Tobias Vaage

Control Architecture Design for the Path-Following Problem and Attitude/Speed Control of a fixed-wing Unmanned Aerial Vehicle

Master's thesis in Industrial Cybernetics
Supervisor: Tor Arne Johansen
Co-supervisor: Dirk Reinhardt

May 2021

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

Tobias Vaage

Control Architecture Design for the Path-Following Problem and Attitude/ Speed Control of a fixed-wing Unmanned Aerial Vehicle

Master's thesis in Industrial Cybernetics
Supervisor: Tor Arne Johansen
Co-supervisor: Dirk Reinhardt
May 2021

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

PROJECT DESCRIPTION SHEET

Name: Tobias Vaage
Department: Engineering Cybernetics
Thesis title (English): Control Architecture Design for the Path-Following Problem and Attitude/Speed Control of a fixed-wing Unmanned Aerial Vehicle
Thesis title (Norwegian): Design av reguleringsarkitektur for banefølgingsproblemet og orienterings/hastighets regulering av et ubemannet luftfartøy

Thesis Description: Conventional control architectures in the aircraft literature are based on a set of nested control loops with increasing bandwidth from the outer to inner level based on successive loop-closure design. This architecture includes a path-following control loop which has the objective to steer the Unmanned Aerial Vehicle (UAV) towards a predefined path by passing references on e.g. airspeed, altitude, heading to a low-level autopilot. The autopilot then uses the available actuators, e.g. aileron, elevator, rudder, throttle, to achieve convergence towards the reference attitude and speed. In practice, the autopilot often consists of PI/PID controllers in the lateral and longitudinal direction based on linearizations of the dynamic model around a trimmed flight condition. The resulting controllers naturally yield good performance in the vicinity of the trim state, with less desirable behavior at more distant regions of the state space. A possible remedy is to use gain-scheduling design where a set of controllers based on different trim states can be designed and a scheduling variable such as airspeed can be used to interpolate between their respective parameters. The goal of the project is to apply this controller design for the attitude/speed control and use Line-of-Sight (LOS) guidance for path-following control. A parameterized model of the UAV is to be identified to facilitate the design of the low-level autopilot PI/PID loops that the gain-scheduling controller is based on.

The following tasks should be considered:

1. Perform a literature review on previous/related work (system identification, gain scheduling, LOS guidance) with focus on the application to fixed-wing UAVs.
2. Apply your found system identification method(s) to identify a dynamic model based on simulated sensor data.
3. Design a nominal speed/attitude PI/PID controller based on a chosen trim condition (e.g. horizontal wings-level flight).
4. Extend the nominal controller with other flight conditions using gain-scheduling design.
5. Based on your findings from the literature, extend the low-level controller with a LOS guidance controller on the outer loop.
6. Conduct a simulation study and compare the performance of the nominal controller with the gain-scheduling controller.
7. Conclude and suggest future work.

Abstract

Unmanned aerial vehicles (UAVs) are continuously becoming more autonomous, and this thesis aims to illuminate some aspects of this. The main objectives in this thesis are the identification of an aircraft model, airspeed and attitude control and how gain scheduling affects controller performance, and also deriving a path-following guidance system. The identification problem addresses how one can determine a set of aerodynamic coefficients by using the output-error method. This thesis also derives several PID control systems for control of airspeed and attitude. The control systems are extended by implementing gain scheduling, and the differences in performance are then studied further in a case study. The guidance problem is solved by using a Line-of-sight (LOS) method, where the distance between the aircraft and a desired path is minimized.

The system identification method showed ambiguous results. Some coefficients were easier to identify than others, and the longitudinal aerodynamic coefficients in general were harder to identify than the lateral coefficients. While the lateral dynamics with the identified coefficients was similar to the dynamics with the true coefficient values, the identified longitudinal coefficients rendered the dynamics unstable. Both the lateral and longitudinal control systems was able to control their respective domains. They consist of pitch/altitude, roll/course, and airspeed control, which are made of different variations of PID control. The control loops were extended by implementing gain scheduling, and the performance improved even further. That is especially true for the longitudinal dynamics. Gain scheduling did damp out unwanted oscillations, which lead to a more stable and accurate flight. The guidance system was able to follow the waypoints in a smooth and accurate manner. The system seems very sensitive to the tuning parameters, which exposes the system when the waypoints are not as easy to navigate through. However in nominal flight, the guidance system operates without issues. The system reduces the human workload down to determining positions in the world the aircraft is to fly through.

Sammendrag

(Norwegian translation of the Abstract)
Ubemannede fly blir mer og mer autonome, og denne avhandlingen har som formål å belyse noen aspekter ved dette. Hovedoppgavene i denne avhandlingen baserer seg på identifisering av en matematisk modell for UAV, orienterings- og hastighets-regulering og hvordan "Gain scheduling" påvirker ytelsen til reguleringssystemene, og til slutt utviklingen av et banefølgingssystem. Identifikasjonsproblemet adresserer hvordan man kan finne et sett med aerodynamiske koeffisienter ved å bruke en metode som er omtalt som "Output-error method". Denne avhandlingen utvikler også diverse PID reguleringssystem for regulering av fart og orientering. Reguleringssystemene blir videre utvidet ved at man legger til "Gain scheduling", hvor forskjellene i ytelse blir studert i et casestudie. Styringsproblemet blir løst ved å ta i bruk en "Line-of-sight" (LOS) metode, hvor distansen mellom flyet og en bane blir minimert.

Metoden for systemidentifikasjon viste tvetydige resultater. Noen koeffisienter var vanskeligere å identifisere enn andre, mens de longitudinale koeffisientene generelt var vanskeligere å identifisere enn de laterale. Mens den laterale dynamikken med de identifiserte koeffisientene samsvarte med dynamikken med de sanne koeffisient-verdiene, klarte ikke det longitudinale systemet å simuleres med de identifiserte koeffisientene. Både det laterale og longitudinale reguleringssystemet klarte å regulere deres respektive domener. De består av pitch/høyde-, rull/kurs-, og hastighetsregulatorer, som er designet med variasjoner av PID regulering. Reguleringssløyfene ble utvidet med "Gain scheduling", noe som forbedret ytelsen ytterligere. Dette er spesielt tilfelle for den longitudinale dynamikken. "Gain scheduling" dempet bort uønskede svingninger, noe som førte til en mer stabil og presis flyvning. Banefølgingssystemet klarte å følge gitte veipunkter på en stabil og presis måte. Systemet virker meget sensitivt til innstillings-parameterene, noe som gjør systemet sårbart ved mer krevende veipunkter. Ved normal flyvning derimot, opererer systemet uten problemer. Banefølgingssystemet reduserer den menneskelige arbeidsmengden til å definere noen ønskede posisjoner i verden man ønsker at flyet skal fly gjennom.

