Phillip Larsen Tronstad

Model-Based In-Flight Icing Detection for Fixed-Wing UAVs

Master's thesis in Cybernetics and Robotics
Supervisor: Kristoffer Gryte
Co-supervisor: Bogdan Løw-Hansen, Tor Arne Johansen
July 2023

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Phillip Larsen Tronstad

Model-Based In-Flight Icing Detection for Fixed-Wing UAVs

Master's thesis in Cybernetics and Robotics
Supervisor: Kristoffer Gryte
Co-supervisor: Bogdan Løw-Hansen, Tor Arne Johansen
July 2023

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

NTNU
Norwegian University of
Science and Technology

Abstract

The commercial market has witnessed a remarkable surge in the prevalence of small, low-cost unmanned aerial vehicles (UAVs) in recent times. However, this rapid growth raises a crucial concern: ensuring the safe operation of these UAVs. Among the various hazards they face, in-flight icing emerges as a significant threat that can seriously impact the performance and stability of UAVs. Consequently, it poses an escalating risk to their overall safety. To address this issue, the primary objective of this master's thesis is to devise a cost-effective and efficient methodology for model-based in-flight icing detection specifically designed for fixed-wing UAVs.

The research begins with the design of preliminary system identification experiments optimized for the identification of the dynamics at the dominant frequencies in the lateral and longitudinal modes. The data gathered is analyzed by utilizing conventional aircraft system identification algorithms such as the equation-error method and the output error method. These methods are designed to identify stability and control derivatives of the Skywalker X8 fixed-wing UAV in order to model its dynamics in normal and icing conditions. Despite the basic sensor suite and strong winds during the flight experiments, the equation-error method resulted in models with R^2 scores of around 0.6 when compared with the observed dynamics.

The methodology investigated in this thesis combines the identified models of clean and iced airframes with a multiple-model adaptive estimation framework. This is driven by a bank of extended Kalman filters (EKFs) in order to detect and quantify the presence of icing in flight. Due to the UAV crashing during flight experiments, the iced airframe model was not identified. Instead, the icing detection scheme proved its potential in simulations using other sets of coefficients. It was able to correctly detect and quantify icing levels in a selection of simulated scenarios of flight in icing conditions, identifying the correct icing severity after about 5 s on average.

Sammendrag

Det kommersielle markedet har vært vitne til en bemerkelsesverdig økning i forekomsten av ubemannede fly (UAV-er) i nyere tid. Denne raske veksten har ført til økende bekymring rundt sikker bruk av UAV-er. Blant de ulike farene knyttet til bruken av UAV-er, er ising under flygning en trussel som kan få alvorlige konsekvenser for UAV-ers ytelse og stabilitet. Målet med denne masteroppgaven er å utvikle en modellbasert metode for å detektere ising på UAV-er under flyging.

Det første steget i denne prosessen omhandler utformingen av optimaliserte eksperimenter der hensikten er å samle data som best beskriver den laterale og longitudinale dynamikken. Dataen som samles inn, analyseres ved hjelp av konvensjonelle metoder for systemidentifikasjon av flydynamikk, som equation-error-metoden og output-error-metoden. Disse metodene anvendes til å estimere stabilitets- og kontrollderivatene til Skywalker X8 UAV-en, som modellerer den underliggende dynamikken i nominell og iset tilstand. Til tross for den primitive sensorpakken ombord på UAV-en og den sterke vinden under flyeksperimentene, resulterte modellene fra equation-error metoden i R^2-verdier på rundt 0,6 sammenliknet med den observerte dynamikken.

Metoden som brukes i denne masteroppgaven kombinerer de identifiserte modellene av flyet med og uten ising med en multi-modell adaptiv estimeringsstrategi. Denne metoden er drevet av en bank av utvidede Kalman-filtre (EKF-er) for å detektere og kvantifisere graden av ising under flyging. På grunn av at UAV-en krasjet og brant opp under forsøket, ble ikke den isede flymodellen identifisert. I stedet viste isdeteksjonsmetoden sitt potensial i simuleringer ved bruk av tidligere identifiserte koeffisienter. Estimatoren var i stand til å oppdage og kvantifisere isingsnivåer i ulike simulerte flyscenarier under isede forhold og identifisere riktig alvorlighetsgrad av ising med en gjennomsnittlig forsinkelse på ca. 5 s.

Preface

This thesis is submitted as a part of the requirements for the Master's degree at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology.

The master's thesis is a continuation of my specialization project [1] from the autumn semester of 2022. As the project thesis is not published, the theoretical background concerning UAV dynamics and aircraft system identification, as well as parts of the method will be restated in full.

I would like to thank my supervisors; Kristoffer Gryte for giving his feedback and guidance during the project thesis and for proposing the initial project idea, Bogdan Løw-Hansen for consistently finding time to offer advice and assistance at every stage of the project, and Professor Tor Arne Johansen for his willingness to step in as a supervisor in this project. Additionally, I would also like to thank Erlend Magnus Lervik Coates and Dirk Peter Reinhardt whose work and assistance with the Skywalker X8 hardware and onboard software were instrumental to the progress of this research. Furthermore, I would like to thank drone pilots Pål Kvaløy and Artur Zolich for their help with the execution of the flight experiments.

Phillip Larsen Tronstad

Trondheim, July 2023

Contents

Acronyms

ADS air data system.

AHRS attitude and heading reference system.

AOA angle of attack (α).

CFD computational fluid dynamics.

CG center of gravity.

CIFER Comprehensive Identification from FrEquency Responses.

DOF degrees of freedom.

DUNE DUNE: Unified Navigation Environment.

EE equation-error.

EKF extended Kalman filter.

FD frequency-domain.

FNN feedforward neural network.

GNA adaptive Gauss-Newton.

GNSS global navigation satellite system.

GOF goodness of fit.

IMC inter-module communication.

IMU inertial measurement unit.

INS inertial navigation system.

KF Kalman filter.

LSTS Laboratório de Sistemas e Tecnologia Subaquática.

ML maximum likelihood.

MMAE multiple model adaptive estimation.

NARX nonlinear autoregressive network with exogenous inputs.

NED north-east-down.

NEES normalized estimation error squared.

NIS normalized innovations squared.

ODE ordinary differential equation.

OEM output error method.

OLS ordinary least squares.

PE persistence of excitation.

PEM prediction error minimization.

PWM pulse width modulation.

RC radio-control.

RMSE root-mean-square error.

RTL return to launch.

SBC single board computer.

SINDy sparse identification of nonlinear dynamics.

SSA side-slip angle (β).

SVD singular value decomposition.

TD time-domain.

TIC Theil's inequality coefficient.

UAV unmanned aerial vehicle.

UKF unscented Kalman filter.

Chapter 1

Introduction

The first chapter of this thesis will provide the motivation behind the exploration of model-based in-flight icing detection for fixed-wing UAVs and why it is a relevant research topic. Recent accomplishments in the research community and the current state of the art are presented and reviewed. Finally, the research objective is defined, and the contributions made toward it throughout the study are highlighted. Section 1.5 provides a concise overview of the remaining components outlined in the thesis.

1.1 Motivation

UAVs have gained immense popularity in recent years due to their adaptability and extensive uses in a variety of industries. UAVs cover a wide range of aircraft that are remotely controlled or autonomously operated without a human onboard. These range from small handheld drones to large fixed-wing aircraft with long-range capabilities. UAVs are mostly known for their uses in military surveillance and reconnaissance, aerial photography, scientific research and agriculture, and more recently, in the emerging field of drone delivery services. The latter technology allows autonomous drone operation in populated areas, posing a direct threat to the safety of customers unless all failure modes of the drones have been properly addressed.

The usage of UAVs in adverse weather conditions can pose serious safety risks, especially in cold climates. One significant challenge is the accumulation of ice on the airframe due to encounters with supercooled water droplets. This phenomenon modifies the shape of the airframe and adds mass, altering the aerodynamic capabilities of the UAV, degrading performance. This is especially problematic for small aircraft due to their absolute size [2]. Hence, autonomous operation of UAVs beyond visual line of sight is not common if there is a risk of icing.

With manned aerial vehicles, such as passenger airplanes, ice is removed and prevented in-flight by routing bleed air from the combustion engines to the wings [3]. For most small fixed-wing UAVs, the propulsion system is electric and does not generate enough excess thermal energy for ice shedding and anti-icing. Instead, anti-icing systems for electric fixed-wing UAVs require valuable battery power for thermocouples [4], directly reducing flight time. Autonomous operation in icing conditions would instead need compensation in terms of model-based adaptive control laws, capable of handling changes in the underlying dynamics of the UAV. A necessary condition for this is a reliable icing detection system that can estimate model parameters in icing conditions in real-time.

For many years, commercial manned aircraft have utilized various specialized systems to detect icing. These systems can range from visually observing ice accumulation on a specific probe located outside the cockpit window to sensors that detect changes in the airframe's resonant frequency when in icing conditions [5]. However, these icing detection systems typically require pilot intervention or are specialized and costly. This poses a challenge for the emerging field of commercial, low-cost UAVs equipped with basic sensor suites. Instead, indirect ice detection systems based on the changing dynamics in icing conditions are currently being researched as a low-cost solution to the problem. This will be reviewed in the next section.

This thesis aims to contribute to the recently emerging icing detection and anti-icing research field. The particular issue examined is the model-based in-flight icing detection of fixed-wing UAVs using low-cost sensor suites. The specific UAV used in this thesis is the Skywalker X8 fixed-wing UAV, shown Figure 1.1.

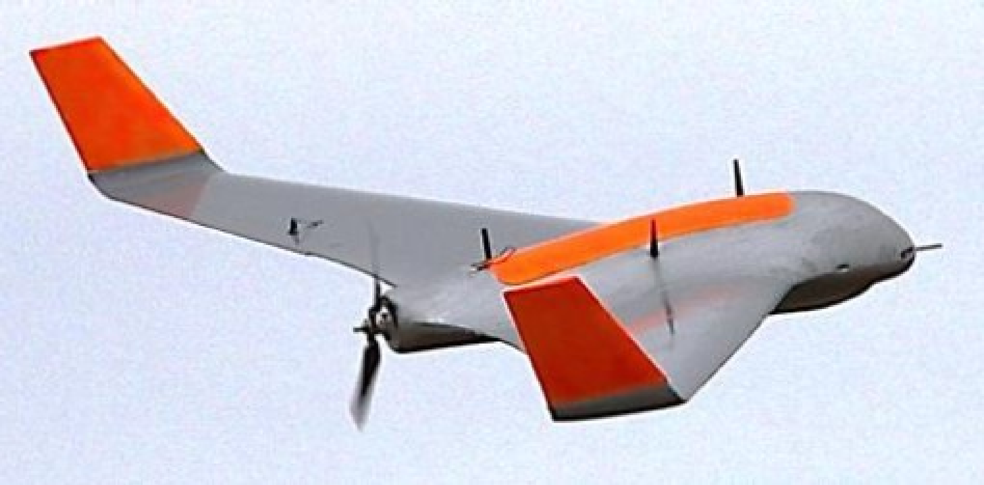

Figure 1.1: Skywalker X8 flying wing UAV. Source: NTNU UAV-Lab

1.2 Literature Review

As this research aims at developing a model-based approach to icing detection, the underlying equations and parameters defining the aircraft dynamics have to be identified. Literature regarding system identification of fixed-wing UAVs is particularly relevant for this. This section will review and present the current state of the art and how it has been tied to the problem of detecting ice accumulation on the airframe. The main sources of information on system identification in this thesis are Klein and Morelli [6] and Jategaonkar [7]. Modeling of aircraft dynamics necessary to perform system identification is well described in Beard and McLain [8], which is the primary source for the modeling in this thesis.

System identification of fixed-wing aircraft is a widely covered topic in the literature, with methods ranging from computational fluid dynamics (CFD) to neural network-based approaches. To better understand the applications and requirements of the various methods, the most interesting aspects of recent research are summarized in Table 1.1. The survey table highlights which payloads are mainly used for the different identification methods and which parameters in the dynamic model they are able to estimate. The input signal chosen for experimental flights determines the information contained in the measured dynamic response and is also included in the table. Note that system identification of fixed-wing UAVs is mainly performed with time-domain (TD) methods using multistep signals as control surface inputs. This is the most viable option when the system dynamics can be approximated, such that the frequency content of the multistep input signals can be optimized for the frequency response of the dynamic modes [6, 7]. Another approach is to use frequency sweep signals as input to the control surfaces. This provides a wide band of frequencies in the input signal but is not optimized for the aircraft dynamics, yielding less accurate identification estimates. However, this is a good approach when there is little to no *a priori* information about the system dynamics [6]. Frequency sweeps are mainly used with frequency-domain (FD) system identification methods.

When the desired frequencies have been excited in the preliminary model, the output data has to be measured with the tools necessary to describe and reconstruct the key dynamics of the aircraft properly. The most primitive sensor suite that has been successfully used with system identification of both longitudinal and lateral dynamics of fixed-wing UAVs consists of global navigation satellite system (GNSS), inertial measurement unit (IMU), and an air pressure sensor for measuring the airspeed. The latter sensor varies a lot in air measuring capabilities, from simple pressure sensors to advanced air data systems (ADSs). Simple pressure sensors, such as pitot-static tubes, are only capable of measuring the relative air velocity in the body x-direction (u_r), while 5-hole ADSs can also measure the angle of attack (α) (AOA) and side-slip angle (β) (SSA). Many avionic configurations, such as in [9, 10, 11, 12] also include an inertial navigation system (INS) based on EKFs for estimating the Euler angles. Others [13, 14, 15] employ an attitude and heading reference system (AHRS) for estimating the Euler angles.

Verification of the system identification varies a lot, from the general root-mean-square error (RMSE) to Theil's inequality coefficient (TIC), which is frequently used in the validation of simulated models. Klein and Morelli [6] use the t-statistic, standard error, and the R^2-score to evaluate all system identification results. As this thesis mainly focuses on data gathered from real flight experiments, similar metrics to Klein and Morelli are more relevant.

The NTNU UAV Icing Lab works with research in a multitude of topics concerning icing prevention, detection, and removal [16, 17, 18]. Wenz and Johansen [19] studied the application of a moving horizon estimator and unscented Kalman filter (UKF) to estimate lift coefficients for icing detection. Their payload consisted of a basic sensor suite, and simulations showed that their estimator is able to detect changes in lift coefficients due to icing.

Seron et al. [20] employed a bank of unknown input observers representing the variety in dynamics imposed by various levels of icing severity. It was designed to exploit the varying operating points in icing conditions. The icing severity was estimated with an accuracy inversely proportional to the quantization level by selecting the model with the smallest residual. Their model is based on the linearized longitudinal state-space model developed in Beard and McLain [8, Chapter 5], where the structural effects of icing are added to the state-space model to generate the bank of unknown input observers.

Cristofaro et al. [21] proposed a nested multiple-model framework for icing detection for the Aerosonde UAV. This method was based on a bank of possible system models for different icing severities, and the estimators were based on Krener min-max observers. Their method showed a fast transient response to changes in the unknown parameters.

Rotondo et al. [22] proposed a linear parameter varying proportional integral unknown input observer for detection of icing and actuator faults in UAVs. Their method correctly identified unexpected changes in the aircraft dynamics, both due to actuator fault and icing.

Table 1.1: Survey table on system identification of fixed-wing aircraft.

Author	Platform	Identification Method	Input Signal	Payload	Estimated Parameters	Validation (metric)
Dorobantu et al. [23]	Ultra Stick 25e UAV	FD OEM	Frequency sweep	IMU	$X_{(\cdot)}, Y_{(\cdot)}, Z_{(\cdot)}, L_{(\cdot)}, M_{(\cdot)}, N_{(\cdot)}$	Time-domain (TIC)
Simsek et al. [9]	TAI Pelikan	FD and TD OEM	Frequency sweep (FD), Multisteps (TD)	GNSS, INS (EKF), pressure sensor	$C_{L_{(\cdot)}}, C_{m_{(\cdot)}}, C_{Y_{(\cdot)}}, C_{l_{(\cdot)}}, C_{n_{(\cdot)}}$	Time-domain (TIC)
Scheper et al. [24]	Multiplex Twinstar II	TD OEM	Multisines	GNSS, IMU, pitot-static tube	$X_{(\cdot)}, Y_{(\cdot)}, Z_{(\cdot)}, L_{(\cdot)}, M_{(\cdot)}, N_{(\cdot)}$	N/A
Chowdhary et al. [10]	HFB-320 [a]	TD EKF, UKF, OEM	3-2-1-1, Step	INS, GNSS, ADS	$C_{D_{(\cdot)}}, C_{L_{(\cdot)}}, C_{m_{(\cdot)}}$	N/A
Tian et al. [25]	WVU Phastball UAV, Khawk 55" UAV	TD LS, EKF, CF	Doublets	GNSS, IMU, ADS, flow vanes	α, β	ADS measurements
Arifianto et al. [13]	Hobby Express Telemaster	TD OEM	3-2-1-1, 1-2-1, doublets	GNSS, AHRS, ADS	$C_{X_{(\cdot)}}, C_{Y_{(\cdot)}}, C_{Z_{(\cdot)}},$ $C_{l_{(\cdot)}}, C_{m_{(\cdot)}}, C_{n_{(\cdot)}}$	TD Multisteps
Ahsan et al. [26]	Taurus RC UAV [b]	PEM, GNA (OEM)	3-2-1-1, doublets	GNSS, IMU, Compass, Pressure Sensor	$C_{Y_{(\cdot)}}, C_{l_{(\cdot)}}, C_{n_{(\cdot)}}$	TD % match
Ahmad et al [27]	Taurus RC UAV [b]	FD LS	Doublets	GNSS, IMU, Compass, Barometer	$\mathcal{H}_{\phi/\delta_a}(s), \mathcal{H}_{p/\delta_a}(s), \mathcal{H}_{r/\delta_a}(s),$ $\mathcal{H}_{v/\delta_a}(s)$	TD % match
McGrail [28]	WVU Phastball UAV	TD OEM, TD LS, FD LS	Multisines	GNSS, IMU, pitot tube, Gyroscope, Pressure sensors	$C_{L_{(\cdot)}}, C_{D_{(\cdot)}}, C_{m_{(\cdot)}},$ $C_{Y_{(\cdot)}}, C_{l_{(\cdot)}}, C_{n_{(\cdot)}}$	TD/FD Multisines
Simmons et al. [14]	HobbyKing Bix3	TD EE and OEM	Doublets, 1-2-1	GNSS, IMU, AHRS	$C_{X_{(\cdot)}}, C_{Y_{(\cdot)}}, C_{Z_{(\cdot)}},$ $C_{l_{(\cdot)}}, C_{m_{(\cdot)}}, C_{n_{(\cdot)}}$	Simulation (TIC)
Oliveria et al. [29]	Cessna Citation II [a]	TD OEM, TD Two Step Method	3-2-1-1	GNSS, Accelerometers Gyroscopes, pitot tube	$C_{X_{(\cdot)}}, C_{Z_{(\cdot)}}, C_{m_{(\cdot)}}$	goodness of fit (GOF)
Simsek et al. [11]	TAI Pelikan	CIFER$^{®}$[30] (FD LS and OEM)	Frequency sweep	GNSS, INS, pressure sensor	$Y_{(\cdot)}, Z_{(\cdot)}, L_{(\cdot)}, M_{(\cdot)}, N_{(\cdot)}$	TD Doublets and 1-2-1 (TIC and RMSE)
Venkataraman et al. [31]	Sentera Vireo UAV	TD PEM	Frequency sweep	GNSS, IMU Magnetometer, pitot tube	$X_{(\cdot)}, Y_{(\cdot)}, Z_{(\cdot)}, L_{(\cdot)}, M_{(\cdot)}, N_{(\cdot)}$	TD Frequency sweep (% fit)
Badwan et al. [32]	HobbyKing Bix3	NARX-FNN	N/A	N/A	$\mathbf{W}^1, \mathbf{V}^1, \mathbf{W}^2$ (FNN weight matrices)	Simulation
Rao et al. [33]	E-flite Apprentice S 15e	SIDPAC [6] (TD OEM)	Doublets, 1-2-1, 3-2-1-1	N/A	$Y_{(\cdot)}, Z_{(\cdot)}, L_{(\cdot)}, M_{(\cdot)}, N_{(\cdot)}$	N/A
Grymin et al. [15]	Hobby Express Telemaster	TD Two-step method and EE	Doublets, 1-2-1, 3-2-1-1	GNSS, pressure sensor, AHRS, 5-hole ADS	$C_{X_{(\cdot)}}, C_{Y_{(\cdot)}}, C_{Z_{(\cdot)}},$ $C_{l_{(\cdot)}}, C_{m_{(\cdot)}}, C_{n_{(\cdot)}}$	Simulation (Theil and GOF)
Rhudy et al. [12]	WVU Phastball UAV	UKF	Doublets	GNSS, INS, IMU, pitot tube, wind vanes	α, β	Ground weather station

[a] Manned research plane

[b] Self-made/unique

1.3 Purpose and Objectives

With the insights gained from reviewing recent accomplishments in the field, the research objectives for this thesis can be defined. The main objective of this thesis is to investigate whether icing can be indirectly detected and estimated in flight based on changes in the observed dynamics of UAVs using a basic sensor suite. The thesis focuses on analyzing the dynamics of UAVs in icing conditions and aims to develop a methodology for detecting and estimating the severity of icing using the observed changes in the UAV's dynamics. This is achieved through system identification, which models the differences in dynamics between clean and iced airframes. The system identification is carried out on data gathered from flight experiments where the inputs are designed to optimize the information content in the data based on a preliminary model of the UAV. Based on the insights gained from system identification, an estimation algorithm is then developed and applied to correctly identify the severity of icing encountered during UAV flight.

1.4 Contributions

This thesis consists of three main parts, starting with the design of maneuvers for experimental flights optimizing the information content in the measured response of the UAV dynamics. The second part regards data processing and system identification of the data gathered from experimental flights. Lastly, a multiple model adaptive estimation (MMAE) framework for detecting and quantifying the effects of icing on the UAV dynamics is developed. The contributions made by the work during these stages are:

- An improved interface for simple configuration and execution of flight test maneuvers in-flight using the onboard single-board computer running the DUNE environment through the Neptus infrastructure. This builds on the foundational work accomplished by Coates et al. [34].

- Frequency analyses of dynamic modes in the decoupled longitudinal and lateral dynamics of the Skywalker X8 UAV development platform for the design of optimal inputs for maneuvers. This improves the system identification accuracy by exciting the UAV dynamics at its dominant frequencies, increasing the information content for system identification.

- A multiple-model adaptive estimation scheme employing a bank of EKFs. This leverages the variations in the dynamical system model in icing conditions for in-flight icing detection and icing severity estimation. The variations in the system model are described by the stability and control derivatives, which determine the response to the environment and actuator inputs in flight.

1.5 Thesis Outline

- **Chapter 1** presents the motivation for pursuing the research topic of this thesis and presents the current state of the art and recent relevant accomplishments in the fixed-wing UAV research field.

- **Chapter 2** presents the foundational theoretical material which the research conducted in this thesis builds upon.

- **Chapter 3** describes the development platform and the methods applied in the data acquisition and analysis for the simulations and experiments. This chapter also describes the post-processing techniques applied to the experimental data to synchronize and transform the data into a format compatible with system identification.

- **Chapter 4** presents and discusses the results from the flight experiments conducted at Breivika and from simulations with a multiple-model adaptive estimator for icing detection and severity estimation.

- **Chapter 5** concludes and discusses the experiences gathered about system identification and model-based in-flight icing detection for fixed-wing UAVs.

- **Appendix A** describes the experiment plan and how the maneuvers are executed.

- **Appendix B** contains results from system identification of data from various maneuvers.

Chapter 2

Theoretical Background

The relevant theoretical background for model-based in-flight icing detection for fixed-wing UAVs is presented here, starting with airframe ice formation. The theory on aircraft modeling, coordinate frames and aerodynamic forces is mostly derived from Beard and McLain [8]. Theory on various techniques for system identification is mostly based on Klein and Morelli [6] and Jategaonkar [7]. This chapter is mostly directly taken from the project thesis [1], with some additions related to state estimation and icing detection.

2.1 Airframe Ice Formation

When an aircraft's airframe encounters supercooled water droplets, they freeze upon impact, leading to in-flight icing. The resulting ice accretion takes on different shapes, with rime ice and glaze ice being the two main types of icing observed on the airframe, as well as mixed ice, which is a combination of the two.

At the lowest temperatures, supercooled water droplets freeze almost immediately on impact with the airframe. This causes the water to freeze before fully settling on the airframe, trapping small air pockets between the freezing water droplets, resulting in a rough and opaque surface. This type of icing is called rime ice. As the ice is not allowed to travel along the wing before freezing. It usually occurs only on the tip of the leading edge of the wing, accumulating in a streamlined manner. The implications of rime ice on the airfoil characteristics are therefore limited under normal conditions [35].

Glaze ice, however, occurs at higher temperatures. Unlike rime ice, the water droplets travel along the airframe before freezing gradually, resulting in a smooth and transparent film of ice. This has a tendency to form irregular shapes called horn ice that can significantly degrade the aerodynamics of the airfoil [35], separating the airflow across the airfoil.

In normal icing conditions a combination of these types is often observed.

This is characterized by the streamlined, rough leading edge icing as well as the slowly freezing glaze ice on the surface. Similarly to glaze icing conditions, this combination of ice accretion has the tendency to grow large horns that can have a substantial impact on the aerodynamics [16].

For fixed-wing UAVs, the most critical aspect of ice accretion is on the wings and propeller. The main concern is the degradation of aerodynamic performance due to the changing geometry introduced by the icing, increasing drag and decreasing lift. Due to the relative size of small UAVs, large amounts of airframe ice can shift the center of gravity (CG) and the pitching moment [16]. The geometrical implications of ice accretion can cause the airfoil of a fixed-wing aircraft to suffer from flow separation, leading to lower stall AOAs [16, 35].

Bragg et al. [36] developed an additive model for modeling the effect of icing on the various aerodynamic coefficients. The model is based on the nominal flight conditions. The effect on each coefficient is then modeled in terms of the nominal coefficients $C_{(\cdot)}$, an icing severity parameter η_{ice} and the constant $K_{C_{(\cdot)}}$ which is dependent on each aerodynamic coefficient $C_{(\cdot)}$ and the specific aircraft [36]. The notation $(\cdot)$ is a placeholder for various states and inputs of an aircraft, which will be explained later in this chapter. This model is provided in Equation (2.1).

$$C_{(\cdot)}^{iced} = (1 + \eta_{ice} K_{C_{(\cdot)}}) C_{(\cdot)}^{clean} \tag{2.1}$$

These aerodynamic coefficients are also called stability and control derivatives, as they relate changes in the states and actuator inputs to the forces and moments acting on the aircraft. They are often related to either the longitudinal or lateral dynamics. The longitudinal and lateral aerodynamics will later be assumed separated. With this assumption in mind, Melody et al. [37] state that icing on the tailplane of an aircraft mostly impacts the longitudinal dynamics of the aircraft, while icing on wings primarily affects the lateral dynamics. On conventional aircraft, the elevator is part of the tailplane, while this is part of the fuselage of flying wing-type aircraft. This leads to the initial assumption that icing on flying wing-type aircraft significantly impacts both longitudinal and lateral dynamics, and both dynamics are interesting to examine.

2.2 Airspeed, Angle of Attack, and Sideslip Angle

Inertial forces experienced by an aircraft are related to the velocities and accelerations relative to the inertial frame $\mathcal{F}^i$. However, the aerodynamic forces acting on an aircraft are caused by the relative velocity and direction of the surrounding airmass. It is therefore advantageous to express such quantities relative to the body frame $\mathcal{F}^b$, as this makes the derivation of aerodynamic equations simpler. The relationship can be described by

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix} = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} \tag{2.2}$$

This implies that when wind is absent, $\mathbf{V}_a = \mathbf{V}_g$. Since the wind velocity may have any direction relative to the airframe, the directional air motion is described by the rotation matrix $\mathcal{R}_w^b$, consisting of the AOA α and the SSA β:

$$\mathbf{V}_a = \mathcal{R}_w^b \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\beta\cos\alpha & -\sin\beta\cos\alpha & -\sin\alpha \\ \sin\beta & \cos\beta & -\sin\beta\sin\alpha \\ \cos\beta\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} \tag{2.3}$$

This expression can now be inverted to find an expression for the V_a, AOA, and SSA. These are the quantities that will be used for calculating forces acting on the aircraft and for estimating aerodynamic parameters and are given by

$$\begin{aligned} V_a &= \sqrt{u_r^2 + v_r^2 + w_r^2} \\ \alpha &= \tan^{-1}\left(\frac{w_r}{u_r}\right) \\ \beta &= \sin^{-1}\left(\frac{v_r}{V_a}\right) \end{aligned} \tag{2.4}$$

2.3 UAV Kinematics and Dynamics

2.3.1 State Variables

The kinematic state variables of an aircraft are position ($\mathbf{p}^n$), velocity ($\mathbf{v}_{nb}^b$), attitude ($\boldsymbol{\Theta}_{nb}^n$) and angular velocity ($\boldsymbol{\omega}_{nb}^n$), where

$$\mathbf{p}^n = \begin{bmatrix} p_n & p_e & p_d \end{bmatrix}^\top \tag{2.5a}$$

$$\mathbf{v}_{nb}^b = \begin{bmatrix} u & v & w \end{bmatrix}^\top \tag{2.5b}$$

$$\boldsymbol{\Theta}_{nb}^n = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top \tag{2.5c}$$

$$\boldsymbol{\omega}_{nb}^n = \begin{bmatrix} p & q & r \end{bmatrix}^\top \tag{2.5d}$$

which results in a 6-degrees of freedom (DOF), 12 states dynamic model.

