence, the BMP280, when used in a differential setup, is assumed to have an accuracy of ±12 Pa. Throughout the wind tunnel test the logging frequency was 7 Hz and for the flights it was increased to 20 Hz. The BMP280 comes with built-in low pass filters, but these filters have not been employed, since an objective of the presented method is to be able to estimate air data parameters in high dynamics.

A total of 16 BMP280 sensors have been embedded in the surface of the Skywalker X8 used in all the tests. The sensors are connected to an Arduino Mega 2560 through Serial Peripheral Interface (SPI). The wire-based communication usable for a MEMS-based sensor relieves the conventional setup of rubber tubes connected to a pressure scanner, which will simplify the implementation. The Arduino Mega 2560 collects the pressure and temperature readings and outputs the readings through serial to USB to either a laptop (the setup used the wind tunnel tests) or an ODROID-XU4 that logs the data (the setup employed in the flight tests). The 16 sensors have been distributed in a symmetric pattern with 5 sensors on a cross section of each wing that is parallel to the (longitudinal) xy-plane. The remaining 6 sensors have been embedded in the nose of the Skywalker X8 in a pattern roughly resembling a circle. Figure 3.2 shows the implementation of the pressure sensors on the wings, where the sensor installment has been attempted to result in as small an intrusion as possible to the aircraft surface. The wing is shown before a layer of film is applied to reduce the roughness of the surface. After applying the layer of film, small rectangles were cut in the film to leave the BMP280 sensors open to the environment. Figure 3.3, shows the placement of all BMP280 sensors on the portside of a 3D model of the Skywalker X8.

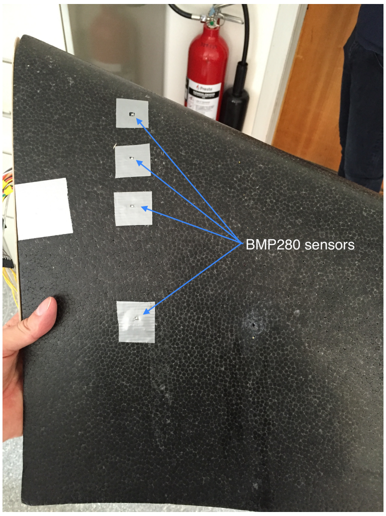


Figure 3.2: A top view of the Skywalker X8 starboard wing showing four BMP280 pressure sensors.

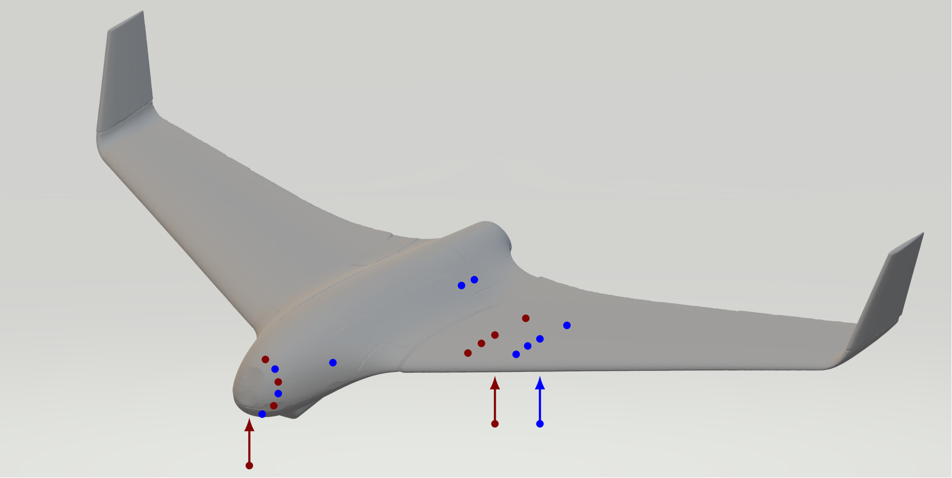


Figure 3.3: Sensor placement illustrated on the Skywalker X8 UAV. The red dots correspond to the BMP280 sensors and the blue dots correspond to the DSA3217 pressure scanner measurement points described further down. The dots with connected line and arrow denotes a sensor or measurement point, placed on the underside. Note that besides the two BMP280 middle nose sensors, the remaining sensors and measurement points are only shown on the port side, however, the placement is symmetric on the starboard side.

3.3.1 Wind tunnel tests setup

A part of the development and verification of the results in this chapter is based on wind tunnel testing. The tests were conducted in the BLTW Slovak University of Technology (STU) wind tunnel in Bratislava [44]. The BLTW STU wind tunnel allows testing in both laminar and turbulent wind flow depending on the section of the tunnel used. For the purpose of the method presented in this paper, laminar wind flow was chosen and all results presented are from tests conducted in the designated laminar flow area of the tunnel. The BLTW STU wind tunnel is 14.6 m long and has a cross section of 2.6 x 1.6 m.

For the wind tunnel tests a PTU-D48 pan-and-tilt unit by FLIR Systems was employed as a part of the mount of the Skywalker X8 in order to control AOA and SSA. The PTU-D48 offers a precision up to 0.006° and is controlled through a computer, which enabled altering the AOA and SSA during the tests without having to stop/start the wind tunnel for every single AOA and SSA. Since the wind tunnel has a few minutes transition time before reaching a steady state airspeed, this allowed for a much faster data collection rate. A picture of the wind tunnel setup is shown in Figure 3.4. The data from the PTU-D48 and the BMP280 sensors were collected and synced using DUNE: Unified Navigation Environment, developed by the Underwater Systems and Technology Laboratory [70].

The BLTW STU is equipped with three Scanivalve DSA3217 pneumatic pressure scanners. The DSA3217 pressure scanners measures with a frequency of 10 Hz and offer a ±1.25 Pa full scale long term accuracy, corresponding to an accuracy improvement compared to the BMP280 of almost a factor 10. The DSA3217 pressure scanners have been connected through rubber tubes to thin cobber pipes embedded in the Skywalker X8, flush with the surface. The pressure scanners have been used to measure the pressure at the points at the tips of the cobber pipes at the surface of the Skywalker X8. The high accuracy pressure scanners allows insight into the impact of sensor accuracy on the estimation method by comparing results obtained using the high accuracy DSA3217 pressure scanners to the results obtained using the lower accuracy BMP280s. Since it is not possible to place both types of pressure sensors directly on top of each other and to avoid cross influence between sensors, the DSA3217 pneumatic cobber tubes have been placed in different positions from the BMP280s and furthermore, a total of 22 pneumatic sensors has been placed on the Skywalker X8 and the placement is shown in Figure 3.3. It should be noted that since the placement and numbers of sensors will influence the results, the comparison between DSA3217 pneumatic and MEMS-based pressure sensors is not ideal, but should still serve to give insight into the advantage of a higher number of higher accuracy sensors on the presented method. Table 3.1 lists the extra equipment used in the wind tunnel tests along with its description.

The wind tunnel tests were conducted the 7th and 8th of September, 2016. The tests encompassed six different airspeeds: 11.01 m/s (T = 27.4 °C, $\rho = 1.1573$ m/kg^3), 12.09 m/s (T = 27.4 °C, $\rho = 1.1573$ m/kg^3), 13.12 m/s (T = 27.5 °C, $\rho = 1.1568$ m/kg^3), 14.14 m/s (T = 27.6 °C, $\rho = 1.1564$ m/kg^3), 15.21 m/s (T

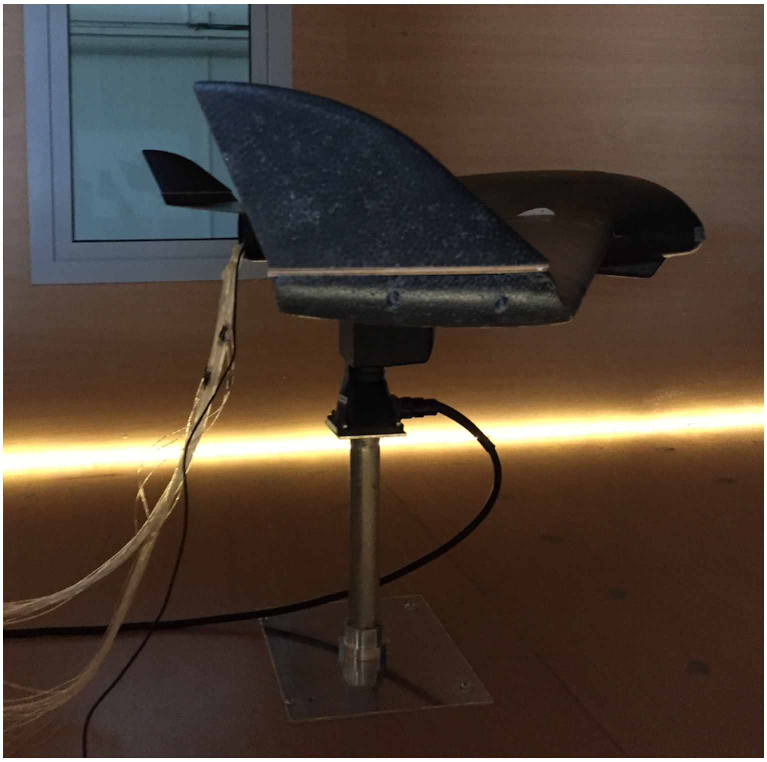


Figure 3.4: A sideways view of the tunnel setup with the Skywalker X8 mounted on the PTU-D48.

= 27.6 °C, $\rho = 1.1559$ m/kg^3), 16.23 m/s (T = 27.7 °C, $\rho = 1.1555$ m/kg^3), and 17.32 m/s (T = 27.8 °C, $\rho = 1.1555$ m/kg^3). At high airspeeds, the Skywalker X8 looked to be under some flutter and a high static wing load. For that reason, no tests with airspeed higher than 17.32 m/s were attempted, although the Skywalker X8 is usually flown in airspeeds up to 25 m/s. The inability to cope with high airspeeds in the wind tunnel are assumed to primarily be attributed to two different factors. Firstly, the Skywalker X8 blocks out a sizeable amount of the tunnel cross section, this will result in the airspeed being higher around the aircraft because of the blockage effect. Secondly, the free stream turbulence is approximately 5% and since the shear stress in turbulent flow is higher than in laminar [6], this results in higher drag on the X8.

The AOA and SSA values were chosen to span a large range of different values, since ML algorithms are better suited for interpolation than extrapolation. The Skywalker X8 was therefore also tested in stall conditions. The nonlinear relation between lift coefficient and AOA for the stall regions, is assumed to also influence the pressure distribution in a nonlinear fashion, making the ML modeling more difficult. The AOA values tested for extend up to 35 °, which is beyond the AOA values normally identified with regular flight. AOA values in this range are usually

Equipment	Description
3 DSA3217 pressure scanners	A high accuracy pressure sensor used for comparison with the lower accuracy BMP280 setup
22 1.5 cm long thin copper pipes	Embedded in the surface of the Skywalker and connected to the DSA3217 pressure scanners by rubber tubes
Laptop	Runs DUNE and connected to the PTU-D48 and the Arduino Mega 2560 which sends the collected BMP280 pressure measurement
DUNE: Unified Navigation Environment software	Logs the PTU-D48 output and time stamps together with the BMP280 measurements and timestamps
PTU-D48	Used to adjust and accurately measure the AOA and SSA of the Skywalker X8 during tests
Wind tunnel PC	Logs the DSA3217 measurements

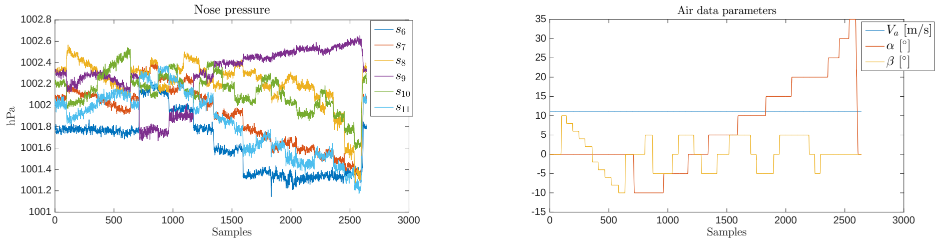
Table 3.1: The extra equipment used in the wind tunnel tests that is not a part of the flight test setup described in Section 3.3.2 and Appendix B.

only associated with special maneuvers such as deep stall landings, or other agile maneuvers or high turbulence. The intended purpose of the air data parameter measurement system should be considered when selecting the training data range for AOA, SSA and airspeed. Given the discussion in Section 3.1, it is also important to have data spanning a large range of Re numbers, and since the Re number is directly proportional to the airspeed, this was attempted fulfilled by varying the airspeed.

The raw pressure data for the Skywalker X8 nose BMP280 sensors and the corresponding air data parameters for the test with airspeed 11.01 m/s are shown in Figure 3.5. It is worth noticing that some of the sensors appear to have a very small response to changes in SSA while other sensors likewise appear to largely not be affected by the changes in AOA. This is related to the geometry of the Skywalker X8 relative to the wind and indicates that these sensors would be less valuable in estimating the corresponding parameter compared to the sensors with a larger change in response.

3.3.2 Flight test setup

For the flight test, the same Skywalker X8 as employed in the wind tunnel tests was used. However, since flight does not allow for connection by cable, the test setup was different from the tunnel test setup in a few ways. The Scanivalve DS3217 pressure



a The Skywalker X8 nose pressure measured by the six BMP280s embedded in surface.

b The air data parameters, where α and β are the angles of the PTU-D48 pan-and-tilt unit.

Figure 3.5: The raw data from the wind tunnel test with an airspeed of 11.01 m/s.

scanners have not been possible to use, since they are too heavy and cumbersome for the Skywalker X8. The 16 embedded BMP280 pressure sensors are therefore the only embedded pressure sensors used during flight. The BMP280 sensors are read using an Arduino Mega 2650 and are sent through a serial connection to an ODROID-XU4 where it is logged along with GNSS and IMU data with precise timestamps using a synchronization board developed by Albrektsen and Johansen [3]. As a source of ground-truth air data parameters for ML and testing, the Micro Air Data System by the Aeroprobe Corporation has been used. The Micro Air Data System consists of a 5-port air data probe connected through rubber tubes to a small pressure scanner. The Micro Air Data System has an accuracy of ±1° on flow angles and has a total flow velocity accuracy <1 % or 1 m/s (whichever is larger). The Micro Air Data System air data probe is mounted on the nose section of the Skywalker X8. The mounting mechanism can result in a slight misalignment between the probe and the aircraft axes from which the AOA and SSA are defined with respect to, and this should be compensated for. Another error source is that the Aeroprobe saturates beyond ±21° for AOA and SSA and any measurements outside this range will be truncated to ±21°. This will have a negative influence when using ML to estimate AOA and SSA values above 21°, and will also inhibit result assessments for angles outside of this range.

The flight test was conducted on the 20th of September 2016 on a field in Udduvoll, Norway. For the period of the flight, there was a temperature of approximately 16°C. The data set begins at the catapult launch of the Skywalker X8 and ends directly after landing. During the flight that lasts just below 37 minutes, the pilot operated the UAV in a large square with some changes in pitch and altitude. For the last part of the flight, the pilot engaged in some high dynamic maneuvers. The flight has been divided into two separate segments. The first five minutes of the flight has been separated from the remaining 31 minutes and will throughout the test be used as a flight benchmark test in order to assess the ML algorithms capabilities in estimating air data parameters for a flight scenario.

3.4 Results

This section contains the results obtained using the ML approaches on both wind tunnel test data and flight data. First, the BMP280 wind tunnel data is used to train and test the NN and LR approaches to see if the ML methods are capable of modeling the wind tunnel data input and output relations. Afterwards, a comparison with the more accurate DS3217 follows. The main benchmark test of the ML method capabilities will be a five minute test segment of flight with the Skywalker X8. The wind tunnel trained ML algorithms will be tested on this data to assess whether it is feasible to use the obtained wind tunnel data in order to train the ML algorithms for estimating air data parameters during flight. Afterwards, ML algorithms trained using the other segment of flight will be tested on the five minute benchmark flight and the results will be compared and discussed. Then, the impact of choice of sensor configuration on the accuracy of the results will be evaluated, as well as using a pseudo Re number as an extra input.

3.4.1 Wind tunnel test results

The focus of this section will be on the results obtained using the BMP280 sensors on the wind tunnel data and whether the ML algorithms are capable of modeling the air data parameters from the measured pressure. Results obtained with the BMP280 data using different LR and NN configurations are displayed in Table 3.2 and 3.3, respectively. The tables contain the RMSE for the air data parameter estimates from both the training data and the test data, however, the validation data results from the NNs have been left out.