Preface

This thesis is submitted in partial fulfillment of the requirements for the Master of Science in Engineering (Msc) at NTNU - Norwegian University of Science and Technology. The work has been conducted at the Department of Engineering Cybernetics.

Having a background in mechanical engineering, I had an intuition of how mechanical components operate, but the communication between those systems remained a mystery. Throughout the last few years I have been fortunate enough to uncover this fascinating topic. The more I learned, the more I became interested in the control of vehicles. This thesis is an extension of a related project conducted last semester, where control of unmanned aircraft was central. The cybernetic way of thinking and my interest for airborne vehicles are the main reasons why this thesis came to be. I hope that my cybernetic journey does not end along with this academic degree, as it has been very intriguing.

Acknowledgments

I would like to thank my supervisor Tor Arne Johansen for giving me the opportunity to study the control of UAVs in my master thesis. I am grateful for the useful feedback he has given. I would also like to extend my gratification to my co-supervisor Dirk Reinhardt. The many discussions and support he has given has been of great value to me. I would also like to thank my fellow students for keeping spirits high when the pressure mounts, and also my family for the support. Without these people, this thesis would have been an impossible task, and I am very grateful for all the support and help they have provided throughout the semester.

Tobias Vaage

Trondheim, May 2021

Contents

List of Figures

List of Tables

Nomenclature

DOF	Degrees of freedom
ILOS	Integral line of sight
INS	Inertial navigation system
LOS	Line of sight
MSS	Marine Systems Simulator
NLP	Nonlinear optimization problem
PID	Proportional-Integral-Derivative (controller)
SYSID	System Identification
UAV	Unmanned aerial vehicle
WP	Waypoint

1 Introduction

Unmanned aerial vehicles (UAV) are aircraft without a human pilot. Usually UAVs are remote controlled and/or piloted by autonomous systems. These autonomous systems are under constant improvement, which ultimately leads to a more efficient and safe aviation industry. There is also a use for UAVs in the military industry, where sophisticated autonomous systems play a significant role in the outcome of military operations. For instance the guidance system that is dealt with in this thesis is derived from the same ideas as missile guiding systems. The aircraft in question in this thesis is the Skywalker X8 aircraft. The Skywalker X8 is a fixed-wing UAV in a flying wing configuration. This means that it has no tail and no clear distinction between the wing and the fuselage [12]. The control surfaces are a set of elevons, one on each wing, and a propeller. The elevons are used for both rolling and pitching motions, and these motions usually originates from ailerons (δ_a) and elevators (δ_e). The former controlling roll, and the latter controlling pitch. There is however a mathematical relationship which can be used to translate elevon signals to aileron and elevator signals [4]. This translation is necessary because the literature used for control design and modeling in this thesis is based on aileron and elevator deflections. Equation (1) describes this relationship. Even though the physical aircraft does not have ailerons and elevators, those are used to represent the elevons throughout the various tasks in this thesis.

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

where δ_{er} is the right elevon and δ_{el} is the left elevon. There are several types of control architectures in the aircraft literature which can be applied to a fixed-wing unmanned aerial vehicle. In the case of Skywalker X8 there has been developed control architectures based on optimization techniques and PID controllers. This thesis will dig deeper into the latter, and explore the possibilities within nested control loops based on successive loop-closure design. In relation to control, there will be developed a guidance system in which the control system receives course commands from. So the idea is to feed waypoints describing where the aircraft is to fly into the guidance system, and from there compute the necessary course angles. The desired course is then fed into the control system, which manifests into an input sequence that is fed to the aircraft actuators.

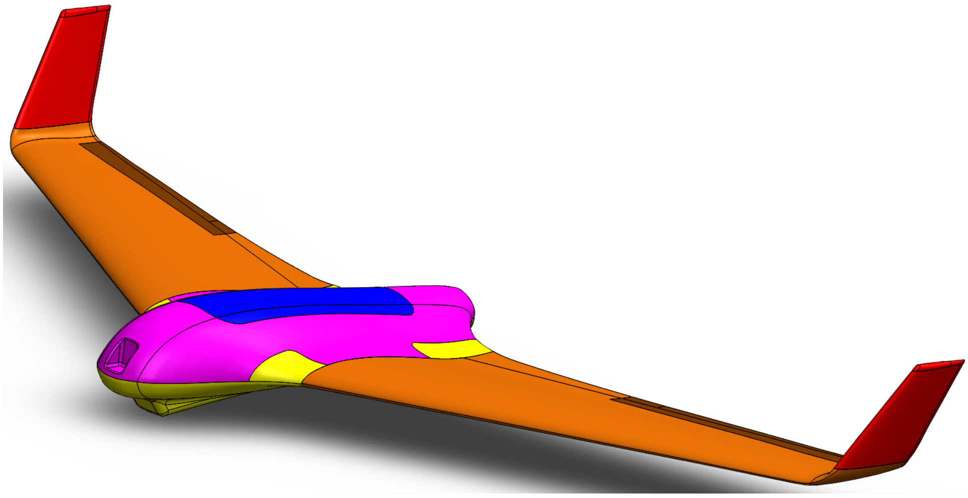


Figure 1: Illustration of the Skywalker X8 aircraft [1].

In addition to control of the aircraft, it is just as important to identify an accurate mathematical model of it. There exist some papers already which have proposed a model for the X8 aircraft, for instance [13] and [12]. In this thesis however, the ideas presented in these papers will be implemented in the derivation of a preliminary model used for system identification. System identification in this context is based on fitting the aerodynamic coefficients to the aircraft dynamics by comparing the output of a model with a set of sensor data. Describing aircraft dynamics in 6-DOF is relatively straight-forward to do generally. However the issue arises when aircraft dynamics of different sizes and shapes are to be described accurately. The surrounding air affects different aircraft in different ways, and that is where system identification becomes useful.