2.3.2 Kinematics

Since the position is often measured in the NED frame, while the translational velocity is expressed in the body frame, a rotation matrix is needed to express the kinematic relationship between the position and velocity.

$$\frac{d}{dt}\begin{bmatrix} p_n \\ p_e \\ p_d \end{bmatrix} = \mathcal{R}_b^v \begin{bmatrix} u \\ v \\ w \end{bmatrix} = (\mathcal{R}_v^b)^\top \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Inserting the rotation matrix using the shorthand notations $c_x \triangleq \cos x$ and $s_x \triangleq \sin x$, yields

$$\begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{p_d} \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ [8, equation 3.1].} \tag{2.6}$$

Similarly to translational kinematics, rotational kinematics needs rotation matrices to express the relationship between position and velocity. The angular velocities are defined in the body frame, while the angular positions are defined in three separate frames. The angles ϕ, θ, and ψ define the coordinate frame transformations from the vehicle frame to the body frame by $\mathcal{R}_{v2}^b$, $\mathcal{R}_{v1}^{v2}$, and $\mathcal{R}_v^{v1}$, respectively. Using this information, we can define the rotational kinematic relationship as

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathcal{R}_{v2}^b(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathcal{R}_{v2}^b(\phi)\mathcal{R}_{v1}^{v2}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

With some reordering, this can be used to express the Euler rates as a function of the Euler angles and rotational velocities in the body frame:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.7}$$

2.3.3 Rigid-Body Dynamics

The rigid-body dynamics of the UAV can be derived from Newton's second law on translational and rotational degrees of freedom. Translational motion is described by the relationship

$$m\left(\frac{d\mathbf{V}_b}{dt_b} + \omega_{b/i}^b \times \mathbf{V}_b\right) = \mathbf{f}_b \tag{2.8}$$

where m is the aircraft's mass, $\omega_{b/i}^b$ represents the angular velocity of the UAV in the inertial frame, and $\mathbf{V}_b$ is the ground speed in $\mathcal{F}^b$. $\mathbf{f}_b = [f_x, f_y, f_z]^\top$

is the sum of forces acting on the UAV in each of the components of $\mathcal{F}^b$. The term $\frac{d\mathbf{V}_b}{dt_b}$ represents the accelerations in each direction and can be written as $\frac{d\mathbf{V}_b}{dt_b} = [\dot{u}, \dot{v}, \dot{w}]^\top$. $\omega^b_{b/i}$ is the vector of angular rates in the body frame $[p, q, r]^\top$. With this information, Equation (2.8) can be rearranged to derive an expression for the acceleration components in $\mathcal{F}^b$ as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{2.9}$$

Similarly, rotational motion is expressed by applying Newton's second law for rotational motion about the CG of the UAV.

$$\frac{d\mathbf{h}_b}{dt_b} + \omega^b_{b/i} \times \mathbf{h}_b = \mathbf{m}_b \tag{2.10}$$

where $\mathbf{h}_b \triangleq \boldsymbol{I}\omega^b_{b/i}$ is the angular velocity vector and $\mathbf{m}_b$ is the sum of external moments. Here, $\boldsymbol{I}$ is the inertia matrix. The inertia matrix for the Skywalker X8 used in this thesis was found by Gryte [38]:

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} = \begin{bmatrix} 0.335 & 0 & -0.029 \\ 0 & 0.140 & 0 \\ -0.029 & 0 & 0.400 \end{bmatrix} \tag{2.11}$$

The off-diagonal elements related to the y-axis are zero, which follows from the assumption that the Skywalker X8 is symmetric about the x-z plane. Inserting this information into Equation (2.10) and rearranging to express rotational accelerations as a function of inertia and external forces, we get

$$\dot{\omega}^b_{b/i} = \boldsymbol{I}^{-1}[-\omega^b_{b/i} \times (\boldsymbol{I}\omega^b_{b/i} + \mathbf{m}^b)] \tag{2.12}$$

Inverting the inertia matrix $\boldsymbol{I}$ results in a cumbersome expression, which can be written more elegantly by defining a set of constants representing different combinations of inertial constants. In the aerodynamics literature, it is common to use the constants

$$\begin{aligned} \Gamma_1 &= \frac{I_{xz}(I_{xx} - I_{yy} + I_{zz})}{\Gamma} & \Gamma_2 &= \frac{I_{zz}(I_{zz} - I_{yy}) + I_{xz}^2}{\Gamma} \\ \Gamma_3 &= \frac{I_{zz}}{\Gamma} & \Gamma_4 &= \frac{I_{xz}}{\Gamma} \\ \Gamma_5 &= \frac{I_{zz} - I_{xx}}{I_{yy}} & \Gamma_6 &= \frac{I_{xz}}{I_{yy}} \\ \Gamma_7 &= \frac{(I_{xx} - I_{yy})I_{xx} + I_{xz}^2}{\Gamma} & \Gamma_8 &= \frac{I_{xx}}{\Gamma} \end{aligned} \tag{2.13}$$

with $\Gamma = I_{xx}I_{zz} - I_{xz}^2$, we can write the rotational accelerations as

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_5 n \\ \frac{1}{I_{yy}} m \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix} \tag{2.14}$$

2.4 Forces and Moments

In this subsection the various forces and moments effecting a UAV during flight are described. This includes the effects of gravity, aerodynamics and the propulsion system.

2.4.1 Gravitational Forces

The gravitational force acting on the UAV in the vehicle frame is simply $\mathbf{f}_d = mg$ along the $\mathbf{k}^v$ axis, assuming the earth's CG is located in the earth center. The gravitational force is often expressed in the body frame using the transformation $\mathcal{R}_v^b$:

$$\mathbf{f}_g^b = \mathcal{R}_v^b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{bmatrix}$$

2.4.2 Aerodynamic Forces and Moments

The lift, drag, and moment generated by the airfoil are dependent on the dynamic pressure around the wing. The dynamic pressure is given by $\frac{1}{2}\rho V_a^2$, where ρ is the air density and V_a is the airspeed. A key assumption when developing the dynamic model of a UAV is that the longitudinal and lateral dynamics are decoupled. In practice, this is not true, but it is widely used in the literature and is generally accepted as a tolerable approximation when performing mild maneuvers [8, 6, 39, 40]. This simplifies the equations for the aerodynamic forces and moments in equations (2.15) and (2.19), and also allows connecting lateral and longitudinal maneuvers to their respective dynamical models, as described later in Chapter 3. The longitudinal aerodynamic model consists of the forces and moments that impact translational motion along the $\mathbf{i}^b$ direction and rotational motion about the $\mathbf{j}^b$axis. The notation for the roll, pitch, and yaw moments will be borrowed from Fossen [41], which uses K, M, and N, respectively. The lift force, drag force, and pitch moment are usually defined as

$$\begin{bmatrix} F_{lift} \\ F_{drag} \\ M \end{bmatrix} = \frac{1}{2}\rho V_a^2 S_{wing} \begin{bmatrix} C_L(\alpha, q, \delta_e) \\ C_D(\alpha, q, \delta_e) \\ cC_m(\alpha, q, \delta_e) \end{bmatrix} \tag{2.15}$$

where C_L, C_D, and C_m are nondimensional aerodynamic coefficients. S_{wing} is the planform area of of the wing and c is the mean chord length. The parameters α, q, and δ_e are the angle of attack, angular rate about the y-axis, and elevator deflection, respectively. As the Skywalker X8 UAV only has a set of elevons as control surfaces, the elevon deflection has to be converted to aileron-elevator signals. This is done by

$$\begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_{er} \\ \delta_{el} \end{bmatrix} \tag{2.16}$$

where δ_{el} and δ_{er} are the left and right elevons.

The forces and moments in Equation (2.15) are generally nonlinear, but through the assumption of small angles of attack, can be approximated by first-order Taylor series.

$$F_{lift} = \frac{1}{2}\rho V_a^2 S_{wing}\left[C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\tfrac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e\right] \tag{2.17a}$$

$$F_{drag} = \frac{1}{2}\rho V_a^2 S_{wing}\left[C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\tfrac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e\right] \tag{2.17b}$$

$$M = \frac{1}{2}\rho V_a^2 S_{wing}c\left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\tfrac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e\right] \tag{2.17c}$$

Equations (2.17a), (2.17b), and (2.17c) form the longitudinal aerodynamic model. The pitch rate q is expressed in $\mathrm{rad\,s^{-1}}$ and is nondimensionalized by multiplicating with $\frac{c}{2V_a}$. The lift and drag forces are expressed in the stability frame and can be expressed in the body frame through rotating by the AOA:

$$\begin{bmatrix} f_x \\ f_z \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} -F_{drag} \\ -F_{lift} \end{bmatrix} \tag{2.18}$$

The lateral aerodynamic model consists of the forces and moments that impact translational motion along the $\mathbf{j}^b$ direction and rotational motion about the $\mathbf{i}^b$ and $\mathbf{k}^b$ axes. Similarly to the AOA affecting longitudinal motion, the SSA β affects the lateral motion, along with the lateral angular velocities and the aileron deflection angle δ_a:

$$\begin{bmatrix} f_y \\ K \\ N \end{bmatrix} = \frac{1}{2}\rho V_a^2 S_{wing} \begin{bmatrix} C_Y(\beta, p, r, \delta_a, \delta_r) \\ bC_l(\beta, p, r, \delta_a, \delta_r) \\ bC_n(\beta, p, r, \delta_a, \delta_r) \end{bmatrix} \tag{2.19}$$

where b is the wingspan of the aircraft. Similarly to the longitudinal forces and moments, the lateral forces and moments can be approximated by a first-order Taylor series.

$$\begin{split} f_y = \frac{1}{2}\rho V_a^2 S_{wing}\Bigg[& C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r \\ & + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\Bigg] \end{split} \tag{2.20a}$$

$$\begin{split} K = \frac{1}{2}\rho V_a^2 S_{wing}b\Bigg[& C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r \\ & + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r\Bigg] \end{split} \tag{2.20b}$$

$$\begin{split} N = \frac{1}{2}\rho V_a^2 S_{wing}b\Bigg[& C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r \\ & + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r\Bigg] \end{split} \tag{2.20c}$$

Equations (2.20a), (2.20b), and (2.20c) form the lateral aerodynamic model. For flying wing UAVs such as the Skywalker X8, the rudder input δ_r is not relevant.

The forces and moments acting on the UAV can now be nondimensionalized and written in terms of nondimensional stability and control derivatives:

$$C_s = C_{s_0} + C_{s_\alpha}\alpha + C_{s_q}\frac{c}{2V_a}q + C_{s_{\delta_e}}\delta_e \quad \text{for } s \in \{L, D, m\} \tag{2.21a}$$

$$C_s = C_{s_0} + C_{s_\beta}\beta + C_{s_p}\frac{b}{2V_a}p + C_{s_r}\frac{b}{2V_a}r + C_{s_{\delta_a}}\delta_a \quad \text{for } s \in \{Y, l, n\} \tag{2.21b}$$

2.4.3 Observed Forces and Moments

When performing system identification techniques derived later, the stability derivatives in Equation (2.21) will be identified. It is therefore useful to compare the estimated nondimensional forces and moments against the observed forces and moments in flight. These are found by using measured accelerations, angular rates, angular accelerations $\boldsymbol{a} = [a_x, a_y, a_z]^\top$, AOA α, SSA β, and thrust forces T. The equations for these forces are found by applying Newton's second law for an aircraft in-flight. This is done by Klein and Morelli [6, equations 5.99 and 5.100], and are repeated here:

$$C_X = \frac{1}{\bar{q}S_{wing}}(ma_x - T) \tag{2.22}$$

$$C_Y = \frac{ma_y}{\bar{q}S_{wing}} \tag{2.23}$$

$$C_Z = \frac{ma_z}{\bar{q}S_{wing}} \tag{2.24}$$

$$C_L = -C_Z \cos\alpha + C_X \sin\alpha \tag{2.25}$$

$$C_D = -C_X \cos\alpha - C_Z \sin\alpha \tag{2.26}$$

$$C_l = \frac{1}{\bar{q}S_{wing}b}[i_{xx}\dot{p} - i_{xz}(pq + \dot{r}) + (i_{zz} - i_{yy})qr] \tag{2.27}$$

$$C_m = \frac{1}{\bar{q}S_{wing}\bar{c}}[i_{yy}\dot{q} + (i_{xx} - i_{zz})pr + i_{xz}(p^2 - r^2)] \tag{2.28}$$

$$C_n = \frac{1}{\bar{q}S_{wing}b}[i_{zz}\dot{r} - i_{xz}(\dot{p} - qr) + (i_{yy} - i_{xx})pq] \tag{2.29}$$

where the torque generated by the propeller has been neglected.

2.4.4 Propulsion Forces and Moments

The thrust model for a propeller can be found by calculating the pressure difference between the upstream and downstream wind velocities and applying it to the propeller area using Bernoulli's principles. The propeller and thrust vector of the Skywalker X8 is assumed to be aligned with the $\mathbf{i}^b$ axis, resulting in zero moments generated by the propulsion system. The torque generated by the propeller is neglected. Coates et al. [42] discussed two models for propulsion systems. The Fitzpatrick model gave satisfactory thrust estimates and was used by Reinhardt [39]. This reduced-order model assumes a relationship between thrust force, throttle, and airspeed given by

$$F_{prop} = \frac{1}{2}\rho S_{prop} C_{prop} (V_a + \delta_t(k_{motor} - V_a))\delta_t(k_{motor} - V_a) \tag{2.30}$$

where S_{prop} is the area swept by the propeller, C_{prop} is the propeller aerodynamic coefficient, ρ is the air density and $k_{motor} = 37.5$ is the motor coefficient. The force vector generated by the propulsion can then be written as

$$\mathbf{f}_p = \frac{1}{2}\rho S_{prop} C_{prop} \begin{bmatrix} (V_a + \delta_t(k_{motor} - V_a))\delta_t(k_{motor} - V_a) \\ 0 \\ 0 \end{bmatrix} \tag{2.31}$$

The forces acting on the UAV can be summarized by [8, eq. 4.18]

$$\begin{aligned} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta cos\phi \end{bmatrix} + \begin{bmatrix} C_X(\alpha) + C_{X_q}(\alpha)\frac{c}{2V_a}q + C_{X_{\delta_e}}(\alpha)\delta_e \\ C_{Y_0} + C_{Y_\beta} + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a \\ C_Z(\alpha) + C_{Z_q}(\alpha)\frac{c}{2V_a}q + C_{Z_{\delta_e}}(\alpha)\delta_e \end{bmatrix} \\ + \frac{1}{2}\rho S_{prop} C_{prop} \begin{bmatrix} (V_a + \delta_t(k_{motor} - V_a))\delta_t(k_{motor} - V_a) \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{2.32}$$

and the torques are summarized by [8, eq. 4.20]

$$\begin{bmatrix} K \\ M \\ N \end{bmatrix} = \frac{1}{2}\rho V_a^2 S_{wing} \begin{bmatrix} b(C_{l_0} + C_{l_\beta} + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a) \\ c(C_{m_0} + C_{m_\alpha} + C_{m_q}\frac{c}{2V_a}p + C_{m_{\delta_e}}\delta_e) \\ b(C_{n_0} + C_{n_\beta} + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a) \end{bmatrix} \tag{2.33}$$

2.5 System Identification

According to Klein and Morelli [6], system identification, particularly in the context of aircraft systems, refers to the process of quantitatively determining the mathematical models that describe the behavior of the aircraft's dynamic systems. It involves analyzing the input-output relationships of the aircraft's various components and subsystems to derive accurate and reliable mathematical representations of their dynamics. The goal of system identification is to develop models that capture the essential characteristics of the aircraft system [6]. System identification requires postulating the class or general structure of the system under test to which the system is equivalent. Hence, the problem boils down to determining the set of equations and unknown parameters such that the model response matches the measured system response given the same input signals [7]. The output error method (OEM) and equation-error (EE) are two common methods for aircraft system identification, and their TD implementations will be presented.

2.5.1 Simulation

Certain methods of system identification, such as the output-error method, require simulation in order to determine a mathematical model of the system. Simulations are performed by computing the next state given a differential equation of the dynamics. In order to perform the simulation on a computer, the continuous physical model of the dynamics needs to be discretized to conform to the digital framework of the computer. The most precise method for the discretization of linear systems of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\end{aligned} \tag{2.34}$$

is the exact discretization

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{A}_d\mathbf{x}[k] + \mathbf{B}_d\mathbf{u}[k]\\ \mathbf{y}[k] &= \mathbf{C}_d\mathbf{x} + \mathbf{D}_d\mathbf{u}[k]\end{aligned} \tag{2.35}$$

where:

$$\mathbf{A}_d = e^{\mathbf{A}T}, \quad \mathbf{B}_d = \left(\int_0^T e^{\mathbf{A}T} d\tau\right)\mathbf{B}, \quad \mathbf{C}_d = \mathbf{C}, \quad \mathbf{D}_d = \mathbf{D} \quad [43] \tag{2.36}$$

Although this integration yields high accuracy for linear systems, it is not applicable to nonlinear systems. Exact discretization is also problematic for large linear system matrices due to the high computational cost of complex matrix exponential integrals. In such cases, the discrete dynamics are approximated, and the simplest alternative is the first-order forward Euler method:

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}(t+T) - \mathbf{x}(t)}{T} \tag{2.37}$$

where T is the sampling period. This method explains the change in state $\mathbf{x}$ between timesteps t and $t+T$, relative to the time difference. We can define $\mathbf{x}[k] := \mathbf{x}(kT) = \mathbf{x}(t)$ where $kT \leq t < (k+1)T$. Inserted into the forward Euler approximation in Equation (2.37), the discretized system becomes

$$\mathbf{x}[k+1] = \mathbf{x}[k] + Tf(\mathbf{x}, \mathbf{u}, k) \tag{2.38}$$

The major limitation of the forward Euler method is the risk of being numerically unstable, where the numerical solution diverges although the exact solution does not. One solution to achieve higher stability is to evaluate multiple points between each sample. This leads to the family of Runge-Kutta methods. While the forward Euler is a first-order method due to the single function evaluation, Runge-Kutta methods achieve higher orders by calculating multiple approximations of the function between each sample. Fourth-order Runge-Kutta is usually recommended for aircraft dynamics, but reasonably good results were achieved with real flight data with second and third-order methods by both Jategaonkar [7] and Klein and Morelli [6]. The equations for the second-order Runge-Kutta method are

$$k_1 = \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k], k) \tag{2.39}$$

$$k_2 = \mathbf{f}(\mathbf{x}[k], Tk_1, \mathbf{u}[k], k + T) \tag{2.40}$$

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \frac{T}{2}(k_1 + k_2) \tag{2.41}$$

2.5.2 Output-Error

The OEM is a method of system identification that adjusts the model parameters iteratively in order to minimize residuals between the measured variables and the estimated model response. That is, it minimizes the error in output between the model and the measurements [6]. This leads to a nonlinear optimization problem, for which a nonlinear optimization method is required. This makes the OEM a maximum likelihood estimator, as it attempts to maximize a likelihood function, making the observed data the most probable given the model. It has been the most used time-domain method of aircraft system identification since it was introduced in the 1960s [7].

Since the measurement noise is assumed Gaussian with probability $p(\mathbf{z})$ and $\mathbf{R}$ is the measurement covariance, the likelihood function is

$$\mathbb{L}[\mathbf{z}_N; \boldsymbol{\theta}] = (2\pi)^N |\mathbf{R}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{v}^\top \mathbf{R}^{-1}\mathbf{v}\right] \tag{2.42}$$

where $\mathbf{z}_N = [z(1), z(2), \dots, z(N)]$ are measurements, $\mathbf{v}(i) = z(i) - \hat{y}(i)$ is the residual and the estimate of the maximum likelihood is given by

$$\hat{\boldsymbol{\theta}} = \max_{\boldsymbol{\theta}} \mathbb{L}(\mathbf{z}_N; \boldsymbol{\theta}) \tag{2.43}$$

Equation (2.42) is optimized by setting the derivative of the right-hand side equal to zero, and solving for $\mathbf{R}$:

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{z}(i) - \mathbf{y}(i))(\mathbf{z}(i) - \mathbf{y}(i))^\top \tag{2.44}$$

For computational efficiency, the maximum likelihood (ML) estimator is found by minimizing the negative logarithm of the likelihood function, such that

$$\hat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} \sum_{i=1}^{N} -\ln(\mathbb{L}[\mathbf{z}(i)|\mathbf{z}_{i-1};\boldsymbol{\theta}])) \tag{2.45}$$

where $\hat{\boldsymbol{\theta}}$ minimizes the cost function

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{v})^{\top}\mathbf{R}^{-1}(\mathbf{v}) \tag{2.46}$$

Here, a linear observation model is assumed.

Klein and Morelli [6] use a modification of the Newton-Raphson nonlinear optimization, as this yields good convergence rates. The modified Newton-Raphson algorithm is developed by obtaining a two-term Taylor series expansion of the cost function in Equation (2.46), as showed by Iliff and Taylor [44]. The goal is to find a $\boldsymbol{\theta}$ such that $\nabla J(\boldsymbol{\theta}) = 0$:

$$J(\boldsymbol{\theta}_o + \Delta\boldsymbol{\theta}) \approx J(\boldsymbol{\theta}_o) + \Delta\boldsymbol{\theta}^{\top}\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\theta=\theta_o} + \Delta\boldsymbol{\theta}^{\top}\left.\frac{\partial^2 J}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^{\top}}\right|_{\theta=\theta_o}\Delta\boldsymbol{\theta} \tag{2.47}$$

An optimum is obtained when

$$\frac{\partial}{\partial \boldsymbol{\theta}} J(\boldsymbol{\theta}_o + \Delta\boldsymbol{\theta}) = \mathbf{0} \tag{2.48}$$

Solving this for $\Delta\boldsymbol{\theta}$ yields the Newton-Raphson algorithm

$$\Delta\boldsymbol{\theta} = -\left[\frac{\partial^2 J}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^{\top}}\right]^{-1}\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\theta=\theta_o} \tag{2.49}$$

Comparing the Newton-Raphson algorithm in Equation (2.49) to the gradient descent optimization, which performs a step towards the fastest descending direction, the Newton-Raphson algorithm attempts to predict the local minimum and steps directly towards it. This results in much better convergence rates at the cost of computation of the Hessian [6, 44]. However, the Hessian can be approximated by neglecting the second-order term:

$$\begin{aligned}\frac{\partial^2 J(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_k} &= \sum_{i=1}^{N}\frac{\partial \mathbf{y}^{\top}(i)}{\partial \theta_j}\hat{\mathbf{R}}^{-1}\frac{\partial \mathbf{y}(i)}{\partial \theta_k} - \sum_{i=1}^{N}\frac{\partial^2 \mathbf{y}(i)}{\partial \theta_j \theta_k}\hat{\mathbf{R}}^{-1}\mathbf{v}(i) \\ &\approx \sum_{i=1}^{N}\frac{\partial \mathbf{y}^{\top}(i)}{\partial \theta_j}\hat{\mathbf{R}}^{-1}\frac{\partial \mathbf{y}(i)}{\partial \theta_k}\end{aligned} \tag{2.50}$$

This approximation removes the need to compute the second derivative, saving computational power. Finally, inserting the terms for the first and second derivative into the Newton-Raphson algorithm, the change in the parameter vector $\hat{\boldsymbol{\theta}}$ is

$$\Delta\boldsymbol{\theta} = -\left[\sum_{i=1}^{N} \frac{\partial \mathbf{y}^\top(i)}{\partial \boldsymbol{\theta}} \hat{\mathbf{R}}^{-1} \frac{\partial \mathbf{y}(i)}{\partial \boldsymbol{\theta}}\right]^{-1}_{\boldsymbol{\theta}=\boldsymbol{\theta}_o} \left[\sum_{i=1}^{N} \frac{\partial \mathbf{y}^\top(i)}{\partial \boldsymbol{\theta}} \hat{\mathbf{R}}^{-1} \mathbf{v}(i)\right]_{\theta=\theta_o} \tag{2.51}$$

The approximation gets more accurate as the innovations $\mathbf{v}(i) = \mathbf{z}(i) - \mathbf{y}(i)$ go towards zero since the neglected Hessian matrix was multiplied with $\mathbf{v}(i)$. Since the innovations are the residuals between the output and the model - or the output errors, this estimation is called the output-error method. The matrix $\frac{\partial \mathbf{y}(i)}{\partial \boldsymbol{\theta}}$ in the Newton-Raphson equation describes the change in output $\mathbf{y}$(i) from a change in the parameter vector $\boldsymbol{\theta}$, and is also called the sensitivity matrix. The partial derivative of $\mathbf{y}$ with respect to $\boldsymbol{\theta}$ can be found numerically by numerical partial differentiation,

$$\frac{\partial \mathbf{y}}{\partial \theta_j} = \frac{\mathbf{y}(\boldsymbol{\theta}_o + \delta\boldsymbol{\theta}_j) - \mathbf{y}(\boldsymbol{\theta}_o)}{|\delta\boldsymbol{\theta}_j|} \tag{2.52}$$

2.5.3 Equation-Error

Ordinary least squares (OLS) estimation is based on fitting curves to data by minimizing the sum of the squared residuals between a model and the corresponding measurements. Equation-error methods can be mathematically simple, as the OLS method can be solved with matrix algebra operations. Equation-error methods do not rely on probability theory, in contrast to the output-error method derived earlier, making them much simpler to implement and computationally efficient.

The model consisting of equations (2.17) and (2.20) is suited for linear regression, as the approximation assumes constant aerodynamic coefficients in $C_L(\alpha, q, \delta_e)$, $C_D(\alpha, q, \delta_e)$, $C_Y(\beta, p, r, \delta_a)$, $C_l(\beta, p, r, \delta_a)$, $C_m(\alpha, q, \delta_e)$ and $C_n(\beta, p, r, \delta_a)$.

By defining a vector $\mathbf{z} \in \mathbb{R}^N$ consisting of N measurements, a matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$ consisting of regressors, and $\boldsymbol{\theta}$ as a vector of the unknown model parameters we can define a model of the form

$$\mathbf{z} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\nu} \tag{2.53}$$

where $\boldsymbol{\nu}$ is introduced as a residual to capture the measurement errors that cannot be explained by the linearized model in (2.17) and (2.20). As is shown by Klein and Morelli [6], the least squares estimator $\hat{\boldsymbol{\theta}}$ can be found by minimizing the cost function of quadratic differences between the model outputs and measured outputs:

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\mathbf{z} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{z} - \mathbf{X}\boldsymbol{\theta}) \tag{2.54}$$

The parameter $\hat{\boldsymbol{\theta}}$ that minimizes the cost function must satisfy

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -\mathbf{X}^\top\mathbf{z} + \mathbf{X}^\top\mathbf{X}\hat{\boldsymbol{\theta}} = 0 \tag{2.55}$$

which leads to the least-square estimator

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{z} \tag{2.56}$$

As an example, the setup for estimating the lift coefficient model $C_L(\alpha, q, \delta_e)$ would be

$$\mathbf{X}_L = \begin{bmatrix} 1 & \alpha(1) & \frac{c}{2V_a(1)} q(1) & \delta_e(1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha(N) & \frac{c}{2V_a(N)} q(N) & \delta_e(N) \end{bmatrix}, \mathbf{z} = \begin{bmatrix} C_L(1) \\ \vdots \\ C_L(N) \end{bmatrix} \tag{2.57}$$

where N is the number of samples and $\mathbf{z}$ consists of the observed, or measured, nondimensional force coefficients given by C_L from Equation (2.22).

As explained by Jategaonkar [7], least squares methods do not rely on the temporal relation defining explicit, sequential ordering in time of states and inputs. Therefore, it is well suited for data preprocessing methods related to data partitioning and assembling data collected from multiple runs and maneuvers.

2.5.4 Data Collinearity

If the vectors of the regressor in Equation (2.56) are linearly independent, $\mathbf{X}^\top \mathbf{X}$ is positive definite, and the inverse exists. However, in the case where the regressors are linearly depended, collinearity is introduced. This means that there are several ways to combine the regressors to model the same variation in the dependent variable [6]. It is therefore difficult to compute $(\mathbf{X^T X})^{-1}$ for use in equation-error system identification. Collinearity can be detected by examining the regressor matrix with standardized regressors [6]. The standardized regressors are scaled versions of the regressors and response variables, with zero mean and unit length:

$$\xi_j^* = \frac{\xi_j(i) - \bar{\xi_j}}{\sqrt{S_{jj}}} \qquad j = 1, 2, ..., N \tag{2.58}$$

where

$$S_{jj} = \sum_{i=1}^{N} [\xi_j(i) - \bar{\xi_j}]^2 \qquad i, j = 1, 2, ..., N \tag{2.59}$$

The new regressor matrix $\mathbf{X}^* = [\boldsymbol{\xi}_1^* \quad \boldsymbol{\xi}_2^* \quad \dots \quad \boldsymbol{\xi}_n^*]$ consists of the standardized regressors, where n is the number of the regressor. $\mathbf{X}^{*\top} \mathbf{X}^*$ now results in a scaled correlation matrix with ones along the diagonal. Klein and Morelli [6] recommends regressor correlations below 0.9 since the problems of collinearity become more prominent above this threshold.

Collinearity may be prevented by performing rank-deficient least squares described in [6, Ch. 6.3.2]. This method calculates the inverse of the $\mathbf{X}^{*\top} \mathbf{X}^*$

matrix using singular value decomposition (SVD) and eliminates terms associated with singular or near-singular values from the SVD decomposition. The SVD is a unique decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\dagger} \tag{2.60}$$

for any complex-valued matrix $\mathbf{X} \in \mathbb{C}^{n\times m}$ [45], where $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V} \in \mathbb{C}^{m\times m}$ are unitary matrices with orthonormal columns, such that $\mathbf{U}^{\dagger}\mathbf{U} = \mathbf{V}^{\dagger}\mathbf{V} = \mathbf{I}$, where † denotes the complex conjuagate transpose. $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ is a diagonal matrix with positive values on the diagonal, which are called the singular values of $\mathbf{X}$. In the case of $\mathbf{X}^{\top}\mathbf{X}$, we have the following equality

$$\mathbf{X}^{\top}\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^{\top}\mathbf{U}^{\top}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top} = \mathbf{V}\mathbf{\Sigma}^{2}\mathbf{V}^{\top} \tag{2.61}$$

where the singular values are the on the diagonal of $\mathbf{\Sigma}^2$. The singular values are ordered from largest to smallest, and the number of nonzero singular values is the rank of $\mathbf{X}$. As described by Klein and Morelli [6], each element of $\mathbf{\Sigma}$ is evaluated against the criterion

$$\frac{\mu_j}{\mu_{max}} < N\varepsilon \tag{2.62}$$

where μ_j is an element on the diagonal of $\mathbf{\Sigma}$, N is the number of data points, and ε is the precision of the computing machine. When a singular value is dropped, its element in $\mathbf{\Sigma}$ is set to zero before calculating the inverse of the information matrix as

$$(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\top} \tag{2.63}$$

As long as the regressors are linearly independent, this method is identical to the normal information matrix inversion performed in Equation (2.56). However, when a singular value fails the criterion in Equation (2.62), the singular value is dropped, and the related regressor is not a part of the least squares estimate.