	Linear Regression				
	B	**BX**	**BQ**	**BC**	**BXQC**
$\check{V}_a$ [m/s]	0.4519	0.1541	0.3005	0.3280	0.1465
$\bar{V}_a$ [m/s]	0.4541	0.1558	0.2987	0.3260	0.1505
$\check{\alpha}$ [°]	1.2490	0.4833	0.8252	0.8967	0.4371
$\bar{\alpha}$ [°]	1.2650	0.4889	0.8333	0.9044	0.4704
$\check{\beta}$ [°]	1.0462	0.6079	0.8033	0.7972	0.5950
$\bar{\beta}$ [°]	1.0647	0.6384	0.8073	0.8088	0.6094

Table 3.2: The results obtained using LR with different basis function expansions on the wind tunnel data. Vee denotes the training set RMSE and bar denotes the test set RMSE.

Looking into the numerical LR results, it appears that the basis function expansion of adding first order cross terms is the single function augmentation that yields the best numerical results. However, it does also expand the input vector with 105 extra entries, since

$$\text{number of extra entries} = \binom{n}{k} \tag{3.8}$$

	Neural Networks					
	$\mathbf{F}_5\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_0$	$\mathbf{F}_{15}\mathbf{S}_0$	$\mathbf{F}_{20}\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_5$	$\mathbf{F}_{10}\mathbf{S}_{10}$
$\check{V}_a$ [m/s]	0.1369	0.1171	0.0991	0.0877	0.0829	0.0498
$\bar{V}_a$ [m/s]	0.1391	0.1239	0.1060	0.0986	0.0938	0.0689
$\check{\alpha}$ [°]	0.3950	0.3522	0.3491	0.3275	0.3303	0.3286
$\bar{\alpha}$ [°]	0.4086	0.3860	0.3799	0.3675	0.3716	0.3642
$\check{\beta}$ [°]	0.5605	0.4830	0.4456	0.4138	0.3942	0.3690
$\bar{\beta}$ [°]	0.5738	0.5200	0.5018	0.4650	0.4608	0.4305

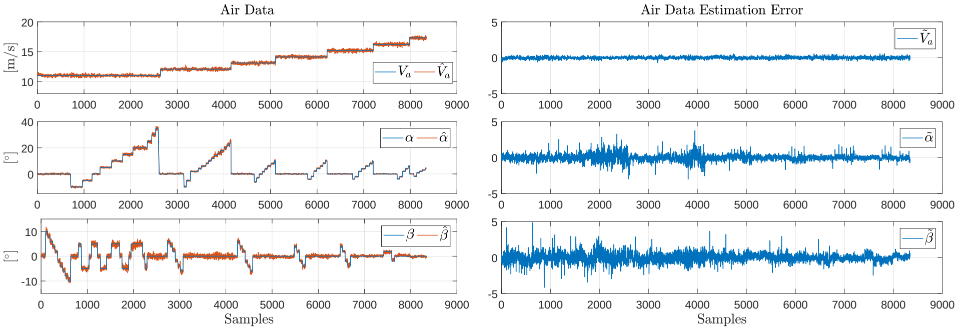
Table 3.3: The results obtained using NNs with different structures on the wind tunnel data. Vee denotes the training set RMSE and bar denotes the test set RMSE.

where n=15 and $k = 2$, whereas the other feature expansions only increase the vector size with an additional 15 entries each. The remaining basis function expansions yield similar results. The best numerical results are obtained for the basis function expansion **BXQC**, but the RMSE values are relatively close to the results obtained expanding only with the first order cross terms. The choice of LR basis function expansion will therefore be a trade-off between accuracy and computational cost, but for this data set it appears that the chosen function expansions leave a limit on the level of obtainable accuracy when compared to the NNs.

The results obtained using NNs are from an accuracy perspective superior to the LR results. The estimate RMSE decreases with the increase in the NN structure complexity which might hint that more neurons and more layers provide more accurate estimates. It appears that the difference between training and test set RMSE grows with NN complexity, which could indicate a small degree of overfitting for the more complex NN structures, as expected.

Figure 3.6 consists of two plots of the results obtained using LR and Figure 3.7 consists of the two corresponding plots using NNs. These plots contain the ground truth values along with the estimates using the complete data set, i.e. both training data and test data (and validation data for the NN), and the error between ground truth values and estimates. For the LR results, it appears that the accuracy of the airspeed is relative constant, but for NN results, the accuracy is slightly higher for lower airspeeds. For higher AOA and SSA, both methods exhibit less accurate estimates compared to lower AOA and SSA values.

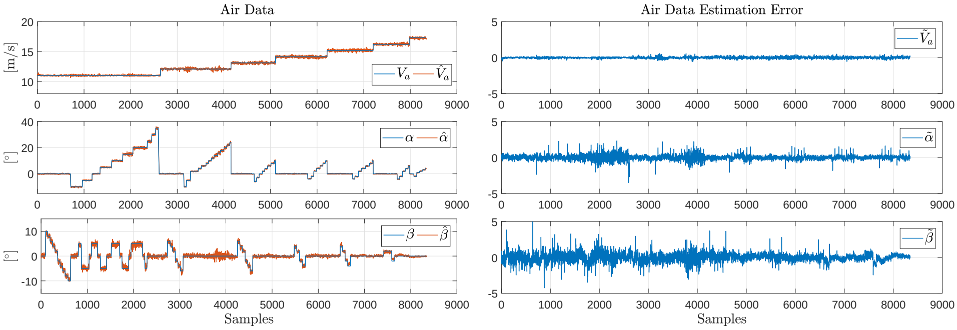
Figure 3.8 shows the difference between the NN and LR estimates from Figure 3.6 and Figure 3.7 and both sets of estimates together with the ground truth values. It appears that for the wind tunnel tests, the biggest difference between the two ML algorithms occurs when the AOA assumes high values and this is assumed to be attributed to nonlinearities and turbulence introduced when the wing is close to or under stall conditions.



a True air data parameter values and the corresponding LR estimates, denoted by the hat accent.

b The LR air data parameter estimation error, denoted by the tilde accent.

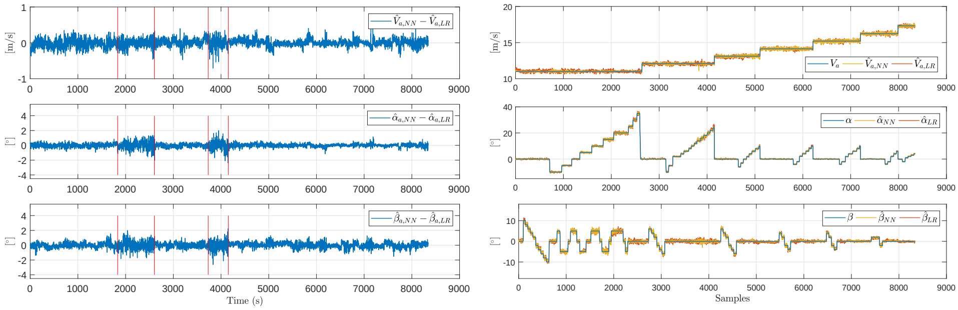
Figure 3.6: Results obtained using LR with the basis input vector augmented with the first order cross terms, the absolute function term, and the cubed term (**BXQC**). The results displayed consists of the complete wind tunnel data set.



a True air data parameter values and the corresponding NN estimates, denoted by the hat accent.

b The NN air data parameter estimation error, denoted by the tilde accent.

Figure 3.7: Results obtained using NNs with 10 neurons in a single hidden layer ($\mathbf{F}_{10}\mathbf{S}_0$). The results displayed consists of the complete wind tunnel data set.



a The difference between the NN and LR estimates. The two sets of red vertical lines indicate an airspeed of 12 m/s or higher in the two interior segments.

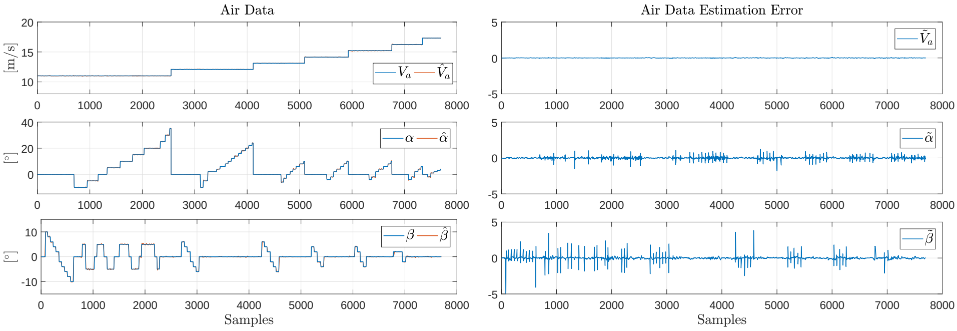
b The results from both the NN and LR plotted together with the ground truth values.

Figure 3.8: Comparison between the NN and LR results, respectively denoted by $\hat{V}_{a,NN}$ and $\hat{V}_{a,LR}$, from Figure 3.6 and Figure 3.7.

3.4.2 Comparison with higher accuracy pressure sensor

To obtain insight into the advantage of using more sensors of higher accuracy on the method, a comparison with the Scanivalve DSA3217 pneumatic pressure scanner setup is presented. As mentioned previously, the DSA3217 is too heavy and large to install on the Skywalker X8 and was therefore not possible to use during flight. Furthermore, the setup utilizes more sensors than the BMP280 setup, and the measurement points are not identical. The comparison between the two sets of sensors can therefore as such not provide a complete picture of the influence of sensor amount and sensor accuracy on the method, but it can function towards an impression of the highest attainable estimation accuracy given the used sensor accuracy, as well as an understanding of the influence of ML algorithm choice on the results. For the comparison between BMP280 sensors and the DSA3217, only a single LR and NN structure has been chosen, **BXQC** and $\mathbf{F}_{10}\mathbf{S}_0$, respectively. A plot of the $\mathbf{F}_{10}\mathbf{S}_0$ NN estimation results are shown in Figure 3.9 and the numerical results are listed in in Table 3.4.

The spikes in estimation error seen in Figure 3.9, is assumed primarily to be a result of the built-in low pass filter of the DSA3217 pressure scanner, resulting in a transient phase between reaching the PTU-D48 pan-and-tilt unit set points and the pressure measurements. There are therefore two sets of results in Table 3.4. In the first set of results, the data has been used without any processing (labeled as *raw data*). In the second set of results, the transition phase and has been removed from the data before training and testing (labeled as *stationary data*). It appears that the higher number of sensors, and higher accuracy of the DSA3217 pressure scanners improves the numerical results for the ML approaches when compared to the less accurate BMP280 measurements. Based on these results, it is expected that an increase in number of sensors and sensor accuracy will improve the output



a True air data parameter values and the corresponding NN estimates, denoted by the hat accent.

b The NN air data parameter estimation error, denoted by the tilde accent.

Figure 3.9: Results for the NN ($\mathbf{F}_{10}\mathbf{S}_0$) estimation method using the DSA3217 pressure scanner measurements and displaying the complete wind tunnel data set.

	BXQC	$\mathbf{F}_{10}\mathbf{S}_0$	**BXQC**	$\mathbf{F}_{10}\mathbf{S}_0$
	raw data	raw data	stationary data	stationary data
$\check{V}_a$ [m/s]	0.0205	0.0172	0.0169	0.0128
$\bar{V}_a$ [m/s]	0.0213	0.0179	0.0182	0.0140
$\check{\alpha}$ [°]	0.2213	0.1796	0.0285	0.0235
$\bar{\alpha}$ [°]	0.2347	0.2062	0.0328	0.0247
$\check{\beta}$ [°]	0.3497	0.3311	0.0447	0.0370
$\bar{\beta}$ [°]	0.3678	0.3610	0.0452	0.0373

Table 3.4: The results obtained using LR and NN on the DSA3217 pressure scanner measurements. Vee denotes the training set RMSE and bar denotes the test set RMSE.

of the ML algorithms, and that the choice of sensor and quantity of sensors will be a trade-off between price, weight, size, and power consumption of the sensors and the accuracy of the air data parameter estimates.

The results presented so far indicates that a more complex structure of the ML approach provides better results and that the NNs have slightly lower RMSE values than the LR. Also, there is reason to believe that using higher accuracy sensors results in higher accuracy estimates. However, a weakness with the approach so far is that the estimates are trained and tested with a set of data with identical conditions and only a rather limited set of different air data parameter points. Should the trained LR and NN structures be tested with a different set of data with air data parameters differing from the points trained for, there is no guarantee on the quality of the results. The test and training of ML methods on flight benchmark test data is therefore the primary focus of the remaining part of the results section.

3.4.3 Wind tunnel trained ML algorithms on benchmark flight test

This section contains the results obtained using the ML methods trained using the wind tunnel data on the benchmark flight test. For this comparison, 100% of the wind tunnel data has been used in training the LR method and for the NN method, the wind tunnel data has been divided into 85% training data set and a 15% validation data set. The initial numerical results using the wind tunnel data for training (not included here) showed low estimation accuracy on the benchmark flight test. Given a significant difference between the RMSE of the training set results and the test set results, it is assumed that the poor results are not primarily attributable to poor sensor accuracy, but instead stems from one or several of the following factors:

- There might have been a misalignment between the AOA and SSA in of the Skywalker X8 in the wind tunnel and the mounting of the Aeroprobe sensor. This would result in biases on the AOA and SSA estimates corresponding to the misalignment.
- As mentioned previously, the Skywalker X8 blocked out a sizeable amount of the wind tunnel. This could therefore result in the airspeed around the Skywalker X8 being higher than the wind tunnel airspeed used in training the ML methods. This could in turn result in the estimates provided by the ML methods being too low.
- ML methods are not suited for extrapolation and a considerable fraction of the airspeeds experienced by the Skywalker X8 in the flight test, was higher than the highest value tested for in the wind tunnel tests. Courrieu [23] suggested a geometric approach of using a convex hull polytope to find the domain of validity of a feedforward NN and this limit is obviously violated by the airspeed, without even having to consider the remaining values of the estimated parameters.
- Too few training data points to properly model the (biased) input-to-output relationship. Rohloff et al. [77] discusses a binning technique to choose a proper training set from a larger data set. The training data is divided into bins in the different dimensions and choosing data following this technique then ensures a decent density distribution of the training data. However, for the wind tunnel data this technique could be misleading since the data could appear to be evenly divided into different bins, while in reality, many of the points in the bins would be from the same static air data parameter wind tunnel set points. Instead, data collected from flight tests will add a stochastic element that ensures that no air data parameter points are exactly the same, and that should result in a finer resolution and a better distribution compared to the wind tunnel.

Furthermore, training the ML methods using data containing stall conditions introduce strong nonlinearities into the system, and modeling will require even more data and represents an unnecessary difficulty if the intention is to not fly under stall conditions[2].

In order to remove the effect of blockage and misalignment on the results, a new set of results have been generated where the training data have had bias correction terms added to it. The bias have been estimated as the mean of the error for 10 instances of training $\mathbf{F}_{10}\mathbf{S}_0$ NNs with the wind tunnel data and testing on the benchmark flight. The biases were found to be 3.24 m/s for the airspeed, 10.59° for the AOA, and -1.63° for the SSA. The results are listed in Table 3.5 and 3.6.

Linear Regression

	B	**BX**	**BQ**	**BC**	**BXQC**
$\check{V}_a$ [m/s]	0.4521	0.1541	0.3001	0.3275	0.1468
$\bar{V}_a$ [m/s]	1.2075	1.4815	1.7577	3.5591	1.0415
$\check{\alpha}$ [°]	1.2512	0.4828	0.8260	0.8974	0.4406
$\bar{\alpha}$ [°]	8.6618	2.4905	3.4954	17.0525	7.7749
$\check{\beta}$ [°]	1.0488	0.6115	0.8036	0.7986	0.5959
$\bar{\beta}$ [°]	4.1666	5.1462	2.8018	5.3993	4.9833

Table 3.5: The results obtained using LR trained with wind tunnel data with added bias corrections on the flight benchmark test. Vee denotes the training set RMSE and bar denotes the test set RMSE.

Neural Networks

	$\mathbf{F}_5\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_0$	$\mathbf{F}_{15}\mathbf{S}_0$	$\mathbf{F}_{20}\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_5$	$\mathbf{F}_{10}\mathbf{S}_{10}$
$\check{V}_a$ [m/s]	0.1341	0.1154	0.0961	0.0818	0.0773	0.0567
$\bar{V}_a$ [m/s]	0.7337	0.9642	1.0603	1.2318	1.0871	1.3066
$\check{\alpha}$ [°]	0.4004	0.3566	0.3411	0.3190	0.3334	0.3148
$\bar{\alpha}$ [°]	1.0877	1.5216	1.7246	3.7936	2.2111	3.8493
$\check{\beta}$ [°]	0.5665	0.5003	0.4471	0.4220	0.4122	0.3784
$\bar{\beta}$ [°]	2.2016	2.4157	2.5866	3.1320	2.6473	2.0639

Table 3.6: The results obtained using NNs trained with wind tunnel data with added bias corrections on the flight benchmark test. Vee denotes the training set RMSE and bar denotes the test set RMSE.