This thesis is organized in the following manner. Section 2 puts this work in a larger context by discussing related work and literature. Section 3 derives a preliminary model of the aircraft, and relates it to the perspective of an inertial environment. The resulting equations of motion which govern the aircraft dynamics are identified. In section 4, system identification is used to determine the aerodynamic properties of the aircraft. Section 5 deals with the control system used in this project. A lateral and longitudinal controller is derived, and gain scheduling is also introduced and implemented. In section 6 the guidance system is implemented. This is a system which takes a set of waypoints, and outputs desired course angles necessary to achieve those waypoints based on the ever-changing position of the aircraft. Section 7 addresses the case study where the difference in performance between a control system with and without gain scheduling is studied. Lastly in section 8 the thesis is concluded.

2 Literature Review

As mentioned in the introduction, there are three main problems that are addressed in this thesis; System identification, attitude/speed control, and guidance systems. These themes are all discussed in relationship to the Skywalker X8 aircraft, where the larger objective is to gain more knowledge about control and flight of small unmanned aircraft. The system identification process entails finding out how to best describe the aircraft mathematically such that it coincides with the aerodynamic behavior of a physical flight. Identifying a good mathematical model is central in the design of flight controllers. This thesis uses PID control to steer the attitude and speed of the aircraft towards a reference. This type of controller is usually used on linear systems, but it has great potential when gain scheduling is introduced. The last of the main problems is guidance. A guidance system is a system which is situated between the human and the control system. The problem is to guide the aircraft through a path that is predefined by a human interface. The various methods and ideas regarding this topic will be discussed further.

2.1 System Identification

Identifying a solid mathematical model is not a trivial task. There has been numerous efforts in doing so in the past, and some of which have had an impact of the work in this thesis. This subject can be divided into two subcomponents; modelling and system identification. Modelling entails using kinematics and dynamics in order to describe the relationship between position/orientiation and velocity/rates as well as between acceleration and forces. For this, Beard & McLain [4] has been used as a reference. This book discusses several topics regarding small unmanned aircraft. This book is a good introduction to the subject of aircraft in a technical perspective. It covers many topics broadly, which may lead to an exclusion of complex ideas. The resulting mathematical model from this book is represented by 12 differential equations. The framework given by [4] allows for further research within interesting topics such as system identification and control. Gryte [13] uses this model structure to develop a model of the X8 aircraft. In addition, this paper collected wind tunnel data in order to identify the aerodynamic coefficients. [13] describes some of the same elements as [4], and the two papers have been used together in order to design a mathematical model of the aircraft.

Given a preliminary model derived from trigonometric relationships, Newton–Euler equations, and first principles, one has to fit this to a specific aircraft. Simmons et. al [17] introduces system identification methodology for this type of aircraft. The paper discusses how one can describe a specific aircraft's dynamics by comparing a preliminary model to a set of measurement data. It is a paper which focuses on the experimental identification of a HobbyKing Bix aircraft. The findings can therefore not be continued to this project, but the methodology remains. The paper uses largely the same notation as [4], which is good for better intuition regarding the methodology. The paper discusses the identification of aerodynamic coefficients, which has to be found experimentally.

Simmons [17] uses excitation in the lateral and longitudinal direction and collects the resulting sensor data in a physical flight. These measurements are then used to compare with a preliminary model which is simulated using the same excitation. Then it is possible to use an optimization scheme to fit the aerodynamic coefficients such that the model is adapted to the measurements of the physical flight. A limitation that occurs when continuing that research to this thesis is the measurements of a physical flight. In this project, simulation is used to obtain those measurements. [17] also introduces ideas that would be interesting to investigate further, such as modal analysis for excitation frequency.

As mentioned, [13] has experimentally identified approximations of the aerodynamic coefficients, and is a source that has been used as a factual basis. The coefficients have been identified using a wind tunnel among other methods. The findings in this paper are used to determine the performance of the system identification method. It is questionable how accurate these findings are, as there are relatively large discrepancies between for instance wind tunnel results and XFLR5 results. The latter is a tool used for analyzing airfoils, planes and wings at low Reynolds numbers. However, the identified coefficients do result in a reasonable aerodynamic behavior, and are therefore considered adequate.

2.2 Attitude/speed Control and Gain Scheduling

There are numerous ways to control attitude and speed of small unmanned aircraft using model-based methods, for example optimization techniques or PID control. Model Predictive Control (MPC) and Linear Quadratic Regulator (LQR) are examples of optimization-based control [8]. This thesis is a continuation of previous work done on the Skywalker X8 aircraft, where MPC methods have been researched [18]. Beard & McLain [4] is a source that introduces PID control and successive loop closure. This method is based on creating separate control systems for separate motions, primarily lateral and longitudinal motions. The factual basis regarding successive loop closure used in this thesis is inspired by this book. This book also has a unique perspective on tuning the controller gains, where the aircraft model plays a significant role. Along with technical insight, this book provides a decent intuition about the important factors that come in to play when designing a control system for aircraft.

One of the larger objectives in this thesis is to extend a nominal attitude/speed controller by introducing gain scheduling. For this, Bett [5] has been a useful resource. This paper introduces the subject of gain scheduling in an understandable way for persons without prior expertise on the subject. The paper is part a theoretical framework for implementing gain scheduling in control systems, and part discussion on the topic. As the controller framework from [4] is already given, the theoretical basis was not easily continued in the implementation. However, the extensive discussions allowed for a good understanding of the subject and the ways it can be tied into existing control systems.

2.3 Guidance Systems

Guidance systems come in many shapes. There is *Pure Pursuit guidance* [7] and *Constant Bearing guidance* [6], but the guidance problem at hand concentrates on *Line-of-sight guidance* (LOS). Fossen [10] has been the main resource when uncovering and implementing this feature to the control system. This book is the supplementary material used in a subject at NTNU called "Guidance and Control of Vehicles", which has been an inspiration in the work conducted in this thesis. The ideas presented in the guidance part of the book allows for an intuitive understanding before delving into the theoretical basis of guidance systems. The book for instance draws associations to how a predator follows a prey in the explanation of path-following guidance systems. The intuitive understanding of guidance systems and the inherent potential regarding autonomy if implemented properly, is interesting and helpful in regards to the guidance problem.