2.5.5 Model Validation

In order to measure and compare the quality of the parameter estimation, mathematical methods for model validation are required. One common metric for determining the closeness of an estimate to the true value that is commonly used for aircraft system identification [6], is the coefficient of determination R^2. It is given by the equation

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{\hat{\boldsymbol{\theta}}^{\top}\mathbf{X}^{\top}\mathbf{z} - N\bar{z}^2}{\mathbf{z}^{\top}\mathbf{z} - N\bar{z}^2} \tag{2.64}$$

where SS_R and SS_T are the regression sum of squares, and the total sum of squares, respectively. R^2 is then a measure of the proportion of the variation that can be explained by the model and is a score between 0 and 1, where 1 indicates a perfect model for the data [6].

The t_0-statistic is frequently used for evaluating the significance of a regressor and can be seen as a test for a parameter $\hat{\theta}$ being significantly different from zero [6]. It is defined by

$$t_0 = \frac{\hat{\theta}_j}{s(\hat{\theta}_j)} \tag{2.65}$$

where

$$s = \sqrt{\frac{SS_E}{N-p}} = \sqrt{\frac{\sum_{i=1}^{N}(z(i) - \hat{y}(i))^2}{N-p}} \tag{2.66}$$

is the standard deviation, N is the number of data samples, p is the number of terms in the model and SS_E is the sum of squared residuals.

Following the assumption that the measurement errors are normally distributed, the linearity of the OLS estimator with respect to the measurements, together with the deterministic property of the OLS estimator, the estimated parameters are

$$\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}) \tag{2.67}$$

This leads to the $\hat{\theta}_j$ being normally distributed, and consequently the t-statistic of Equation (2.65) being t-distributed with $N-p$ DOF. Confidence intervals for the parameter $\hat{\theta}_j$ can then be calculated by

$$\hat{\theta}_j - t(\frac{\alpha}{2}, N-p)s(\hat{\theta}_j) \leq \theta_j \leq \hat{\theta}_j + t(\frac{\alpha}{2}, N-p)s(\hat{\theta}_j) \tag{2.68}$$

Klein and Morelli [6] also states that in flight testing and system identification of flight test data, $N-p \gg 100$, such that the t-distribution approximates the normal distribution following the central limit theorem.

2.6 State Estimation

EKFs are widely used for nonlinear state estimation and have a wide range of uses in aircraft state estimation [6]. The EKF predicts the next state using the nonlinear equations governing the dynamics of the aircraft and corrects the state predictions using sensor measurements and a measurement model. With the linearized state-space model of an aircraft available, it is used for error covariance propagation [6]. Kalman filters (KFs) in general require tuning of the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, estimating the process noise and the measurement noise, respectively.

2.6.1 Tuning the Kalman Filters

EKFs are tuned in order to optimize performance by modifying the values along the diagonals in the **Q** and **R** matrices, corresponding to the estimated process noise and measurement noise, respectively [46]. Although time-consuming, this is often done by trial and error as conventional optimization techniques for tuning of EKFs can get stuck in local minima [47]. One systematic approach for tuning the **Q** matrix is by evaluating the *filter consistency.* Brekke [46] states that a filter is consistent if its errors, on average are well described by the output of the filter. This is usually tested by checking that "The state errors should have magnitude commensurate with the state covariance yielded by the filter" (Brekke 2020 [46, p. 63]) and that "The innovations should have magnitude commensurate with the innovation covariance yielded by the filter" (Brekke 2020 [46, p. 63]). These conditions are assessed by examining the normalized estimation error squared (NEES)

$$\boldsymbol{\epsilon}_k = (\hat{\mathbf{x}}_k - \mathbf{x}_k)\mathbf{P}_k^{-1}(\hat{\mathbf{x}}_k - \mathbf{x}_k) \tag{2.69}$$

and the normalized innovations squared (NIS)

$$\boldsymbol{\epsilon}_k^{\nu} = (\boldsymbol{\nu}_k^{\top})\mathbf{S}_k^{-1}(\boldsymbol{\nu}_k) \tag{2.70}$$

where **P** is the posterior covariance matrix, $\hat{x}_k$ is the posterior state estimate, x_k is the ground truth, $\boldsymbol{\nu}$ is the innovation $\mathbf{y}_k - \mathbf{g}(\check{\mathbf{x}}_\mathbf{k}, 0)$ of the meaurement and the predicted measurement, and **S** is the innovation covariance. Both the NEES and the NIS can be shown to be chi-square distributed random variables with N degrees of freedom [48], where N is the number of time steps.

Chapter 3

Method and Data Acquisition Design

This chapter explains the method used for data acquisition and analysis in the research performed. The experimental platform including the hardware and the software for flight experiments is presented, as well as the specifications used for its components and sensors. Both the hardware and software architectures are explained here to indicate the capabilities and limitations of the experimental platform. The *a priori* frequency analysis using well-established linearized modes of the aircraft model is performed in order to optimize the inputs during flight testing and system identification. The data processing techniques used on the data gathered from flight experiments are presented to evaluate the quality and validity of the data employed in system identification. This includes the estimation scheme for estimating the wind velocity, as wind velocities are essential for calculating key states for system identification. Finally, the EKF-based MMAE structure used for the detection of icing and icing severity estimation is developed.

3.1 Experimental Platform

This section describes the experimental platform used for data gathering. The hardware includes a single board computer (SBC), sensors, actuators, servo drivers, telemetry, and the ground station, as shown in Figure 3.1. The software includes DUNE: Unified Navigation Environment (DUNE) and Neptus for control and execution. The experimental platform is highly influenced by the setup used by Reinhardt [39], and most of the hardware is the exact same used in his Ph.D. thesis.

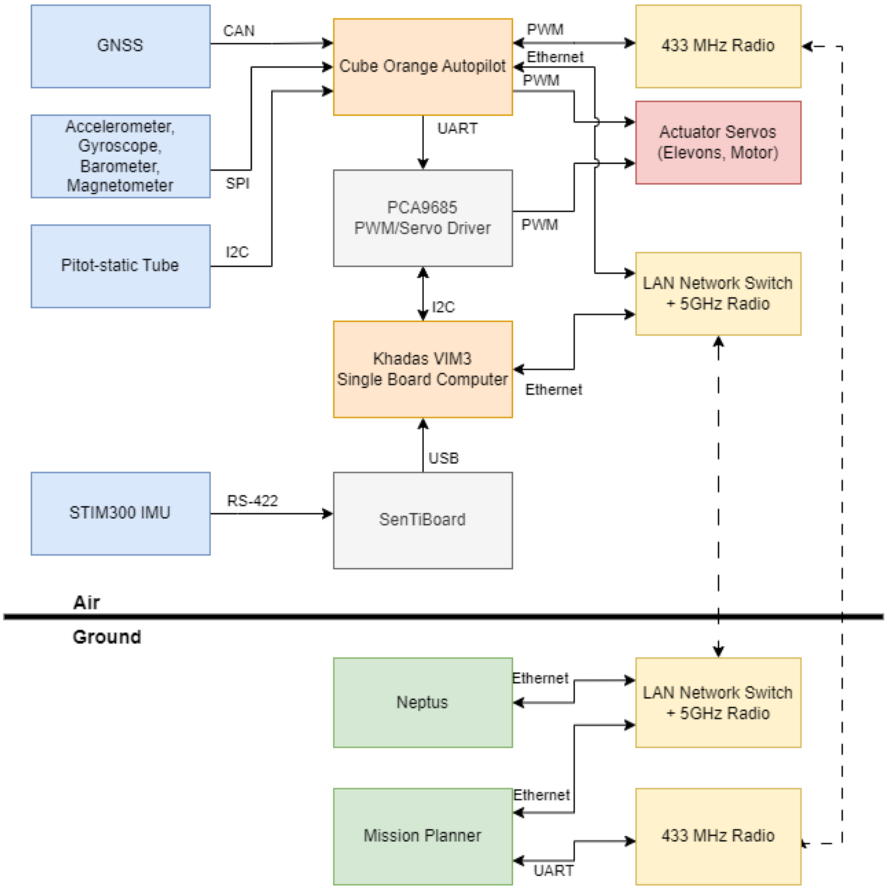

Figure 3.1: Payload and ground station hardware configuration of the experimental platform.

3.1.1 Skywalker X8 Fixed-Wing UAV

The UAV used in this thesis is the Skywalker X8 fixed-wing UAV. It is a flying wing, as it does not have a tail or rudder. Instead of the normal configuration of elevators, ailerons, and rudder, its control surfaces consist of elevons, which combines the functionality of ailerons and elevators. For propulsion, it is equipped with a brushless DC servo motor and a propeller. The relevant specifications are listed in Table 3.1. The servo motor constants used for mapping actuation angles of the elevons to pulse width modulation (PWM) signals were calculated by Reinhardt [39] in lab experiments using cameras. The mapping from the desired angle to PWM signal is given by:

$$PWM = \frac{\delta_{el} - b_0}{b_1} \tag{3.1}$$

where δ_{el} is the desired elevon angle and the constants b_0 and b_1 are given in Table 3.1.

Artificial ice shapes have been modeled by Richard Hann and Damiano Varagnolo [49] and can be attached to the leading edge of the wings to simulate flight in icing conditions. The profile was found through testing in an icing wind tunnel and is designed to resemble mixed ice.

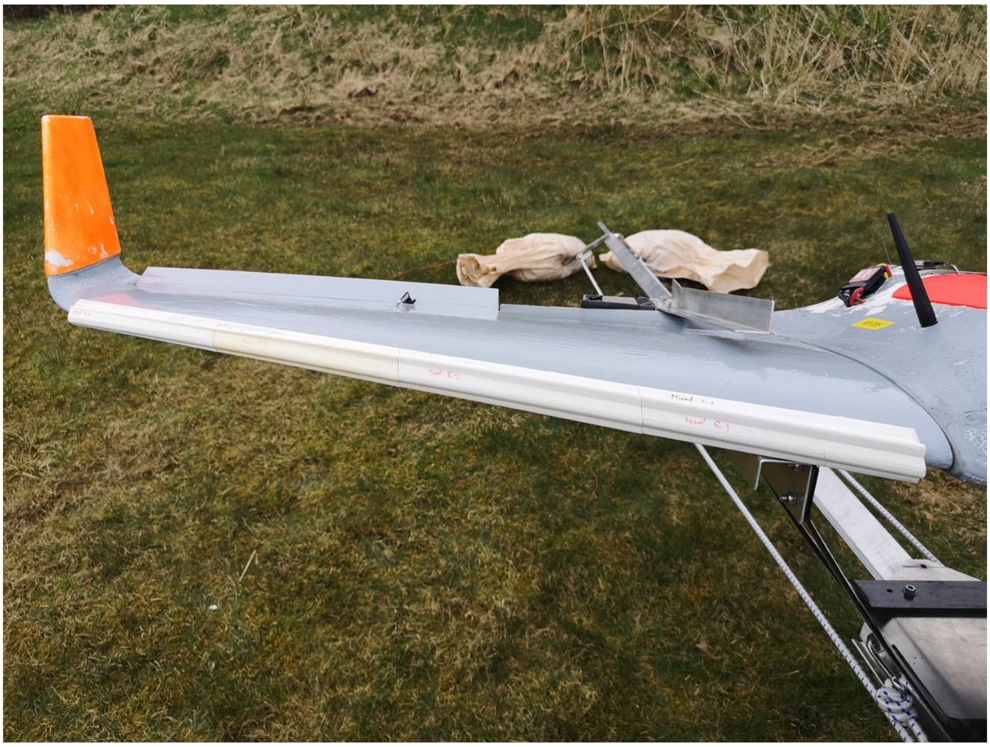

Figure 3.2: Right wing of the Skywalker X8 at Breivika equipped with 3D printed ice models. The image is taken pre-flight during safety checks.

3.1.2 System Architecture

The system architecture illustrated in figure 3.1 includes two possibilities for controlling the actuators; the Cube Orange Autopilot and DUNE through Neptus on the ground station. The Cube Orange Autopilot runs the ArduPilot [50] autopilot software where mission control can set waypoints in ArduPilot's Mission Planner software. The radio-control (RC) controller operated by the pilot has a switch that is programmed to give control to DUNE using a MUX switch. Similarly to Coates et al. [34], the fail-safe functionality included in ArduPilot is augmented to handle fail-safe when the SBC running DUNE is in control. This way, return to launch (RTL) is activated in the event of the RC transmitter losing signal regardless of which entity is in control.

Table 3.1: Specifications for the Skywalker X8 used for flight tests.

Parameter	**Value**	**Description**
m	3.3600 kg	Mass of the UAV with standard payload
b	2.1000 m	Wing span
c	0.3571 m	Mean chord width
k_{motor}	37.5	Motor constant describing the relationship between throttle and propeller discharge velocity at zero airspeed, $k_{motor} = V_d/\delta_t$ [8]
S_{prop}	0.1018	Area swept by the propeller
S_{wing}	0.7500	Area of wing
C_{prop}	0.5	Propeller constant for scaling propeller efficiency
$b_{0,left}$	108.016 55°	Left servo offset for 0° elevon actuation.
$b_{1,left}$	−0.072 64°	Left servo angle change per PWM pulse width.
$b_{0,right}$	−109.484 595 15°	Right servo offset for 0° elevon actuation.
$b_{1,right}$	0.072 871 62°	Right servo angle change per PWM pulse width.

3.1.3 SentiBoard

The sensor timing board (SentiBoard) is a user-configurable sensor-timing system developed by Albreksen [51]. The initial purpose for the SentiBoard in this development platform was to receive sensor output from both a STIM300 IMU and an Aeroprobe 5-hole ADS, synchronizing their measurements. However, the ADS at the UAV lab was not calibrated and cannot be used until it is calibrated by Aeroprobe staff. Instead, only the STIM300 is connected to the SentiBoard, depicted in Figure 3.3. The SentiBoard is still used as it is supported by the SentiUtils package, with useful tools for logging and parsing STIM data.

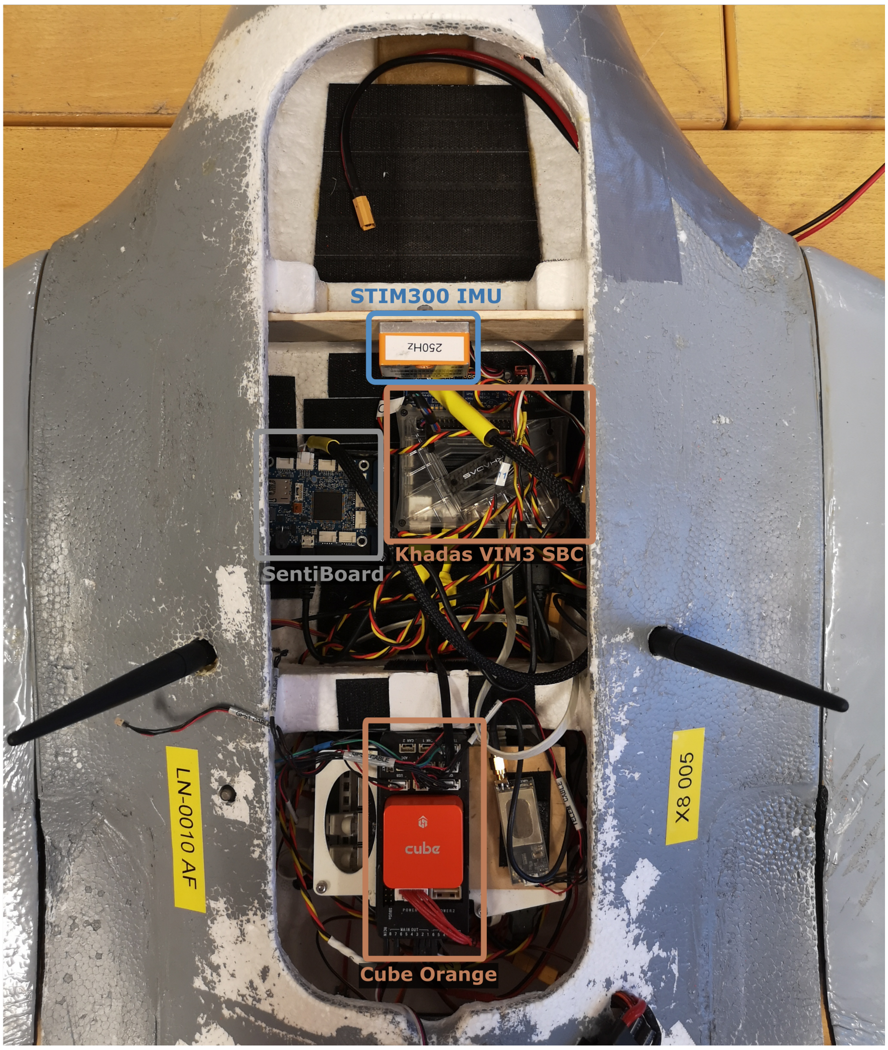

Figure 3.3: Payload installed in the Skywalker X8 fixed-wing UAV at the NTNU UAV lab.

3.1.4 STIM300

The STIM300 IMU was configured to sample data at a rate of 250 MHz using the Sensonor EVK configuration tool through a USB interface. The bit rate was set to 921 600 bit s^{-1}, which is the default bit rate.

Due to the limited space in the avionics bay of the Skywalker X8 UAV, the IMU is mounted on the wall separating the avionics bay and the battery area. This is shown in Figure 3.3. It is mounted firmly with screws instead of velcro which is used for the rest of the payload. Although the velcro would serve as a lowpass filter, this is to prevent the IMU from wiggling on the soft velcro. For the STIM300 IMU data to be comparable to the IMU in the Cube Orange's IMU, the pose relative to the Cube Orange has to be found. The translational displacement between the STIM300's measurement frame m_I and the Cube Orange's measurement frame m_{II} was measured with calipers assuming the inertial center of the STIM300 and the Cube Orange is in the respective geometric centers. The measured displacement in meters was found to be approximately

$$\mathbf{r}_{m_{II}}^{m_I} = [0.239, \quad 0.000, \quad -0.008] \tag{3.2}$$

The relative orientation is found by recording data with the same setup as used in flight testing. Using an Error-State Kalman Filter (ESKF) in Matlab, an estimate of the orientation was found. Then, the average orientation was found according to [52] by minimizing the weighted sum of the squared Frobenius norms of the difference of the attitude matrices:

$$\overline{\mathbf{q}} \triangleq \arg\min_{\mathbf{q}\in\mathbb{S}^3} \sum_{i=1}^{n} w_i \left\| A(\mathbf{q}) - A\left(\mathbf{q}_i\right)\right\|_F^2 \tag{3.3}$$

where the attitude matrix is given by the equation

$$A(\mathbf{q}) = (\eta^2 - ||\boldsymbol{\varepsilon}||^2)\boldsymbol{I}_{3\times 3} + 2\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top - 2\eta\mathbf{S}(\boldsymbol{\varepsilon}) \tag{3.4}$$

In the latter equation, $\mathbf{S}(\boldsymbol{\varepsilon})$ is the cross-product matrix of the vector part of the quaternion $\mathbf{q} = \left[\eta \ \boldsymbol{\varepsilon}^\top\right]^\top = [\eta \ \varepsilon_1 \ \varepsilon_2 \ \varepsilon_3]^\top$.

The average quaternion form the filtered data was normalized for both the Cube Orange IMU and the STIM300 IMU. Then, the quaternion representing the rotation from the Cube Orange IMU to the STIM300 IMU was found by

$$\mathbf{q}_{m_{II}}^{m_I} = \overline{\mathbf{q}}_{m_{II}}^* \otimes \overline{\mathbf{q}}_{m_I} \tag{3.5}$$

where $\mathbf{q}^*$ denotes the conjugate of the quaternion and $\otimes$ is the quaternion multiplication operator. Transforming the quaternion to Euler angles using [41, eq. 2.97-2.99] yields

$$\phi = \arctan 2(2(\varepsilon_2\varepsilon_3 + \varepsilon_1\eta), 1 - 2(\varepsilon_1^2 + \varepsilon_2^2)) = 0.372\,56^\circ \tag{3.6}$$

$$\theta = -\arcsin(2(\varepsilon_1\varepsilon_3 - \varepsilon_2\eta)) = 89.0008^\circ \tag{3.7}$$

$$\psi = \arctan 2(2(\varepsilon_1\varepsilon_2 + \varepsilon_3\eta), 1 - 2(\varepsilon_2^2 + \varepsilon_3^2)) = -9.864^\circ \tag{3.8}$$

3.1.5 Accelerometer Position Correction

For the case of translational accelerometers, it is important to consider the mounting positions relative to the CG. When these accelerometers are installed at locations other than the CG, they are susceptible to detecting both tangential and centripetal accelerations. These additional accelerations arise due to the relative motion between the mounting position and the CG [6]. Gryte et al. [40] found that the CG is located 44.0 cm behind the nose of the Skywalker X8 based on experimental flight data. Measuring the position of the STIM300 IMU, it is found that it is located 30.0 cm behind the nose of the UAV's hull. It is assumed that the difference in payload does not significantly shift the CG. A lever arm between the STIM300 IMU measurement frame $\{m_I\}$ and the CG is then approximated to

$$\mathbf{r}_{m_I g} = \begin{bmatrix} -0.44 \\ 0.0 \\ 0.0 \end{bmatrix} - \begin{bmatrix} -0.30 \\ 0.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} -0.14 \\ 0.0 \\ 0.0 \end{bmatrix} \tag{3.9}$$

Using Fossen [41, eq. 14.5], the lever arm compensation of the linear acceleration can be calculated by

$$\mathbf{a}^b_{ng} = \mathbf{a}^b_{nm_I} + \dot{\boldsymbol{\omega}}^b_{nb} \times \mathbf{r}_{m_I g} + \boldsymbol{\omega}^b_{nb} \times (\boldsymbol{\omega}^b_{nb} \times \mathbf{r}^b_{m_I g}) \tag{3.10}$$

where $\boldsymbol{\omega}^b_{nb} = \boldsymbol{\omega}^b_{nm_I}$ for a rigid body rotating about the inertial frame $\{n\}$. The hardware configuration presented in Chapter 3 does not include any device for directly measuring the angular acceleration $\dot{\boldsymbol{\omega}}^b_{nb}$, and will have to be calculated.

Although the payload does not include sensors for measuring the angular acceleration, Klein and Morelli [6] states that it can be estimated by differentiation. The Matlab function `gradient()` uses central difference approximation

$$\frac{\partial f}{\partial x} = \frac{f(x+h) - f(x-h)}{2h} \tag{3.11}$$

to calculate the derivative.

3.1.6 DUNE

The software framework of the Skywalker X8 at the NTNU UAV lab is based on DUNE, developed by Laboratório de Sistemas e Tecnologia Subaquática (LSTS) at the University of Porto [53]. DUNE is an onboard software for running C++ code during unmanned operations. It encourages structuring complex plans into smaller maneuvers consisting of smaller atomic tasks. In this thesis, DUNE governs sensors, actuators, communication, and maneuvers during flight experiments with the Skywalker X8 UAV and runs on the Khadas VIM3 SBC. Multistep and chirp input signals for system identification maneuvers can be generated by a DUNE task that is programmed to be highly configurable during operation. This is needed, as the magnitudes of the input signals have to be found while testing due to the stability of the UAV being unknown and varying depending on the wind velocity and the impact of the ice shapes.

DUNE uses the Inter-Module Communication (IMC) protocol, also developed by LSTS at the University of Porto, for message passing onboard unmanned vehicles. A redundant DUNE task passes data from ArduPilot running on the Cube Orange to DUNE running on the SBC over inter-module communication (IMC) at 40 Hz, in case the SD card of the Cube Orange is corrupted or lost.

3.1.7 Neptus, Command and Control Framework

Neptus is a command and control software framework also developed by LSTS at the University of Porto [54]. It can be used with all kinds of autonomous and unmanned vehicles, as it utilizes IMC to communicate with DUNE running onboard unmanned vehicles. In this experiment, Neptus is used to modify parameters in the onboard DUNE tasks to execute the aforementioned multisteps and chirp maneuvers after the pilot has transferred control of the UAV to Neptus. Neptus runs on a ground control laptop computer, sending IMC messages to the UAV through ethernet and radio communication, as depicted in Figure 3.1.

3.2 Synchronization and Interpolation

The IMC messages containing various sensor data are sent irregularly due to sensor drift and varying transmission delays. This has to be synchronized to a universal time series such that the data can be evaluated at these time intervals. Klein and Morelli [6] suggests different types of interpolation, from linear interpolation to implementations of the sampling theorem. Linear interpolation is a special case of polynomial interpolation and is defined by Cheney and Kincaid [55] as

$$p(x) = y_0 + \left(\frac{y_1 - y_0}{x_1 - x_0}\right)(x - x_0) \tag{3.12}$$

where x is the new query point between the points (x_0, y_0) and (x_1, y_1) [55]. This is done in Matlab using the `interp1()` function by inputting the sampled data and timestamps, using t_{sync} as the new sequence of query points. t_{sync} is defined as a time vector starting and ending at the same timestamps as the sampled data but with a fixed frequency of 40 Hz, corresponding to the sampling rate of ArduPilot. Hence, all data will be synchronized with the same sample times, making the system identification more accurate than the alternative, which is to use the nearest samples.

The sampling theorem states that the value of the underlying continuous function can be reconstructed at any time using the measured samples by:

$$z(t) = \sum_{k=0}^{N-1} z(k)\,\mathrm{sinc}\left(\pi(\frac{t}{\Delta t} - k)\right) \tag{3.13}$$

if the underlying continuous function is bandwidth-limited such that all components have frequencies below the Nyquist frequency $f_N = 1/(2\Delta t)$ [6]. Note that Equation (3.13) is a simplification of the theorem, as the actual summation interval $[-\infty, \infty]$ requires an infinite amount of data. $\text{sinc}(x)$ is defined as $\frac{\sin(x)}{x}$. Although more rigorous, this method is sensitive to noise, as it does not include any method for distinguishing the deterministic signal from stochastic noise.

3.3 Linearization of the equations of motion

A linearized state-space model of equations (2.7), (2.9), and (2.14) is required for the implementation of an EKF and for designing optimal input signals for system identification. Linearization is performed in the same manner as in [6, 8, 7] by assuming that all states consist of a steady part with an added perturbation. Hence, all states can be written as $\bar{x} \triangleq x - x^*$ where the superscript asterisk denotes the steady part. The bar denotes the deviation from this steady state. Different textbooks use different values for x^*, as Klein and Morelli [6] use the average of the state time series, while Beard and McLain [8] use the trim conditions. This disparity might be due to the application of the linear model, as Klein and Morelli have a more practical approach with real flight data generated from flight testing of an unknown system, while Beard and McLain tend to focus on modeling and simulations, where trim conditions are easily found in simulations. In this thesis, trim conditions for different airspeeds and AOA are found easily in simulations in Simulink by setting an airspeed and altitude reference and letting the simulator's PID controllers make the UAV reach a steady state.