Modifying the training data in an attempt to account for the effects of blockage and misaligned sensors improved the numerical results. However, the high AOA and SSA RMSE for the LR method, may indicate that there are significant nonlinearities in the wind tunnel data. The NN results indicate that the more advanced structures are prone to overfitting. As mentioned previously, the numerical results are found as the mean of the results obtained from 10 different instances of NNs with identical structure. The results from each of these 10 NNs show a large variation in the RMSE of the different estimates. The best NNs were capable of obtaining

[2] It should be noted that the comparison sensor, the Micro Air Data System by the Aeroprobe Corporation, is not capable of measuring in the stall region either.

a RMSE on the flight benchmark test of 0.54 m/s for the airspeed, 0.77° for the AOA, and 1.37° for the SSA, which is significantly lower than the mean of the results from the 10 NNs. This indicates that the results could be improved by a training data choosing algorithm, perhaps using a binning technique as described previously. The high variance on the accuracy of the results is assumed to be primarily due to a lack of a high resolution wind tunnel training data set that covers all possible air data parameter values that the Skywalker X8 can be expected to experience in flight.

3.4.4 Flight trained ML algorithms on benchmark flight

This section contains the results of training the ML methods with a 31 minute flight training set and evaluating the methods on the flight benchmark test. The last part of the training segment involved the Skywalker X8 flying in high dynamic maneuvers and the training data set contains data that eclipses the test set and the algorithms will therefore not have to rely on extrapolation. The results are shown in Table 3.7 and 3.8. The results obtained by training the ML algorithms using flight data gives much better results than the ones obtained using the wind tunnel data for training. It appears that the test LR results obtained are not that dependent on the structure of the LR, although it seems beneficial to add a single basis function expansion to increase the the accuracy of the estimates. For the NN, the results appear to be relatively independent on the chosen structure, which indicates that for the conditions trained and tested in, the relation between pressure and air data parameters must be possible to model by three five-neuron NNs. Furthermore, the numerical results are relatively close in accuracy and both methods seems viable to use for similar flight conditions if properly trained.

Linear Regression

	B	**BX**	**BQ**	**BC**	**BXQC**
$\check{V}_a$ [m/s]	0.3815	0.2865	0.3213	0.3227	0.2760
$\bar{V}_a$ [m/s]	0.4439	0.3587	0.3655	0.3696	0.4088
$\check{\alpha}$ [°]	0.3112	0.2010	0.2325	0.2465	0.1896
$\bar{\alpha}$ [°]	0.2736	0.2379	0.2609	0.2591	0.2328
$\check{\beta}$ [°]	1.3654	0.9162	1.0928	1.1371	0.8652
$\bar{\beta}$ [°]	1.3358	1.0058	0.9997	1.0572	1.0161

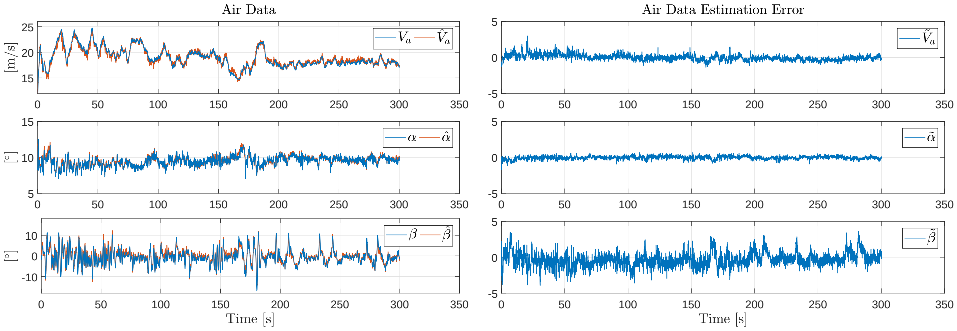
Table 3.7: The numerical results obtained using LR trained with flight data on the flight benchmark test set. Vee denotes the training set RMSE and bar denotes the test set RMSE.

Figure 3.10 and 3.11 contains plot of the results obtained using the flight data trained **BXQC** LR and $\mathbf{F}_{10}\mathbf{S}_0$ NN algorithms. The plots show relatively low estimation errors from both algorithms. The improvements of the results from using flight data compared to wind tunnel data is expected to be caused by the removal of the error points discussed in the previous section - there is no misalignment of angles, the airspeed used in the training data is not biased due to blockage effect,

Neural Networks						
	$\mathbf{F}_5\mathbf{S}_0$	$\mathbf{F}_1 0\mathbf{S}_0$	$\mathbf{F}_{15}\mathbf{S}_0$	$\mathbf{F}_{20}\mathbf{S}_0$	$\mathbf{F}_{10}\mathbf{S}_5$	$\mathbf{F}_{10}\mathbf{S}_{10}$
$\check{V}_a$ [m/s]	0.2745	0.2516	0.2388	0.2305	0.2437	0.2326
$\bar{V}_a$ [m/s]	0.3503	0.3497	0.3460	0.3478	0.3372	0.3492
$\check{\alpha}$ [°]	0.1905	0.1735	0.1654	0.1613	0.1677	0.1622
$\bar{\alpha}$ [°]	0.2341	0.2298	0.2415	0.2445	0.2298	0.2313
$\check{\beta}$ [°]	0.8504	0.7584	0.7055	0.6592	0.6922	0.6676
$\bar{\beta}$ [°]	0.9165	0.9329	0.9585	0.9674	0.9322	0.9180

Table 3.8: The numerical results obtained using NNs trained with flight data on the flight benchmark test set. Vee denotes the training set RMSE and bar denotes the test set RMSE.

and there is no extrapolation. However, the biggest improvement is assumed to be attributed to the data being distributed in relevant flight conditions and with a much finer parameter resolution which allows the methods a better chance at correctly modeling the input/output relations.



a True air data parameter values and the corresponding LR estimates, denoted by the hat accent.

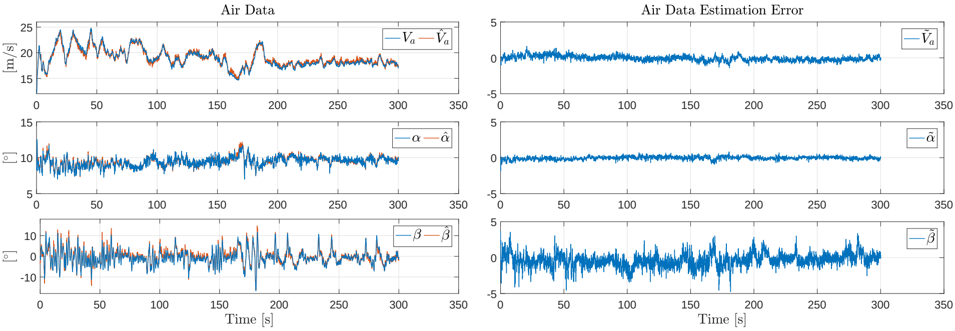
b The LR air data parameter estimation error, denoted by the tilde accent.

Figure 3.10: Results obtained using a **BXQC** LR trained with flight data on the flight benchmark test set.

Figure 3.12 shows the difference between the NN and LR algorithm estimates Figure 3.10 and Figure 3.11. The plot indicates that the two algorithms provide the largest differences in estimates when the air data ground truth values deviate the furthest from steady flight conditions. This is assumed to be due to less training data covering these air data values.

3.4.5 Sensor configuration influence

So far, all the algorithms have been trained using the data from all the BMP280 sensors. However, a part of the novelty of the presented method lies in the option of adjusting the sensor configuration to whichever aircraft is the desired platform of implementation. This section investigates the relation between different sensor



a True air data parameter values and the corresponding NN estimates, denoted by the hat accent.

b The NN air data parameter estimation error, denoted by the tilde accent.

Figure 3.11: Results obtained using a $\mathbf{F}_{10}\mathbf{S}_0$ NN trained with flight data on the flight benchmark test set.

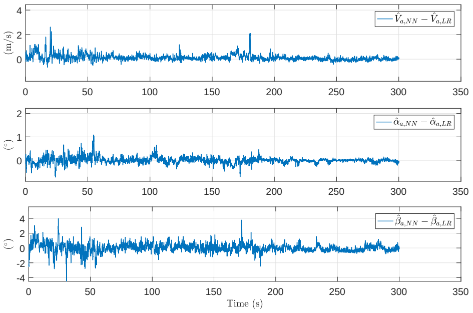


Figure 3.12: The difference between the NN and LR estimates, respectively denoted by $\hat{V}_{a,NN}$ and $\hat{V}_{a,LR}$, from Figure 3.10 and Figure 3.11.

configurations and their corresponding results in estimating the air data parameters. The results will obviously only be valid for the Skywalker X8 UAV, but can perhaps serve as an indicator of how the results would be on similar platforms (which would have to be trained using data acquired using that specific platform). The flight data trained **BXQC** LR and $\mathbf{F}_{10}\mathbf{S}_0$ NN algorithms have been chosen as the basis ML algorithms for this sensor configuration assessment. Furthermore, a single test is included where the the full sensor configuration is augmented with a pseudo Re number. The pseudo Re number uses ambient pressure and temperature from the autopilot, but with the relative velocity in the forward direction, u_r, obtained from the Pitot-static tube as an approximation for the airspeed.

Reverting back to Figure 3.2 and 3.3, the naming convention of the sensors is that the first five sensors, p_{s_1} - p_{s_5} are located in the starboard wing. The topside sensor at the leading edge is p_{s_1} and the sensor numbers then increment towards and around the trailing edge with p_{s_5} as the sensor on the bottom of the wing. For the nose, the sensor located at twelve o'clock in Figure 3.3 is the reference sensor and the sensor on the right is p_{s_6}. Incrementing in a clock-wise pattern, the sensor to the left of the reference sensor is $p_{s_{10}}$. The port wing follows the same system as the starboard wing starting with sensor $p_{s_{11}}$ and ending with $p_{s_{15}}$ on the bottom side of the wing. For the tests, the LR and NN structures **BXQC** and $\mathbf{F}_{10}\mathbf{S}_0$ have been used once again. Regarding the choice of sensor configurations, it appears intuitive to evaluate the configurations that involves using only nose or wing sensors, since there might be limitations on a UAV platform related to these, e.g. propellers that hinders the use of pressure sensors in either nose or wings. It would also be advantageous to get an impression of the influence of using fewer sensors on the attainable accuracy. Fewer sensors would reduce the cost of the setup, reduce complexity of the implementation, and lower the risk of a single sensor malfunctioning.

The numerical results from different sensor configurations and augmenting with the pseudo Re number are shown in Table 3.9 and 3.10, where the numbers are used for denoting the sensors used in the configuration, i.e. 5 corresponds to using sensor p_{s_5} in the configuration. Both tables show the same trends in accuracy with respect to the different sensor configurations. The nose sensors alone are relatively poor at estimating the airspeed, whereas adding a sensor from each wing greatly increase the accuracy. The wing sensors alone are on the other hand decent at estimating the airspeed, but suffer from poor SSA estimates, since the sensors only measure in two parallel planes, both perpendicular to the SSA. The choice of sensor configuration should be dependent on the UAV platform of implementation and will be a trade-off between accuracy, cost, and ease of implementation. Adding the pseudo Re number only appears to provide a small benefit to the results. However, the flight results contain relatively constant ambient conditions, and the pseudo Re number can potentially provide a higher level of robustness with respect to these conditions or be used in non-dimensionalizing the pressure measurements into pressure coefficients.

BXQC LR

	All sensors + Re	**All sensors**	**Nose sensors**	**Wing sensors**	**5,6,7,8, 9,10,15**	**5,7,9,15**
$\check{V}_a$ [m/s]	0.2593	0.2760	0.9329	0.6121	0.5120	0.7303
$\bar{V}_a$ [m/s]	0.4043	0.4088	0.8858	0.7293	0.5765	0.7371
$\check{\alpha}$ [°]	0.1809	0.1896	0.4213	0.2890	0.2955	0.3228
$\bar{\alpha}$ [°]	0.2271	0.2328	0.3837	0.4086	0.2737	0.3185
$\check{\beta}$ [°]	0.8546	0.8652	1.1743	2.2266	1.0059	1.1251
$\bar{\beta}$ [°]	1.0104	1.0161	1.0874	3.7345	0.9274	0.9826

Table 3.9: The numerical results from using different sensor configurations and adding a pseudo Re number when training and testing the **BXQC** LR algorithm. Vee denotes the training set RMSE and bar denotes the test set RMSE.

$\mathbf{F}_{10}\mathbf{S}_0$ NN

	All sensors + Re	**All sensors**	**Nose sensors**	**Wing sensors**	**5,6,7,8, 9,10,15**	**5,7,9,15**
$\check{V}_a$ [m/s]	0.2375	0.2534	0.8512	0.5680	0.4534	0.6684
$\bar{V}_a$ [m/s]	0.3402	0.3535	0.8363	0.7891	0.4557	0.6713
$\check{\alpha}$ [°]	0.1655	0.1741	0.4030	0.2688	0.2815	0.3225
$\bar{\alpha}$ [°]	0.2167	0.2318	0.3873	0.4167	0.2857	0.3277
$\check{\beta}$ [°]	0.7439	0.7575	1.0746	2.0189	0.9051	1.0188
$\bar{\beta}$ [°]	0.9326	0.9507	1.0568	3.7503	0.9168	1.0232

Table 3.10: The numerical results from using different sensor configurations and augmenting with a pseudo Re number when training and testing the $\mathbf{F}_{10}\mathbf{S}_0$ NN algorithm. Vee denotes the training set RMSE and bar denotes the test set RMSE.

3.5 Chapter Summary

This chapter presented a method for estimating air data for small fixed-wing unmanned aerial vehicles. The method comprises a set of low-cost MEMS-based pressure sensors embedded in the surface of the unmanned aerial vehicle combined with machine learning algorithms. A strength of the presented method is the flexibility of the pressure sensor placement. Two different machine learning algorithms, neural networks and linear regression, have been implemented and tested. Both algorithms have been evaluated on data obtained through wind tunnel experiments and experimental flight data. The neural network algorithm was found to generally provide a slightly lower estimation error than the linear regression approach presented. However, linear regression allows for basis function expansions that could potentially improve the results further. By comparing the results obtained from using the low-cost sensors on the aircraft against the results from the expensive pressure scanner from the wind tunnel, the influence on sensor amount and sensor accuracy on the results was assessed. Training the machine learning algorithms using only wind tunnel data was found to have several error sources, namely the wind tunnel blockage effect, misalignment in the aircraft mount, as well as a too sparse (not very rich) training data set. However, trying to account for these error sources showed potential in decreasing the estimation error. Using flight data to train the machine learning algorithms was found to be a feasible approach that allowed estimating the air data parameters for both neural networks and linear regression. Finally, a study of the sensor number and placement influence on the result was conducted along with an assessment of augmenting the system input by using a pseudo Reynolds number.

Chapter 4

Kalman Filters for Air Data System Bias Correction

This chapter proposes two different Kalman filter approaches for correcting air data systems with low-frequency drift or a constant error bias for fixed-wing UAVs. The presented estimators use a standard sensor suite consisting of a GNSS receiver, an IMU, and a heading reference, combined with an air data system that is assumed to provide measurements with an unknown additive slowly time-varying bias. Neither estimator is dependent on the UAV model. The estimators are, except for the effect of noise, proven to have globally exponentially stable (GES) equilibrium points of the error dynamics if provided with persistence-of-excitation (PE) of the angular velocity of the UAV. This chapter is initiated by Section 4.1 in which the problem formulation is stated along with the models and assumptions. Section 4.2 contains the wind velocity estimator and analyses the observability properties of the system, and Section 4.3 describes the relative velocity estimator along with a corresponding observability analysis. The two estimators are tested in simulation in Section 4.4 and using experimental flight data in Section 4.5. The results from the experimental flight data is obtained using the ML approach presented in Chapter 3. The results indicates that a certain amount of PE is needed in order to have converging bias estimates during turbulent wind conditions.

4.1 Problem Formulation

The velocity over ground of a UAV can be expressed as the sum of the relative velocity and the wind velocity according to

$$v_g^n = R_b^n v_r^b + v_w^n \tag{4.1}$$

where $v_g^b = [v_n, v_e, v_d]^\top$ is the velocity over ground of the UAV decomposed in the NED frame, $v_r^b = [u_r, v_r, w_r]^\top$ is the relative velocity of the UAV with respect to the wind decomposed in the BODY frame and $v_w^n = [w_n, w_e, w_d]^\top$ is the wind velocity decomposed in the NED frame. The goal is to estimate either v_w^b or v_r^b, since knowledge of one allows computing the other using only known measurements.