[10] is the last iteration of many versions of the book. And so it is safe to assume that the ideas and implementation of these guidance systems are continually revised, which speaks for the factual quality. In addition to the literature, there is a related toolbox [9] which can be used as an aid in the implementation of the guidance system(s). It is important to notice that [10] discusses marine crafts, not aircraft. Therefore it is up to reader in some cases to identify what can be transferred to a aerodynamic environment, and what only coincides with a hydrodynamic environment. The book paints a picture where a guidance system is put in context to a larger system. This is useful when the guidance system is a subcomponent of a system, and makes it easier for the designer to expand on different ideas without making significant changes to the larger system. Those extensions could be observers, filters, control systems, navigation systems, or just another guidance system.

2.4 Conclusion

Based on the objectives in this thesis, there are three major components where external literature is very helpful. Those are system identification and modeling, attitude/speed control and gain scheduling, and guidance systems. There are many ways of going about these topics individually, and the literature which have been used is largely based on recommended papers and previous experience with the topics. As mentioned, the specialization project [18] serves a precursor to this thesis. That paper has been useful in the structural element of the thesis, as well as some modeling aspects.

In regards to system identification, [17] has provided methods for doing so in an understandable manner. The method is easily transferable to the X8 aircraft, and provides some interesting ideas in regards to further work. A preliminary model is provided by [4] and [13], where the latter also brings a solution for the aerodynamic parameters, which can be used to check the performance of the system identification algorithm. Identifying a good mathematical model is important for the work done throughout the thesis, because all system tests are conducted by simulating the model.

When transitioning into the control of the X8 aircraft, both [4] and [5] provide important intuition and theory. Beard [4] is used for the design of cascaded control loops and computation of controller gains. The book provides relevant and digestible theory regarding the control loops, and for this particular implementation the methodology is a good match. When extending the control system to include gain scheduling, [5] was the primary resource. This paper delves into what gain scheduling is, and how to mathematically describe its characteristics. The intuition formed by this paper was useful, but the mathematical notation and descriptions are of limited value, as the implementation is adapted to the existing control system.

The last major objective in this thesis is implementing a guidance system, and for that Fossen [10] has been used for inspiration and guidance theory. The term Line-of-sight guidance is discussed in detail and lays the groundwork for how the guidance system is implemented in this thesis. The book also provides ideas on how the system can be expanded to consider weather and collision avoidance, which could be possible subjects to investigate further. It is necessary to emphasize that this book is based on marine applications, and one has to make sure that the ideas and theory are transferable to aircraft before implementing them. The toolbox [9] which goes along with [10] is also a useful element in the implementation of guidance systems in *matlab*.

All in all, the literature supports the objectives in this thesis such that they are achievable. The combination between intuition of different topics, as well as theory describing relevant systems are what characterizes the literary sources used in this project. Some papers and books are used due to earlier experiences such as the specialization project [18] and the subject "Guidance and Control of Vehicles" at NTNU. In addition, several of the mentioned papers was recommended to the author of this thesis by faculty members with more experience in the subject of UAVs.

3 Modeling and Coordinate frames

The modeling of the Skywalker X8 aircraft is mainly based on deriving the equations of motion in six degrees of freedom (DOF) with the help of various coordinate frames. These coordinate frames are elementary in order to both derive and understand the equations of motion. They are also useful in order to include the effect of for example wind and angle of attack on the dynamics. This section is largely based on the work done in [4] and [13], and the previous work done in the spezialization project [18].

3.1 Coordinate frames

There are two relevant frames in this case, the first one being the inertial frame $\mathcal{F}^i$. It is often referred to as a North-East-Down (NED) frame. NED is a geographical reference frame which manifests in tangent planes on the surface of Earth. The axis are shown in figure 2. The i-axis points to the true North, the j-axis points towards the true East, and the k-axis points downwards towards the Earth´s surface. This reference frame is useful to get an understanding of the position and altitude of an aircraft, but it is not suitable for modeling the dynamics.

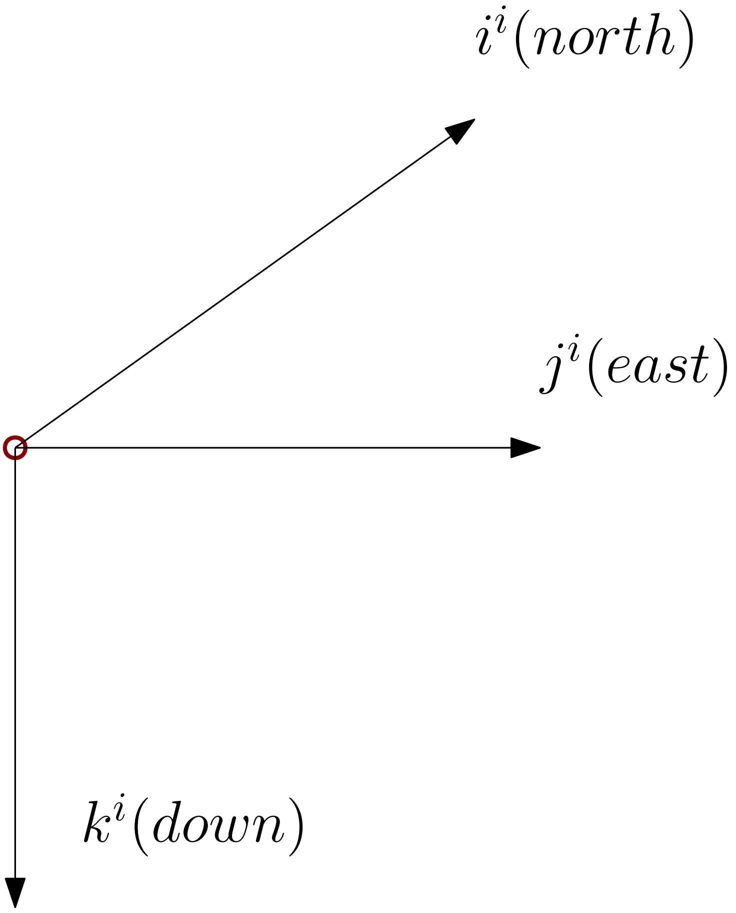


Figure 2: The inertial frame.