The linearized models are only valid near the points of linearization. Therefore, the maneuvers used in flight for system identification must not deviate too far away from these trim conditions chosen as points of linearization. This principle is also frequently used in control systems design [7]. The dynamic model is developed using the equations of motion developed in Beard and McLain [8]. These are reduced-order modes developed under the assumptions. The longitudinal short-period mode assumes constant altitude and thrust input, while the lateral roll mode assumes constant pitch angle and neglects heading dynamics. The linearization of longitudinal and lateral dynamics is performed by computing the Jacobians of the respective longitudinal and lateral state equations in (2.7), (2.9) and (2.14). The longitudinal Jacobians are calculated by

$$\mathbf{A}_{long.} = \frac{\partial f_{long.}}{x_{long.}} = \begin{bmatrix} \frac{\partial \dot{u}}{\partial u} & \frac{\partial \dot{u}}{\partial w} & \frac{\partial \dot{u}}{\partial q} & \frac{\partial \dot{u}}{\partial \theta} \\ \frac{\partial \dot{w}}{\partial u} & \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial q} & \frac{\partial \dot{w}}{\partial \theta} \\ \frac{\partial \dot{q}}{\partial u} & \frac{\partial \dot{q}}{\partial w} & \frac{\partial \dot{q}}{\partial q} & \frac{\partial \dot{q}}{\partial \theta} \\ \frac{\partial \dot{\theta}}{\partial u} & \frac{\partial \dot{\theta}}{\partial w} & \frac{\partial \dot{\theta}}{\partial q} & \frac{\partial \dot{\theta}}{\partial \theta} \end{bmatrix} \text{ and } \mathbf{B} = \frac{\partial x_{long.}}{\partial u_{long.}} \begin{bmatrix} \frac{\partial \dot{u}}{\partial \delta_e} \\ \frac{\partial \dot{w}}{\partial \delta_e} \\ \frac{\partial \dot{q}}{\partial \delta_e} \\ \frac{\partial \dot{\theta}}{\partial \delta_e} \end{bmatrix} \tag{3.14}$$

and the lateral Jacobians are found by calculating

$$\mathbf{A}_{lateral} = \frac{\partial f_{lateral}}{x_{lateral}} = \begin{bmatrix} \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial p} & \frac{\partial \dot{v}}{\partial r} & \frac{\partial \dot{v}}{\partial \phi} \\ \frac{\partial \dot{p}}{\partial v} & \frac{\partial \dot{p}}{\partial p} & \frac{\partial \dot{p}}{\partial r} & \frac{\partial \dot{p}}{\partial \phi} \\ \frac{\partial \dot{r}}{\partial v} & \frac{\partial \dot{r}}{\partial p} & \frac{\partial \dot{r}}{\partial r} & \frac{\partial \dot{r}}{\partial \phi} \\ \frac{\partial \dot{\phi}}{\partial v} & \frac{\partial \dot{\phi}}{\partial p} & \frac{\partial \dot{\phi}}{\partial r} & \frac{\partial \dot{\phi}}{\partial \phi} \end{bmatrix} \text{ and } \mathbf{B} = \frac{\partial f_{lateral}}{u_{lateral}} \begin{bmatrix} \frac{\partial \dot{v}}{\partial \delta_a} \\ \frac{\partial \dot{p}}{\partial \delta_a} \\ \frac{\partial \dot{r}}{\partial \delta_a} \\ \frac{\partial \dot{\phi}}{\partial \delta_a} \end{bmatrix} \tag{3.15}$$

Calculating the Jacobians result in a set of model coefficients denoted as $X_{(\cdot)}, Z_{(\cdot)}, M_{(\cdot)}, Y_{(\cdot)}, L_{(\cdot)}$, and $N_{(\cdot)}$, where $(\cdot)$ is a placeholder for the state which the coefficient is related to. The resulting linearized longitudinal and lateral state space equations can then be assembled as

Longitudinal

$$\underbrace{\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} X_u & X_w V_a^* \cos\alpha^* & X_q & -g\cos\theta^* \\ \frac{Z_u}{V_a^* \cos\alpha^*} & Z_w & \frac{Z_q}{V_a^* \cos\alpha^*} & \frac{-g\sin\theta^*}{V_a^* \cos\alpha^*} \\ M_u & M_w V_a^* \cos\alpha^* & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} X_{\delta_e} \\ \frac{Z_{\delta_e}}{V_a^* \cos\alpha^*} \\ M_{\delta_e} \\ 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix}}_{\mathbf{u}} \tag{3.16}$$

Lateral

$$\underbrace{\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} Y_v & \frac{Y_p}{V_a^* \cos\beta^*} & \frac{Y_r}{V_a^* \cos\beta^*} & \frac{g\cos\theta^* \cos\phi^*}{V_a^* \cos\beta^*} \\ L_v V_a^* \cos\beta^* & L_p & L_r & 0 \\ N_v V_a^* \cos\beta^* & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \frac{Y_{\delta_a}}{V_a^* \cos\beta^*} \\ L_{\delta_a} \\ N_{\delta_a} \\ 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\delta_a}_{\mathbf{u}} \tag{3.17}$$

Here, the equations are written in terms of α and β instead of w and v, by the relations $\dot{\alpha} = \frac{\dot{w}}{V_a^* \cos\alpha^*}$ and $\dot{\beta} = \frac{\dot{v}}{V_a^* \cos\beta^*}$, respectively. The equations used for calculating the stability and control derivatives are adopted from Beard and McLain [8, Chapter 5].

3.4 *A Priori* Frequency Analysis of Dynamic Modes

This section aims to utilize the *a priori* linear model of the experimental platform developed in the previous section to perform frequency analysis of the dynamic modes to determine the frequency content of the longitudinal and lateral dynamics. This is advantageous as it is possible to perform some preliminary design of input signals for real-world experiments [7]. This section mostly repeats the process followed in the project thesis [1]. However, the linearized model has been updated with new and more accurate trim conditions, in addition to new stability and control derivatives. It is included here to inform the reader of the steps performed in choosing the input signal for the experimental flights with the Skywalker X8 UAV, as it is nontrivial to the average reader, and since the results slightly differ from the results in the project thesis.

According to Klein and Morelli [6] and Jategaonkar [7], there are two main approaches for designing optimized input signals for system identification. One approach operates with little to no *a priori* information about the system. Here, it is not possible to provide information about the resonant frequencies of the different dynamic modes of the system. When designing input signals for system identification, it is beneficial to make sure the natural frequencies are excited, as this provides the most information about the dynamics of the aircraft [6, 7]. In this case, this is done by exciting a broad range of frequencies, to make sure that the unknown natural frequencies are excited. This is done by applying a sweep across a frequency spectrum as input to the actuators. This frequency range can be defined as a range that is assumed to cover all dominant frequencies of the UAV.

The other approach exploits any *a priori* information about the dynamic modes in order to match the frequency spectrum of the input signal with the expected natural frequencies of the aircraft's dynamic modes. This provides the most information about the underlying system dynamics, making the identified system more accurate. The *a priori* information to be used for the design of the input signal consists of the linearized equations of motion in equations (3.16) and (3.17). This is only possible when the trim conditions and specifications of the UAV are available. This is the case for the Skywalker X8 UAV due to analyses performed by Gryte [40] and Reinhardt [39]. Some commonly used maneuvers for this category of system identification are multistep inputs, which are square-wave inputs of customized periods. Some common types of multisteps found in the survey in Table 1.1 are doublets, 1-2-1, and the 3-2-1-1 maneuver.

Jategaonkar [7] estimates the optimal frequency ranges by utilizing Bode diagrams. The contributions made by each parameter can then be synthesized with this technique. For example, contributions from each term to the force in the Z-direction in the short-period mode of Equation (3.16) can be found by

$$|H_{\dot{u}/\delta_e}(s)| = |\frac{\dot{u}}{\delta_e}(s)|, \quad |H_{u/\delta_e}(s)| = |\frac{X_u u}{\delta_e}(s)|, \quad |H_{\alpha/\delta_e}(s)| = |\frac{X_\alpha \alpha}{\delta_e}(s)|,$$
$$|H_{q/\delta_e}(s)| = |\frac{X_q q}{\delta_e}(s)|, \quad |H_{\theta/\delta_e}(s)| = |\frac{X_\theta \theta}{\delta_e}(s)|, \quad |H_{\delta_e/\delta_e}(s)| = |\frac{X_{\delta_e} \delta_e}{\delta_e}(s)|$$

where the transfer functions $H_{(\cdot/\delta_e)}(s)$ are found using the Laplace transform of the linear time-invariant (LTI) state-space equations in equations (3.16) and

(3.17) [56]

$$
\begin{aligned}
s\mathbf{x}(s) &= \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s) && (3.18)\\
\mathbf{y}(s) &= \mathbf{C}_{(\cdot)}\mathbf{x} + \mathbf{D}_{(\cdot)}\mathbf{u}(s) && (3.19)
\end{aligned}
$$

Solving the first equation of Equation (3.18) for $\mathbf{x}(s)$ and inserting into the second equation, and then solving for $H_{(\cdot/\delta_e)} = \frac{y(s)}{u(s)}$ yields

$$H_{(\cdot/\delta_e)} = \frac{y(s)}{u(s)} = \mathbf{C}_{(\cdot)}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}_{(\cdot)} \tag{3.20}$$

where the matrices $\mathbf{C}_{(\cdot)}$ and $\mathbf{D}_{(\cdot)}$ were defined as

$$
\begin{aligned}
\mathbf{C}_{\dot{u}} &= \begin{bmatrix} X_u & X_w V_a^* \cos\alpha^* & X_q & -g\cos\theta^* \end{bmatrix}, & \mathbf{D}_{\dot{u}} &= \begin{bmatrix} X_{\delta_e} \end{bmatrix}\\
\mathbf{C}_{u} &= \begin{bmatrix} X_u & 0 & 0 & 0 \end{bmatrix}, & \mathbf{D}_{u} &= \begin{bmatrix} 0 \end{bmatrix}\\
\mathbf{C}_{\alpha} &= \begin{bmatrix} 0 & X_w V_a^* \cos\alpha^* & 0 & 0 \end{bmatrix}, & \mathbf{D}_{\alpha} &= \begin{bmatrix} 0 \end{bmatrix}\\
\mathbf{C}_{q} &= \begin{bmatrix} 0 & 0 & X_q & 0 \end{bmatrix}, & \mathbf{D}_{q} &= \begin{bmatrix} 0 \end{bmatrix}\\
\mathbf{C}_{\theta} &= \begin{bmatrix} 0 & 0 & 0 & -g\cos\theta^* \end{bmatrix}, & \mathbf{D}_{\theta} &= \begin{bmatrix} 0 \end{bmatrix}\\
\mathbf{C}_{\delta_e} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, & \mathbf{D}_{\delta_e} &= \begin{bmatrix} X_{\delta_e} \end{bmatrix}
\end{aligned}
\tag{3.21}
$$

The resulting Bode diagrams describing the frequency contributions from each derivative are found in Figures 3.4. At any given frequency, a large magnitude for a term relative to the other terms indicates a dominant influence of that specific derivative. This indicates good information content, which is necessary for accurate identification of the parameter [7]. A rule of thumb formulated by Jategaonkar [7] suggests that a derivative is identifiable when the magnitude of the term is at least 10 % of the magnitude of the largest term. The same analysis was performed for all equations of the state-space formulations in Equation (3.16) and (3.17).

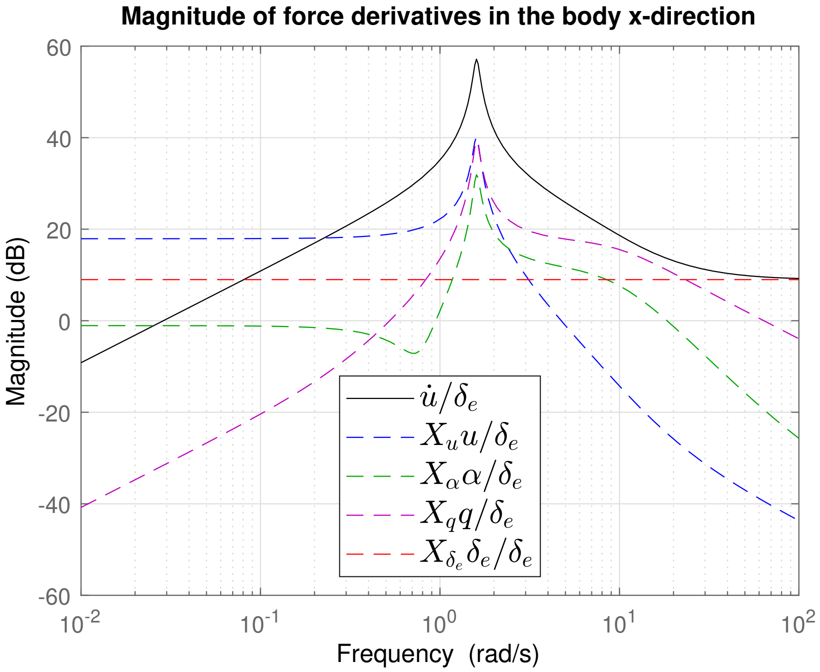

Figure 3.4: Bode magnitude plot of the force contributions made by each derivative of the first equation in Equation (3.16).

By inspecting the Bode magnitude plots in Figure 3.4, it is noted that most of the derivatives are identifiable in the range between $1\,\mathrm{rad\,s^{-1}}$ and $1 \times 10^1\,\mathrm{rad\,s^{-1}}$. This indicates that it might be a good frequency range to optimize the input signals to excite. The same applies to the Bode magnitude plots for the other equations, as well as the Bode magnitude plots for the equations in (3.17).

3.5 Optimal Input Signal Design for System Identification

The second step of designing input signals for system identification is to determine the input signals such that they match the optimal frequency ranges derived in the previous section. Multistep input signals are executed by a series of step inputs in opposite directions with intervals of varying lengths. Both Klein and Morelli [6] and Jategaonkar [7] recommend the doublet (1-1) and the 3-2-1-1 multistep signals, where the numbers denote the number of periods in each direction. That is, the doublet maneuver will last for a total of $2\Delta t$, while the 3-2-1-1 maneuver lasts for $7\Delta t$. The doublet maneuver is very common in the literature due to its simplicity in terms of design and execution. However, the 3-2-1-1 maneuver developed by the Deutsches Zentrum für Luft- und Raumfahrt (DLR), has shown to be superior due to its wider band of frequencies [7]. This is also true to a lesser extent for the 1-2-1 maneuver, which is more often used with lateral maneuvers due to its symmetry of two excitation periods in each direction. Hence, the aircraft will return closer to its initial position after the maneuver. This is arguably more important for the lateral roll angle than the longitudinal pitch angle.

The duration of the different pulses in each direction of the maneuvers is determined by placing the largest mass of the power spectrum of the signal in the desired frequency range of the natural frequency of each dynamic mode. For example, the power spectrum of different pulse widths Δt for a doublet maneuver is shown in the second plot in Figure 3.5.

The normalized frequency $\omega \Delta t$ of the doublet input signal is shown in the third plot in Figure 3.5. By analyzing this power spectrum, it is seen that the peak of the spectrum is located at $\omega \Delta t_{doublet} \approx 2.3$. Choosing $\Delta t_{doublet} = 0.4\,\mathrm{s}$ results in placing the peak well within the optimal frequency range found earlier. The equivalent analysis was performed for the 3-2-1-1 maneuver with the longitudinal state-space model in Equation (3.16) and the 1-2-1 maneuver with the lateral model in Equation (3.17). These periods are chosen slightly lower, at $\Delta t_{3-2-1-1} = 0.3\,\mathrm{s}$ and $\Delta t_{1-2-1} = 0.3\,\mathrm{s}$. These will be used as periods in the multisteps during the experimental flights. The shorter periods for the more complex maneuvers are reasonable, as the power spectra are pushed towards the lower frequencies due to the multiple consecutive periods in the same direction. The 1-2-1 maneuver is used instead of the 3-2-1-1 maneuver in the lateral direction. This is due to it being symmetrical, returning the UAV closer to the initial roll angle after execution of the maneuver [7].

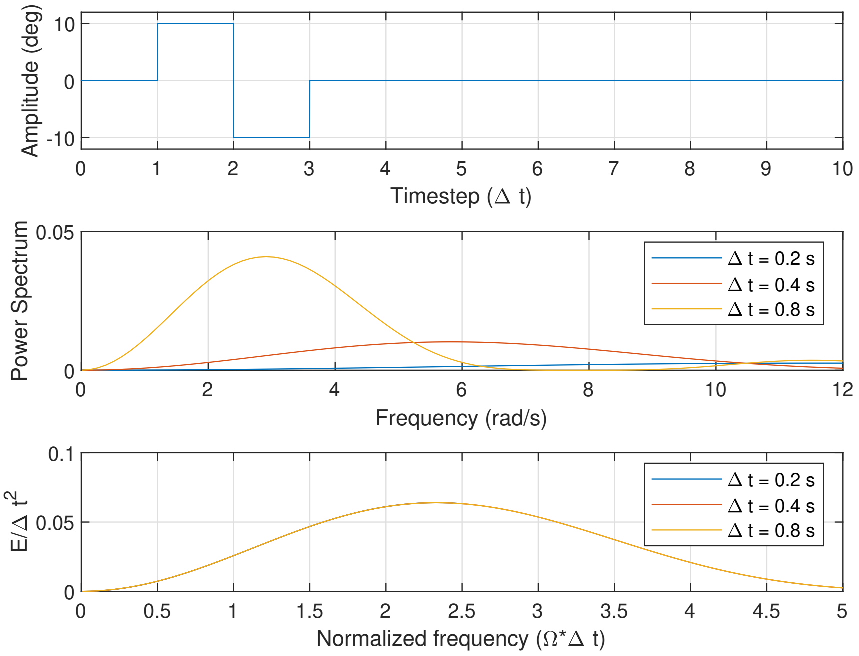

Figure 3.5: The figure shows a doublet input signal with amplitude $\delta = 10°$ (top), and the power spectrum for varying Δt (middle), as well as the normalized frequency content (bottom).

3.6 Estimation of Wind Velocity, AOA, SSA, and Airspeed

Without an air data system, such as a multi-hole air probe, it is impossible to measure the wind-related states directly. However, these can be estimated through sensor fusion. The estimation scheme proposed by Johansen et al. [57], provides estimates of the wind velocity vector $\mathbf{u}_w = [u_w \quad v_w \quad w_w]^\top$, AOA, SSA, and the airspeed V_a. This method requires a pitot-static tube for measuring the air velocity projected on the longitudinal axis of the UAV, an AHRS using IMU, magnetometer, and GNSS measurements, as well as velocity measurements from the GNSS. These measurements are used for a wind velocity observer based on a Kalman filter.

3.6.1 Wind Velocity Observer

The wind velocity observer estimates the wind in all three directions. The estimation scheme makes some assumptions about wind and UAV dynamics. The wind is assumed slowly time-varying in the NED frame

$$\dot{\mathbf{u}}_w^n = 0 \tag{3.22}$$

The pitot-static sensor dynamic is also assumed slowly varying, with $u_r = \gamma u_r^m$ being the measurement equation. u_r^m is the raw measurement from the pitot-static tube, and γ is a sensor scaling factor. The scaling factor is unknown and estimated online. The slowly varying airspeed sensor dynamics can then be modeled by

$$\dot{\gamma} = 0 \tag{3.23}$$

The observer injection term uses the measurement equation

$$u = d_1^\top \mathcal{R}_n^b \mathbf{u}_w^n + u_r^m \gamma \tag{3.24}$$

where $d_1 = [1 \quad 0 \quad 0]^\top$ and u is the measured velocity from the GNSS and AHRS [57].

The state vector is then defined as $\mathbf{x} = \begin{bmatrix}(\mathbf{u}_w^n)^\top & \gamma\end{bmatrix}^\top$ with the system matrix $\mathbf{A} = \mathbf{0}_{4\times4}$ due to the assumptions in equations (3.22) and (3.23). The measurement matrix is time-varying and can be written as

$$\mathbf{C} = \begin{bmatrix} d_1^\top \mathcal{R}_n^b & u_r^m \end{bmatrix}^\top \tag{3.25}$$

The proposed observer by Johansen et al. [57] is given by

$$\begin{bmatrix} \dot{\hat{\mathbf{u}}}_w^n \\ \dot{\hat{\gamma}} \end{bmatrix} = \mathbf{K}\left(u - u_r^m \hat{\gamma} - d_1^\top \mathcal{R}_n^b \hat{\mathbf{u}}_w^n\right) \tag{3.26}$$

where $\mathbf{K}$ is the Kalman gain vector.

The Kalman filter equations are then assembled as

$$\begin{aligned} \mathbf{K}_k &= \hat{\mathbf{P}}_{k-1}\mathbf{C}_k^\top R^{-1} \\ \hat{\mathbf{P}}_k &= \hat{\mathbf{P}}_{k-1} + \mathbf{Q} - \mathbf{K}_k \mathbf{C}_k \hat{\mathbf{P}}_{k-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k(u - u_r^m \hat{\gamma} - d_1^\top \mathcal{R}_n^b \hat{\mathbf{u}}_w^n) \end{aligned} \tag{3.27}$$

where the matrix $\mathbf{C}_k$ is updated according to Equation (3.25) at each timestep k.

Johansen et al. [57] also proved the observability of the wind velocity, as long as sufficient attitude changes are satisfying the persistence of excitation (PE) condition. As the flight experiments aim to perform longitudinal and lateral maneuvers, this condition should be fulfilled without needing any adjustments to the experiment plan.

The same covariance matrices and initial estimates as in Johansen et al. [57] were used, as they used a similar payload and UAV, with

$$
\begin{aligned}
R &= 1 \\
\mathbf{P}_0 &= \mathrm{diag}(10^{-2}, 10^{-2}, 10^{-6}, 10^{-4}) \\
\mathbf{Q} &= \mathrm{diag}(10^{-3}, 10^{-3}, 10^{-6}, 10^{-8}) \\
\hat{\mathbf{u}}_w^n &= [0, 0, 0]^\top \\
\hat{\gamma} &= 1
\end{aligned}
\tag{3.28}
$$

3.6.2 Calculation of AOA, SSA, and Va

With the estimated wind velocities in the NED frame, the relative wind velocities decomposed in the BODY frame can be calculated as in [8] by

$$
\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix}
\tag{3.29}
$$

Finally, the states α, β and V_a are found using Equation (2.4) with the estimated relative velocities.

3.7 Maneuver Data Structure

The maneuvers are extracted from the data set using timestamps of IMC messages announcing the start of the maneuver. These messages contain the type of maneuver performed as well as information about the duration of the maneuver. Hence it was straightforward to extract the IMC messages with the sampled data based on the start and end timestamps of the maneuver.

The data is ordered in Matlab structs by the entity it was sent from in DUNE. The reason for this is that each entity has its own sampling times and frequencies, resulting in unique and unsynchronized time series. Each entity struct in the data structure contains its sampling times and the sampled data. An overview of the maneuver data set is shown in Figure 3.6. The diagram is color coded to distinguish the direct measurements, estimated and filtered states, and metadata describing the configuration of the maneuver. The PWM signals were measured and transmitted by three different entities in-flight due to the DUNE task modifying the PWM signal in order to perform maneuvers. The autopilot entity contains 8 PWM channels from the pilot's RC-transmitter, containing all manual control signals. The fourth PWM entity results from the reconstruction of the throttle PWM signal from the low sampling rate telemetry logs.

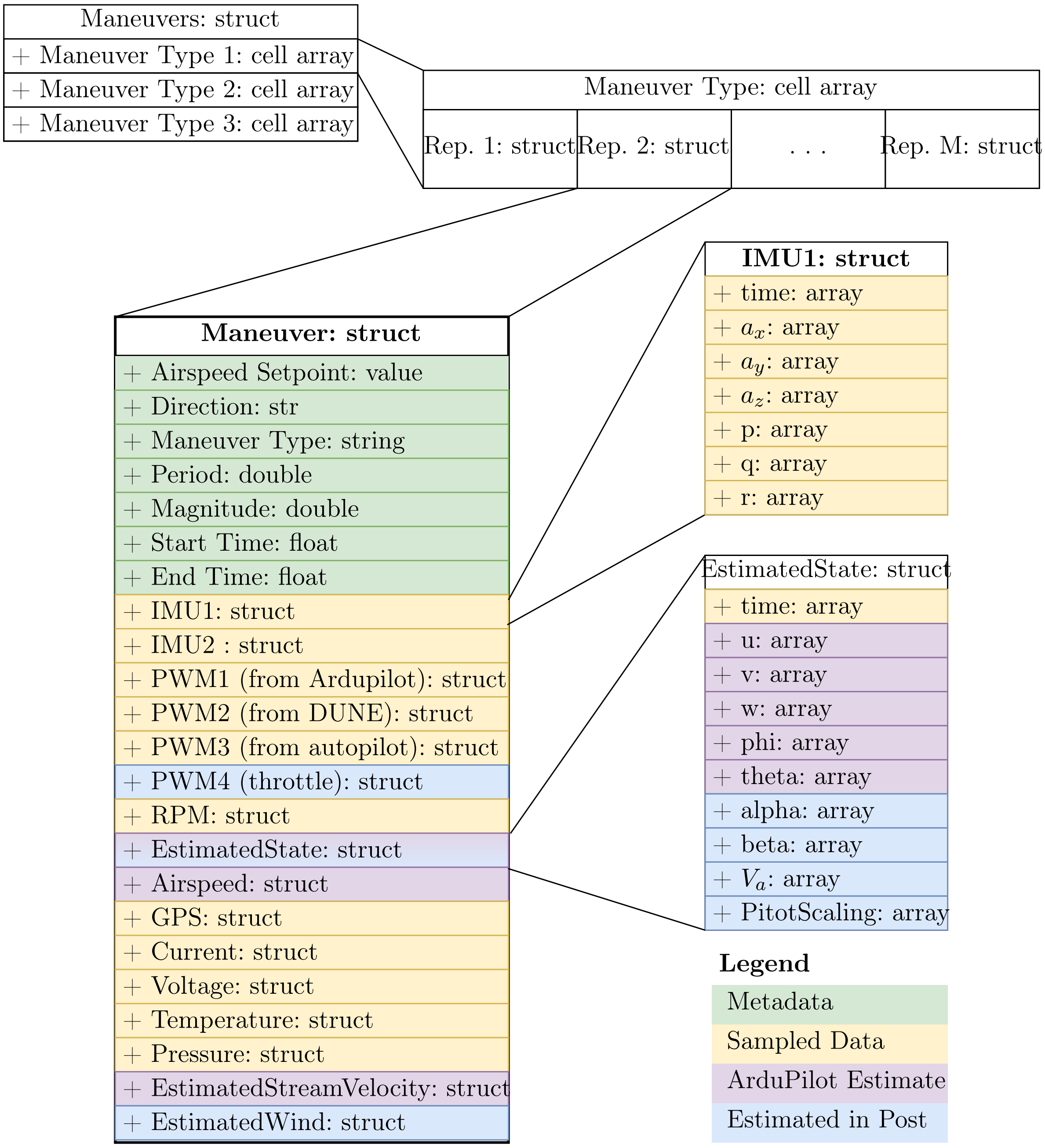

Figure 3.6: Visualization of the data structure of the maneuver data set. Repetitions of maneuvers performed with the same configurations are ordered in Matlab cell arrays. Each maneuver contains metadata, sampled data, and estimated data.

The maneuver types are grouped into cell arrays by the airspeed setpoint, direction (lateral or longitudinal), magnitude of the control surface deflection, and the length of the excitation period Δt. This data structure makes it easy to iterate over maneuvers with the same configuration and makes the data for system identification easily accessible.

The EstimatedState struct in Figure 3.6 originally contains the states estimated by the ArduPilot's INS and various EKFs. The AOA, SSA, V_a, and the pitot scaling γ estimated with the results from the wind velocity observer in Equation (3.27) are inserted into the `EstimatedState` struct.

3.8 System Identification

The system identification methods applied to the flight data estimate the nondimensional aerodynamic coefficients C_L, C_D, C_Y, C_l, C_m, and C_n from equations (2.17) and (2.20). These are reconstructed as the sum of the identified stability and control derivatives multiplied with their respective stability and control parameters, as in Equation (2.21):

$$\begin{aligned}
C_L(\alpha, q, \delta_e) &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e \\
C_D(\alpha, q, \delta_e) &= C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e \\
C_m(\alpha, q, \delta_e) &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e \\
C_Y(\beta, p, r, \delta_a) &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a \\
C_l(\beta, p, r, \delta_a) &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a \\
C_n(\beta, p, r, \delta_a) &= C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a
\end{aligned} \tag{3.30}$$

The two methods applied are the EE using the OLS algorithm and the OEM based on simulating the linearized models in equations (3.16) and (3.17). These are explained in Section 2.5.

3.8.1 Equation-Error Method for Flight Data

The flight data from each maneuver type is assembled into the regressor matrices **X** from Section 2.5 according to Table 3.2.

Table 3.2: Estimated coefficients, the regressor matrix, and the regressand used for OLS in equation-error system identification. N is the number of samples.

$\hat{\boldsymbol{\theta}}_{LS}$	$\mathbf{X}$	$\mathbf{z}$
$\begin{bmatrix} C_{L_0} & C_{L_\alpha} & C_{L_q} & C_{L_{\delta_e}} \end{bmatrix}^\top$	$\begin{bmatrix} 1 & \alpha(1) & \frac{c}{2V_a(1)}q(1) & \delta_e(1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha(N) & \frac{c}{2V_a(N)}q(N) & \delta_e(N) \end{bmatrix}$	$\begin{bmatrix} C_L(1) \\ \vdots \\ C_L(N) \end{bmatrix}$
$\begin{bmatrix} C_{D_0} & C_{D_\alpha} & C_{D_q} & C_{D_{\delta_e}} \end{bmatrix}^\top$	$\begin{bmatrix} 1 & \alpha(1) & \frac{c}{2V_a(1)}q(1) & \delta_e(1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha(N) & \frac{c}{2V_a(N)}q(N) & \delta_e(N) \end{bmatrix}$	$\begin{bmatrix} C_D(1) \\ \vdots \\ C_D(N) \end{bmatrix}$
$\begin{bmatrix} C_{m_0} & C_{m_\alpha} & C_{m_q} & C_{m_{\delta_e}} \end{bmatrix}^\top$	$\begin{bmatrix} 1 & \alpha(1) & \frac{c}{2V_a(1)}q(1) & \delta_e(1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha(N) & \frac{c}{2V_a(N)}q(N) & \delta_e(N) \end{bmatrix}$	$\begin{bmatrix} C_m(1) \\ \vdots \\ C_m(N) \end{bmatrix}$
$\begin{bmatrix} C_{Y_0} & C_{Y_\beta} & C_{Y_p} & C_{Y_r} & C_{Y_{\delta_a}} \end{bmatrix}^\top$	$\begin{bmatrix} 1 & \beta(1) & \frac{b}{2V_a(1)}p(1) & \frac{b}{2V_a(1)}r(1) & \delta_a(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \beta(N) & \frac{b}{2V_a(N)}p(N) & \frac{b}{2V_a(N)}r(N) & \delta_a(N) \end{bmatrix}$	$\begin{bmatrix} C_Y(1) \\ \vdots \\ C_Y(N) \end{bmatrix}$
$\begin{bmatrix} C_{l_0} & C_{l_\beta} & C_{l_p} & C_{l_r} & C_{l_{\delta_a}} \end{bmatrix}^\top$	$\begin{bmatrix} 1 & \beta(1) & \frac{b}{2V_a(1)}p(1) & \frac{b}{2V_a(1)}r(1) & \delta_a(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \beta(N) & \frac{b}{2V_a(N)}p(N) & \frac{b}{2V_a(N)}r(N) & \delta_a(N) \end{bmatrix}$	$\begin{bmatrix} C_l(1) \\ \vdots \\ C_l(N) \end{bmatrix}$
$\begin{bmatrix} C_{n_0} & C_{n_\beta} & C_{n_p} & C_{n_r} & C_{n_{\delta_a}} \end{bmatrix}^\top$	$\begin{bmatrix} 1 & \beta(1) & \frac{b}{2V_a(1)}p(1) & \frac{b}{2V_a(1)}r(1) & \delta_a(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \beta(N) & \frac{b}{2V_a(N)}p(N) & \frac{b}{2V_a(N)}r(N) & \delta_a(N) \end{bmatrix}$	$\begin{bmatrix} C_n(1) \\ \vdots \\ C_n(N) \end{bmatrix}$

Then the equation-error method is applied to data from both single maneuvers and sets of identical maneuvers, according to Section 2.5.