From the relative velocity the airspeed, AOA and SSA are recognized as

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{4.2}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{4.3}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \tag{4.4}$$

4.1.1 Models and assumptions

For the two presented estimators, it is assumed that the UAV is equipped with a standard sensor suite consisting of an inertial measurement unit (IMU), a GNSS receiver, and a heading reference such as a magnetometer. Furthermore, it is assumed that a biased relative velocity sensor is available. Such a sensor could be obtained by using the output of the pressure sensor-array approach presented in Chapter 3 treated as a virtual sensor. The measurements assumed are:

- An attitude estimate from an attitude estimator, typically in either Euler angles or quaternions, used to compute the rotation matrix R_b^n. This estimate is assumed free of noise in the estimator design, but studied in simulation and experiments.
- A GNSS receiver velocity over ground measurement modeled by $v_{g,m}^n = v_g^n + \epsilon_{v_g}$, where $\epsilon_{v_g} \sim \mathcal{N}(0,\, \Sigma_{v_g})$ is a noise term.
- An IMU specific force measurement modeled by $f_m^b = f^b + \epsilon_f$, where $\epsilon_f \sim \mathcal{N}(0,\, \Sigma_f)$ is a noise term.
- An IMU angular rate measurement, compensated for bias and drift, modeled by $\omega_{b,m} = \omega_b + \epsilon_\omega$, where ω_b is the angular velocity of the BODY frame relative to the NED frame, decomposed in the BODY frame, and $\epsilon_\omega \sim \mathcal{N}(0,\, \Sigma_\omega)$ is a noise term.
- A biased relative velocity measurement such as [12] modeled by $v_{r,m}^b = v_r^b + b^b + \epsilon_{v_r}$, where $b^b = [b_u, b_v, b_w]^\top$ is a sensor bias and $\epsilon_{v_r} \sim \mathcal{N}(0,\, \Sigma_{v_r})$ is a noise term.

Note that the argument for time-varying signals have been omitted for notational simplicity, except when needed. An example of a GES attitude and gyro bias observer is found in Grip et al. [33]. The bias-compensated IMU measurements of angular rate and specific force are only used in the relative velocity estimator. Similarly, the GNSS velocity over ground measurement is only used in the relative velocity estimator if wind velocity estimates are also desired.

For both estimators, it is assumed that the wind velocity vector field is slowly time-varying and uniform over the area of flight, i.e.

$$\dot{v}_w^n = \epsilon_{v_w} \tag{4.5}$$

where $\epsilon_{v_w} \sim \mathcal{N}(0,\, \Sigma_{v_w})$ is the wind model noise term, and that the relative velocity sensor bias is slowly time-varying

$$\dot{b}^b = \epsilon_b \tag{4.6}$$

where $\epsilon_b \sim \mathcal{N}(0, \Sigma_b)$ is the bias model noise term. These two model assumptions will be used in both the estimators and the states will be estimated using Kalman filters.

Note on additive bias versus multiplicative error model: Instead of using the bias model in Eq. 4.6, the two estimators can be formulated with an unknown 3-dimensional scaling factor multiplied onto the relative velocity measurement similar to Johansen et al. [47].

4.2 Wind Velocity Estimator

The wind velocity estimators presented in this section is inspired by the estimator presented in Johansen et al. [47]. It estimates the wind velocity and the bias of the relative velocity sensor. Using the assumed measurements, it is possible from the wind velocity to estimate the relative velocity, and in turn the airspeed, AOA, and SSA. The structure of the wind velocity estimator is shown in Fig. 4.1.

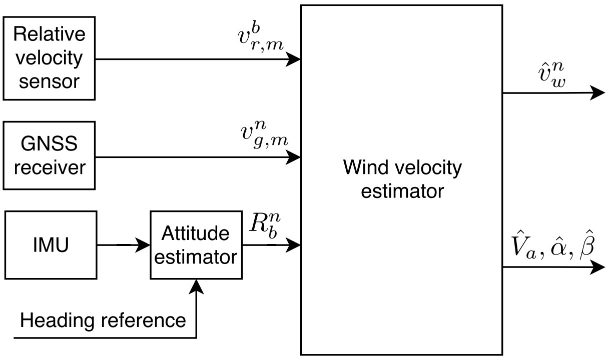


Figure 4.1: Block diagram showing the cascaded structure of the wind velocity estimator.

By inserting the GNSS velocity over ground measurement and the relative velocity sensor measurement into Eq. 2.4, we get

$$v^n_{g,m} - R^n_b v^b_{r,m} = v^n_w + R^n_b b^b + R^n_b \epsilon_{v_r} + \epsilon_{v_g} \tag{4.7}$$

Consider the state vector

$$x_w = \begin{bmatrix} v^n_w \\ b^b \end{bmatrix}$$

and the composed measurement

$$y_w = v^n_{g,m} - R^n_b v^b_{r,m}$$

The linear system can be stated as

$$\dot{x}_w = G_w \epsilon_{x_w} \tag{4.8}$$

$$y_w = C_w(t)x_n + C_w(t)\epsilon_{y_w} \tag{4.9}$$

where

$$G_w = I_6, \quad \epsilon_{x_w} = \begin{bmatrix} \epsilon_{v_w} \\ \epsilon_b \end{bmatrix}$$

$$C_w(t) = \begin{bmatrix} I_3 & R_b^n(t) \end{bmatrix}, \quad \epsilon_{y_w} = \begin{bmatrix} \epsilon_{v_g} \\ \epsilon_{v_r} \end{bmatrix}$$

Also, notice that the system matrix is $A_w = 0_{6\times 6}$.

4.2.1 Wind velocity estimator design

Consider the estimator

$$\dot{\hat{x}}_w = K_w(y_w - C_w\hat{x}_w) \tag{4.10}$$

where K_w is the Kalman filter gain matrix. The process noise covariance matrix is

$$Q_w = E[\epsilon_{x_w}\epsilon_{x_w}^\top] = Q_{x_w}$$

where $Q_{x_w} = \text{diag}(\Sigma_{v_w}, \Sigma_b)$. The measurement covariance matrix is found to be

$$\begin{aligned} R_w(t) &= E[C_w(t)\epsilon_{y_w}\epsilon_{y_w}^\top C_w(t)^\top] = C_w(t)E[\epsilon_{y_w}\epsilon_{y_w}^\top]C_w(t)^\top \\ &= C_w(t)R_{y_w}C_w(t)^\top \end{aligned}$$

where $R_{y_w} = \text{diag}(\Sigma_{v_g}, \Sigma_{v_r})$.

Proposition 1: Assume the angular velocity and angular acceleration of the UAV satisfies $\|\omega^b \times \dot{\omega}^b\| > 0$ for all $t \geq 0$, then the LTV system described by Eq. 4.8 and 4.9 is uniform completely observable (UCO) and uniform completely controllable (UCC).

Proof: First, we show UCO of the pair $(A_w, C_w(t))$. Theorem 6.O12 in Chen [20], states that the continuously differentiable pair $(A(t), C(t))$ is UCO at t, if there exists a finite $t_1 > t$ such that the observability codistribution $d\mathcal{O}_w = [N_0(t_1); ...; N_{n-1}(t_1)]$ has full rank, where

$$N_0(t) = C(t)$$

$$N_{m+1}(t) = N_m(t)A(t) + \frac{d}{dt}N_m(t), \quad m = 0, 1, ..., n-1$$

For the pair $(A_w, C_w(t))$, we have

$$\begin{aligned} d\mathcal{O}_w &= \begin{bmatrix} C_w(t) \\ \dot{C}_w(t) \\ \ddot{C}_w(t) \end{bmatrix} \\ &= \begin{bmatrix} I_3 & R_b^n(t) \\ 0_{3\times 3} & R_b^n(t)S(\omega^b(t)) \\ 0_{3\times 3} & R_b^n(t)S(\omega^b(t))^2 + R_b^n(t)S(\dot{\omega}^b(t)) \end{bmatrix} \end{aligned}$$

Given the form of the $d\mathcal{O}_w$, to verify that it has full rank, it suffices to show that the submatrix

$$\mathcal{O}_{w,1} = \begin{bmatrix} R_b^n(t)S(\omega^b(t)) \\ R_b^n(t)S(\omega^b(t))^2 + R_b^n(t)S(\dot{\omega}^b(t)) \end{bmatrix}$$

is full rank, as proven by Meyer [66]. By linear row operations and scalar multiplications, it is possible to obtain the following matrix that has the same rank as $\mathcal{O}_{w,1}$

$$\mathcal{O}_{w,2} = \begin{bmatrix} R_b^n(t)S(\omega^b(t)) \\ R_b^n(t)S(\dot{\omega}^b(t)) \end{bmatrix}$$

We see that $\mathcal{O}_{w,2}$ has full rank if $\dot{\omega}^b$ and ω^b are non-zero, linearly independent vectors, which corresponds to the requirement $\|\omega^b \times \dot{\omega}^b\| > 0$.

Lastly, we show that the pair (A_w, G_w) is UCC. Theorem 6.12 of Chen [20], states that the continously differentiable pair $(A(t), G(t))$ is UCC at t_0 if there exists a finte $t_1 > t_0$ such that the controllability codistribution $d\mathcal{C}_w = [M_0(t), ..., M_{n-1}(t)]$ has full rank, where

$$\begin{aligned} M_o(t) &= G(t) \\ M_{m+1}(t) &= -A(t)M_m(t) + \frac{d}{dt}M_m(t) \end{aligned}$$

It is easily seen that this is obtained for $d\mathcal{C}_w = M_0 = I_6$. ■

The UCO and UCC properties of the system implies that the equilibrium points of the Kalman filter error dynamics, $\tilde{v}_w^n = v_w^n - \hat{v}_w^n$ and $\tilde{b}^b = b^b - \hat{b}^b$, are GES in the deterministic case, as proven by Anderson [4] and bounded in the mean-square sense in the stochastic case [46]. The definition of the estimator in Eq. 4.10 and the observability analysis is done in continuous time for convenience. However, the Kalman filter should be implemented using a discrete-time algorithm. The conducted analysis showed that in order for the system to be UCO, the requirement $\|\omega^b \times \dot{\omega}^b\| > 0$ has to be fulfilled. This correspond to a PE requirement on the angular rate, that needs to vary over time. The implications of the PE requirement on the estimator performance is investigated in the simulation study in Section 4.4 and using experimental flight data in Section 4.5.

4.3 Relative Velocity Estimator

The relative velocity estimator is distinct from the wind velocity estimator. It does not require GNSS velocity over ground measurements, but angular rate and specific force measurements. The relative velocity estimator structure is shown in Fig. 4.2.

The dynamics of the velocity over ground and attitude is described by

$$\dot{v}_g^n = R_b^n f^b + g^n \tag{4.11}$$

$$\dot{R}_b^n = R_b^n S(\omega^b) \tag{4.12}$$

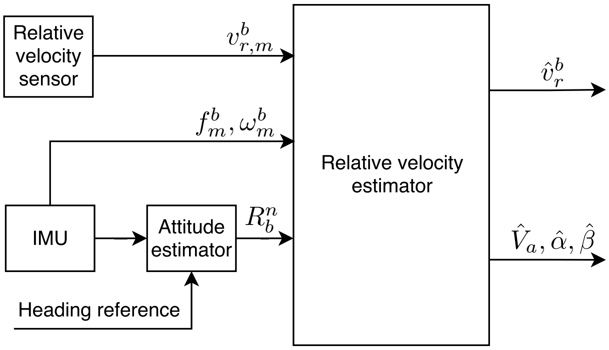


Figure 4.2: Block diagram showing the cascaded structure of the relative velocity estimator.

where f^b is the specific force decomposed in the BODY frame and g^n is the gravity vector decomposed in the NED frame. Differentiating Eq. 2.4 results in

$$\dot{v}^n_g = R^n_b S(\omega^b) v^b_r + R^n_b \dot{v}^b_r + \epsilon_{v_w} \tag{4.13}$$

Inserting Eq. 4.11 into Eq. 4.13 and rearranging gives

$$\dot{v}^b_r = f^b + R^b_n g^n - S(\omega^b) v^b_r - R^b_n \epsilon_{v_w} \tag{4.14}$$

Replacing the specific force and angular rate with the bias-compensated IMU measurements, we have that

$$\dot{v}^b_r = f^b_m + R^b_n g^n - S(\omega^b_m) v^b_r - \epsilon_f - S(v^b_r)\epsilon_\omega - R^b_n \epsilon_{v_w} \tag{4.15}$$

Consider the state vector

$$x_r = \begin{bmatrix} v^b_r \\ b^b \end{bmatrix} \tag{4.16}$$

and the input and measurement vector

$$u_r = \begin{bmatrix} f^b_m \\ g^n \end{bmatrix}, \quad y_r = v^b_{r,m} \tag{4.17}$$

Then the linear time-varying system can be stated as

$$\dot{x}_r = A_r(t) x_r + B_r(t) u_r + G_r(t, x_r) \epsilon_{x_r} \tag{4.18}$$

$$y_r = C_r x_r + \epsilon_{y_r} \tag{4.19}$$

where

$$A_r(t) = \begin{bmatrix} -S(\omega^b_m(t)) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \quad B_r(t) = \begin{bmatrix} I_3 & (R^n_b(t))^\top \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}$$

$$G_r(t, x_r) = \begin{bmatrix} -I_3 & -S(v^b_r(t)) & -(R^n_b(t))^\top & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_3 \end{bmatrix}$$

$$\epsilon_{x_r} = \begin{bmatrix} \epsilon_f \\ \epsilon_\omega \\ \epsilon_{v_w} \\ \epsilon_b \end{bmatrix}, \quad C_r = \begin{bmatrix} I_3 & I_3 \end{bmatrix}, \quad \epsilon_{y_r} = \epsilon_{v_r}$$

4.3.1 Relative velocity estimator design

Consider the estimator

$$\dot{\hat{x}}_r = A_r\hat{x}_r + B_r u_r + K_r(y_r - C_r\hat{x}_r) \tag{4.20}$$

where K_r is a time-varying Kalman gain. The process noise covariance matrix is

$$\begin{aligned} Q_r &= E[G_r(t,\hat{x})\epsilon_{x_r}\epsilon_{x_r}^\top G_r(t,\hat{x})^\top] \\ &= G_r(t,\hat{x})E[\epsilon_{x_r}\epsilon_{x_r}^\top]G_r(t,\hat{x})^\top \\ &= G_r(t,\hat{x})Q_{x_r}G_r(t,\hat{x})^\top \end{aligned}$$

where $Q_{x_r} = \text{diag}(\Sigma_f, \Sigma_\omega, \Sigma_{v_w}, \Sigma_b)$. The measurement covariance matrix is

$$R_r = E[\epsilon_{y_r}\epsilon_{y_r}^\top] = R_{y_r} \tag{4.21}$$

where $R_{y_r} = \Sigma_{v_r}$.

Proposition 2: Given the angular velocity and angular acceleration of the UAV satisfies $\|\omega^b \times \dot{\omega}^b\| > 0$ for all $t \geq 0$, then the LTV system described by Eq. 4.18 and 4.19 is UCO and UCC.

Proof: We start by showing UCO of the pair $(A_r(t), C_r)$ as defined for the LTV system described by Eq. 4.18 and 4.19. Once again employing Theorem 6.O12 of Chen [20], we have

$$\begin{aligned} d\mathcal{O}_r &= \begin{bmatrix} C_r \\ C_rA_r(t) \\ C_rA_r(t)^2 + C_r\dot{A}_r(t) \end{bmatrix} \\ &= \begin{bmatrix} I_3 & I_3 \\ -S(\omega^b(t)) & 0_{3\times3} \\ S(\omega^b(t))^2 - S(\dot{\omega}^b(t)) & 0_{3\times3} \end{bmatrix} \end{aligned}$$

By applying linear row operations and scalar multiplications, we get

$$\begin{aligned} d\mathcal{O}_{r,1} &= \begin{bmatrix} C_r \\ C_rA_r(t) \\ C_rA_r(t)^2 + C_r\dot{A}_r(t) \end{bmatrix} \\ &= \begin{bmatrix} I_3 & I_3 \\ -S(\omega^b(t)) & 0_{3\times3} \\ -S(\dot{\omega}^b(t)) & 0_{3\times3} \end{bmatrix} \end{aligned}$$

Through the identity, $\text{rank}(A) = \text{rank}(A^\top A)$ for any matrix A, we examine the ranks of $d\mathcal{O}_{r,1}$:

$$\begin{aligned} \text{rank}(d\mathcal{O}_{r,1}) &= \text{rank}(d\mathcal{O}_{r,1}^\top d\mathcal{O}_{r,1}) \\ &= \text{rank}\left(\begin{bmatrix} I_3 - S(\omega^b(t))^2 - S(\dot{\omega}^b(t))^2 & I_3 \\ I_3 & I_3 \end{bmatrix}\right) \end{aligned}$$

By applying the Schur complement [89], the full rank requirement is equivalent to

$$\det\left(-S(\omega^b(t))^2 - S(\dot{\omega}^b(t))^2\right) > 0$$

This is obtained if $\dot{\omega}^b$ and ω^b are non-zero, linearly independent vectors, which corresponds to the requirement $\|\omega^b \times \dot{\omega}^b\| > 0$.