The second frame is a body-fixed reference frame $\mathcal{F}^b$. This is the coordinate frame which is used for modeling the dynamics of the aircraft. The origin is the center of mass, and the axis are equal to the inertial frame, only rotated by the yaw, pitch and roll angles. In deriving $\mathcal{F}^b$, the vehicle frame $\mathcal{F}^v$ is used. It is the same coordinate frame, without the three rotations. Rotating $\mathcal{F}^v$ with the yaw angle, one gets what is referred to as the vehicle-1 frame $\mathcal{F}^{v1}$. Rotating from this frame with the pitch angle, one gets the vehicle-2 frame $\mathcal{F}^{v2}$. Rotating a last time with the roll angle results in the body frame $\mathcal{F}^b$. The various rotations are illustrated in figure 3. Doing these rotations in that order is called the zyx-convention, whose purpose is to switch between $\mathcal{F}^i$ and $\mathcal{F}^b$. The mathematical manifestation of this relationship takes the shape of a rotation matrix shown in equation (2). The rotation matrices used in this project are defined as $\mathbf{R}_b^a$, where the subscript represents the frame that is rotated from, while the superscript represents the frame that is rotated to. So in the case of $\mathbf{R}_b^a$, the rotation matrix rotates from $\mathcal{F}^b$ to $\mathcal{F}^a$.

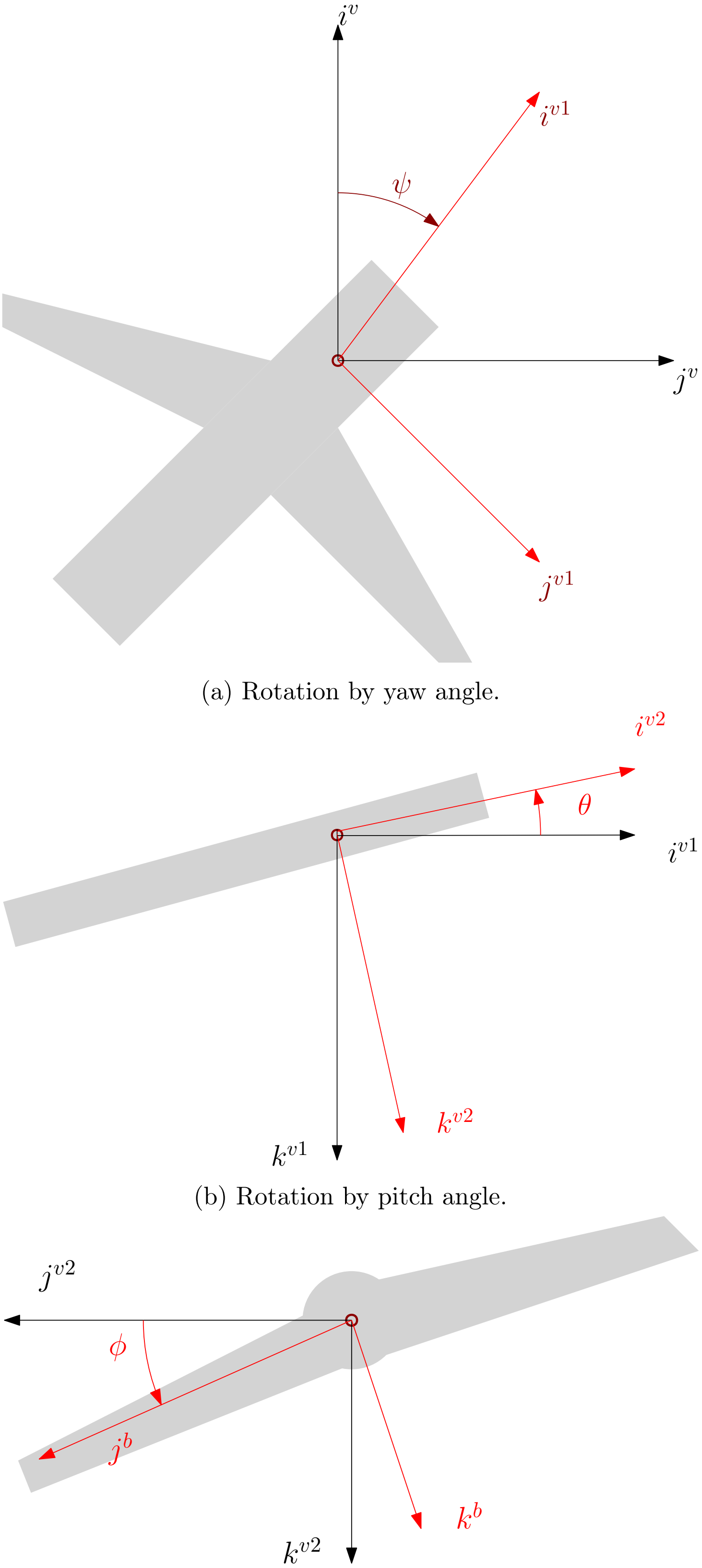
$$\mathbf{R}_b^i = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \tag{2}$$

where $c = cos()$, $s = sin()$, $\phi =$ roll angle, $\theta =$ pitch angle, $\psi =$ yaw angle.

3.2 Modeling

Throughout the project, equation (3) illustrates the states and inputs present in the system. They describe the dynamics of the aircraft over the course of the simulation, and do also serve as the system output in the control system. In addition to these states, there exist some aviation-related states which are derived from the mentioned states, such as airspeed and angle of attack. Although they represent relevant information about the aircraft dynamics, they are still a product of the 12 states given in equation (3). The derivation of these aviation-specific states are described later in the section.

$$x = \begin{bmatrix} p_n \\ p_e \\ p_d \\ u \\ v \\ w \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \text{North position in } \mathcal{F}^i \\ \text{East position in } \mathcal{F}^i \\ \text{Down position in } \mathcal{F}^i \\ \text{Velocity along i-axis in } \mathcal{F}^b \\ \text{Velocity along j-axis in } \mathcal{F}^b \\ \text{Velocity along k-axis in } \mathcal{F}^b \\ \text{Roll angle in } \mathcal{F}^{v2} \\ \text{Pitch angle in } \mathcal{F}^{v1} \\ \text{Yaw angle in } \mathcal{F}^{v} \\ \text{Roll rate in } \mathcal{F}^b \\ \text{Pitch rate in } \mathcal{F}^b \\ \text{Yaw rate in } \mathcal{F}^b \end{bmatrix}, \qquad u = \begin{bmatrix} \delta_e \\ \delta_a \\ \delta_t \end{bmatrix} = \begin{bmatrix} \text{Elevator deflection} \\ \text{Aileron deflection} \\ \text{Thrust command} \end{bmatrix} \tag{3}$$



(a) Rotation by yaw angle.