3.8.2 Output-Error Method for Flight Data

According to Klein and Morelli [6], the initial estimates from the equation-error method can be improved with the OEM. The modified Newton-Raphson optimization algorithm presented in Section 2.5 is used to iteratively improve the estimated coefficients θ from Table 3.2. The linearized state-space models in equations (3.16) and (3.17) are simulated using the measured flight data as states and inputs. This is done by 2nd order Runge-Kutta integration, similar to Klein and Morelli [6]. In contrast to the EE method, the OEM relies on the temporal relation between data points [7] and will only be applied to data from single maneuvers.

3.9 Simulator

The simulator used for simulating flight in normal and icing conditions is based on the Skywalker X8 simulator developed by Gryte [58], and the icing simulation was later added by Winter [59]. The dynamics model of the UAV is based on Fossen [41], while the forces acting on the UAV are based on Beard and McLain [8, Chapter 4] by implementing equations (2.17) and (2.20). The simulator utilizes the default Simulink `ode45` solver for simulating the dynamic system, which is based on the explicit Runge-Kutta-Fehlberg 4(5) method [60].

3.10 Model-Based Estimation of Icing Severity

Now that the modeling of the UAV dynamics and the experimental platform are determined, the final step required to meet the research objective is a state estimation framework based on the UAV model and the sampled data. As the full nonlinear state equations and the linearized state-space model have already been introduced and utilized in the system identification, a natural choice of state estimator would be the EKF. It is also the most widely used tool for nonlinear state estimation. The main advantage of this is that both the nonlinear state equations and the linearized state-space model include the stability and control derivatives. Hence, multiple EKFs can run in parallel with different sets of coefficients. The model fit errors of the different EKFs can then be compared, giving an indication of which set of coefficients gives the best approximation to the dynamics of the UAV. The different sets of coefficients can be chosen as the coefficients representing different icing levels.

3.10.1 EKF for Estimation of UAV Dynamics

The UAV motion and observation models are linearized as follows.

$$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k) \approx \check{\mathbf{x}} + \mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{w}'_k \tag{3.31}$$

$$\mathbf{g}(\mathbf{x}_k, \mathbf{n}_k, \mathbf{w}_k) \approx \check{\mathbf{y}} + \mathbf{G}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k-1}) + \mathbf{n}'_k \tag{3.32}$$

where

$$\begin{aligned}\check{\mathbf{x}}_k &= \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{0}\right)\\ \mathbf{F}_{k-1} &= \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right)}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{0}}\\ \mathbf{w}'_k &= \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right)}{\partial \mathbf{w}_k}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{0}} \mathbf{w}_k\end{aligned} \tag{3.33}$$

and

$$\check{\mathbf{y}} = \mathbf{g}\left(\check{\mathbf{x}}_k, \mathbf{0}\right), \mathbf{G}_k = \left.\frac{\partial \mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right)}{\partial \mathbf{x}_k}\right|_{\check{x}_k, \mathbf{0}}, \mathbf{n}'_k = \left.\frac{\partial \mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right)}{\partial \mathbf{n}_k}\right|_{\check{x}_k, \mathbf{0}} \mathbf{n}_k \tag{3.34}$$

where $\mathbf{F}$ is the same as the system matrices $\mathbf{A}$ of equations (3.16) and (3.17), except that they are linearized for each iteration of the EKF rather than the trimmed states.

Then the classic recursive update equations for the EKF are applied [46]

$$\begin{aligned}
\text{Prediction: } \check{\mathbf{P}}_k &= \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}'_k \\
\check{\mathbf{x}}_k &= \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{0}\right) \\
\text{Kalman gain: } \mathbf{K}_k &= \check{\mathbf{P}}_k\mathbf{G}_k^T\left(\mathbf{G}_k\check{\mathbf{P}}_k\mathbf{G}_k^T + \mathbf{R}'_k\right)^{-1} \\
\text{Correction: } \hat{\mathbf{P}}_k &= \left(\mathbf{1} - \mathbf{K}_k\mathbf{G}_k\right)\check{\mathbf{P}}_k \\
\hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k + \mathbf{K}_k\underbrace{\left(\mathbf{y}_k - \mathbf{g}\left(\check{\mathbf{x}}_k, \mathbf{0}\right)\right)}_{\text{innovation}}
\end{aligned} \tag{3.35}$$

In the prediction step, the full nonlinear equations (2.9) and (2.14) for the aircraft dynamics are used by applying the force and torque equations (2.32) and (2.33).

3.10.2 Multiple Model Adaptive Estimation for Icing Detection

The EKF implementation proposed in the previous section can easily be customized for different sets of known stability and control derivatives. This is true for both the coefficients used to calculate the linearized propagation model in equations (3.16) and (3.17), as well as the full nonlinear state propagation model. Combining this with the system identification performed for the UAV in normal and iced flight conditions, it is possible to simultaneously evaluate the performance of different EKFs loaded with different sets of coefficients. This is the idea behind the MMAE scheme used in this research.

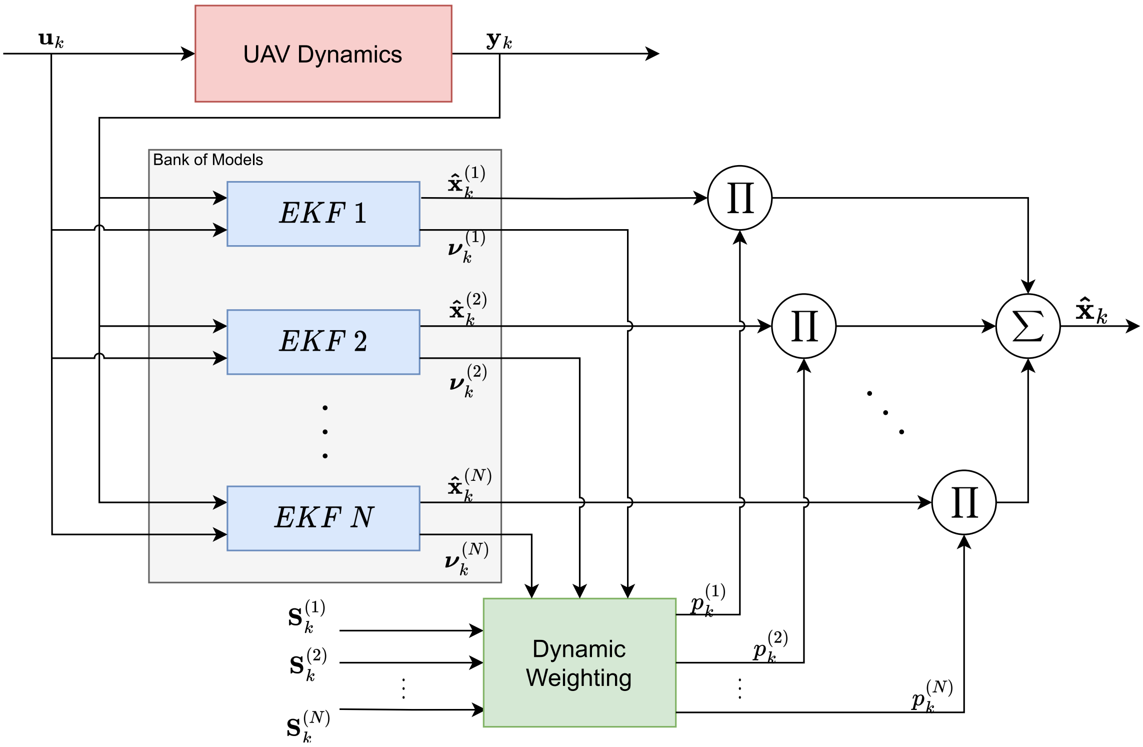

Figure 3.7: MMAE structure for icing detection and classification. The bank of possible models is marked in the area towards the left.

The MMAE will here be adopted from [61] by employing a bank of N different EKFs, each performing state estimation for different models. The chosen models represent the sets of the identified stability and control derivatives with and without icing. These are supplemented by intermediate models that are created through interpolation between the identified parameter sets. The stability and control derivatives are chosen to vary according to the model by Bragg in Equation (2.1), using identified parameters from the clean airframe ($C_{(\cdot)}^{clean}$) and the iced airframe ($C_{(\cdot)}^{iced}$):

$$C_{(\cdot)} = (1 - \eta_{ice})C_{(\cdot)}^{clean} + \eta_{ice}C_{(\cdot)}^{iced} \tag{3.36}$$

where $\eta_{ice} \in [0, 1]$ is the icing severity factor. This model assumes that each stability and control derivative varies linearly between the clean and iced values.

Figure 3.7 describes the structure of the MMAE for N models. The posterior probability is calculated with the using the innovations $\boldsymbol{\nu}_k^{(i)}$ and the innovation covariance $\mathbf{S}_k^{(i)}$ for each model $i \in \{1, ..., N\}$ at each timestep k. The dynamic weights $p_{k+1}^{(i)}$ are generated with the dynamic recursion

$$p_{k+1}^{(i)} = \frac{\beta_i e^{-\sigma_k^i}}{\sum_{j=1}^{N} p_k^j \beta_j e^{-\sigma_k^i}} p_k^i \qquad \text{for } i = \{1, ..., N\} \tag{3.37}$$

where β_i is a positive constant coefficient given by

$$\beta_j = \frac{1}{\sqrt{||\mathbf{S}_k^i||}} \tag{3.38}$$

and σ_k^i is called an *error measuring function* and is defined by

$$\sigma_k^i = \frac{1}{2}||\boldsymbol{\nu}_k^i||_{(\mathbf{S_k^i})^{-1}} \tag{3.39}$$

where the notation $||x||_Q = \sqrt{x^\top Qx}$ is used. As shown by Cristofaro et al. [21], one property of this recursion is the convergence of the weights:

$$p_k^{i^\star} \to 1, \quad p_k^j \to 0 \; \forall j \neq i^\star \tag{3.40}$$

with

$$i^\star := \arg \max_{i \in \{1,...,N\}} p_k^i \tag{3.41}$$

The icing conditions represented by the icing severity factor η are usually continuous and time-variant. For the dynamic weights being able to respond to changing η, the saturation of Equation (3.40), where $p_k^j = 0$ is undesirable. Cristofaro et al. [21] provide an updated procedure that restricts saturation and hence is able to adapt to changing icing severity in all situations. This method can be described by

$$q_{k+1}^{(i)} = \frac{\beta_i e^{-\sigma_k^i}}{\sum_{j=1}^{N} p_k^j \beta_j e^{-\sigma_k^i}} p_k^i \tag{3.42}$$

$$r_{k+1}^{(i)} = \sum_{j \neq i} \text{dead}_{1-\epsilon}(q_{k+1}^{(j)}) \tag{3.43}$$

$$p_{k+1}^{(i)} = \text{sat}_{1-\epsilon}(q_{k+1}^{(i)}) + \frac{r_{k+1}^{(i)}}{N-1} \tag{3.44}$$

where the operators

$$\text{sat}_\Delta(s) = \min(\Delta, \max(s, -\Delta)) \tag{3.45}$$

$$\text{dead}_\Delta(s) = s - \text{sat}_\Delta(s) \tag{3.46}$$

have been defined, and $\epsilon > 0$ is acting as a buffer preventing saturation. ϵ must be chosen sufficiently small, i.e. $\epsilon < 1/N$.

Chapter 4

Results and Discussion

The results from all phases of the project are presented here, starting with analyzing and discussing the execution of the flight experiments at Breivika. The data gathered in the flight tests was used for estimating the wind velocity at the experiment location, as this is needed for the system identification of the UAV. Finally, the results from simulations with an icing detection MMAE are presented.

4.1 Flight Experiments at Breivika

Flight experiments were conducted on the 9th of May 2023 at Agdenes Flyplass in Breivika. The purpose was to gather data for system identification using the methods described in the previous chapter. The pilot programmed the autopilot in Mission Planner to follow a racetrack pattern, seen in Figure 4.1. The maneuvers were performed on the long edges of the racetrack after the pilot gave the control to DUNE by toggling a switch on the RC-controller. These edges were about 500 m long when measured with the measurement tool in Google Maps, effectively translating to about 20 seconds of time for maneuvering assuming airspeeds of $20\,\mathrm{m\,s^{-1}}$ and zero wind.

The experimentation started with a calibration flight, where the intention was to gradually increase the amplitude of all maneuvers to find how large the amplitudes could become before unintended behavior was provoked. A threshold of 45 deg in both pitch and roll was set, as the pilots regarded that as an upper limit of what could be deemed safe. The maximum amplitudes of the different maneuvers can be found in Table 4.1. Also, normal flights at various airspeeds were performed in the same run in order to get some reference data.

After the calibration flight, the next flight was performed to collect data for system identification under normal conditions, to serve as a baseline model for comparison with the iced model. A maneuver plan can be found in Appendix

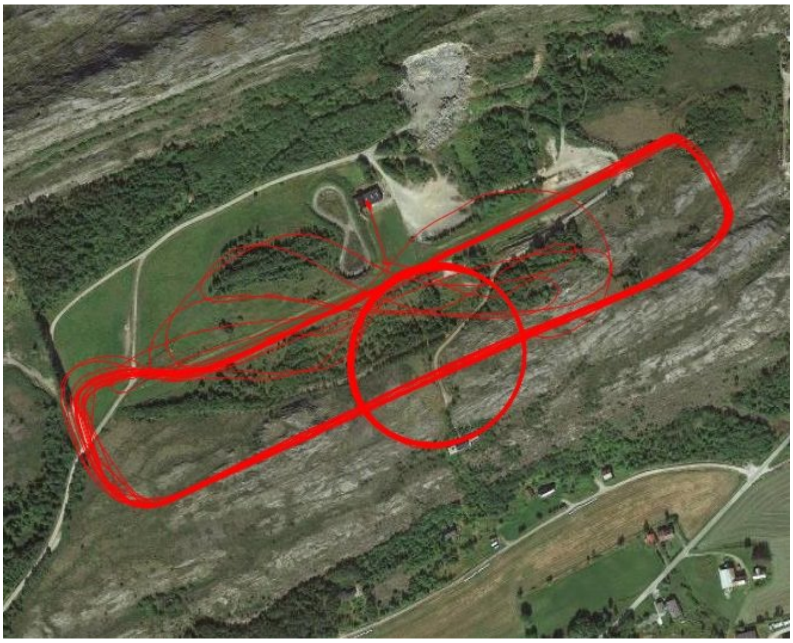

Figure 4.1: The racetrack pattern for the system identification experiments. Reconstructed by Ardupilot estimates during flight experiments. Maneuvers were performed on the long sides of the rectangle.

A.2, where the purpose is to collect data from 3 to 5 maneuvers of each type, depending on the time frame and battery level. The same maneuvers were performed at different airspeeds. Varying the airspeed changes the trim condition due to the dependency of the aerodynamic coefficients on the AOA, which in turn is dependent on the airspeed. Also, the maneuvers were performed with different magnitudes, one normal and one high. The reason for this is that the high amplitudes have the same frequency content but with a higher power, which could make the stability and control derivatives easier identifiable. Normal amplitudes were used in case the high amplitudes shifted the UAV too far away from the linearization points for working with the linearized state-space model in equations (3.16) and (3.17). Additionally, the low-magnitude maneuvers are more comparable in case the high magnitudes are too aggressive for the icing runs.

Table 4.1: Summary of maneuver magnitudes and periods used for system identification experiments. The large magnitudes were found experimentally by ensuring that the angles did not exceed 45°.

Maneuver	Normal Mag.	Large Mag.	Δt	Max angle [state]
Elevator Doublet	15°	25°	0.4 s	30° [θ]
Elevator 3-2-1-1	10°	20°	0.3 s	40° [θ]
Aileron Doublet	15°	25°	0.4 s	35° [ϕ]
Aileron 1-2-1	15°	25°	0.3 s	35° [ϕ]

The 3D-printed ice shapes were attached firmly to the leading edge of the wing using double-sided tape. The profile of the ice imitates the shape of mixed ice in Figure 4.2, representing flight in the most intrusive icing conditions. After takeoff, the UAV seemed to be highly unstable and was doing erratic and uncontrolled movements. It was too unreliable for the pilot to give control to DUNE, and the onboard autopilot was not able to stick to the trajectory programmed in Mission Planner. Shortly later, contact with the UAV was lost from ground control. A few minutes later, the UAV was found crashed and burning in a nearby gravel pit, as seen in Figure 4.3 and 4.4. All parts and components of the payload of the UAV were then already burnt beyond repair and recovery, and the Skywalker X8 UAV was discarded in its entirety.

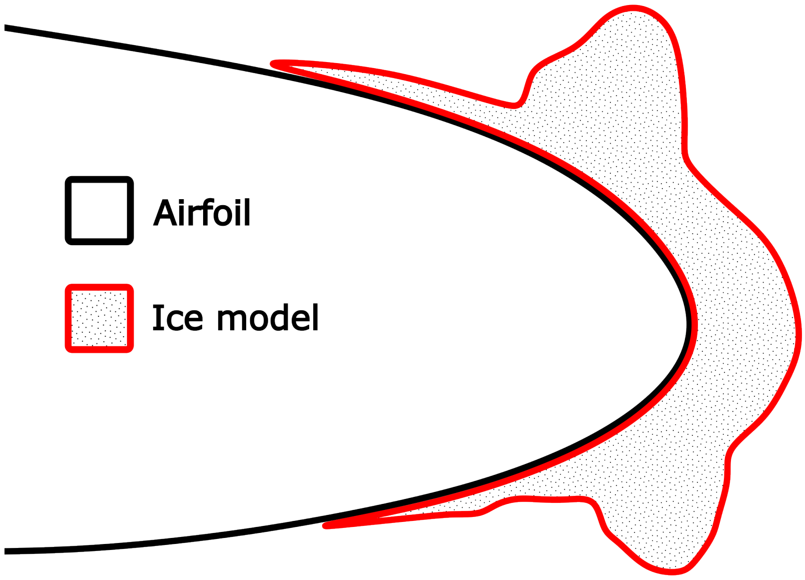

Figure 4.2: Profile of the 3D-printed ice mockup attached to the leading edge of the Skywalker X8's airfoil. The figure is based on images of the actual ice mockups.

The data collected prior to the crash was limited to multistep maneuvers conducted under normal flight conditions. This data was primarily intended as reference information for comparing with the data collected in icing conditions. As a result, no usable data for system identification could be obtained from the icing run. This was due to the absence of maneuvers performed during the very short icing run and the fact that the only available data was from the telemetry log, which had lower sampling rates.

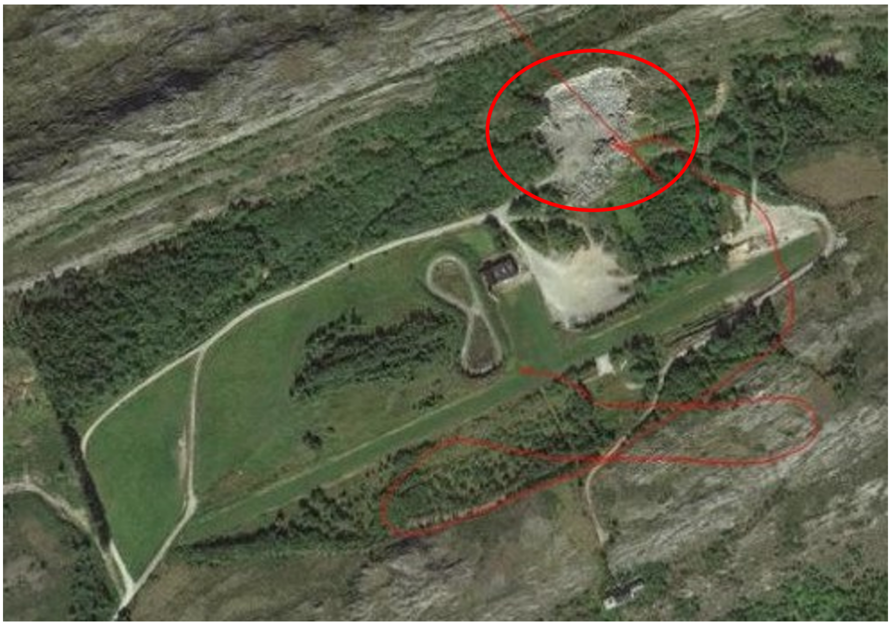

Figure 4.3: The image shows the final telemetry log updates from the Skywalker X8 in the gravel pit by the ground station. The red circle shows the crash site where it was found burning.

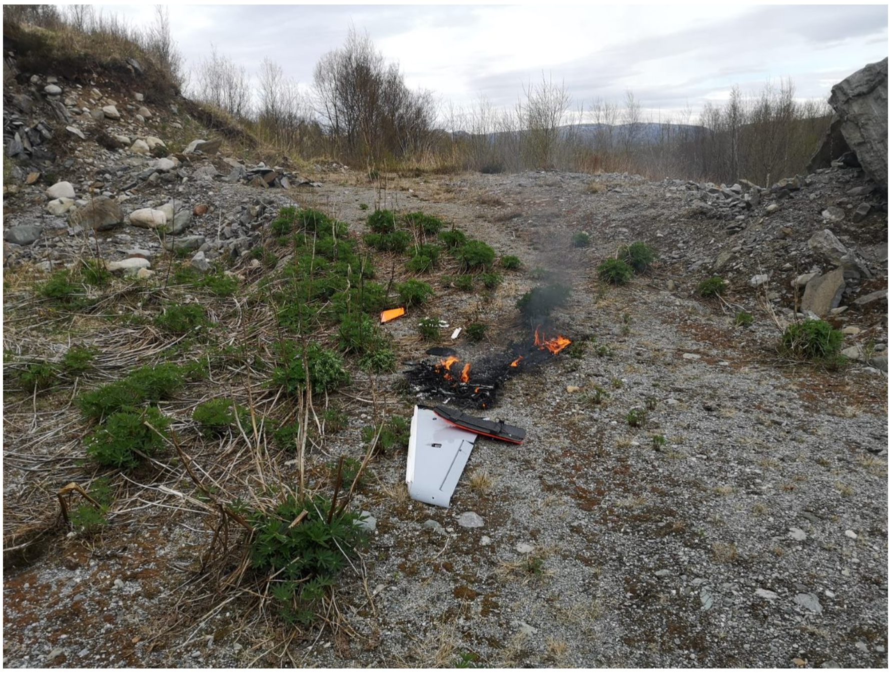

Figure 4.4: The burning Skywalker X8 UAV post-flight, with the 3D-printed ice shapes attached to the leading edge of the wing.

4.2 Synchronization of Experiment Data

Unfortunately, the SD card containing the high-resolution data with the least delay was unsalvageable, as it crumbled upon inspection in the burned ruins of the Skywalker X8 pictured in Figure 4.4.

Luckily, the redundant IMC messages passed from ArduPilot to DUNE contain most of the data required by the system identification. This data has a slightly lower resolution and is expected to have some more transmission delay. However, some messages, such as the state estimation from ArduPilot had a quite high sample rate of about 40 Hz. After each flight, the DUNE data was transferred to Neptus and saved securely on the ground station computer. The DUNE data also contained the high-resolution STIM300 IMU data sampled at 250 Hz from the SenTiBoard DUNE task.

The only data that was not transmitted and saved through the redundant IMC messages is the throttle PWM signal. This is likely due to an error in the DUNE task, where the PWM messages listened to the wrong PWM channel on the Cube Orange. This was instead gathered from the telemetry log and needs to be synchronized with the DUNE log. Luckily, the elevon servo PWM signals were transmitted by both the redundant logs, such that this signal can be used to synchronize the telemetry log with the DUNE log.

4.3 Wind Data Estimation

The wind velocity observer in Section 3.6 was used to estimate the wind at Breivika during the flight tests. The measurements from the pitot-static tube, the estimated Euler angles, and the velocities from ArduPilot were used for this. The Euler angles estimated by ArduPilot were in the interval $[-180°, 180°]$. These were remapped to the interval $[0°, 360°]$ by using the Matlab functions `unwrap()` to unwrap the Euler angles, followed by `wrapTo2Pi()` to wrap the angles around the desired interval. The estimated wind velocities are shown in Figure 4.5. To assess the quality of the estimated wind velocity, it can be compared to the weather data from the time period at the location. Weather history data from `http://yr.no` reports north-western winds of $8.1\,\mathrm{m\,s^{-1}}$ at 12:00, to $7.2\,\mathrm{m\,s^{-1}}$ at 13:00 at the Ørland III weather station, This is the nearest weather station to Breivika with a distance of approximately 10.8 km across the Trondheim Fjord. The absolute wind velocity of the estimates in Figure 4.5 are shown in Figure 4.6. The average estimated wind velocity lies between $8\,\mathrm{m\,s^{-1}}$ and $10\,\mathrm{m\,s^{-1}}$ during the period the experiments were conducted. This fits well with the reports from Ørland III, considering the high uncertainty in the weather report due to the long distance. While the weather report says that the wind is north-western, the estimated winds are in the north-eastern direction. This disagreement is hard to settle, as this could be caused by local variations in wind direction. This is magnified by the geography at Breivika, where the airfield lies by a rocky outcrop, pictured towards the top of Figure 4.3, which affects both wind direction and turbulence.

The scaling factor $\hat{\gamma}$ is close to 1 during the experimental flight, indicating a good calibration of the pitot-static tube and that it is operating without faults.

The estimated wind velocities were used to calculate the estimated AOA, SSA, and V_a according to Equation (2.4). The results are shown in Figure 4.7, along with a more detailed view of two consecutive longitudinal doublet maneuvers at $V_a = 18\,\mathrm{m\,s^{-1}}$ in Figure 4.9. The SSA seems very large, fluctuating between 50° and −50°. However, this is expected considering the strong wind. The AOA is highly correlated to the pitch angle, as expected. This is seen in Figure 4.8 and the more detailed Figure 4.9. It is also generally quite low compared to the findings by Winter [59] at similar airspeeds. This should lead to a weaker lift force according to Equation (2.17a). Since the UAV could still fly at a constant height, this must have been compensated for by some other factor, assuming the estimate is accurate. Looking at the wind estimation plot in Figure 4.5, a negative wind velocity would hit the UAV's airframe from below. It could help explain the lower AOA as sufficiently strong upward winds could compensate for the lift. The reference value for the V_a started at $18\,\mathrm{m\,s^{-1}}$ after takeoff, lowering to $16\,\mathrm{m\,s^{-1}}$, before increasing to $24\,\mathrm{m\,s^{-1}}$ in the final stage of the flight. The estimated V_a is slightly higher than these values during the experimental flight, which seem to lie about $2\,\mathrm{m\,s^{-1}}$ above.

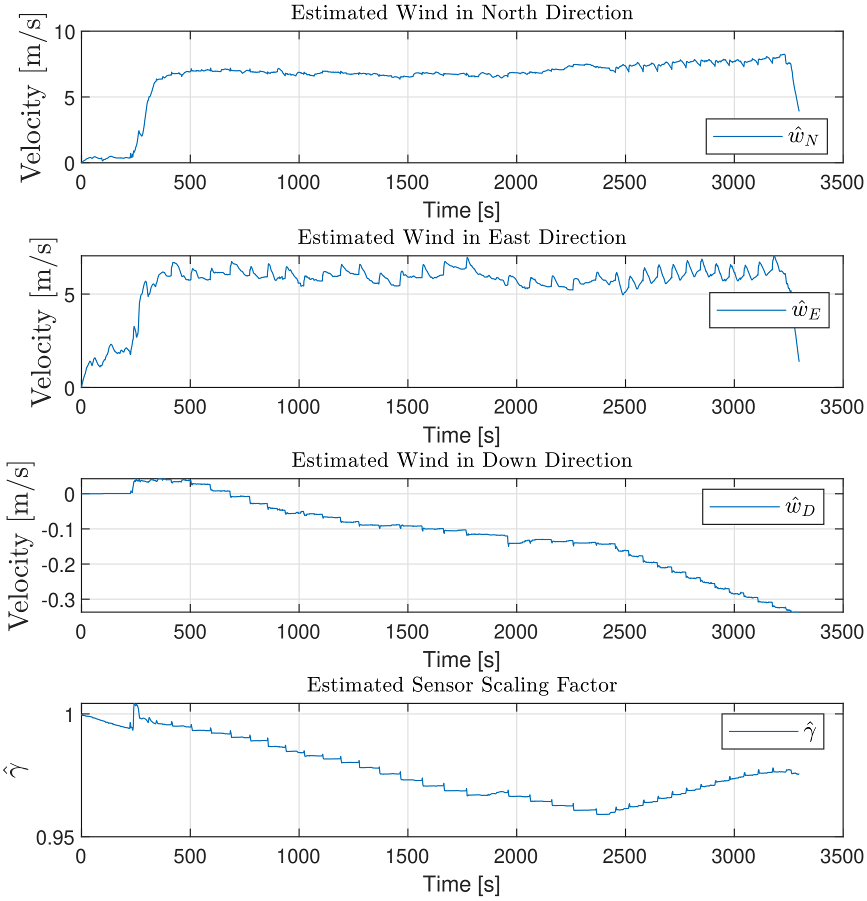

Figure 4.5: Estimated wind velocities in the north-east-down (NED) directions, and the pitot scaling factor $\hat{\gamma}$, which provides an estimate of the calibration of the pitot-static tube sensor.