Applying Theorem 6.12 of Chen [20] to show UCC property of the pair $(A_r(t), G_r)$, it is sufficient to examine the first set of columns of controllability codistribution $\mathcal{C}_r = M_0 = G_r(t)$, where it is easy to see that this has full rank. ■

The UCO and UCC properties of the system implies that the equilibrium points of the Kalman filter error dynamics, $\tilde{v}_r^b = v_r^b - \hat{v}_r^b = 0$ and $\tilde{b}^b = b^b - \hat{b}^b = 0$, are GES in the deterministic case, as proven by Anderson [4] and bounded in the mean-square sense in the stochastic case [46]. Similar to wind velocity estimator, the definition of the relative velocity estimator in Eq. 4.20 and the observability analysis is done in continuous time for convenience.

4.4 Simulation Study

This section presents simulation results for the wind velocity estimator and the relative velocity estimator. The results are obtained using the UAV model presented in Beard and McLain [6] with the Aerosonde UAV model parameters. Since both estimators are kinematic, the aerodynamic model of the UAV does not influence the estimation properties of the estimators.

The wind is modeled as a uniform, constant wind field with added turbulence. The turbulence is generated as white noise filtered through a Dryden model, with the transfer functions

$$H_u(s) = \sigma_u \sqrt{\frac{2V_a}{L_u}} \frac{1}{s + V_a/L_u} \tag{4.22}$$

$$H_v(s) = \sigma_v \sqrt{\frac{3V_a}{L_v}} \frac{\left(s + V_a/(\sqrt{3}L_v)\right)}{(s + V_a/L_u)^2} \tag{4.23}$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_a}{L_w}} \frac{\left(s + V_a/(\sqrt{3}L_w)\right)}{(s + V_a/L_w)^2} \tag{4.24}$$

where σ_u, σ_v, σ_w and L_u, L_v, L_w are the turbulence intensities and spatial wavelengths along the vehicle frame axes as defined in [67]. The simulation assumes low altitudes and moderate gusts. Suitable Dryden model parameter values for these conditions was presented by Langelaan et al. [52] and the Dryden model has been implemented with those parameters values and a constant nominal airspeed $V_a = V_{a_0}$. The parameter values are listed in Table 4.1.

The simulated trajectory has been chosen to provide insight into the behavior of the two estimators with respect to PE requirements. The simulation is initiated with trim conditions at an altitude of 50 m and an airspeed of 26 m/s. After 50 seconds, the course control command objective is chosen as a sinusoidal signal with

altitude	50	m
L_u,L_v	200	m
L_w	50	m
σ_u,σ_v	2.12	m/s
σ_w	1.4	m/s
V_{a_0}	26	m/s

Table 4.1: Dryden gust model parameters used in simulation.

amplitude of 50 degrees and a frequency of 0.04 Hz. This course control command pattern is continued for 100 seconds, corresponding to four periods. At time 225 s to 325 s, the altitude objective was similarly chosen as a sinusoidal signal with an amplitude of 10 m and a frequency of 0.04 Hz. It is worth noticing that the wind influence on the UAV, will result in a non-perfect tracking of control objectives. Plots of the position, angular rates, and Euler angles obtained through simulations are shown in figures 4.3 - 4.5. The angular rates indicate that the PE assumption of $\|\omega^b \times \dot{\omega}^b\| > 0$ only holds between time 50 s to time 150 s.

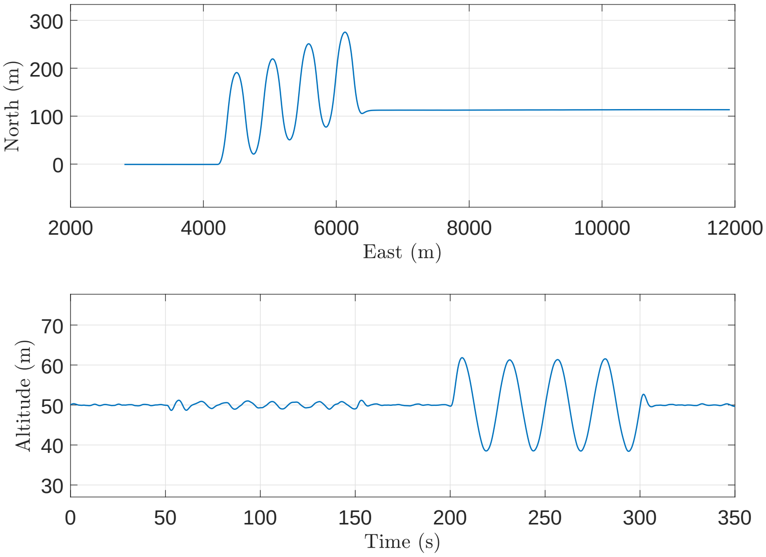


Figure 4.3: The trajectory described by the UAV in simulation.

For both estimators, the sensors were assumed corrupted by additive, uncorrelated, zero-mean white noise and sampled at 100Hz. For the biased relative velocity measurement, a standard deviation of 0.05 m/s was applied and an arbitrarily chosen constant bias $b = [2.0, -1.5, 1.3]$ was added. The rotation matrix was

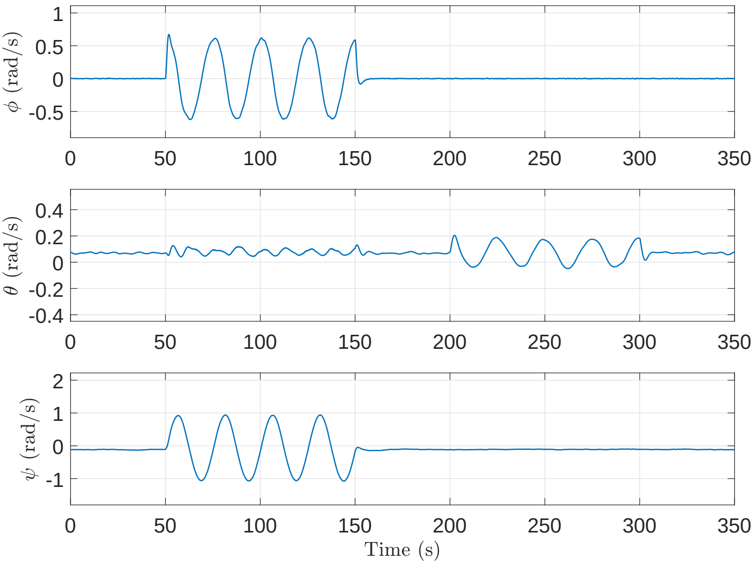


Figure 4.4: The Euler angles obtained from simulations.

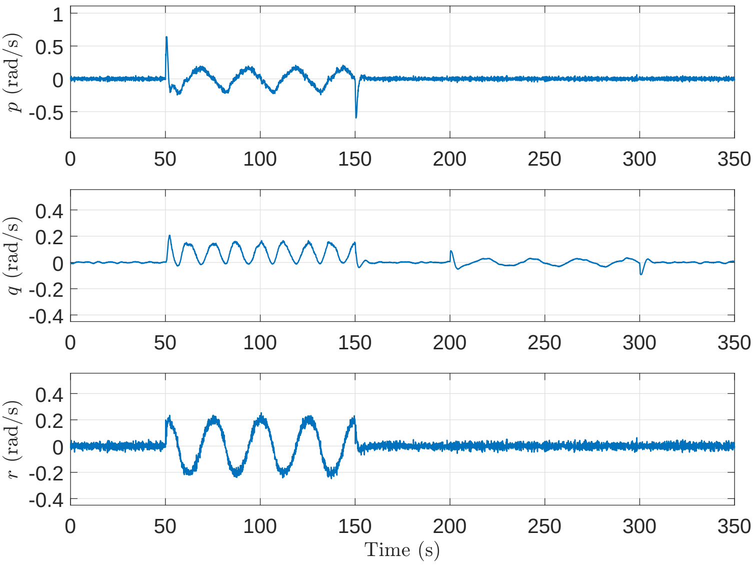


Figure 4.5: The angular rates obtained from simulation.

parametrized by Euler angles with a white noise term with standard deviation 1.0 deg. For the wind velocity estimator, the GNSS velocity over ground measurement white noise term was assumed to have a standard deviation of 0.05 m/s, and for the relative velocity estimator, the specific force measurements and the angular rate measurements were assumed to have white noise terms with standard deviations of $2.5{\times}10^{-3}$ deg/s and $2.5{\times}10^{-3}\text{m/s}^2$.

4.4.1 Wind velocity Estimator simulation results

For the wind velocity estimator, the wind model noise standard deviations were chosen by tuning as $\Sigma_{v_w} = 1{\times}10^{-2} I_3$, and the bias model noise standard deviations were chosen by tuning as $\Sigma_b = 1 \times 10^{-4} I_3$, and the error covariance matrix was initialized as $P_{w,0} = 1 \times 10^{-2} I_6$. The tuning was conducted with a prioritization of steady state performance over fast convergence. The simulation results are shown in Fig. 4.6 - 4.8.

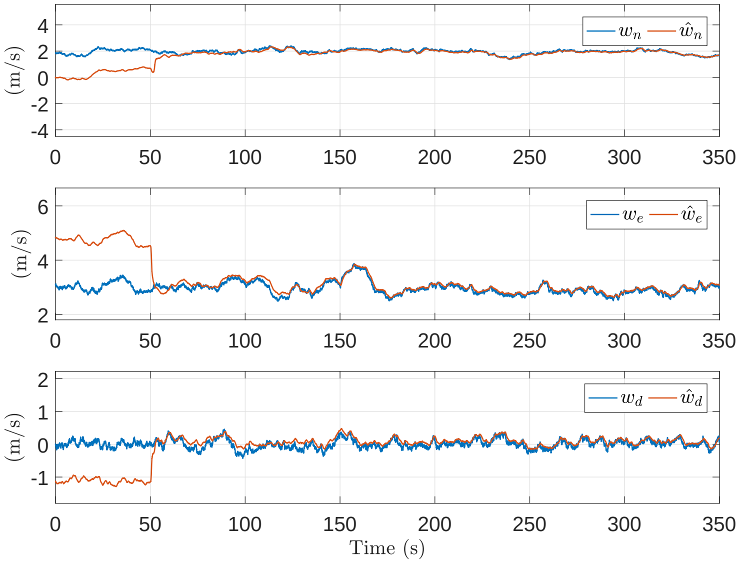


Figure 4.6: The wind velocity obtained from simulation using the Dryden wind model and the corresponding wind velocity estimator estimates.

The results show that during the first 50 seconds of trim flight conditions, the wind velocity estimator is not provided with sufficient PE and the estimates do not converge. Once the UAV engages in a sinusoidal course pattern, the estimates converge towards the true values. After convergence, there is no divergence even in periods without PE. RMSE results from the period after t = 100 seconds is shown in table 4.2.

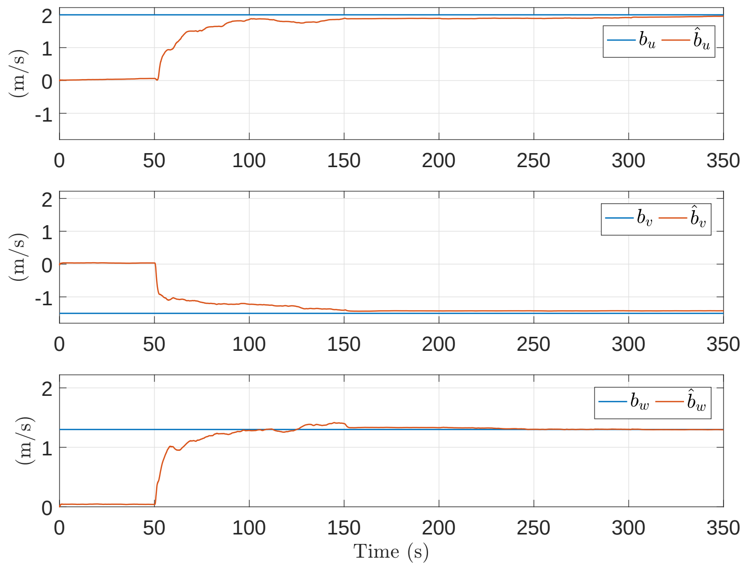


Figure 4.7: The relative velocity sensor bias and the wind velocity estimator bias estimates.

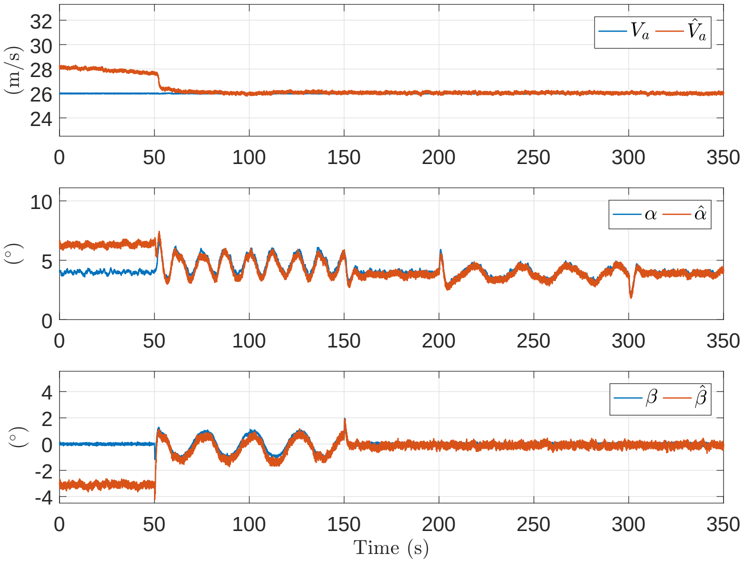


Figure 4.8: The air data obtained from simulation and the air data estimates computed from the wind velocity estimator.

$\tilde{v}_n$	0.0720	m/s
$\tilde{v}_e$	0.0903	m/s
$\tilde{v}_d$	0.1048	m/s
$\tilde{b}_u$	0.0659	m/s
$\tilde{b}_v$	0.0761	m/s
$\tilde{b}_u$	0.0863	m/s

Table 4.2: The RMSE obtained from simulation for the wind velocity estimator after t = 100 seconds.

4.4.2 Relative velocity estimator simulation results

For the relative velocity estimator, the wind model noise standard deviations, the bias model noise standard deviations, and the initial error covariance matrix were chosen as the same values as used in the wind velocity estimator simulation. The simulation results are shown in Fig. 4.9 - 4.11.

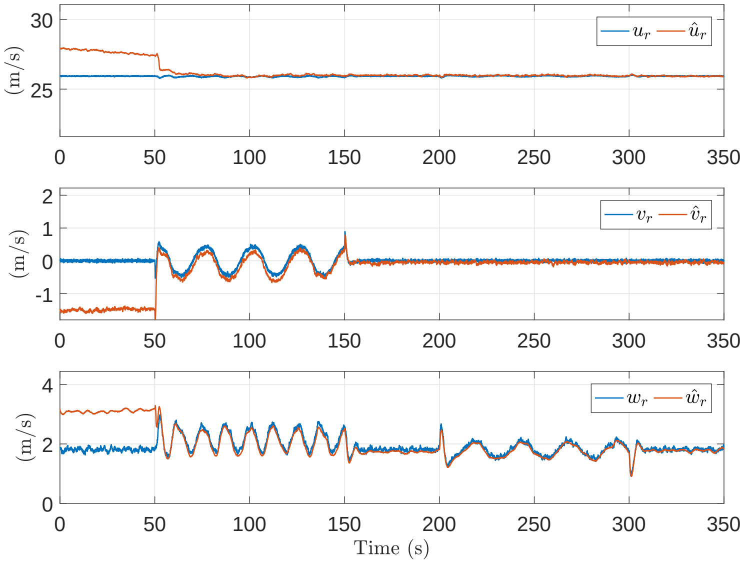


Figure 4.9: The relative velocity obtained from simulation and the corresponding relative velocity estimator estimates.

Similarly to the wind velocity estimator, the relative velocity estimator does not show convergence of estimates for the first 50 seconds. After the UAV initiates the sinusoidal course pattern, the estimates starts converging towards the true values and does not diverge in the following periods without PE. RMSE results from the simulation period after 100 seconds are shown in table 4.3.

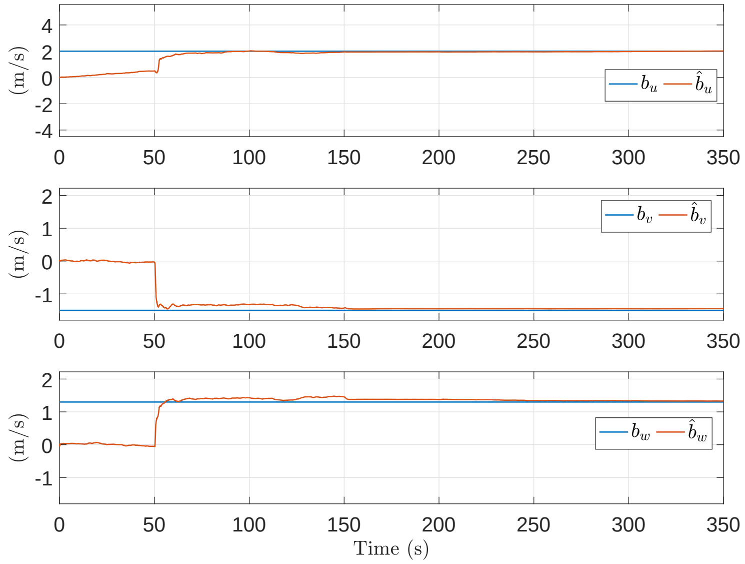


Figure 4.10: The relative velocity sensor bias and the relative velocity estimator bias estimates in simulation.