(b) Rotation by pitch angle.

(c) Rotation by roll angle.

Figure 3: Rotating from the inertial frame to the body frame.

The dynamics of the aircraft are dependent upon velocities and accelerations relative to the inertial frame $\mathcal{F}^i$ and the body frame $\mathcal{F}^b$. Because the aerodynamic forces and moments depend on the surrounding air, one has to take into account the relativity between the velocity of the aircraft and velocity of the surrounding air as well. This introduces the need for identifying some relative variables. [4] defines some relevant and some not relevant variables from this project's perspective, which is why only the relevant ones are defined in this thesis.

Airspeed is one of those variables, and it is the difference between ground speed and wind speed. Airspeed is the velocity with respect to the surrounding air.

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w$$

where $\mathbf{V}_g$ is ground speed, and $\mathbf{V}_w$ is wind speed, both velocities are with respect to the inertial frame. Using this relationship, one can define the relative velocities as:

$$\begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix} \tag{4}$$

The relative velocities defined in equation (4) serves as the body frame components of the airspeed. It is in many cases useful to express the magnitude of the airspeed, which is illustrated in equation (5).

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{5}$$

There are two more relative terms that are necessary in order to describe the equations of motion, and those are angle of attack (α) and side-slip angle (β). Angle of attack is defined as the angle between the $i^{\mathbf{b}}$-$j^{\mathbf{b}}$ plane and the airspeed vector. The side-slip angle is defined as the angle between the $i^{\mathbf{b}}$-$k^{\mathbf{b}}$ plane and the airspeed vector. Their mathematical representations are found in equation (6).

$$\alpha = tan^{-1}(\frac{w_r}{u_r}) \tag{6a}$$

$$\beta = sin^{-1}(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}) \tag{6b}$$

In order to utilize trim states and course control, which will be discussed in detail later in the thesis, crab angle and flight path angle need to be defined. Crab angle (χ_c) is defined as $\chi_c = \chi - \psi$, where χ or course angle is the angle between inertial north and the ground speed vector projected to the $i^{\mathbf{i}}$-$j^{\mathbf{i}}$-plane. The flight path angle (γ) is defined as $\gamma = \theta - \alpha$. For further information and illustrations related to these variables, see [4].

3.2.1 Dynamics

In this section the equations governing to motion of the aircraft are derived. A total of 12 differential equations, one for each state in equation (3) can be divided into two subcategories; kinematics and kinetics. The main difference between them is that the kinematics treat the geometrical aspect of motion, and kinetics describe how the motion is affected by external forces [10].

Kinematics

Using the zyx-convention, one can describe the inertial position dynamics:

$$\begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{p_d} \end{bmatrix} = \mathbf{R}_b^i \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

The angular rates are defined in $\mathcal{F}^b$, while the Euler angles are defined in the vehicle-1, vehicle-2 and body frame as discussed earlier. With the correct rotations, the relationship between Euler angle dynamics and angular rates is defined as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi) \cdot tan(\theta) & cos(\phi) \cdot tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Kinetics

Through Newtons laws of motion it is possible to derive a model for the aircraft dynamics. Because kinetics describe the effects of external forces on the aircraft, the forces are defined in the body frame. By relating the external forces $\mathbf{F} = [X, Y, Z]$ and the resulting moments $\mathbf{M} = [l, m, n]$ to a rigid-body, being the aircraft, the Newtons law of motion results in the following dynamics for the aircraft [13]:

$$m\dot{\mathbf{V}} + \omega \times m\mathbf{V} = \mathbf{F} \tag{7a}$$

$$\mathbf{I}\dot{\omega} + \omega \times \mathbf{IV} = \mathbf{M}, \tag{7b}$$

where $\mathbf{I}$ is the inertia matrix, m is mass, $\mathbf{V}$ are the linear velocities $[u, v, w]$ and ω are the angular rates $[p, q, r]$. The inertia matrix is defined as:

$$\begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix}$$

By calculating (7a) the linear velocity dynamics are described by:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{8}$$

By calculating (7b) the angular rate dynamics are described by:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{J_y} m \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix}, \tag{9}$$

where

$$
\begin{aligned}
\Gamma &= I_x I_z - I_{xz}^2 \\
\Gamma_1 &= \frac{I_{xz}(I_x - I_y + I_z)}{\Gamma} \\
\Gamma_2 &= \frac{I_z(I_z - I_y) + I_{xz}^2}{\Gamma} \\
\Gamma_3 &= \frac{I_z}{\Gamma} \\
\Gamma_4 &= \frac{I_{xz}}{\Gamma} \\
\Gamma_5 &= \frac{I_z - I_x}{I_y} \\
\Gamma_6 &= \frac{I_{xz}}{I_y} \\
\Gamma_7 &= \frac{(I_x - I_y)I_x + I_{xz}^2}{\Gamma} \\
\Gamma_8 &= \frac{I_x}{\Gamma}
\end{aligned}
$$

The 6-DOF, 12-state model that describes the kinematics and kinetics in a small UAV are as follows:

$$
\begin{bmatrix} \dot{p_n} \\ \dot{p_e} \\ \dot{p_d} \end{bmatrix} = \begin{bmatrix} c_\theta c\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{10a}
$$

$$
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{10b}
$$

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi) \cdot tan(\theta) & cos(\phi) \cdot tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{10c}
$$

$$
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6(p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{J_y} m \\ \Gamma_4 l + \Gamma_8 n \end{bmatrix} \tag{10d}
$$

In equation (10b) and (10d), external forces and moments affect the accelerations of the aircraft.

These external forces and moments are defined as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{f}_g + \mathbf{f}_a + \mathbf{f}_p \tag{11a}$$

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \mathbf{m}_a + \mathbf{m}_p \tag{11b}$$

The subscripts g, a, p stands for gravity, aerodynamic and propulsion, respectively. In deriving their contribution to the dynamics, the modeling method used in Beard & McLain [4] has been utilized.