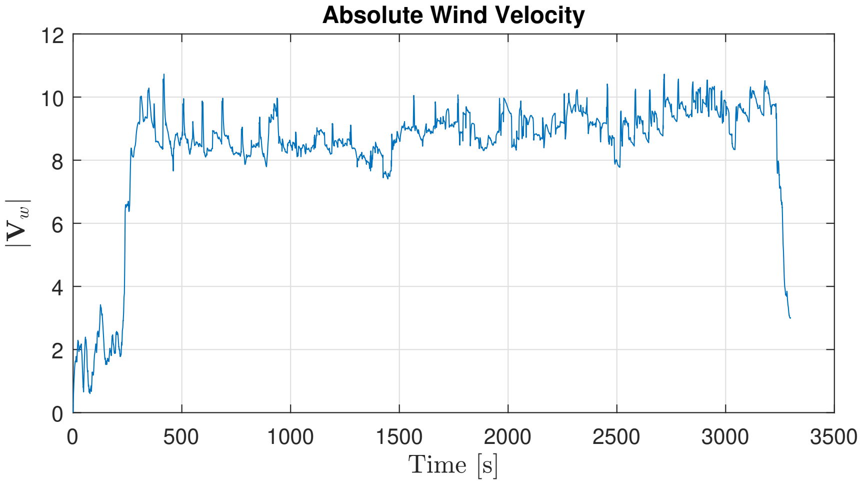

Figure 4.6: The absolute estimated wind velocity at Breivika during the entire experiment flight from takeoff to landing.

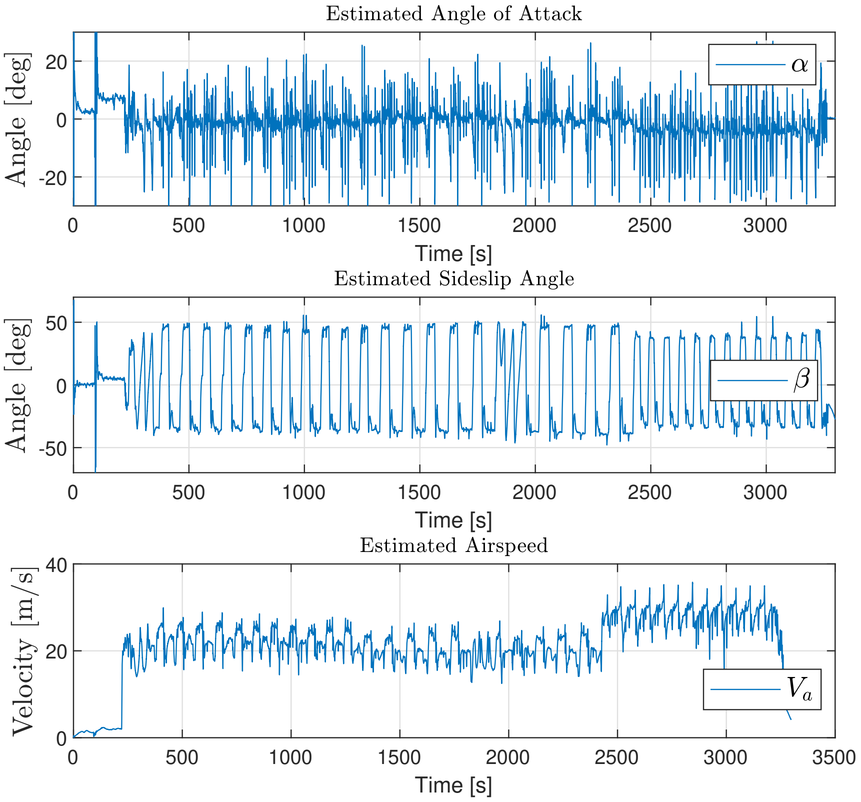

Figure 4.7: Estimated AOA, SSA and V_a.

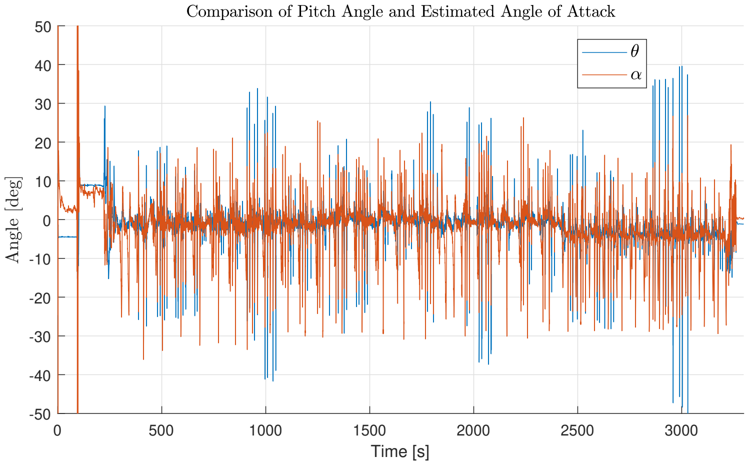

Figure 4.8: Comparison of the pitch angle and the estimated AOA for the entire experimental flight.

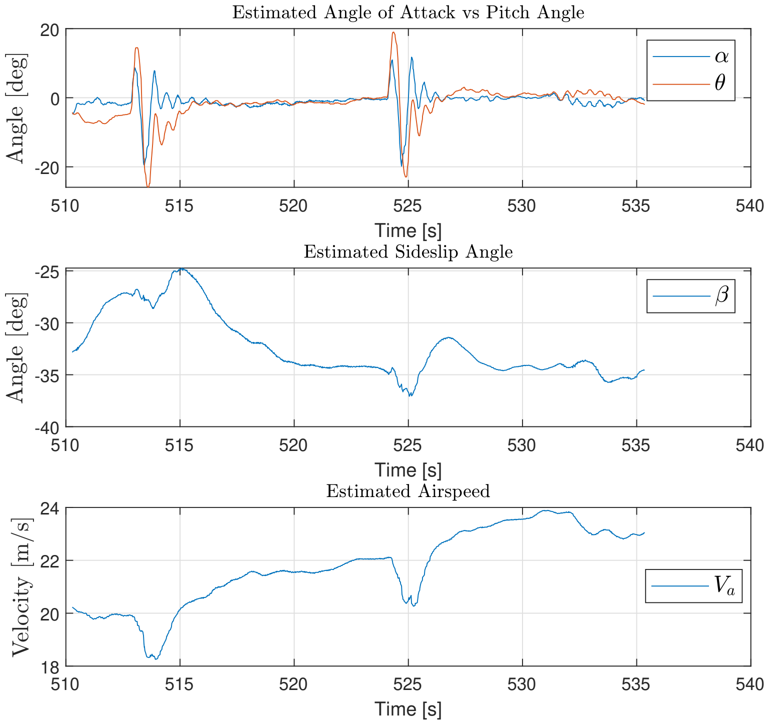

Figure 4.9: Detailed view of the estimated AOA, pitch angle, SSA and V_a for two consecutive longitudinal doublet maneuvers.

Validation of the estimated AOA and SSA is difficult without any reference data of local wind direction or measurements with a 5-hole ADS. However, for the AOA, the relationship between the AOA and the specific force f_z from Equation (2.18) measured by the IMU can be investigated. Combining f_z with the simplified model for the lift coefficient $C_L(\alpha) = C_{L_0} + C_{L_\alpha}\alpha$ from Beard and McLain [8, eq. 4.12], the f_z can be approximated by the linear relationship

$$f_z = k_0 + k_1\alpha \tag{4.1}$$

where k_0 and k_1 are paramters dependent on the airspeed V_a, rotational velocity q and the elevator deflection δ_e [57]. This approximation is valid for low AOA [8]. f_z and the AOA α are plotted against each other in Figure 4.10. The plot indicates a linear relationship between f_z and α, which would be expected. The highest concentration of markers lies about $f_z \approx 10\,\mathrm{m/s^2}$, which stems from most of the data being collected in trimmed flight between the maneuvers. In this case, the accelerometer should measure a specific force close to the gravitational acceleration $g = 9.81\,\mathrm{m/s^2}$.

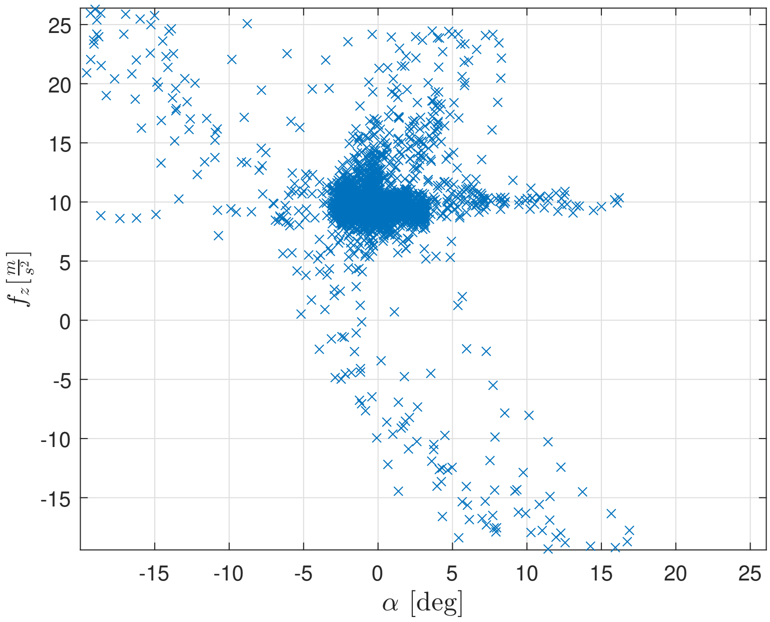

Figure 4.10: AOA correlations from the set of longitudinal doublet maneuvers at $V_a = 18\,\mathrm{m\,s^{-1}}$ with 15° deflection angles.

The same linear approximation can be used for f_y and its dependency on the SSA β for low β. However, as the data was collected in strong sideways winds, the magnitude of β mostly lies between 35° and 50°, which is too large.

4.4 System Identification of Flight Data

The flight data was exported to Matlab and assembled into the data structure presented in Figure 3.6. All different configurations of maneuvers are listed in Appendix A, where they are separated by airspeed, deflection angle, and actuated control surface. The maneuvers were performed by stabilizing the UAV in trim at a desired airspeed before executing the maneuvers. These were executed by sending maneuver configurations to the onboard DUNE software through Neptus on the ground station computer, depicted in Figure 3.1. The maneuvers were executed both upwind and downwind in case there are significant differences in the aerodynamics in each direction.

4.4.1 Equation-Error

The EE system identification was performed according to Section 3.8 on the compiled data from maneuvers of the same configuration to help prevent overfitting to the data from a single maneuver.

Starting with the most basic maneuver, the estimated stability and control derivatives from the EE method applied to the dataset of longitudinal and lateral doublet maneuvers are listed in Table 4.2. The commanded airspeed for this maneuver is $V_a = 18\,\mathrm{m\,s^{-1}}$, which is the nominal cruise speed according to the pilot. The nondimensional force and moment coefficients were calculated according to Equation (3.30) and are plotted in Figure 4.11. The estimated C_L and C_m fit the observed data quite well, with R^2 score of 0.65 and 0.67, respectively, displayed in the residual plot in Figure 4.13. The C_D is less accurate in matching the observed force coefficients with an R^2 score of 0.39. It becomes negative at the largest excitations, meaning that it somehow models a negative drag force at certain points, which makes no sense physically. Hence, all drag derivatives should be positive, which is not the case in Table 4.2. The drag force coefficient is more dependent on the airspeed V_a and the modeled propulsion force F_{prop} from Equation (2.30). Hence, the bad model fit might be explained by either inaccurate estimates of the V_a or inaccurate modeling of the propulsion force. The residual plot shows that the residuals are significantly larger during the rapid increase in AOA as the maneuvers are executed. The linearized force coefficients in Equation (2.32) are only valid for small perturbations in AOA, which could explain the bad model fit when far away from these conditions. Analyzing the collinearity of the regressor data according to Equation (2.58) reveals low linear dependency, where all off-diagonal elements are far below the recommended maximum absolute value of 0.9 by Klein and Morelli [6].

$$\mathbf{X}^{*\top}\mathbf{X}^{*} = \begin{bmatrix} 1 & -0.3189 & 0.0077 \\ -0.3189 & 1 & -0.4857 \\ 0.0077 & -0.4857 & 1 \end{bmatrix} \tag{4.2}$$

where

$$\mathbf{X} = \begin{bmatrix} \alpha(1) & \frac{c}{2V_a(1)}q(1) & \delta_e(1) \\ \alpha(2) & \frac{c}{2V_a(2)}q(2) & \delta_e(2) \\ \vdots & \vdots & \vdots \\ \alpha(N) & \frac{c}{2V_a(N)}q(N) & \delta_e(N) \end{bmatrix} \tag{4.3}$$

and N is the number of samples.

The same maneuver was performed in the lateral direction by applying the same input to the aileron. The estimated stability and control derivatives are also listed in Table 4.2. The resulting lateral force coefficients calculated according to Equation (3.30) are shown in Figure 4.12. From the residual plot in Figure 4.13, it is seen that the resulting model fit from lateral doublet maneuvers with $\delta_a = 15°$ are generally better than for the longitudinal models, and the residuals are more distributed in the 95 % confidence interval. As these are not directly dependent on the F_{prop} and are assumed decoupled from the AOA, this strengthens the hypothesis about the bad modeling of these. Similarly to the longitudinal regressors, the lateral regressors also show low collinearity, with

$$\mathbf{X}^{*\top}\mathbf{X}^{*} = \begin{bmatrix} 1 & -0.0119 & -0.0012 & 0.0731 \\ -0.0119 & 1 & 0.2009 & 0.1214 \\ -0.0012 & 0.2009 & 1 & -0.4392 \\ 0.0731 & 0.1214 & -0.4392 & 1.0000 \end{bmatrix} \tag{4.4}$$

where

$$\mathbf{X} = \begin{bmatrix} \beta(1) & \frac{b}{2V_a(1)}p(1) & \frac{b}{2V_a(1)}r(1) & \delta_a(1) \\ \beta(2) & \frac{b}{2V_a(2)}p(2) & \frac{b}{2V_a(2)}r(2) & \delta_a(2) \\ \vdots & \vdots & \vdots & \vdots \\ \beta(N) & \frac{b}{2V_a(N)}p(N) & \frac{b}{2V_a(N)}r(N) & \delta_a(N) \end{bmatrix} \tag{4.5}$$

and N is the number of samples.

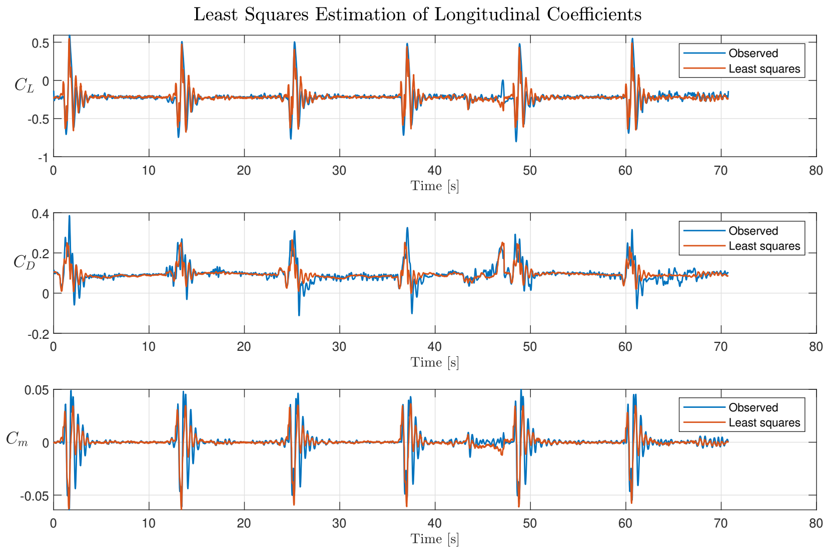

Figure 4.11: Longitudinal aerodynamic coefficients calculated according to Equation (3.30) using least squares estimates. They are plotted against the observed coefficients calculated with Equation (2.22) using flight data from elevator doublets with $|\delta_e| = 15°$ at $V_a = 18\,\mathrm{m\,s^{-1}}$.

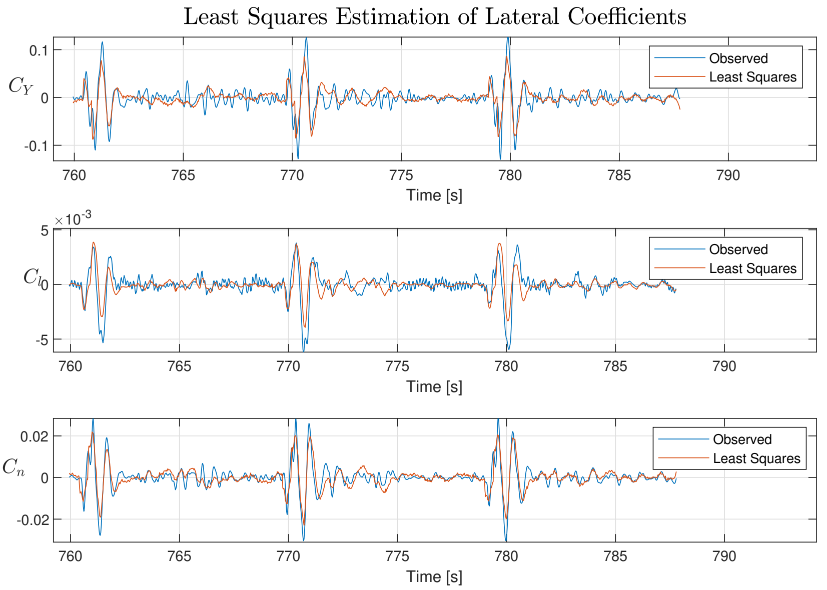

Figure 4.12: Estimated and measured lateral force coefficients from least squares estimation of doublet maneuvers with $|\delta_a| = 15°$.

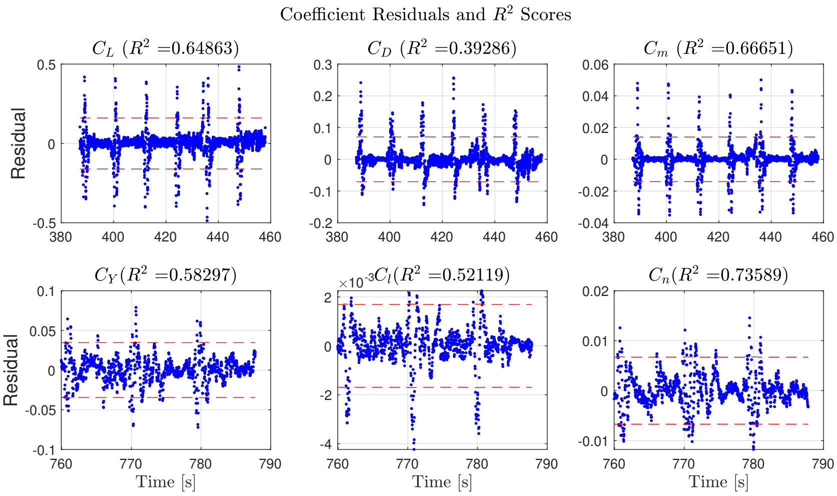

Figure 4.13: Residuals and R^2 scores from longitudinal and lateral least squares estimation of force coefficients of doublet maneuvers with amplitudes $\delta_e = \delta_a = 15°$. The red lines show the bounds of the 95 % confidence interval.

Moving on to the more complex 3-2-1-1 longitudinal and the 1-2-1 lateral maneuvers, the results are very similar to the results from the doublet maneuvers. Only the R^2 scores of $C_m = 0.71$ and $C_n = 0.67$ differ from the previous least squares estimates with any significance, where the pitch moment force coefficient model fit improved, while the yaw moment force coefficient worsened. The lateral residuals are in this case more similar to the longitudinal residuals, with larger errors during maneuvers. Comparing the standard errors of the doublets in Table 4.2 with the new standard errors in Table B.1, there is a general improvement in the confidence in the estimates, while the values of the estimated coefficients are very similar.

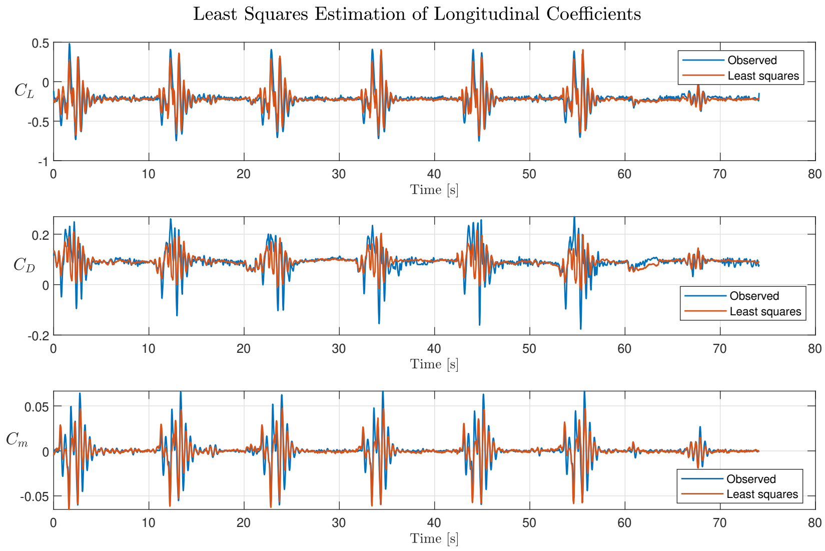

Figure 4.14: Estimated and measured longitudinal force coefficients from least squares estimation of 3-2-1-1 maneuvers with $|\delta_a| = 10°$.

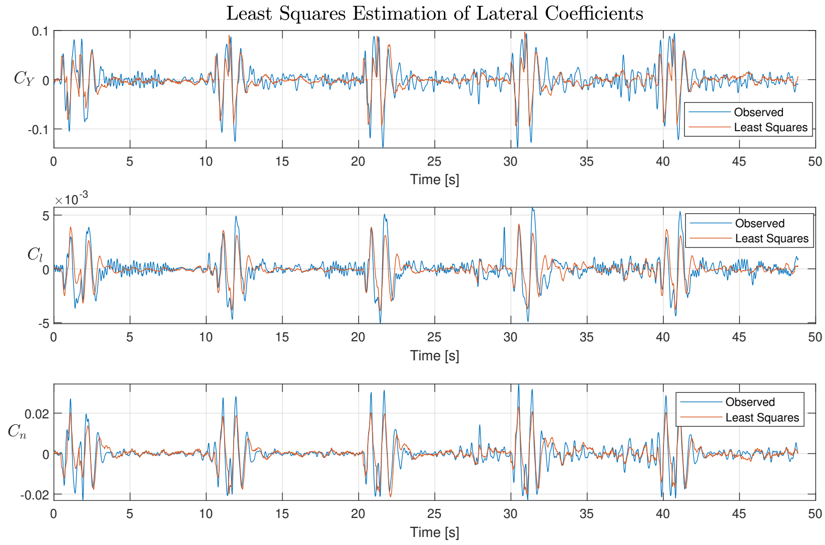

Figure 4.15: Estimated and measured lateral force coefficients from least squares estimation of 1-2-1 maneuvers with $|\delta_a| = 15°$.

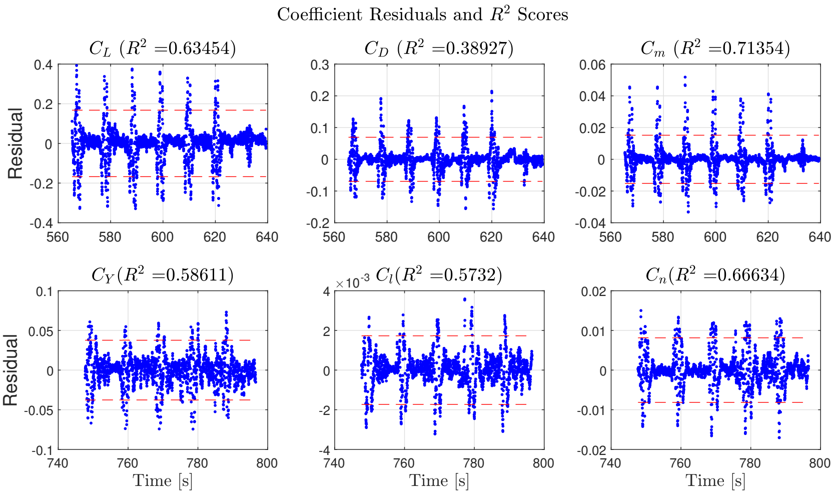

Figure 4.16: Residuals and R^2 scores from longitudinal and lateral least squares estimation of force coefficients from longitudinal 3-2-1-1 and lateral 1-2-1 maneuvers. The actuator deflection angles were 10° and 15°, respectively. The red lines show the bounds of the 95 % confidence interval.

Table 4.2: Identified derivatives from EE and OEM system identification based on longitudinal and lateral doublet maneuvers with 15° amplitude and $\Delta t = 0.4\,\mathrm{s}$. The reference values are taken from wind tunnel and XFLR5 analyses by Gryte et al. [40].

	Reference	**EE**	σ^2	$\lvert t_0 \rvert$	**OEM**
C_{L_0}	0.0867	-0.2382	0.0017	144.27	2.7206
C_{L_α}	4.02	0.1737	0.0198	8.7858	0.2032
C_{L_q}	3.87	-20.9666	0.3293	63.6708	17.9577
$C_{L_{\delta_e}}$	0.278	-0.6828	0.0224	30.4266	1.9206
C_{D_0}	0.0197	0.0837	0.0008	108.0128	0.3450
C_{D_α}	0.0791	-0.2767	0.0083	33.1704	0.0175
C_{D_q}	-	-3.3025	0.1215	27.1791	0.1670
$C_{D_{\delta_e}}$	0.0633	0.1902	0.0097	19.6728	3753
C_{m_0}	0.0302	-0.0016	0.0001	10.8699	0735
C_{m_α}	-0.126	0.0135	0.0017	7.7792	-0.456
C_{m_q}	-1.3	1.3018	0.0288	45.1843	-0.5873
$C_{m_{\delta_e}}$	-0.206	-0.0549	0.0020	27.9751	-0.0083
C_{Y_0}	0.0032	0.0033	55.5×10^{-4}	5.9295	0.0034
C_{Y_β}	-0.224	-0.0019	8.06×10^{-4}	2.3548	-0.0574
C_{Y_p}	-0.137	1.0365	0.0545	19.0318	1.0220
C_{Y_r}	0.0839	0.1409	0.0165	8.5561	0.1673
$C_{Y_{\delta_a}}$	0.0433	0.0028	9.8×10^{-5}	28.9924	0.0027
C_{l_0}	0.0041	-6.5×10^{-5}	2.7×10^{-5}	2.3859	-0.0001
C_{l_β}	-0.0849	-6.2×10^{-6}	3.95×10^{-5}	0.1579	-0.0755
C_{l_p}	-0.404	-0.0036	0.0027	1.3395	-0.8839
C_{l_r}	0.0555	0.0243	8.1×10^{-4}	30.1471	0.1013
$C_{l_{\delta_a}}$	0.12	-7.3×10^{-7}	4.8×10^{-6}	0.1511	-0.0026
C_{n_0}	-4.7×10^{-4}	-8.6×10^{-4}	1.1×10^{-4}	7.9461	-0.0009
C_{n_β}	0.0283	2.4×10^{-4}	1.6×10^{-4}	1.5078	0.0076
C_{n_p}	0.0044	-0.2447	0.0106	23.1419	-0.0992
C_{n_r}	-0.012	0.0840	0.0032	26.2654	0.0652
$C_{n_{\delta_a}}$	-0.0034	-5.0×10^{-4}	1.9×10^{-5}	26.1157	-0.0003

Equation error system identification was also performed on the same maneuvers with larger control surface deflection angles. The results from these are listed in Appendix B. The R^2 scores were generally slightly lower than for the low-amplitude maneuvers, with higher standard deviations for the estimated derivatives, especially for the longitudinal maneuvers. The same applied to the maneuvers executed with an airspeed $V_a = 24\,\mathrm{m\,s^{-1}}$. The explanation is likely that these more aggressive maneuvers drove the UAV even further away from the linearization points, making the inherent assumption of small AOA from the Taylor series expansion from the equations in (2.17) invalid. The maneuvers from the flights at $V_a = 16\,\mathrm{m\,s^{-1}}$ with the low amplitudes were very similar to the results above at $V_a = 18\,\mathrm{m\,s^{-1}}$. This makes sense, as the only difference is $2\,\mathrm{m\,s^{-1}}$ in airspeed. Since the results were so similar, only the results at the nominal airspeed $V_a = 18\,\mathrm{m\,s^{-1}}$ will be considered.

4.4.2 Output-Error Method

The estimated stability and control derivatives from the EE method in Table 4.2 were used as initial estimates for the OEM. Since the OEM requires simulation of a dynamical system, the state-space equations (3.16) and (3.17) were used, linearized around trim conditions. Since these were not found during flight experiments due to the strong winds, the mean values of the measured states were used, similarly to Klein and Morelli [6].

The OEM was applied to the data from a longitudinal and a lateral doublet maneuver at $V_a = 18\,\mathrm{m\,s^{-1}}$ using control surface deflections of 15°. The estimated coefficients are shown in Table 4.2 and differ vastly from the EE estimates. The estimated longitudinal states in Figure 4.17 show that the model fits the data well with R^2 scores well above 0.8 for all estimated states. Reconstructing the nondimensional force and moment coefficients, and comparing them with the EE estimates and the observed coefficients, reveals that the OEM estimates differ drastically. Figure 4.18 and Table 4.2 show that the constant terms from the OEM estimation C_{L_0}, C_{D_0}, and C_{m_0} have lifted the force coefficients away from the observed estimates. This indicates that the linearized model used does not fit the measured data well.

The same was also true for the lateral linearized model, where the R^2 scores were negative, implying a bad fit. This is likely due to the large SSA, which deviates far away from the small angles assumed when linearizing the nonlinear model.

The linear models were simulated with an RK2 integrator in the OEM, as in Klein and Morelli [6]. For most of the maneuvers, these simulations diverged, with only the simulations of doublet maneuvers at low deflection angles being stable. For that reason, the coefficients estimated by OEM for other maneuvers are not included with the EE estimates in Appendix B. Other ordinary differential equation (ODE) solvers, such as Matlab's `ode45`, were tested. These simulations generally did not diverge, but the estimated stability and control derivatives were abnormally large as well as the simulation times grew significantly. Hence, the OEM was not usable for this data, and the EE estimates are assumed to be the most reliable estimates.