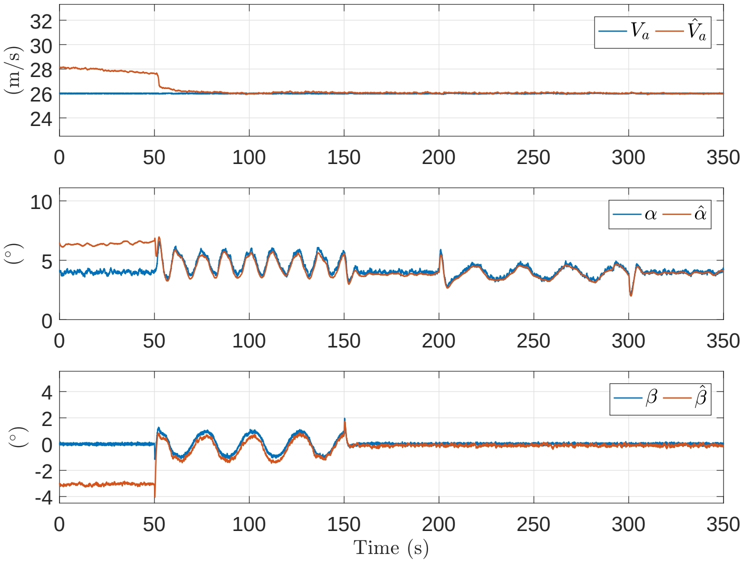


Figure 4.11: The air data obtained from simulation and the air data estimates computed from the relative velocity estimator.

$\tilde{u}_r$	0.0670	m/s
$\tilde{v}_r$	0.0843	m/s
$\tilde{w}_r$	0.0898	m/s
$\tilde{b}_u$	0.0598	m/s
$\tilde{b}_v$	0.0740	m/s
$\tilde{b}_u$	0.0777	m/s

Table 4.3: The RMSE obtained from simulation for the relative velocity estimator after t = 100 seconds.

4.5 Experimental Flight Test Results

This section contains results from testing the air data system bias correction estimators on experimental flight data. The flight data was obtained on the 30th of January, 2017, outside of Agdenes in Norway using a Skywalker X8 Flying Wing UAV (see Appendix A). A STIM300 IMU was used to provide angular rate and specific force measurements at 500 Hz. The distributed pressure sensor approach presented in Chapter 3 has been used as a virtual relative velocity sensor. Three neural networks were trained to provide relative velocity virtual sensor measurements at 20 Hz. As a ground truth reference, the Micro Air Data System by the Aeroprobe Corporation has been used to provide relative velocity measurements at 100 Hz. The sensor measurements has been logged with an accurate timing of less than 10 ns using the SyncBoard [3] (described in Appendix B). In addition, a Pixhawk PX4 Autopilot was used to provide EKF attitude and velocity over ground estimations. The estimators are tested on a flight segment that is 10 minutes long. The relative velocity virtual sensor measurements and the ground truth values are shown in Fig. 4.12 and 4.13. The bias error in the neural network output are assumed to be attributed to one of two things. The BMP280 sensors are subjects to a slowly time-varying drift that a pre-flight calibration process is used to compensate for. This calibration uses the mean of the BMP280 sensor measurements for a shorter period and noise from wind during this process will negatively influence the results. In addition, [12] assumes that the pressure distribution, given enough data, is invariant to ambient conditions, and the amount of data used to train the NNs may not be sufficient.

The flight is spent loitering in a circle with an approximate radius of 100 m, where better PE conditions would have been achieved with sinusoidal patterns or figure eight flying. The position, angular rates, and Euler angles are shown in Fig. 4.14 - 4.16. Also, note that the mounting of the Micro Air Data System probe on the Skywalker UAV is a potential error source, since an error in the alignment of the probe with respect to the definitions of the virtual relative velocity sensor axes will result in estimation errors.

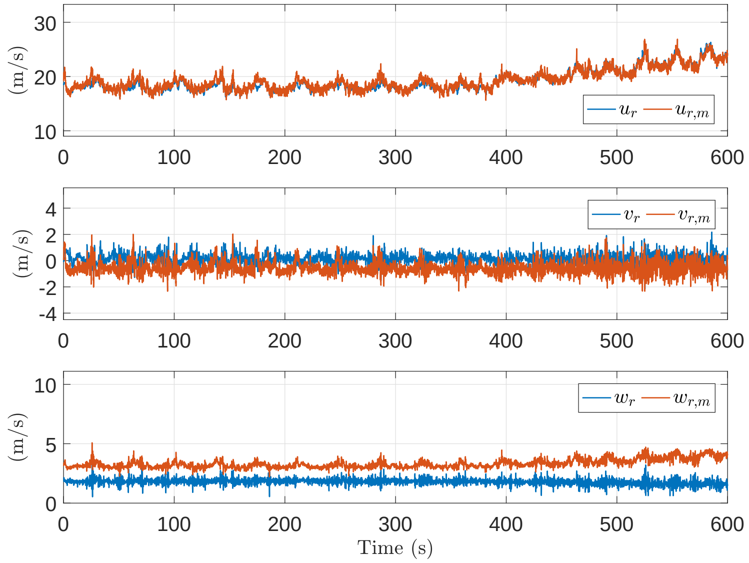


Figure 4.12: The relative velocity virtual sensor measurements and the Micro Air Data System ground truth values. It is apparent that the biases between the two sets of signals are not constant.

4.5.1 Wind velocity estimator flight results

For the experimental flight data, the wind velocity estimator was tuned as $\Sigma_{v_w} = \text{diag}(3 \times 10^{-2}, 3 \times 10^{-2}, 8 \times 10^{-2})$, and $\Sigma_b = 9 \times 10^{-6} I_3$. The error covariance matrix was initialized as $P_{w,0} = \text{diag}(1 \times 10^{-5}, 1 \times 10^{-5}, 1 \times 10^{-5}, 3 \times 10^{-6}, 3 \times 10^{-4}, 3 \times 10^{-4})$. The sensor noise covariance matrices were chosen by tuning as $\Sigma_{v_g} = \text{diag}(1 \times 10^{-2}, 1 \times 10^{-2}, 2 \times 10^{-2})$ and $\Sigma_{v_r} = \text{diag}(6 \times 10^{-3}, 6 \times 10^{-3}, 3 \times 10^{-3})$. The tuning was conducted with a prioritization of steady state performance over fast convergence. The experimental flight test results are shown in Fig. 4.9 - 4.11. The wind velocity estimator RMSE results from the flight after the initial 100 seconds have passed are listed in table 4.4.

$\tilde{V}_a$	0.6911	m/s
$\tilde{\alpha}$	1.1432	deg
$\tilde{\beta} r$	2.2395	deg
$\tilde{v}_n$	0.7254	m/s
$\tilde{v}_e$	0.6804	m/s
$\tilde{v}_d$	0.4364	m/s

Table 4.4: The RMSE obtained from flight data for the wind velocity estimator after t = 100 seconds.

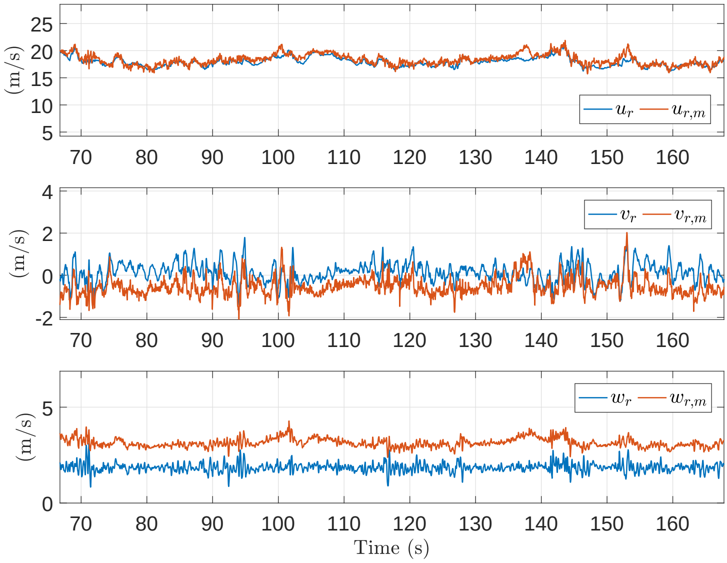


Figure 4.13: A closer look at the virtual sensor relative velocity measurements and the ground truth values. It is evident that the virtual sensor does not output a perfectly biased relative velocity measurement, but it does appear to capture the high-frequency dynamics.

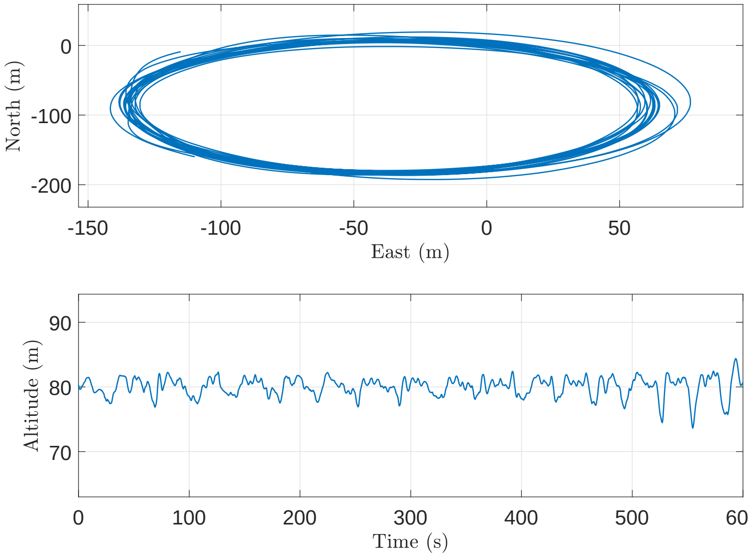


Figure 4.14: The trajectory described by the UAV during the experimental flight.

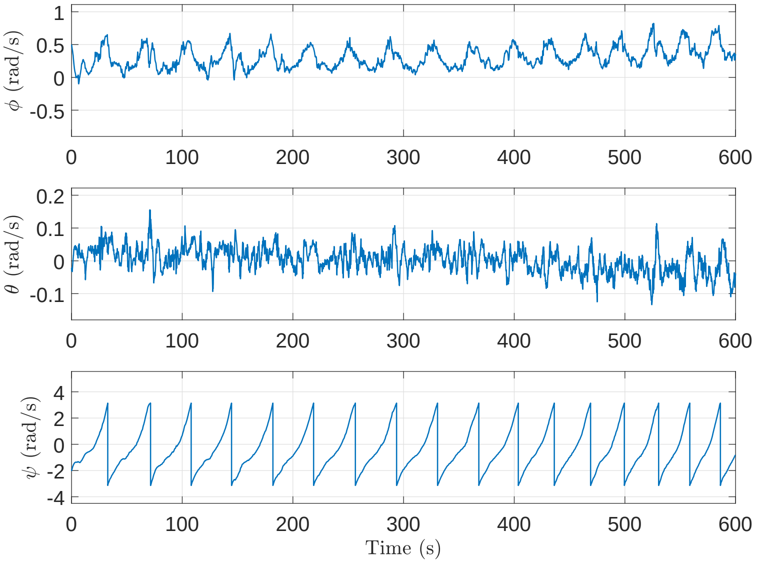


Figure 4.15: The Euler angles obtained from the Pixhawk PX4 Autopilot during the experimental flight test.

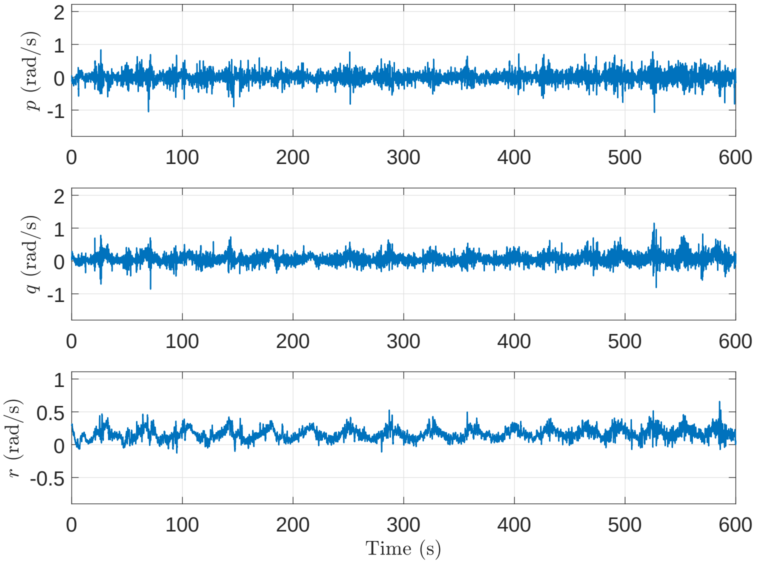


Figure 4.16: The angular rates obtained from the STIM300 IMU during the experimental flight test. It is not obvious whether the angular rates and angular acceleration from this test flight provides sufficient PE for the estimators to have UCO properties.

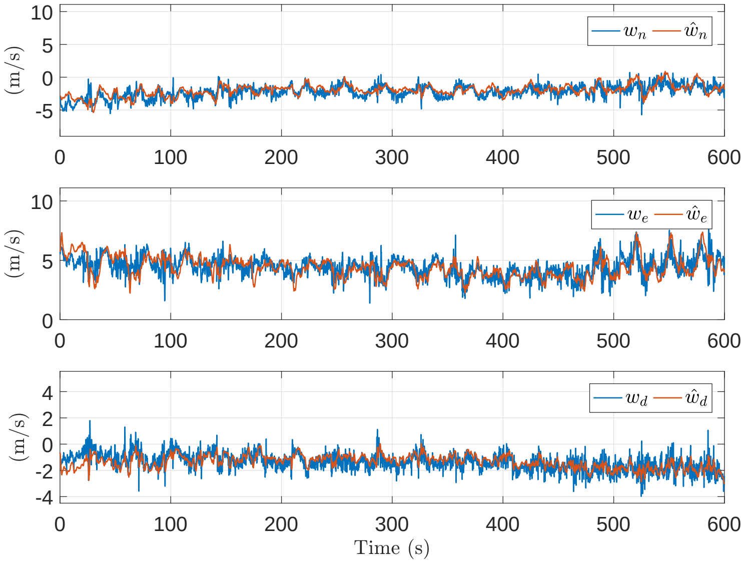


Figure 4.17: The wind velocity computed from from the Micro Air Data System using the relations in Eq. 2.4 and the wind velocity estimates from the wind velocity estimator.

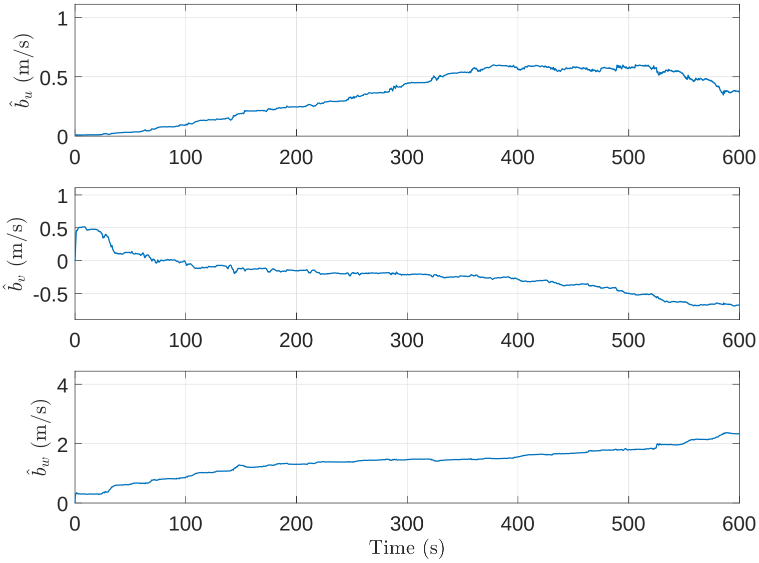


Figure 4.18: The relative velocity sensor bias estimates from the wind velocity estimator.