The gravitational force is modeled as a force proportional to the mass acting at the center of mass. The gravitational force given in $\mathcal{F}^b$ is described by equation (12).

$$\mathbf{f}_g^b = \begin{bmatrix} -mg \cdot s\theta \\ mg \cdot c\theta \cdot s\phi \\ mg \cdot c\theta \cdot c\phi \end{bmatrix} \tag{12}$$

The aerodynamic forces and moments are usually decomposed into two categories; longitudinal and lateral. The longitudinal modes (DOF 1,3,5) or (*Forward/Back*, *Down/Up*, *Pitch*) are decoupled from the lateral modes (2,4,6) or (*Right/Left*, *Roll*, *Yaw*) by assuming:

- The fuselage is slender (length is much larger than width and height).
- The longitudinal velocity is much larger than the vertical and transversal velocities.

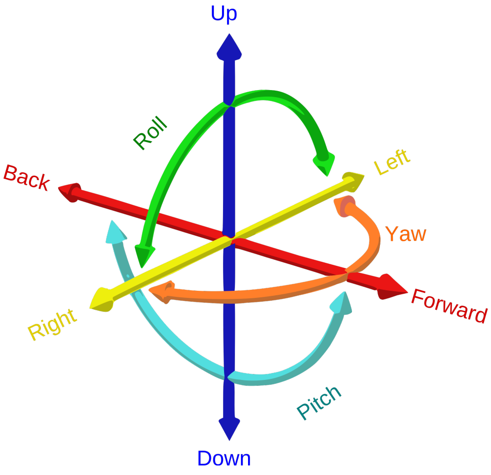


Figure 4: Illustration of six degrees of freedom [11].

Longitudinal Aerodynamics

The lift and drag forces are highly dependent on angle of attack. These functions with respect to α are nonlinear, but it is possible to express these functions as linear versions over a wide range of angles of attack. In most cases if the angle of attack does not cause wing stall or is around zero, it is fine to use the linear expressions below.

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha}\alpha \tag{13a}$$

$$C_D(\alpha) = C_{D_0} + C_{D_\alpha}\alpha \tag{13b}$$

Equation (13) is a linear approximation of the nonlinear dynamics, which is accurate for small values of α. The various flight maneuvers performed throughout the thesis should not result in large angles of attack, and this approximation is therefore considered sufficient. The coefficients in equation (13) must be identified, which will be discussed later in the section.

The longitudinal forces and moments from lift and drag is then described by:

$$F_{\text{lift}} = \frac{1}{2}\rho V_a^2 S(C_L(\alpha) + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e) \tag{14}$$

$$F_{\text{drag}} = \frac{1}{2}\rho V_a^2 S(C_D(\alpha) + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e) \tag{15}$$

$$m = \frac{1}{2}\rho V_a^2 Sc(C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e), \tag{16}$$

where c is the straight line between the airfoil leading- and trailing edge, S is the wing area, and b is the wingspan. In order to include lift and drag in simulations, it is necessary to convert them to the body frame $\mathcal{F}^b$, which is done in equation (17). This relationship substantiates the claim of their dependency of angle of attack. When α increasingly deviates from zero, the effect of lift increases, which coincides with the intuition of pitching up/down for creating lift. Equation (17) assumes a side-slip angle $\beta = 0$.

$$\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \cdot \begin{bmatrix} -F_{\text{drag}} \\ -F_{\text{lift}} \end{bmatrix} \tag{17}$$

Lateral Aerodynamics

The lateral aerodynamics are largely influenced by side-slip angle β. Following the same approach as with longitudinal aerodynamics, we end up with the forces and moments below:

$$Y = \frac{1}{2}\rho V_a^2 S(C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a) \tag{18}$$

$$l = \frac{1}{2}\rho V_a^2 Sb(C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a) \tag{19}$$

$$n = \frac{1}{2}\rho V_a^2 Sb(C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a) \tag{20}$$

By analyzing equations (13) through (20) it can be observed that the aerodynamic forces and moments are dependent on a set of coefficients (denoted as C), as well as the states, inputs and aircraft-specific parameters. It is this set of coefficients that has to be identified in order for the aircraft model to be as close to the physical aircraft as possible.

Lastly, the equations of motion are also affected by propulsion. The propulsion force and moment act only in the i-axis, as is illustrated in equations (21) and (22).

$$\mathbf{f}_p = \frac{1}{2}\rho S_{prop} C_{prop} \begin{bmatrix} (k_{motor}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

$$\mathbf{m}_p = \begin{bmatrix} -k_{T_p}(k_\Omega \delta_t)^2 \\ 0 \\ 0 \end{bmatrix}, \tag{22}$$

where S_{prop}, C_{prop}, k_{motor}, k_{T_p} and k_Ω are propeller characteristics.

3.3 Trim State

Trim is something widely used in the aviation industry, and has been used in this project as well. When the aircraft is in trim state, the control surfaces are in a state where no change is necessary in order to proceed with the flight determined by the trim settings. These settings are further explained in the following paragraph. All external forces and moments are compensated for. The system is in equilibrium at the state x^* and input u^* if $\dot{x} = f(x^*, u^*) = 0$. This is the condition for a trimmed flight. The method used for computing trim is obtained from [4], which this subsection is largely based on.

When calculating the trim states, the wind is treated as an unknown disturbance, and the trim states is therefore found assuming the wind speed is zero. This means that $V_a = V_g$, $\psi = \chi$, and $\gamma = \gamma_a$. These three parameters serves as inputs/settings to the trim calculations, along with initial conditions. There are some conditions that needs to be fulfilled in order for the aircraft to be in trim:

- Constant speed, V_a^*
- Constant flight path angle, γ^*
- Constant orbit of radius R^*

The states and inputs for the fixed-wing aircraft are the same as before, see equation (3), and $f(x,u)$ is represented by the right side of the equations of motion (10). These equations are independent of p_n, p_e and p_d. Therefore, trimmed flight is independent of position. Since only $\dot{p}_n$ and $\dot{p}_e$ are dependent on ψ, trimmed flight is also independent of ψ. In a constant-climb orbit, the speed of the aircraft is constant, which means that $\dot{u}^* = \dot{v}^* = \dot{w}^* = 0$. The roll and pitch angles are constant as well, so $\dot{\phi}^* = \dot{\theta}^* = \dot{p}^* = \dot{q}^* = 0$. The yaw rate is constant and given by:

$$\dot{\psi}^* = \frac{V_a^*}{R^*} cos(\gamma^*) \tag{23}$$

Equation (23) implies that $\dot{r}^* = 0$. The climb rate is also constant and given by:

$$\dot{h}^* = V_a^* sin(\gamma^*) \tag{24}$$

It is now possible to specify $\dot{x}^*$ as a function of V_a^*, γ^* and R^*, see equation (25).

$$\dot{x}^* = \begin{bmatrix} \dot{p_n}^* \\ \dot{p_e}^* \\ \dot{p_d}^* \\ \dot{u}^* \\ \dot{v}^* \\ \dot{w}^* \\ \dot{\phi}^* \\ \dot{\theta}^* \\ \dot{\psi}^* \\ \dot{p}^* \\ \dot{q}^* \\ \dot{r}^* \end{bmatrix} = \begin{bmatrix} \times \\ \times \\ -V_a^* sin(\gamma^*) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{V_a^*}{R^*} cos(\gamma^*) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{25}$$

where × is used to represent the "don't care" conditions.