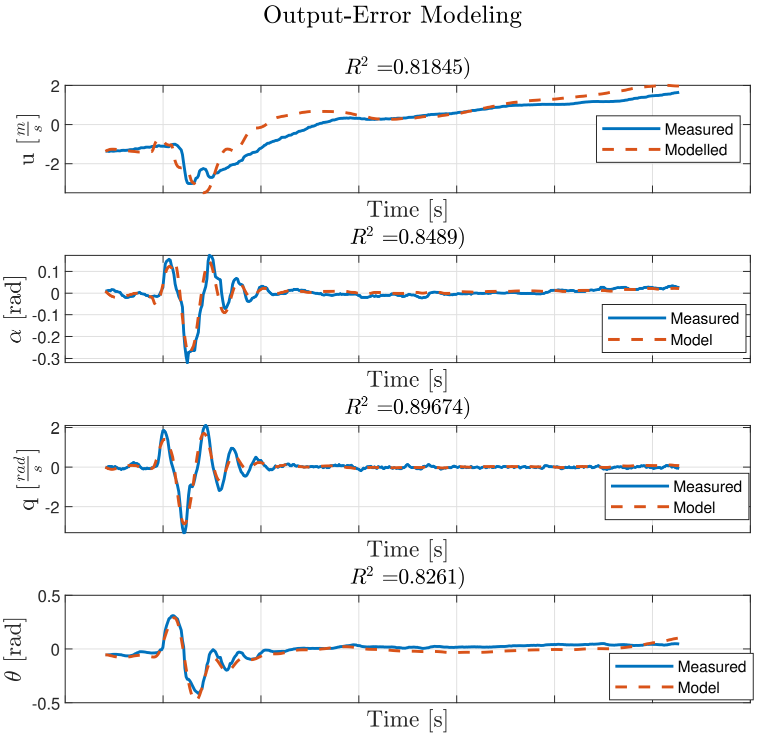

Figure 4.17: Model fit and R^2 score from the output-error method based on the linearized state-space model in Equation (3.16).

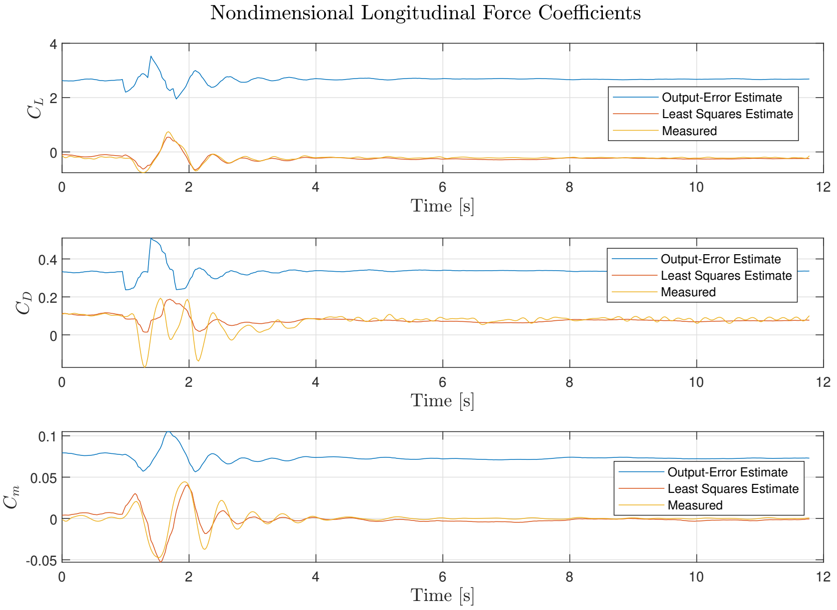

Figure 4.18: Estimated longitudinal nondimensional force coefficients from the output-error method compared to the EE estimates and the observed values.

4.4.3 Compatibility with the Simulink Simulator

The identified parameters from the flight experiments in nominal conditions were tested in the simulator implemented in Simulink from Section 3.9. Unfortunately, the simulations with these parameters inserted were highly unstable, as the control systems were not designed for this configuration. Major upgrades to the simulator are outside the scope of this thesis, and the usage of the parameters identified from real flight experiments will not be pursued further.

4.5 Multiple Model Adaptive Estimator for Icing Detection

As the parameters from the flight experiment with ice mockups from Section 4.4 could not be identified, other parameters were chosen for simulations with the MMAE. Winter [59] has performed system identification experiments of the Skywalker X8 in software-in-the-loop (SITL) simulations using a similar Simulink setup as in this project. The values used for the clean and iced stability and control derivatives are borrowed from his work and are presented in Table 4.3. For the proof of concept in this research, only the longitudinal model will be considered. The nonlinear equations for the longitudinal dynamics, together with the linearized short-period mode state-space model from Equation

(3.16) are used in the bank of EKFs.

Table 4.3: Clean and iced stability and control derivatives based on the simulations by Winter [59].

Derivative	Clean	Iced
C_{L_0}	0	0
C_{L_α}	0.2654	0.2194
C_{L_q}	4.6323	-3.3484
$C_{L_{\delta_e}}$	0.2780	0.2780
C_{D_0}	0	0
C_{D_α}	0.0188	0.0539
C_{D_q}	0	0
$C_{D_{\delta_e}}$	0.0633	0.0633
C_{m_0}	0	0
C_{m_α}	-0.331	-0.0147
C_{m_q}	-1.9920	-2.0407
$C_{m_{\delta_e}}$	-0.2060	-0.2060

With these parameters, the MMAE detection strategy presented in Section 3.10 was implemented in Matlab and Simulink. Band-limited white Gaussian measurement noise was added to the simulations according to [19], with

$$\sigma^2_{acc} = 0.001\,\mathrm{m}^2/\mathrm{s}^4 \quad \sigma^2_{gyro} = 0.001\,\mathrm{rad}^2/\mathrm{s}^2 \quad \sigma^2_{GPS} = 0.1\,\mathrm{m}^2/\text{ second}^2 \tag{4.6}$$

in addition to the noise on the simulated airspeed measurements by a pitot-static tube $\sigma^2_{pitot} = 0.001\,\mathrm{m}^2/\mathrm{s}^2$. As in [19], it is assumed that the AHRS estimates noise-free attitude angles.

Each EKF in the bank of models was tuned with the same covariance matrices and initial conditions. These were found experimentally by trial and failure. The values used in the simulations for all test scenarios were

$$\begin{aligned} R &= \mathrm{diag}(1 \times 10^{-1}, 1 \times 10^{-1}, 1 \times 10^{-1}, 1 \times 10^{-1}) \\ \mathbf{Q} &= \mathrm{diag}(7 \times 10^{-7}, 1.5 \times 10^{-7}, 3.5 \times 10^{-5}, 1 \times 10^{-6}) \\ \mathbf{P}_0 &= \mathrm{diag}(1, 1, 1, 1) \end{aligned} \tag{4.7}$$

The ground truth icing level in the simulations η_{gt} was set to vary with time in several different icing scenarios. This icing level was used to determine the stability and control derivatives according to Equation (3.36) using the parameters from Table 4.3. The dynamic weights were initialized with $p^i = 1/N$ for $i \in \{1, 2, \ldots, N\}$. The estimation works by assuming that the model with the highest dynamic weight at any given time is the one that best approximates the actual dynamics. The method was tested in three different hypothetical icing detection or estimation scenarios, demonstrating different possible applications. Results from the different scenarios are presented below.

4.5.1 Scenario 1: Binary Icing Detection

The first and simplest scenario regards detecting whether the airframe is suffering from icing or not. Icing conditions are, in this case, defined as flight with any icing severity above $\hat{\eta} > 0.25$ to allow some buffer in case of other disturbances, such as small modeling errors due to actuator inputs. This setup uses the MMAE estimator with a bank of only $N = 2$ models, which are configured with control and stability derivatives corresponding to icing severities of $\eta_1 = 0$ and $\eta_2 = 0.25$.

The ground truth icing severity factor η_{gt} is chosen to start at 0 before linearly increasing to 1 to represent a gradual accumulation of ice. Then it is kept constant for a while before decreasing linearly back to zero, simulating the ice melting off due to flight in warmer air masses. The simulation was performed by keeping the UAV in trimmed flight for 300 s. The result from the estimation presented in Figure 4.19 shows that the MMAE detector is able to detect almost exactly when the ground truth icing exceeds the threshold of where icing occurs. This is explained by the dynamic weights in Figure 4.20, calculated by Equation (3.42). The weights quickly adapt to the changing estimation accuracies of the EKFs due to the changing ground truth icing severity η_{gt}. It is noted that the detector is only detecting a fully clean airframe after the ground truth icing severity has fallen to zero. The desired behavior would be to detect changes in the icing earlier.

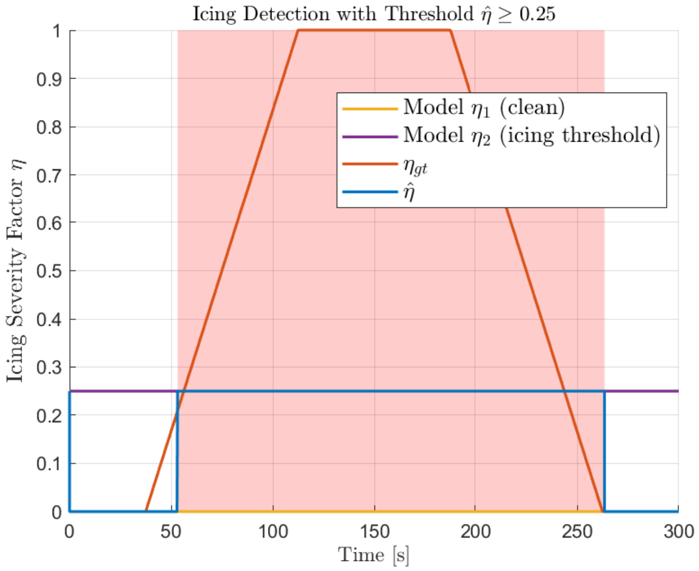

Figure 4.19: MMAE for icing detection with threshold $\hat{\eta} \geq 0.25$. The red area marks the area where icing is detected. The vertical lines represent the icing severities of the two EKFs, where the estimated icing severity $\hat{\eta}$ will always attach to one of these.

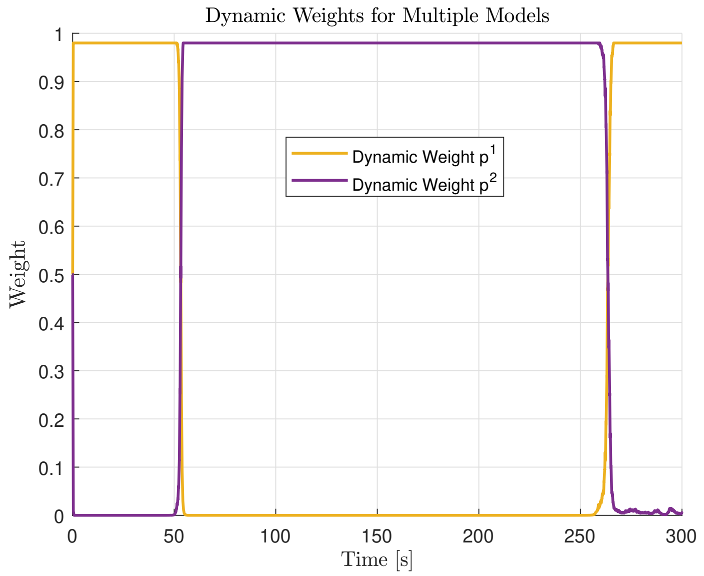

Figure 4.20: The dynamic weights during icing detection in the case of two MMAE models. The weights quickly swap places around $t = 50\,\mathrm{s}$ and $t = 260\,\mathrm{s}$ due to the two EKFs varying in accuracy as the icing severity changes.

4.5.2 Scenario 2: Estimation of Icing Severity

This scenario simulates a flight in icing conditions with gradual ice accretion and gradual ice decretion to simulate the ice melting off again due to encountering warmer air masses, similar to Scenario 1. This is repeated with two different icing severity peaks of $\eta = 0.6$ and $\eta = 1$. These were selected to assess the performance when dealing with the icing severity stabilizing between two levels, and when the icing severity changes across the range from no icing to fully iced. The result in Figure 4.21 shows that the discrete icing severity estimator with a bank of $N = 5$ EKFs with icing severities $\eta_{ice} = \{0, 0.25, 0.5, 0.75, 1\}$ is able to follow the continuous ground truth icing severity. An optimal behavior needs to be defined to quantify the performance of the discrete state estimation of the continuous signal. One possibility is to define the optimal estimate as the nearest icing severity to the ground truth η_{gt} in the predefined bank of possible icing severities η_{ice} at any time

$$\hat{\eta}_t^* = \min(|\eta_{gt,t} - \eta_{ice}^i|) \text{ for } i = \{1, 2, \dots, N\} \tag{4.8}$$

where t is the timestep and N is the number of icing severities in the MMAE model bank. The residual between the optimal and actual estimates is plotted in Figure 4.23. It is seen that the estimate is almost always slower than the optimal behavior, but it always finds the correct icing severity. The identification time difference between the optimal transitions in estimated icing severities and the actual transitions in estimated icing severity is included in the figure.

These confirm the time difference observation, as the average time difference is approximately $\Delta t = 4.45\,\mathrm{s}$, with the average of the absolute time differences being $5.47\,\mathrm{s}$. This indicates that the estimation is lagging behind the optimal behavior. Although lagging by some seconds, the estimator's performance can be considered good.

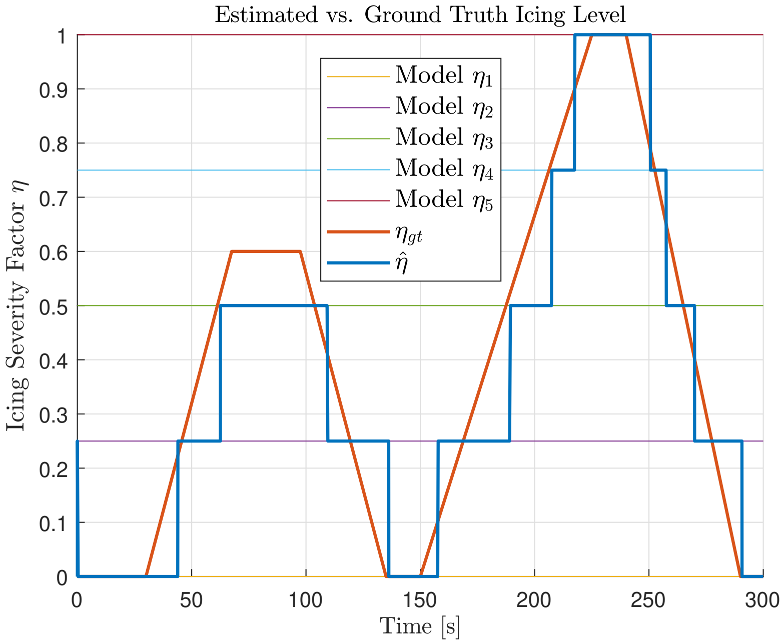

Figure 4.21: Estimated and ground truth icing severity factors for linear accretion and melting of ice. The horizontal lines display the icing severity factors of each model in the MMAE bank of models.

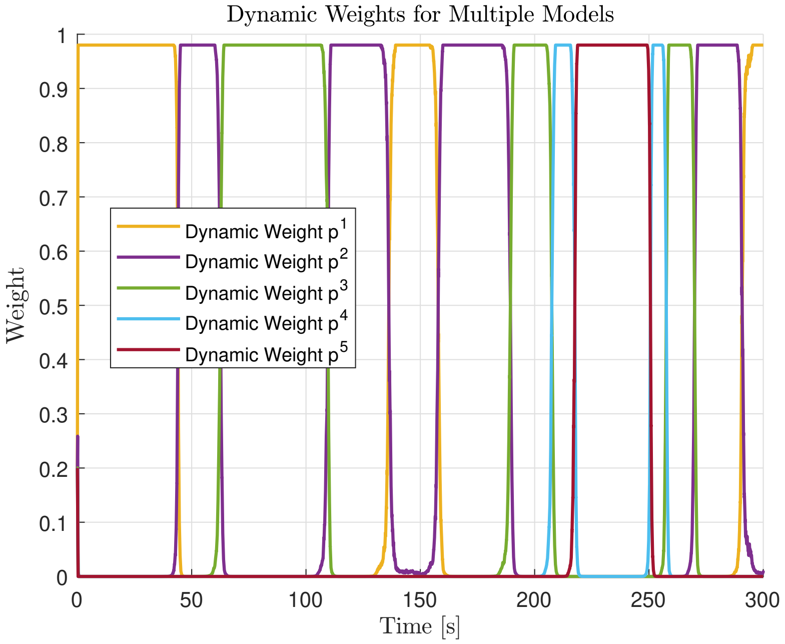

Figure 4.22: The dynamic weights during a simulated icing run with a linear accretion and melting of ice. The dominant dynamic weight at each timestep is assumed to represent the best-fitting icing severity factor.

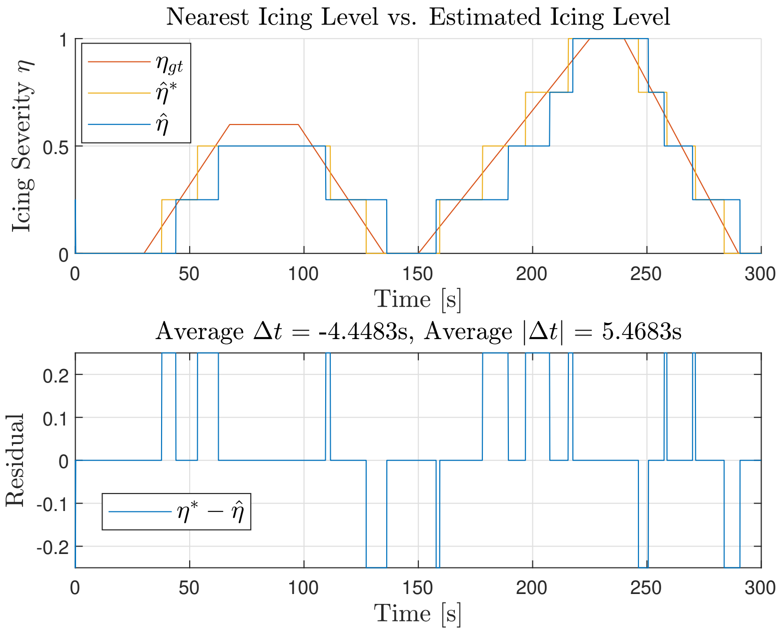

Figure 4.23: The residual between the estimated icing level and the optimal estimate.

4.5.3 Scenario 3: Estimation of Icing Severity With Shedding

In this scenario, the icing gradually builds up from $\eta_{ice} = 0$ at a steady rate and is held constant for a while before suddenly falling back to zero. This leads to a sudden change in dynamics due to the virtual ice shedding off. This simulates icing being actively removed by heating the airframe in-flight. For this case, the same bank of $N = 5$ icing models from the previous scenario were chosen with the icing severities set to $\eta_{ice} = \{0, 0.25, 0.5, 0.75, 1\}$. The resulting icing estimation is shown in Figure 4.24, with the corresponding dynamic weights displayed in Figure 4.25.

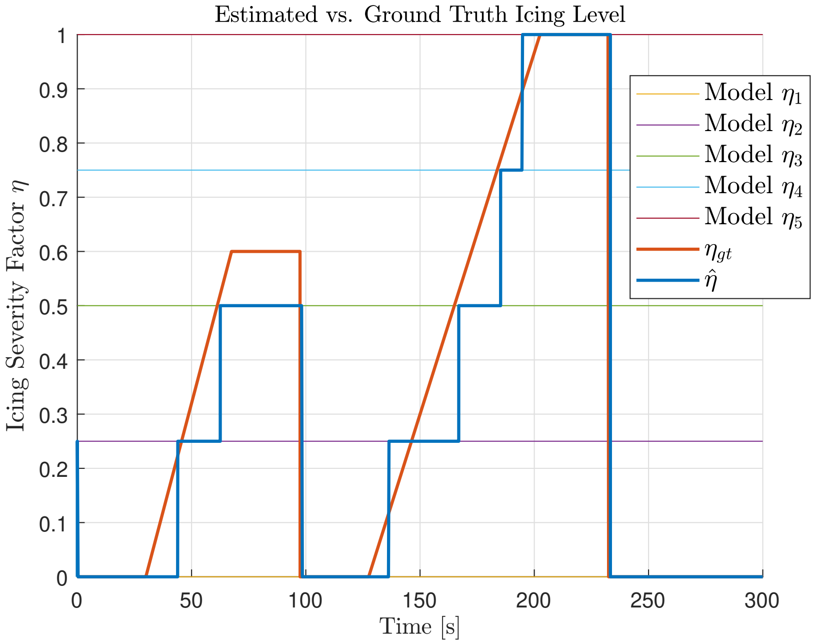

Figure 4.24: Estimated and ground truth icing severity factors for a linear ice buildup with shedding. The horizontal lines display the icing severity factors of each model in the MMAE bank of models.

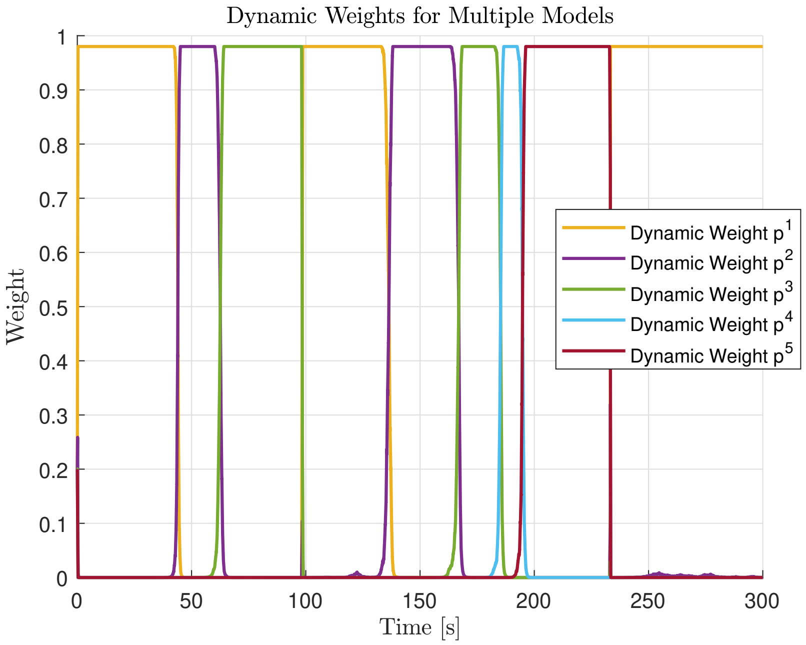

Figure 4.25: The dynamic weights during a simulated icing run with a linear ice buildup with shedding. The dominant dynamic weight at each timestep is assumed to represent the best-fitting icing severity factor.

The figures show that the MMAE estimator is robust towards rapid changes in the dynamics caused by ice shedding. The estimated icing severity factor $\hat{\eta}$ responds rapidly to the falling η_{gt} and correctly identifies the icing severity. The response to the linearly increasing η_{gt} is a bit slower, with the average time difference between the optimal estimator and the estimated icing severity of around 5 s. Preferably, it should always switch to the model with the icing severity closest to the ground truth. However, the response in Figure 4.24 shows that it often only switches to the model closest to the ground truth after it has surpassed this icing severity. This means that the estimator almost always underestimates the icing severity, which is undesirable. The residual between the optimal and actual estimates is plotted in Figure 4.26.

The outputs from every EKF in the model bank between the two cases of ice shedding are displayed in Figure 4.27. The progression of the estimates is clearly shown in the slope between $t = 130$ s and $t = 230$ s where the ground truth icing severity increases from $\eta_{ice} = 0$ to $\eta_{ice} = 1$, before the rapid fall back to $\eta_{ice} = 0$ shortly after. This leads to a progression in the residual of each EKF, leading to changes in the dynamic wing. The estimated pitch angles θ of the EKFs seem mostly identical except for being shifted vertically in relation to each other. This could be due to bad modeling or tuning, as the behavior would be expected to be more similar to the AOA α.

Overall, the result is very promising as the lagging identification times of the MMAE estimator would likely be solved with a better tuning of the EKFs.

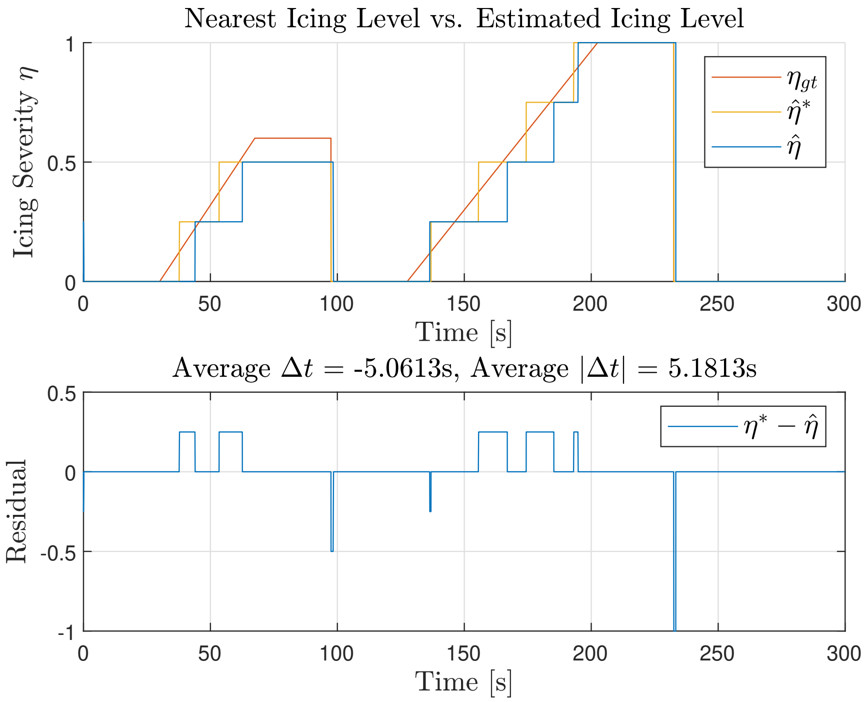

Figure 4.26: The residual between the estimated icing level and the optimal estimate.

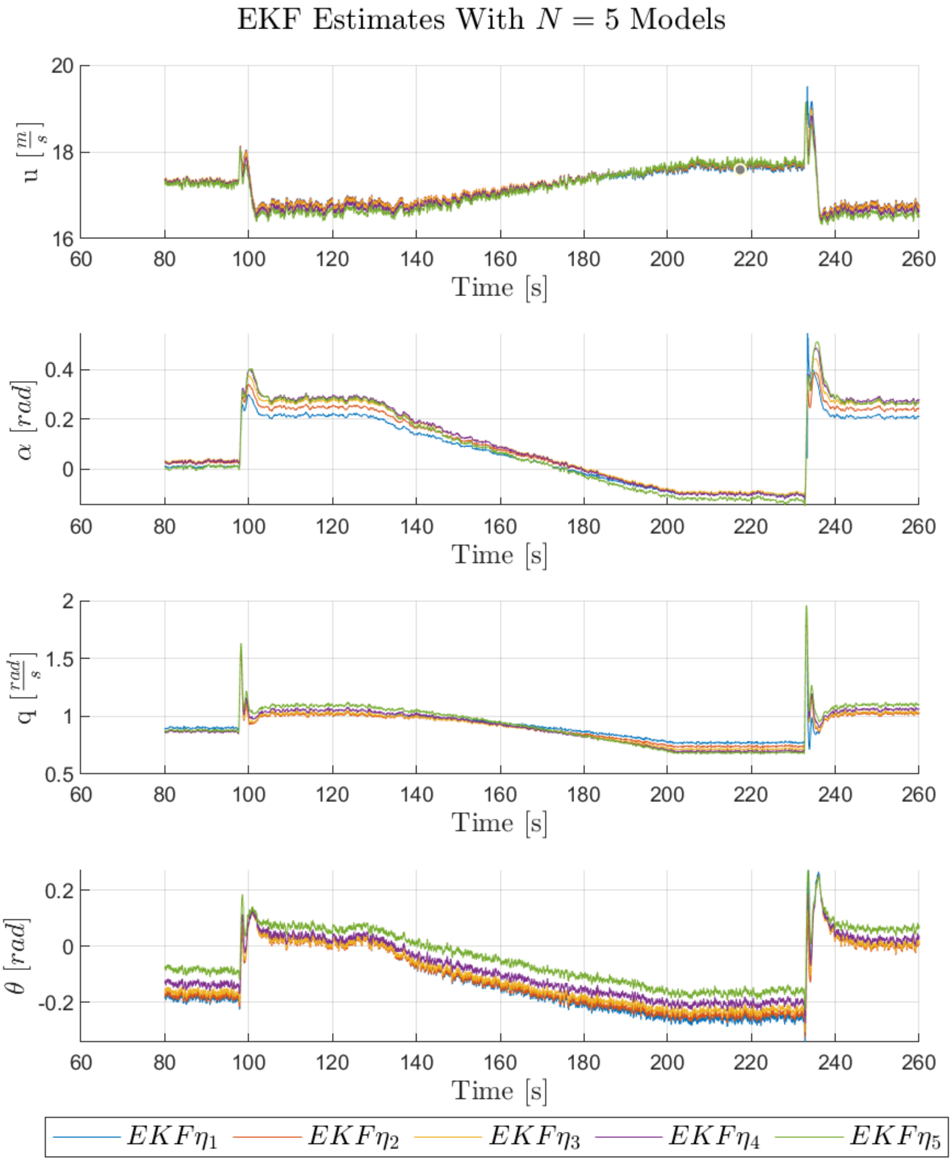

Figure 4.27: EKF estimates from the bank of $N = 5$ models. Each EKF is running with a different set of stability and control derivatives corresponding to the icing severity it evaluates.

Chapter 5

Conculsions and Future Work

This final part of the thesis aims at drawing final conclusions about the most significant results from the experiments and analyses performed during the research of model-based in-flight icing detection for fixed-wing UAVs. A selection of recommendations and some necessary considerations for further exploration of the matter are also discussed.