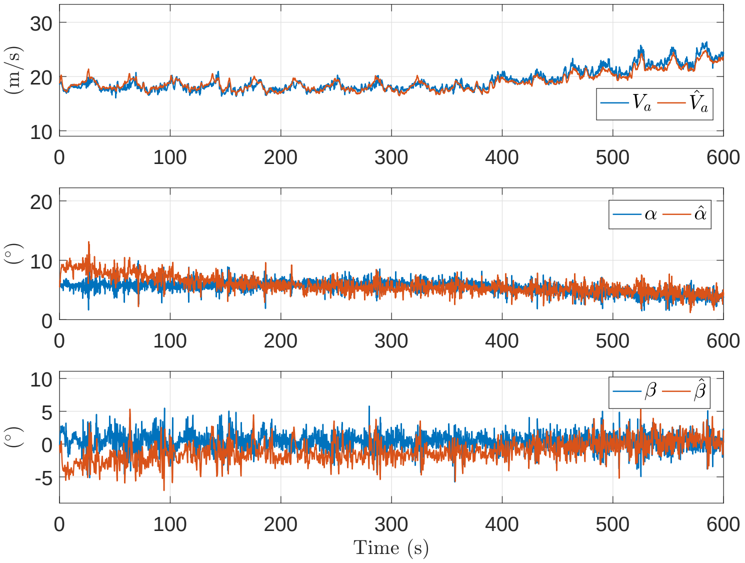


Figure 4.19: The air data obtained from experimental flight and the air data estimates computed from the wind velocity estimator estimates.

4.5.2 Relative velocity estimator flight results

For the experimental flight data, the relative velocity estimator was initialised with the same values for the wind model noise, the bias model noise, the error covariance matrix, and the velocity sensor error covariance as the wind velocity estimator was. The sensor noise covariance matrices were chosen by tuning as $\Sigma_f = 1 \times 10^{-2} I_3$, and $\Sigma_\omega = 1 \times 10^{-3} I_3$. The tuning was conducted with a prioritization of steady state performance over fast convergence. The experimental flight test results are shown in Fig. 4.9 - 4.11. The wind velocity estimator RMSE results from the flight after the initial 100 seconds have passed are listed in table 4.5.

$\tilde{V}_a$	0.7073	m/s
$\tilde{\alpha}$	1.2252	deg
$\tilde{\beta}r$	1.2451	deg
$\tilde{u}_r$	0.7021	m/s
$\tilde{v}_r$	0.4304	m/s
$\tilde{w}_r$	0.4110	m/s

Table 4.5: The RMSE obtained from flight data for the relative velocity estimator after t = 100 seconds.

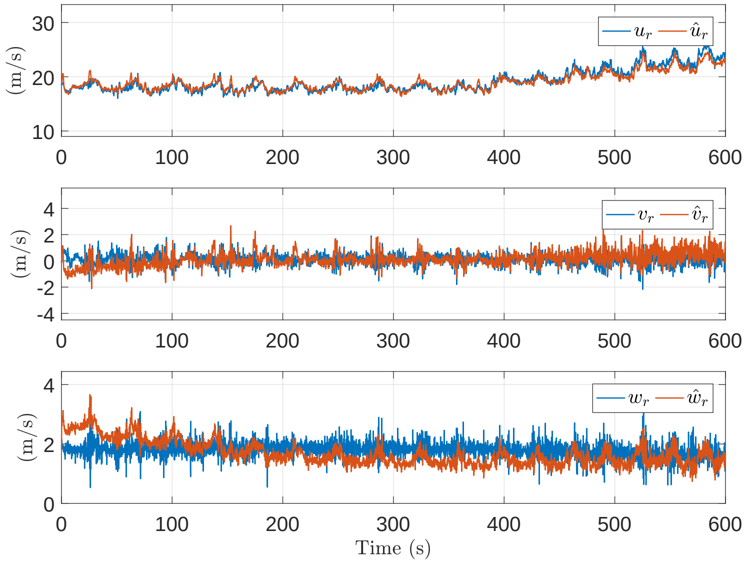


Figure 4.20: The relative velocity obtained from the Micro Air Data System and the relative velocity estimator estimates.

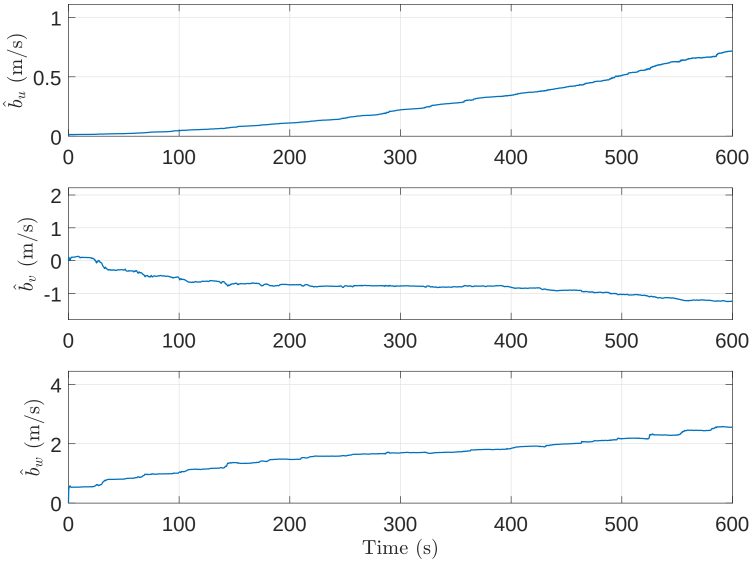


Figure 4.21: The relative velocity sensor bias estimates from the relative velocity estimator.

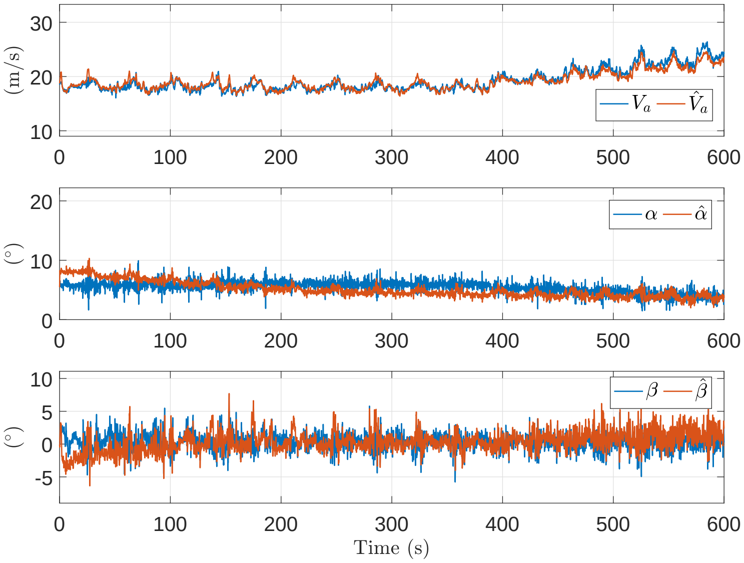


Figure 4.22: The air data obtained from experimental flight and the air data estimates computed from the relative velocity estimator estimates.

Note on faults in sensor input: The method and its results will obviously depend on the faultlessness of the sensor input used. The ground truth sensor, the Micro Air Data System, is factory calibrated and has an accuracy of ±1° on flow angles and a total flow velocity accuracy <1 % or 1 m/s, according to the manufacturer. However, the Micro Air Data System requires annual re-calibration and any faults due to a lack of calibration would result in a faulty comparison with the output of the two estimators. The BMP280 pressure sensors are used after a pre-flight calibration has been conducted, but any drift over time will influence the air data estimation. The influence of this fault on the method depends on both size of sensor drift and the influence of the specific sensor on the ML algorithm output. For sensor drift, the presented method is capable of handling errors that manifests themselves as slowly-varying biases on the virtual relative velocity sensor output. Furthermore, any faults in attitude estimates, GNSS or IMU measurements will negatively influence respectively the wind velocity and relative velocity estimators.

4.6 Chapter Summary

This chapter presented two different Kalman filter approaches for correcting air data systems for a constant error bias. The estimators are designed for fixed-wing unmanned aerial vehicles and use a standard sensor suite consisting of a GNSS receiver, an IMU, and a heading reference, combined with an air data system that is assumed to provide measurements with an unknown additive slowly time-varying bias. The proposed estimators are not model-dependent and are in the absence of noise proven to have globally exponentially stable equilibrium points for the error dynamics if provided with persistence of excitation of the angular rates of the unmanned aircraft. The estimators are verified through simulation and using experimental flight data. The experimental data indicates that a certain amount of PE is necessary to provide converging estimates for turbulent wind conditions.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The contributions and conclusions of the thesis is listed in order of appearance:

Conclusions of Chapter 2 - Nonlinear Model-Based Wind Velocity Observer
In this chapter a nonlinear wind observer for a fixed-wing unmanned aerial vehicle was proposed. The wind observer combines a standard sensor suite consisting of a GNSS receiver, an inertial measurement unit, a Pitot-Static probe, and a heading reference, with model of the aircraft and an airspeed sensor. The observer provides estimates of both the wind velocity and the relative velocity, from which the angle of attack and sideslip angle are computable. The nonlinear wind observer developed does not have any requirements of persistence of excitation of the aircraft, and the nonlinear wind observer has been proven to be exponentially stable. The wind observer is verified through simulations using a realistic wind signal.

Conclusions of Chapter 3 - A Machine Learning Approach to Estimating Air Data
In this chapter, a method for estimating the air data parameters for small fixed-wing unmanned aerial vehicles has been presented. The method compromises a set of low-cost MEMS-based pressure sensors embedded in the surface of the unmanned aerial vehicle combined with machine learning algorithms. The presented method is flexible in placement of sensors and two different machine learning algorithms, neural networks and linear regression, have been implemented and tested. Both algorithms have been evaluated on data obtained through wind tunnel experiments and experimental flight data. The neural network algorithm was found to generally provide a slightly lower estimation error than the linear regression algorithm. However, linear regression allows for basis function expansions that could potentially improve the results further. By comparing the results obtained from using the low-cost sensors on the aircraft against the results from the expensive pressure scanner from the wind tunnel, the influence on sensor amount and sensor accuracy on the results was assessed. Training the machine learning algorithms using only wind

tunnel data was found to have several error sources, namely the blockage effect, misalignment from the aircraft mount, as well as a too sparse (not very rich) training data set. However, trying to account for these error sources showed potential in decreasing the estimation error. Using flight data to train the machine learning algorithms was found to be a feasible approach that allowed estimating the air data parameters for both neural networks and linear regression. Finally, a study of the sensor number and placement influence on the result was conducted along with an assessment of augmenting the system input by using a pseudo Reynolds number.

Conclusions of Chapter 4 - Kalman Filters for Air Data System Bias Correction
This chapter presented two different Kalman filter approaches for correcting air data systems for a constant error bias. The estimators are designed for fixed-wing unmanned aerial vehicles and use a standard sensor suite consisting of a GNSS receiver, an inertial measurement unit, and a heading reference, combined with an air data system that is assumed to provide measurements with an unknown additive slowly time-varying bias. The presented estimators are not model-dependent and are in the absence of noise proven to have globally exponentially stable equilibrium points for the error dynamics if provided with persistence of excitation of the angular rates of the unmanned aircraft. The estimators are verified through simulation and using experimental flight data. The experimental data indicates that a certain amount of PE is necessary to provide converging estimates during turbulent wind conditions.

5.2 Future Work

There are many potential future improvements to the air data estimation methods presented in this thesis. The machine learning approach presented in Chapter 3 has several points that could be improved. During two of the experimental flights conducted, several of the pressure sensors repeatably measured high pressure spikes corresponding to a specific attitude of the aircraft. A deeper look into the aerodynamic theory could perhaps provide a better understanding of the method and how to optimally place the sensors to avoid this sort of behavior. Furthermore, for a new aircraft platform, the need to perform flights with a ground truth sensor or wind tunnel tests in order to gather training data, could potentially be replaced by CFD simulations. The payload could be augmented with a single high-accuracy pressure sensor used for the pre-flight calibration of the low-cost BMP280 pressure sensors, to provide a solution that is more robust to drift of the BMP280 sensors. An algorithm could be designed to detect a pressure sensor failure and in case of this, replace the air data estimation machine learning algorithm with one that uses a different sensor configuration that does not include the faulty sensor. These suggestions could all improve the method presented in Chapter 3, however, it is hard to imagine an end-all air data estimation solution relying on a single method.

A deeper understanding of the presented methods and their shortcomings could be exploited in a solution that combines different observers and estimators into a single component. One such possible combination is shown in Fig. 5.1.

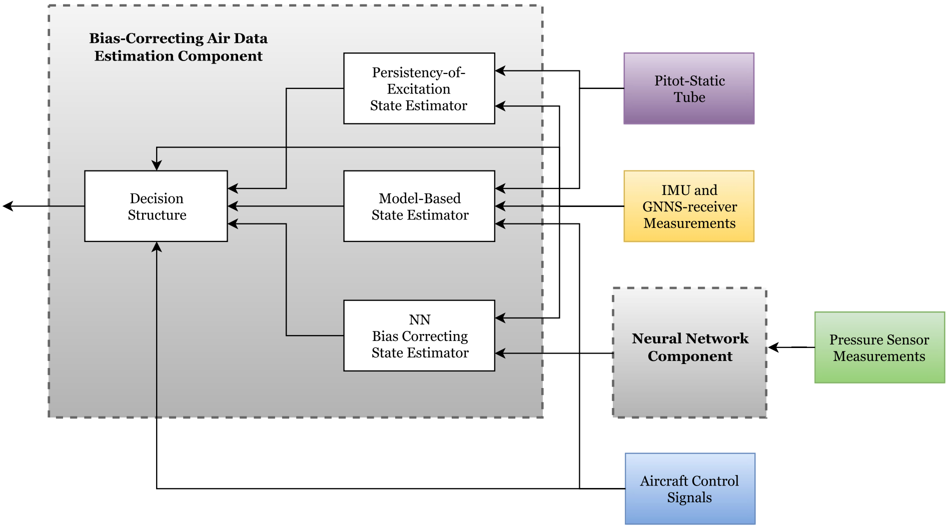

Bias-Correcting Air Data Estimation Component
Persistency-of-Excitation State Estimator
Decision Structure
Model-Based State Estimator
NN Bias Correcting State Estimator
Pitot-Static Tube
IMU and GNNS-receiver Measurements
Neural Network Component
Pressure Sensor Measurements
Aircraft Control Signals

Figure 5.1: A possible combination of different methods for estimating air data combined into a single solution.

The bias-correcting air data estimation component envisioned combines the air data estimation approaches presented in this thesis with the estimator presented in [47]. A decision structure based on the performance of the different methods in different conditions and for different maneuvers is used to switch output estimates of the component. The solution from Fig. 5.1 represents a single possible configuration, however, there are many possible combinations as well as possible improvements to the different subcomponents.

Appendices

Appendix A

Experimental Platform

The experimental results in this thesis have been obtained using a Skywalker X8 Flying Wing UAV. The Skywalker X8 is a consumer grade UAV with a wingspan of 2.12 meters that is usually flown in airspeeds ranging from 15 to 25 m/s, and is capable of carrying a payload of up to 2 kg (including the control system, batteries, etc.). A picture of a Skywalker X8 in flight is shown in A.1.

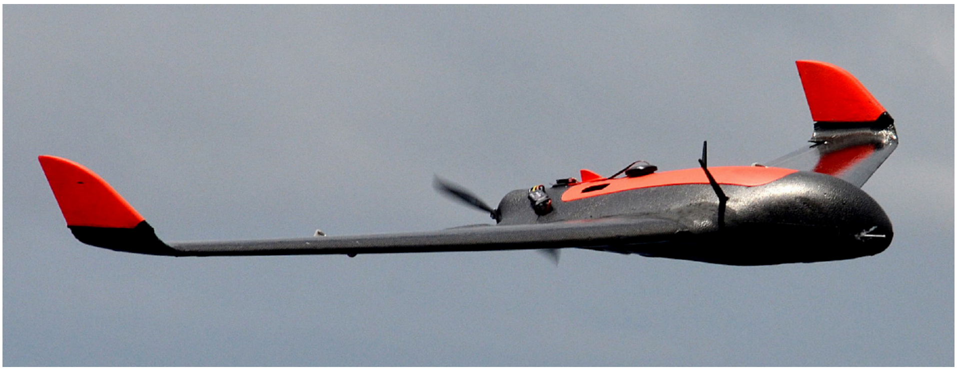


Figure A.1: A Skywalker X8 in flight. Copyright: João Fortuna.

Appendix B

Data Logging Payload

This appendix outlines the method used for logging the experimental flight data used in Chapter 3 and 4. The backbone of the payload consists of the SyncBoard designed by Sigurd Mørkved Albrektsen, NTNU [3]. The SyncBoard is a reconfigurable sensor timing board designed for accurate timing of sensor measurements with an accuracy less than 10 ns. The SyncBoard has been designed to be compatible with different commonly used communcation protocols and uses an onboard microcontroller interrupt capture function combined with time of validity signals to achieve the high accuracy. The SyncBoard is shown in Fig. B.1.