5.1 Conclusion

The main goal of this thesis was to investigate whether icing can be indirectly detected and estimated in flight based on changes in the observed dynamics of UAVs using a basic sensor suite. To develop a model that incorporates parameters varying with icing, it was essential to perform system identification through real flight tests conducted in both nominal and artificial icing conditions. The experiments were carefully designed to optimize system identification by exciting the decoupled longitudinal and lateral dynamics near their dominant frequencies. This step required an *a priori* linearized model of the UAV, which was based on wind tunnel testing performed by the NTNU UAV-Lab, and trim conditions found in simulations. This *a priori* model does not accurately represent real flight experiments, resulting in some uncertainty in the optimal input design. This is partially compensated for by the richer frequency content of the more complex maneuvers, and the input design is assumed to excite the relevant frequencies sufficiently for identifiability. The conditions at Beivika were not ideal for system identification, especially with the primitive pitot tube in the strong winds. As a result, the experimental setup led to different conditions in each direction of the maneuver space, which were not initially considered in the design. A more customized input design, with different excitation periods and amplitudes for different icing and trim conditions, would be difficult to execute during flight tests. This is mainly due to the framework chosen for setting up maneuvers in Neptus and sending them to DUNE in-flight was slow even with the few maneuver configurations chosen for this experiment.

Artificial icing mockups were mounted on the leading edges of the flying wing to emulate flight in icing conditions. Combined with the strong winds, this made the Skywalker X8 highly unstable, leading to an inevitable crash shortly after. Hence, only the data from the experiments in nominal conditions were available for analysis and system identification. The wind, and in turn, the AOA, SSA, and airspeed were estimated by implementing the wind velocity observer proposed by Johansen et al. [57]. This proved successful, almost matching the wind conditions at the nearest weather station. Although the wind direction was not matching the weather data, local winds are assumed to differ slightly from the weather station across the Trondheim Fjord. The estimated AOA was surprisingly low, which might have been caused by upward winds during flight experiments. The entire wind estimation could be avoided by including a 5-hole ADS instead of a pitot-static tube. Then, the AOA, SSA, and airspeed would be easily measured. This would also significantly increase the cost of the development platform but could be justified by the higher data quality for the system identification step. An improved model of the stability and control derivatives in normal and icing conditions would make the real-time state estimation more accurate in flight. Hence, only one 5-hole ADS could suffice for a whole fleet of aircraft with the same aerodynamic characteristics.

The system identification was unfortunately incomplete due to the lack of data from flights in artificial icing conditions. The quality of the data collected in normal conditions was good, although more favorable wind conditions would be preferred. The need for data processing for synchronization, transformations of sensor frames, and estimation highly influenced system identification, which increased the number of sources of error. The IMU data would be more accurate with proper mounting in the CG of the UAV, removing the need for transformation of the measurement frame. Instead, the CubeOrange's embedded IMU could be used to estimate the relative orientation of the coordinate frames, and the translational displacement could be approximated with measurements. After data processing, the data was synchronized and ready for system identification. Analysis of the regressors in the EE method revealed low collinearity, which is favorable as the estimated parameters are more identifiable. Although having low collinearity, the variances of the estimated longitudinal derivatives were a bit high, and the coefficient of determination R^2 was particularly low for the nondimensional drag force coefficient C_D, implying that the model does not fit the data well. This might be due to bad modeling of the propulsion force.

System identification with the OEM revealed that the linearized state-space model did not fit well with the data collected from the experiments, and the estimated stability and control derivatives were not used. This was likely caused by too large deviations from the linearization points, especially for the AOA.

The final step in the proposed strategy for in-flight icing detection was to perform real-time state estimation based on the identified stability and control derivatives. The simulator by Winter [59] proved to be incompatible with the identified derivatives from the flight testing, as the controllers were unable to achieve stable flight with this set of parameters. Without the iced model available, the stability and control derivatives already implemented by Winter in the simulator were used for demonstration. The MMAE proved successful with these sets of parameters and was able to detect icing, as well as estimate the icing severity in simulated icing scenarios. These scenarios were chosen to simulate the gradual accumulation and melting of ice, as well as the rapid shedding of ice, causing a significant change in dynamics. Although the simulated icing conditions were approximated to vary linearly with the icing severity, this is often considered sufficient when the goal is the detection of icing rather than a detailed description of the icing progression. The simulated cases were also only focused on flights in straight lines without any intended maneuvering, which is not representative of actual UAV operations. The performance during regular maneuvering and capability analyses in more extreme conditions is yet to be examined.

The approach of creating a multiple-model adaptive estimator by utilizing a bank of N extended Kalman filters to take advantage of the variations in identified nondimensional stability and control derivatives has demonstrated promising results. Despite identifying numerous opportunities for improvement at every stage of the process, the outcomes have been encouraging.

5.2 Recommendations for Future Work

An obvious next step in the continuation of the development of the icing detection methodology proposed is to connect the system identification with the multiple-model adaptive icing estimation and eventually perform real flight experiments.

Perhaps the most beneficial first step towards further research with the current development platform would be to improve the simulator. A necessary addition would be to implement similar control systems to the ones onboard the Ardupilot autopilot, such that better maneuvers for system identification can be found and tested in simulations. This could also make the identified models compatible with the simulations of the added MMAE in Simulink. The linearized model used for the OEM proved to fit badly with the measured data and should be improved or replaced.

Exploration of other fault detection methods might therefore also be interesting, such as performance-based ice detection methodologies proposed by Deiler et al. [62]. In the project thesis [1], sparse identification of nonlinear dynamics (SINDy), proposed by Brunton et al. [63] showed promising results in the identification of the nonlinear dynamics of the UAV in simulations and could be investigated further. However, this method was far less explainable due to the large, although sparse, identified model. The main issue was the many dynamical couplings which are usually neglected in more traditional aircraft system identification techniques. Hence, this method was not pursued further in this thesis.

It has been established that the quality of the system identification could be improved by adding a 5-hole ADS to measure the AOA accurately, SSA, and airspeed without the need for estimation. The development platform used in the experimentation of this thesis was readily available for the addition of a more advanced air data system as the SentiBoard was also able to parse and synchronize Aeroprobe ADS data from one of the many free interfaces. This could be beneficial, especially during flight testing in the Trondheim area, where wind conditions can be challenging for conducting system identification.

Bibliography

[1] P. Tronstad, *System identification for small fixed-wing uavs: With reference to airframe icing detection*, Unpublished project thesis, 2022.

[2] R. W. Gent, N. P. Dart and J. T. Cansdale, 'Aircraft icing,' *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1776, pp. 2873–2911, 2000, ISSN: 1364503X. (visited on 11/10/2022).

[3] O. Gohardani, *Progress in Aircraft Icing and Aircraft Erosion Research*, eng. Hauppauge: Nova Science Publishers, Incorporated, 2017, ISBN: 1536120308.

[4] R. Hann, A. Enache, M. C. Nielsen, B. N. Stovner, J. van Beeck, T. A. Johansen and K. T. Borup, 'Experimental heat loads for electrothermal anti-icing and de-icing on uavs,' *Aerospace*, vol. 8, no. 3, 2021, ISSN: 2226-4310.

[5] A. G. Hoover, 'Aircraft ice detectors and related technologies for on-ground and inflight applications,' U.S. Department of Transportation, 1993.

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

[7] R. V. Jategaonkar, *Flight Vehicle System Identification - A Time Domain Methodology* (Progress in Astronautics and Aeronautics, Volume 216). American Institute of Aeronautics and Astronautics, 2006, ISBN: 978-1-61583-074-9.

[8] R. W. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. USA: Princeton University Press, 2012, ISBN: 9780691149219.

[9] O. Simsek, S. Haser, E. Orhan and O. Tekinalp, 'Comparison of time and frequency domain identification of a fixed-wing uav,' in Jan. 2016.

[10] G. Chowdhary and R. Jategaonkar, 'Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter,' *Aerospace Science and Technology*, vol. 14, no. 2, pp. 106–117, 2010, ISSN: 1270-9638.

[11] O. Simsek and O. Tekinalp, 'System identification and handling quality analysis of a uav from flight test data,' in *2015 AIAA Atmospheric Flight Mechanics Conference*. Jan. 2015.

[12] M. Rhudy, T. Larrabee, H. Chao, Y. Gu and M. Napolitano, 'Uav attitude, heading, and wind estimation using gps/ins and an air data system,' *AIAA Guidance, Navigation, and Control (GNC) Conference*, Aug. 2013.

[13] O. Arifianto and M. Farhood, 'Development and modeling of a low-cost unmanned aerial vehicle research platform,' *Journal of Intelligent & Robotic Systems*, vol. 80, Oct. 2015.

[14] B. M. Simmons, H. G. McClelland and C. A. Woolsey, 'Nonlinear model identification methodology for small, fixed-wing, unmanned aircraft,' eng, *Journal of aircraft*, vol. 56, no. 3, pp. 1056–1067, 2019, ISSN: 0021-8669.

[15] D. Grymin and M. Farhood, 'Two-step system identification and trajectory tracking control of a small fixed-wing uav,' *Journal of Intelligent & Robotic Systems*, vol. 83, Jul. 2016.

[16] R. Hann and T. A. Johansen, 'Unsettled topics in unmanned aerial vehicle icing,' Apr. 2020. DOI: `https://doi.org/10.4271/EPR2020008`.

[17] R. Hann and T. Johansen, 'Uav icing: The influence of airspeed and chord length on performance degradation,' *Aircraft Engineering and Aerospace Technology*, 2021.

[18] A. Winter, K. Gryte, R. Hann, T. Johansen and A. Wenz, 'Stability of a flying wing uav in icing conditions,' *European Conference for Aeronautics and Aerospace Sciences 2019*, Jun. 2019.

[19] A. Wenz and T. Johansen, 'Icing detection for small fixed wing uavs using inflight aerodynamic coefficient estimation,' in *2019 IEEE Aerospace Conference*. Mar. 2019, pp. 1–9.

[20] M. M. Seron, T. A. Johansen, J. A. D. Doná and A. Cristofaro, 'Detection and estimation of icing in unmanned aerial vehicles using a bank of unknown input observers,' in *2015 5th Australian Control Conference (AUCC), Gold Coast, Australia, November 5-6, 2015*, IEEE, 2015, pp. 87–92.

[21] A. Cristofaro, T. A. Johansen and A. P. Aguiar, 'Icing detection and identification for unmanned aerial vehicles using adaptive nested multiple models,' *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 11, pp. 1584–1607, 2017.

[22] D. Rotondo, A. Cristofaro, T. A. Johansen, F. Nejjari and V. Puig, 'Detection of icing and actuators faults in the longitudinal dynamics of small uavs using an lpv proportional integral unknown input observer,' in *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, 2016, pp. 690–697.

[23] A. Dorobantu, A. Murch, B. Mettler and G. Balas, ‘System identification for small, low-cost, fixed-wing unmanned aircraft,’ *Journal of Aircraft*, vol. 50, no. 4, pp. 1117–1130, 2013.

[24] K. Scheper, G. Chowdhary and E. Johnson, ‘Aerodynamic system identification of fixed-wing uav,’ *AIAA Atmospheric Flight Mechanics (AFM) Conference*, Aug. 2013.

[25] P. Tian, H. Chao, H. P. Flanagan, S. G. Hagerott and Y. Gu, ‘Design and evaluation of uav flow angle estimation filters,’ *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 371–383, 2019.

[26] J. Ahsan, M. Ahsan, A. Jamil and A. Ali, ‘Grey box modeling of lateral-directional dynamics of a uav through system identification,’ in *2016 International Conference on Frontiers of Information Technology (FIT)*, 2016, pp. 324–329.

[27] U. Ahmad, M. Ahsan, A. I. Qazi and M. A. Choudhry, ‘Modeling of lateral dynamics of a uav using system identification approach,’ in *2015 International Conference on Information and Communication Technologies (ICICT)*, 2015, pp. 1–5.

[28] A. K. McGrail, ‘Onboard parameter identification for a small uav,’ English, M.S. thesis, 2012, p. 105, ISBN: 978-1-303-00309-7.

[29] J. Oliveira, Q. Chu, J. Mulder, H. Balini and W. Vos, ‘Output error method and two step method for aerodynamic model identification,’ Aug. 2005, ISBN: 978-1-62410-056-7.

[30] San José State University Research Foundation, *Comprehensive identification from frequency responses*, Accessed: 05.03.2023, 2022. [Online]. Available: `https://www.sjsu.edu/researchfoundation/resources/flight-control/cifer.php`.

[31] R. Venkataraman and P. Seiler, ‘System identification for a small, rudderless, fixed-wing unmanned aircraft,’ *Journal of Aircraft*, vol. 56, pp. 1–9, Mar. 2019.

[32] M. Badwan and T. A. Tutunji, ‘System identification for bixler3 unmanned aerial vehicle,’ in *2018 19th International Conference on Research and Education in Mechatronics (REM)*, 2018, pp. 93–98.

[33] S. Rao, A. Chatterjee and K. Kanistras, ‘System identification of an unmanned aerial vehicle with actuated wingtips,’ *Journal of Intelligent & Robotic Systems*, vol. 105, May 2022.

[34] E. M. Coates, D. Reinhardt, K. Gryte and T. A. Johansen, ‘Toward nonlinear flight control for fixed-wing uavs: System architecture, field experiments, and lessons learned,’ in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 724–734.

[35] M. Bragg, A. Broeren and L. Blumenthal, 'Iced-airfoil aerodynamics,' *Progress in Aerospace Sciences*, vol. 41, no. 5, pp. 323–362, 2005, ISSN: 0376-0421.

[36] M. Bragg, T. Hutchison and J. Merret, 'Effect of ice accretion on aircraft flight dynamics,' in *38th Aerospace Sciences Meeting and Exhibit*. 2000.

[37] J. Melody, T. Başar, W. Perkins and P. Voulgaris, 'Parameter identification for inflight detection and characterization of aircraft icing,' *Control Engineering Practice*, vol. 8, no. 9, pp. 985–1001, 2000, ISSN: 0967-0661.

[38] K. Gryte, 'Precision control of fixed-wing uav and robust navigation in gnssdenied environments,' Ph.D. dissertation, NTNU, 2020.

[39] D. P. Reinhardt, 'On nonlinear and optimization-based control of fixed-wing unmanned aerial vehicles,' Ph.D. dissertation, NTNU, 2022.

[40] K. Gryte, R. Hann, M. Alam, J. Rohac, T. A. Johansen and T. Fossen, 'Aerodynamic modeling of the skywalker x8 fixed-wing unmanned aerial vehicle,' Jun. 2018, pp. 826–835.

[41] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. John Wiley & Sons, Ltd, 2021, ISBN: 978-1-119-57505-4.

[42] E. M. Coates, A. Wenz, K. Gryte and T. A. Johansen, 'Propulsion system modeling for small fixed-wing uavs,' in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 748–757.

[43] C.-T. Chen, *Linear system theory and design* (The Oxford series in electrical and computer engineering), eng, 4th ed. New York: Oxford University Press, 2013, ISBN: 9780199959570.

[44] K. W. Iliff and L. W. Taylor Jr, 'Determination of stability derivatives from flight data using a newton-raphson minimization technique,' Tech. Rep., 1972.

[45] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. USA: Cambridge University Press, 2019.

[46] E. Brekke, *Fundamentals of Sensor Fusion*, 1st ed. Jan. 2020, Unpublished book for the course TTK4250 Sensor Fusion. [Online]. Available: `https://folk.ntnu.no/edmundfo/msc2019-2020/sf13chapters.pdf`.

[47] Z. Chen, C. Heckman, S. Julier and N. Ahmed, *Weak in the nees?: Auto-tuning kalman filters with bayesian optimization*, 2018. arXiv: `1807.08855 [stat.ML]`.

[48] Y. bar-shalom, X.-R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Jan. 2004, ISBN: 047141655X.

[49] E. H. Stray, 'Flight experiment based system identification of the skywalker x8 uav in icing conditions,' M.S. thesis, NTNU, Jun. 2022.

[50] *Ardupilot open-source autopilot software*, Accessed: 17.04.2023. [Online]. Available: `https://ardupilot.org/`.

[51] S. M. Albrektsen and T. A. Johansen, 'User-configurable timing and navigation for uavs,' *Sensors*, vol. 18, no. 8, 2018, ISSN: 1424-8220. [Online]. Available: `https://www.mdpi.com/1424-8220/18/8/2468`.

[52] F. L. Markley, Y. Chen, J. L. Crassidis and Y. Oshman, 'Average quaternions,' *Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 1193–1197, 4 2007.

[53] University of Porto; Underwater Systems and Technology Laboratory, *Dune Unified Navigation Environment*, Accessed: 08.02.2023, 2020. [Online]. Available: `https://lsts.fe.up.pt/software/64`.

[54] University of Porto; Underwater Systems and Technology Laboratory, *Neptus*, Accessed: 08.02.2023, 2020. [Online]. Available: `https://lsts.fe.up.pt/index.php/software/54`.

[55] University of Texas Austin Ward Cheney and D. Kincaid, *Numerical mathematics and computing*, 6th ed. Belmont, CA: Wadsworth Publishing, 2007.

[56] J. G. Balchen, T. Andresen and B. A. Foss, *Reguleringsteknikk*, 6. utgave. NTNU, Trondheim: Institutt for teknisk kybernetikk, 2016, ISBN: 978-82-7842-202-1.

[57] T. A. Johansen, A. Cristofaro, K. Sørensen, J. M. Hansen and T. I. Fossen, 'On estimation of wind velocity, angle-of-attack and sideslip angle of small uavs using standard sensors,' in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 510–519.

[58] K. Gryte, 'High angle of attack landing of an unmanned aerial vehicle,' M.S. thesis, Norwegian University of Science and Technology, 2015.

[59] A. Winter, 'Systems identification, flight performance, and control of a fixed-wing uav in icing conditions,' M.S. thesis, NTNU, 2019.

[60] T. M. Inc., *Ode45*, Natick, Massachusetts, United States, 2023. [Online]. Available: `https://mathworks.com/help/matlab/ref/ode45.html`.

[61] V. Hassani, A. P. Aguiar, M. Athans and A. M. Pascoal, 'Multiple model adaptive estimation and model identification using a minimum energy criterion,' in *American Control Conference, ACC 2009. St. Louis, Missouri, USA, June 10-12, 2009*, IEEE, 2009, pp. 518–523.

[62] C. Deiler and N. Fezans, 'Performance-based ice detection methodology,' in *AIAA Atmospheric Flight Mechanics Conference.* 2017. DOI: `10.2514/6.2017-3394`.

[63] S. L. Brunton, J. L. Proctor and J. N. Kutz, 'Discovering governing equations from data by sparse identification of nonlinear dynamical systems,' *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Mar. 2016.

Appendix A

Experiment Plan

This appendix provides the planned maneuvers and executions of maneuvers for system identification. It consists of separate longitudinal and lateral maneuvers that build on the assumption that they excite each of these dynamics separately and uncoupled, as well as coupled maneuvers. Step sequences and frequency sweeps are preprogrammed in DUNE and executed through Neptus during flight. Maneuvers are performed with and without ice mockups attached to the leading edge of the wings.

A.1 Execution of Maneuvers

Maneuvers are executed while in trimmed flight with interfering controllers turned off. Other controllers can be left on, assuming decoupled dynamics during the execution of maneuvers. The maneuvers are executed by applying various inputs and input sequences to the elevators and ailerons. For the step sequences the *a priori* optimal excitation period is found in Chapter 3.

Step Sequence Maneuvers

- Reach trim conditions with the desired airspeed.
- Configure the maneuver in Neptus and send to DUNE.
- Transfer control of control surfaces to DUNE.
- Wait for the UAV to stabilize at trim speed and record a couple of seconds of trimmed flight.
- Activate the maneuver through Neptus and apply the input step sequence to elevons.
- Let the UAV move freely with the open-loop longitudinal dynamics for 5-10 seconds.

- Repeat 3-5 times.

Frequency Sweep

- Reach trim conditions with the desired airspeed.
- Configure the maneuver in Neptus and send to DUNE.
- Transfer control of control surfaces to DUNE.
- Wait for the UAV to stabilize at trim speed and record a couple of seconds of trimmed flight.
- Apply frequency sweep input to elevons by activating the maneuver through Neptus.
- Let the UAV move freely with the open-loop longitudinal dynamics for 5-10 seconds.
- Repeat 3-5 times.

A.2 Maneuver Plan

The maneuvers executed in the flight experiment are listed here. All maneuvers are performed in normal conditions, repeated 3-5 times. The maximum elevator and aileron control surface deflections $\delta_{e/a}$ were found experimentally. Due to the crash, no frequency sweeps were performed.

Table A.1: Table of maneuvers executed in the flight experiments. All maneuvers were performed with the clean airframe in nominal conditions.

Control Surface	**Maneuver Type**	$V_a [\mathrm{m\,s^{-1}}]$	Δt [s]	$\delta_{e/a} [°]$
Elevator	Doublet	16	0.4	15
Elevator	Doublet	16	0.4	25
Elevator	Doublet	18	0.4	15
Elevator	Doublet	18	0.4	25
Elevator	Doublet	24	0.4	15
Elevator	Doublet	24	0.4	25
Elevator	3-2-1-1	16	0.3	10
Elevator	3-2-1-1	16	0.3	20
Elevator	3-2-1-1	18	0.3	10
Elevator	3-2-1-1	18	0.3	20
Elevator	3-2-1-1	24	0.3	10
Elevator	3-2-1-1	24	0.3	20
Aileron	Doublet	16	0.4	15
Aileron	Doublet	16	0.4	25
Aileron	Doublet	18	0.4	15
Aileron	Doublet	18	0.4	25
Aileron	Doublet	24	0.4	15
Aileron	Doublet	24	0.4	25
Aileron	1-2-1	16	0.3	10
Aileron	1-2-1	16	0.3	25
Aileron	1-2-1	18	0.3	10
Aileron	1-2-1	18	0.3	25
Aileron	1-2-1	24	0.3	10
Aileron	1-2-1	24	0.3	25

Appendix B

Coefficient Tables

The results from equation-error system identification of maneuvers performed at $V_a = 18\,\mathrm{m\,s^{-1}}$ are shown in this appendix. The results from the OEM are not displayed here, as discussed in Section 4.4.

Equation-Error

Table B.1: Identified derivatives from EE system identification based on longitudinal 3-2-1-1 and lateral 1-2-1 maneuvers with 10° and 15° amplitudes, respectively. Both maneuvers used periods of $\Delta t = 0.3\,\mathrm{s}$.

Derivative	**Value**	σ^2	$\lvert t_0 \rvert$	R^2
C_{L_0}	-0.2374	0.0016	144.1655	0.6345
C_{L_α}	0.1943	0.0226	8.5993	0.6345
C_{L_q}	-18.2091	0.2646	68.8115	0.6345
$C_{L_{\delta_e}}$	-0.7054	0.0212	33.2541	0.6345
C_{D_0}	0.0837	6.8×10^{-4}	124.9206	0.3893
C_{D_α}	-0.2743	0.0094	29.2972	0.3893
C_{D_q}	-3.3214	0.1097	30.3803	0.3893
$C_{D_{\delta_e}}$	0.0088	0.0097	16.2814	0.3893
C_{m_0}	4.2×10^{-4}	1.5×10^{-4}	2.8180	0.7135
C_{m_α}	0.0284	0.0020	13.8881	0.7135
C_{m_q}	1.8936	0.0239	79.0673	0.7135
$C_{m_{\delta_e}}$	-0.0207	0.0019	10.7799	0.7135
C_{Y_0}	0.0068	4.8×10^{-4}	14.0358	0.5861
C_{Y_β}	-0.0015	6.5×10^{-4}	2.2239	0.5861
C_{Y_p}	0.8700	0.0374	23.2782	0.5861
C_{Y_r}	0.1027	0.0124	8.2909	0.5861
$C_{Y_{\delta_a}}$	0.0030	7.1×10^{-5}	42.1934	0.5861
C_{l_0}	-9.8×10^{-5}	2.2×10^{-5}	4.0516	0.5732
C_{l_β}	-3.5×10^{-5}	3.00×10^{-5}	1.1904	0.5732
C_{l_p}	-0.0160	0.0017	9.3253	0.5732
C_{l_r}	0.0245	5.7×10^{-4}	43.1129	0.5732
$C_{l_{\delta_a}}$	-1.4×10^{-5}	3.3×10^{-6}	6.0824	0.5732
C_{n_0}	-0.0012	1.1×10^{-4}	11.0478	0.6663
C_{n_β}	1.8×10^{-4}	1.4×10^{-4}	1.2896	0.6663
C_{n_p}	-0.2425	0.0081	30.0017	0.6663
C_{n_r}	0.0969	0.0027	36.1796	0.6663
$C_{n_{\delta_a}}$	-4.0×10^{-4}	1.5×10^{-5}	26.0342	0.6663

Table B.2: Identified derivatives from EE system identification based on longitudinal and lateral doublet maneuvers with 25° amplitudes. Both maneuvers used periods of $\Delta t = 0.4\,\mathrm{s}$.

Derivative	**Value**	σ^2	$\lvert t_0 \rvert$	R^2
C_{L_0}	-0.2366	0.0029	82.9783	0.7466
C_{L_α}	0.2113	0.0312	6.7605	0.7466
C_{L_q}	-21.9648	0.3971	55.3134	0.7466
$C_{L_{\delta_e}}$	-0.5715	0.0265	21.5777	0.7466
C_{D_0}	0.0949	0.0015	64.1148	0.4573
C_{D_α}	-0.4418	0.0162	27.2335	0.4573
C_{D_q}	-5.9125	0.2061	28.6826	0.4573
$C_{D_{\delta_e}}$	-0.1328	0.0137	9.6623	0.4573
C_{m_0}	-0.0023	3.2×10^{-4}	7.1758	0.6363
C_{m_α}	0.0137	0.0035	3.9264	0.6363
C_{m_q}	1.0909	0.0442	24.6660	0.6363
$C_{m_{\delta_e}}$	-0.0667	0.0029	22.6153	0.6363
C_{Y_0}	0.0037	4.6×10^{-4}	8.0789	0.5467
C_{Y_β}	-0.0019	6.7×10^{-4}	2.8410	0.5467
C_{Y_p}	1.0467	0.0411	25.4595	0.5467
C_{Y_r}	0.0730	0.0127	5.7662	0.5467
$C_{Y_{\delta_a}}$	0.0024	6.6×10^{-5}	37.2105	0.5467
C_{l_0}	-1.3×10^{-4}	2.5×10^{-5}	5.2271	0.3975
C_{l_β}	-3.6×10^{-6}	3.7×10^{-5}	0.0988	0.3975
C_{l_p}	-0.0210	0.0023	9.3230	0.3975
C_{l_r}	0.0196	6.9×10^{-4}	28.1900	0.3975
$C_{l_{\delta_a}}$	-2.2×10^{-5}	3.6×10^{-6}	6.0824	0.3975
C_{n_0}	-8.3×10^{-4}	9.2×10^{-4}	9.0983	0.6909
C_{n_β}	3.4×10^{-4}	1.3×10^{-4}	2.5629	0.6909
C_{n_p}	-0.2449	0.0082	29.9515	0.6909
C_{n_r}	0.0869	0.0025	34.5068	0.6909
$C_{n_{\delta_a}}$	-4.3×10^{-4}	1.3×10^{-5}	32.8374	0.6909

Table B.3: Identified derivatives from EE system identification based on longitudinal 3-2-1-1 and lateral 1-2-1 maneuvers with 20° and 25° amplitudes, respectively. Both maneuvers used periods of $\Delta t = 0.3\,\mathrm{s}$.

Derivative	**Value**	σ^2	$\|t_0\|$	R^2
C_{L_0}	-0.2283	0.0022	103.3960	0.6693
C_{L_α}	-0.1225	0.0207	5.9151	0.6693
C_{L_q}	-19.4966	0.2728	71.4637	0.6693
$C_{L_{\delta_e}}$	-0.3430	0.0177	19.3893	0.6693
C_{D_0}	0.0924	0.0013	70.6274	0.3908
C_{D_α}	-0.3235	0.0123	26.3866	0.3908
C_{D_q}	-4.7861	0.1616	29.6238	0.3908
$C_{D_{\delta_e}}$	-0.0170	0.0105	1.6218	0.3908
C_{m_0}	-3.8×10^{-4}	2.5×10^{-4}	1.4714	0.5354
C_{m_α}	0.0112	0.0024	4.6499	0.5354
C_{m_q}	1.4637	0.0316	46.2488	0.5354
$C_{m_{\delta_e}}$	-0.0215	0.0021	10.4978	0.5354
C_{Y_0}	0.0069	4.9×10^{-4}	14.0380	0.5860
C_{Y_β}	-0.0015	6.5×10^{-4}	2.2191	0.5860
C_{Y_p}	0.8713	0.0373	23.3319	0.5860
C_{Y_r}	0.1030	0.0125	8.2466	0.5860
$C_{Y_{\delta_a}}$	0.0030	7.0×10^{-5}	42.1437	0.5860
C_{l_0}	-8.9×10^{-5}	2.2×10^{-5}	3.9743	0.5726
C_{l_β}	-3.5×10^{-5}	3.0×10^{-5}	1.1650	0.5726
C_{l_p}	-0.0158	0.0017	9.2069	0.5726
C_{l_r}	0.0247	5.7×10^{-4}	43.0455	0.5726
$C_{l_{\delta_a}}$	-1.4×10^{-5}	3.2×10^{-6}	4.2415	0.5726
C_{n_0}	-0.0012	1.1×10^{-4}	10.9888	0.6680
C_{n_β}	1.9×10^{-4}	1.4×10^{-4}	1.3150	0.6680
C_{n_p}	-0.2421	0.0081	30.0612	0.6680
C_{n_r}	0.0981	0.0027	36.3989	0.6680
$C_{n_{\delta_a}}$	-3.9×10^{-4}	1.5×10^{-5}	25.9371	0.6680

NTNU
Norwegian University of
Science and Technology