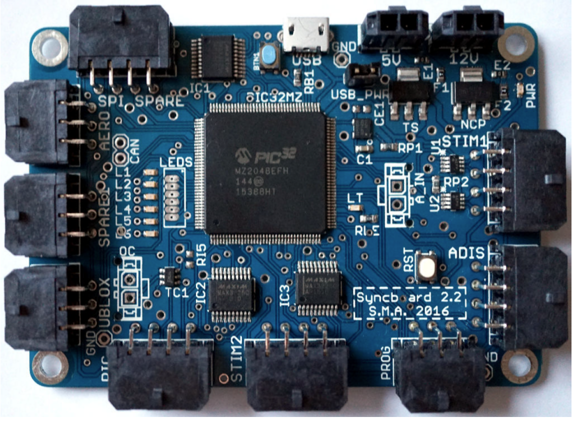


Figure B.1: The topside of the SyncBoard version 2.2. The image is from [3].

The payload used during experimental flights consists of:

- **STIM300:** A tactical grade IMU used to provide measurements of specific force and angular rate at 500 Hz.
- **u-blox LEA-M8T:** GNSS receiver capable of logging position, velocity, range data, and environmental data.
- **Micro Air Data System:** A 5-port probe air data system with an accuracy of ±1° and a total flow velocity $<$ 1% or 1 m/s (whichever is largest) [2].

- **SyncBoard:** A custom printed circuit board designed and built by Sigurd Mørkved Albrektsen, NTNU.
- **ODROID-XU4:** A single-board computer used for handling, parsing and storing the data passed from the SyncBoard onto an SD card.
- **16 Bosch BMP280:** MEMS-based digital pressure and temperature sensors embedded in the surface of the Skywalker X8 UAV.
- **Arduino MEGA 2650:** Microcontroller board connected to the BMP280 pressure sensors via serial peripheral interface bus. The MEGA 2650 collects the pressure measurements and sends them in a package to the ODROID-XU4 component.
- **Pixhawk PX4:** The autopilot used on the Skywalker X8. The PX4 measurements and estimates are accurately synchronized to the rest of the payload using GPS timestamps.

The STIM300, the LEA-M8T, and the Micro Air Data System are directly connected to the SyncBoard. However, the BMP280 pressure sensor measurements are not logged using the accurate timing stamps of the SyncBoard and are related to the other sensors by the timing stamps of the Odroid.

References

[1] O. M. Aamo, M. Arcak, T. I. Fossen, and P. V. Kokotovic. Global output tracking control of a class of Euler-Lagrange systems with monotonic nonlinearities in the velocities. *International Journal of Control*, vol 74, pp. 649-658, 2001.

[2] Aeroprobe. Micro Air Data System [measurement system]. Available from http://www.aeroprobe.com/micro-air-data-computer-models/.

[3] S. M. Albrektsen and T. A. Johansen. SyncBoard - A high accuracy sensor timing board for UAV payloads. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017.

[4] B. D. O. Anderson. Stability Properties of Kalman-Bucy Filters. *Journal of the Franklin Institute*, vol. 2, pp. 137-144, 1971.

[5] J. D. Anderson Jr. *Fundamentals of Aerodynamics: Fifth Edition.* McGraw-Hill, 2011.

[6] R. W. Beard and T. W. McLain. *Small Unmanned Aircraft - Theory and Practice.* Princeton University Press, 2012.

[7] R. Benney, J. Barber, J. McGrath, J. McHugh, G. Noetscher, and S. Tavan. The New Military Applications of Precision Airdrop Systems. In *Infotech@Aerospace*, 2005.

[8] C. M. Bishop. *Pattern Reckognition and Machine Learning.* Springer, 2006.

[9] M. Blades. Analysis of the Global Commercial UAS Market. Technical report, Frost & Sullivan, 2015.

[10] R. Bogue. MEMS sensors: past, present and future. *Sensor Review*, vol. 27, pp. 7-13, 2007.

[11] K. T. Borup, T. I. Fossen, J. Braga, and J. Borge de Sousa. Nonlinear observer for depth-aided INS: Experimental evaluation using an AUV. In *2014 22nd Mediterranean Conference of Control and Automation (MED)*, 2014.

[12] K. T. Borup, T. I. Fossen, and T. A. Johansen. A Machine Learning Approach for Estimating Air Data Parameters for Small Fixed-Wing UAVs Using Distributed Pressure Sensors. Submitted to IEEE Transactions on Aerospace and Electronic Systems. A preprint is available here.

[13] K. T. Borup, T. I. Fossen, and T. A. Johansen. A Nonlinear Model-Based Wind Velocity Observer for Unmanned Aerial Vehicles. In *Non-linear Control Systems - 10th NOLCOS 2016*, 2016.

[14] K. T. Borup, B. N. Stovner, T. I. Fossen, and T. A. Johansen. Kalman Filters for Air Data System Bias Correction for a Fixed-Wing UAV. Submitted to IEEE Transactions on Control Systems Technology. A preprint is available here.

[15] R. G. Brown. Integrated navigation systems and Kalman filtering: a perspective. *Journal of the Institute of Navigation*, 19:355–362, 1972.

[16] K. P. Burnham and D. R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, 2nd Edition.* Springer, 2011.

[17] S. Callegari, A. Talamelli, M. Zagnoni, A. Golfarelli, V. Rossi, M. Tartagni, and E. Sangiorgi. Aircraft Angle of Attack and Air Speed Detection by Redundant Strip Pressure Sensors. *SENSORS, 2004 IEEE*, 2004.

[18] J. Camhi. THE DRONES REPORT. Technical report, BI Intelligence, 2016.

[19] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch. Rigid-Body Attitude Control. *IEEE Control Systems*, vol. pp. 30-51, 2011.

[20] C. Chen. *Linear System Theory and Design, 4th ed.* Oxford University Press, 2013.

[21] A. Cho, Y. Kang, B. Park, and C. Yoo. Airflow angle and wind estimation using GPS/INS navigation data and airspeed. In *2013 13th International Conference on Control, Automation and Systems (ICCAS)*, 2013.

[22] A. Cho, J. Kim, S. Lee, and C. Kee. Wind Estimation and Airspeed Calibration using a UAV with a Single-Antenna GPS Receiver and Pitot Tube. *IEEE Transactions on Aerospace and Electronic Systems*, vol 47, pp. 109-117, 2011.

[23] P. Courrieu. Three algorithms for estimating the domain of validity of feedforward neural networks. *Neural Networks*, vol 7, pp. 169-174, 1994.

[24] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 12-28, 2007.

[25] J. Diebel. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors.* Stanford University, CA, 2006.

[26] M. Erdelj and E. Natalizio. UAV-assisted disaster management: Applications and open issues. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, 2016.

[27] J. A. Farrell. *Aided Navigation: GPS with High Rate Sensors.* McGraw-Hill, 2008.

[28] A. L. Flåten. Experimental Monitoring of Sea Ice Using Unmanned Aerial Systems. Master's thesis, Norwegian University of Science and Technology (NTNU), 2015.

[29] Flying. Qualities of Piloted Aircraft. Technical report, United States of America Department of Defense, 1995.

[30] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* Wiley, 2011.

[31] R. Galeazzi, K. T. Borup, N. K. Niemann, H. Poulsen, and F. Caponetti. Adaptive Backstepping Control of Lightweight Tower Wind Turbine. In *American Control Conference (ACC)*, 2015.

[32] R. W. Gent, N. P. Dart, and J. T. Cansdale. Aircraft Icing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 358(1776):2873–2911, vol 358, issue 1776, pp. 2873-2911, 2000.

[33] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS integration. *Automatica*, vol. 51, pp. 158-166, 2015.

[34] P. D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems.* Artech House, 2013.

[35] 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 *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017.

[36] M. T. Hagan and M. B. Menhaj. Traiining Feedforward Networks With the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, vol. 5 pp. 989 - 993, 1994.

[37] R. Hann, A. Wenz, K. Gryte, and T. A. Johansen. Impact of Atmospheric Icing on UAV Aerodynamic Performance. In *RED-UAS*, 2017.

[38] J. M. Hansen, T. A. Johansen, N. Sokolova, and T. I. Fossen. Nonlinear Observer for Tightly Coupled Integrated Intertial Navigation Aided by RTK-GNSS Measurements. *IEEE Transactions on Control Systems Technology*, vol. PP, pp. 1-16, 2018.

[39] S. Hansen and M. Blanke. Diagnosis of airspeed measurement faults for unmanned aerial vehicles. *IEEE Transactions on Aerospace and Electronic Systems*, 50(1):224–239, 2014.

[40] R. T. Hitt. *Jungle Pilot: The Gripping Story of the Life and Witness of Nate Saint, Martyred Missionary to Ecuador.* Discovery House Publishers, 1997.

[41] J. Hosen, H. H. Helgesen, L. Fusini, T. I. Fossen, and T. A. Johansen. Vision-Aided Nonlinear Observer for Fixed-Wing Unmanned Aerial Vehicle Navigation. *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 1777-1789, 2016.

[42] A. R. Hovenburg, F. A. A. Andrade, C. D. Rodin, T. A. Johansen, and R. Storvold. Contingency Path Planning for Hybrd-electric UAS. In *RED-UAS*, 2017.

[43] A. R. Hovenburg, T. A. Johansen, and R. Storvold. Mission Performance Trade-offs of Battery-Powered sUAS. In *ICUAS*, 2017.

[44] O. Hubová and P. Lobotka. The Multipurpose New Wind Tunnel STU. *Civil and Environmental Engineering*, vol. 10, pp. 1-9, 2014.

[45] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning.* Springer, 2013.

[46] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[47] 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 *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.

[48] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *SPIE Proceedings Series*, vol. 3068, pp. 182-193, 1997.

[49] H. K. Khalil. *Nonlinear Systems, Third Edition.* Prentice Hall, 2002.

[50] V. Klein and E. A. Morelli. *Aircraft System Identification - Theory and Practice.* American Institute of Aeronautics and Astronautics, 2006.

[51] M. Kumon, I. Mizumoto, Z. Iwai, and M. Nagata. Wind Estimation by Unmanned Air Vehicle with Delta Wing. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005.

[52] Jack W. Langelaan, Nicholas Alley, and James Neidhoefer. Wind Field Estimation for Small Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, vol. 34, pp. 1016-1030, 2011.

[53] T. J. Larson and P. M. Siemers III. Subsonic Tests of an All-Flush-Pressure-Orifice Air Data System. *NASA TP-1871*, 1981.

[54] T. J. Larson, S. A. Whitmore, L. J. Ehernberger, J. B. Johnson, and P. M. Siemers III. Qualitative Evaluation of a Flush Air Data System at Transonic Speeds and High Angles of Attack. *NASA TP-2716*, 1987.

[55] R. J. Laurence, B. M. Argrow, and E. W. Frew. Wind Tunnel Results for a Distributed Flush Airdata System. *Journal of Atmospheric and Oceanic Technology*, 2017.

[56] F. A. P. Lie and D. Gebre-Egziabher. Synthetic Air Data System. *Journal of Aircraft*, vol. 50, pp. 1234-1249, 2013.

[57] H. Long and S. Song. Method of estimating angle-of-attack and sideslip angle based on data fusion. In *2009 2nd International Conference on Intelligent Computing Technology and Automation*, 2009.

[58] F. L. Markley. Attitude Error Representation for Kalman Filtering. *Journal of Guidance, Control, and Dynamics*, vol. 26, pp. 311-317, 2003.

[59] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control.* Springer, 2014.

[60] D. W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, pp. 431-441, 1963.

[61] B. Martos and D. F. Rogers. Low Cost Accurate Angle of Attack System. In *55th AIAA Aerospace Sciences Meeting*, 2017.

[62] S. H. Mathisen, V. Grindheim, and T. A. Johansen. Approach Methods for Autonomous Precision Aerial Drop from a Small Unmanned Aerial Vehicle. *IFAC-PapersOnLine*, 2017.

[63] S. H. Mathisen, K. Gryte, T. I. Fossen, and T. A. Johansen. Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing. In *AIAA Infotech @ Aerospace*, 2016.

[64] I. Maza, F. Caballero, J. Capitán, J. R. Martinez-de Dios, and A. Ollero. Experimental Results in Mullti-UAV Coordination for Disaster Management and Civil Security Applications. *Journal of Intelligent & Robotic Systems*, vol 61, pp. 563-585, 2011.

[65] M. Merz and T. A. Johansen. Optimal Path of a UAV Engaged in Wind-influenced Circular Towing. In *RED-UAS*, 2017.

[66] C. D. Meyer Jr. Generalized Inverses and Ranks of Block Matrices. *SIAM Journal on Applied Mathematics*, vol. 25, pp 591-6002, 1973.

[67] D. J. Moorhouse and R. J. Woodcock. MIL-F-8785C. Technical report, Air Force Research Laboratory - Wright-Patterson Air Force Base, 1982.

[68] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012.

[69] P. Paces, K. Draxler, V. Hanzal, T. Censky, and O. Vasko. A combined angle of attack and angle of sideslip smart probe with twin differential sensor modules and doubled output signal. In *2010 IEEE Sensors*, pages 284–289. IEEE, nov 2010.

[70] J. Pinto, Paulo S. Dias, Ricardo Martins, Joao Fortuna, Eduardo Marques, and Joao Sousa. The LSTS toolchain for networked vehicle systems. In *OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension*, 2013.

[71] J. Quindlen and J. Langelaan. Flush Air Data Sensing for Soaring-Capable UAVs. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.

[72] C. Ramprasadh and H. Arya. Multistage-Fusion Algorithm for Estimation of Aerodynamic Angles in Mini Aerial Vehicle. *Journal of Aircraft*, vol. 49, pp. 93-100, 2012.

[73] M. B. Rhudy, M. L. Fravolini, Y. Gu, M. R. Napolitano, S. Gururanjan, and H. Chao. Aircraft model-independent airspeed estimation without pitot tube measurements. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, pp. 1980-1995, 2015.

[74] M. B. Rhudy, T. Larrabee, H. Chao, Y. Gu, and M. Napolitano. UAV Attitude, Heading, and Wind Estimation Using GPS/INS and an Air Data System. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013.

[75] A. Rodriguez, E. Andersen, J. Bradley, and C. Taylor. Wind Estimation Using an Optical Flow Sensor on a Miniature Air Vehicle. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Reston, Virigina, aug 2007. American Institute of Aeronautics and Astronautics.

[76] T. J. Rohloff, S. A. Whitmore, and I. Catton. Fault-Tolerant Neural Network Algorithm for Flush Air Data Sensing. *Journal of Aircraft*, 36(3):541–549, may 1999.

[77] T. J. Rohloff, S. A. Whitmore, and I. Catton. Air Data Sensing from Surface Pressure Measurements Using a Neural Network Method. *AIAA Journal*, vol. 36, pp. 2094-2101, 1998.

[78] D. Rotondo, A. Cristofaro, T. A. Johansen, F. Nejjari, and V. Puig. Diagnosis of Icing and Actuator Faults in UAVs Using LPV Unknown Input Observers. *Journal of Intelligent & Robotic Systems*, pp. 1-15. 2017.

[79] A. Ryan and J. K. Hedrick. A mode-switching path planner for UAV-assisted search and rescue. In *Conference on Decision and Control, CDC*, 2005.

[80] I. Samy, I. Postlethwaite, D. Gu, and J. Green. Neural-Network-Based Flush Air Data Sensing System Demonstrated on a Mini Air Vehicle. *Journal of Aircraft*, vol. 47, pp. 18-31, 2010.

[81] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge. Multipurpose UAV for search and rescue operations in mountain avalanche events. *Geomatics, Natural Hazards and Risk*, vol. 8, pp. 18-33, 2015.

[82] K. L. Sørensen, M. Blanke, and T. A. Johansen. Diagnosis of Wing Icing Through Lift and Drag Coefficient Change Detection for Small Unmanned Aircraft. In *Safeprocess*, 2015.

[83] K. L. Sørensen, S. H.. Helland, and T. A. Johansen. Carbon nanomaterial-based wing temperature control system for in-flight anti-icing and de-icing of unmanned aerial vehicles. In *IEEE Aerospace Conference*, 2015.

[84] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology - 2nd Edition*. The Institution of Enginerring and Technology, 2004.

[85] S. Waharte and N. Trigoni. Supporting Search and Rescue Operations with UAVs. In *2010 International Conference on Emergin Security Technologies (EST)*, 2010.

[86] A. Wenz and T. A. Johansen. Estimation of Wind Velocities and Aerodynamic Coefficients for UAVs using standard Autopilot Sensors and a Moving Horizon Estimator. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017.

[87] A. Wenz, T. A. Johansen, and A. Cristofaro. Combining model-free and model-based Angle of Attack estimation for small fixed-wing UAVs using a standard sensor suite. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.

[88] S. A. Whitmore, R. J. Davis, and J. Fife. In-flight demonstration of a real-time flush airdata sensing system. *Journal of Aircraft*, vol. 33, pp. 970–977, 1996.

[89] F. Zhang. *The Schur Complement and its Applications*. Springer, 2005.