Finding this state vector x^* means solving the system of equations where the equations of motion (10) equals to (25). One method of doing so is to formulate an optimization problem where the difference between the system dynamics and the desired dynamics from equation (25) serves as the objective to be minimized. In this thesis, optimization problems are solved using the casadi tool [2]. Because the objective function has to be scalar, sum of squares error has been used in order to minimize the difference between two vectors. As $\dot{p_n}^*$ and $\dot{p_e}^*$ are not relevant in regards to trim, they are excluded from the sum of squares error computation. The algorithm which performs the trim computation is described in more detail in appendix B.5, and the objective function used in the optimization problem is illustrated in equation (26). As one can see from the sum of squares error computation, the first and second states (p_n and p_e) are excluded from the trim computation, as trim is independent of position.

$$\begin{aligned} \min_{x} \quad & (\dot{x}_i^* - \dot{x}_i(x))^2 \\ \text{where} \quad & i = \mathbb{N}\{3, 4, ..., 12\} \\ & \mathbf{x} = [p_n, p_e, p_d, u, v, w, \phi, \theta, \psi, p, q, r, \delta_e, \delta_a, \delta_t]^T \end{aligned} \tag{26}$$

4 System Identification

System identification (SYSID) is the process of developing a useful mathematical model for a given system from measurements of system inputs and outputs [17]. In this method, there is assumed a separation of longitudinal- and lateral-directional dynamics. This is physically feasible and will potentially aid the convergence of the algorithm. There may be some significant dynamics that are not captured by the decoupled model, which perhaps should be studied further in a later iteration of this work. The overall idea is to treat the aerodynamic coefficients as unknown parameters, and identify them based on comparing a set of sensor data of the dynamics during a flight to a preliminary model of the aircraft. Minimizing the error between the sensor data and the computed states from the model, given a specific input sequence, should yield an estimate of the unknown parameters.

The system identification method is inspired by and based on [17] where the idea is to minimize the difference between the sensor data and the output of a preliminary model of the aircraft. The aerodynamic coefficients serve as the model parameters, and the preliminary model outputs the dynamics with symbolic coefficients. The symbolic output is then compared to the sensor data, so that the aerodynamic coefficients can be determined. Note that the same input sequences are used to obtain the symbolic model output and the sensor data. There are a few things one has to consider when implementing this. For instance, the method may be sensitive to the number of unknowns to approximate, and the dynamics that the sensor data is based on has great significance in how easily the method is able to identify the coefficients. This means that attempting to identify all of the coefficients at once, while using an arbitrary set of sensor data is a next to impossible task. In this thesis, the identification has been conducted in two separate tests, one for the lateral parameters, and one for the longitudinal parameters. As mentioned, this method outputs approximations of these coefficients, meaning that they do not need to be fully accurate. However it is desirable to achieve some sort of coherence between the output of this method and the already identified coefficients in [13]. The identified coefficients in [13] are used to simulate the sensor data, and also functions as a measure to describe the algorithm's ability to converge to the "true" parameters. As mentioned in the literature review, these already identified parameters may not be accurate themselves. However for the purpose of this algorithm, a perfectly identified parameter set is not necessary.

The parameter estimation method used in this thesis is called the "Output-error method" [14]. This method computes a sum of squares error between the sensor data set and the output of the preliminary model. This results in a nonlinear optimization problem (NLP), where casadi [2] has been used to perform the optimization.

$$\begin{aligned} \min_{e} \quad & \frac{1}{2} e^T e \\ \text{where} \quad & e = y_{sensor} - y_{symbolic} \end{aligned} \tag{27}$$

The decision variables in this NLP are the aerodynamic coefficients.

The lateral states, inputs and coefficients are given in table 1 and the longitudinal states, inputs and coefficients are given in table 2.

Table 1: Lateral coefficients and variables.

States	v	p	r						
Inputs	δ_a								
Coefficients	C_{Y_p}	C_{Y_r}	C_{Y_0}	C_{Y_β}	$C_{Y_{\delta_a}}$	C_{l_0}	C_{l_β}	C_{l_p}	C_{l_r}
	$C_{l_{\delta_a}}$	C_{n_0}	C_{n_β}	C_{n_p}	C_{n_r}	$C_{n_{\delta_a}}$			

Table 2: Longitudinal coefficients and variables.

States	u	w	q					
Inputs	δ_e	δ_t						
Coefficients	C_{L0}	$C_{L\alpha}$	$C_{L_{\delta_e}}$	C_{L_q}	C_{D0}	$C_{D\alpha 1}$	$C_{D_{\delta_e}}$	C_{D_q}
	C_{m_α}	$C_{m_{\delta_e}}$	C_{m_q}	C_{m0}				

4.1 Sensor data and excitation maneuvers

To minimize the error between the model output and the sensor data, it is necessary to generate this data set based on a series of flight maneuvers. As this thesis is entirely based on simulations, so are the sensor data. Therefore, all relevant measurements/states for the identification algorithm were fed back by a differential equations solver. Because only dynamic states have been used in the simulations, the remaining states needed to simulate the system were assumed constant using the trim states. The trim settings which were used, was an airspeed of $V_a = 18$ m/s, a flight path angle $\gamma = 0$ degrees, and an orbit of radius $R = \infty$ (straight flight).

For the flight maneuvers that was conducted in the identification process, it was necessary to excite the aircraft in a way that capture the dynamics we wanted to mathematically describe. Two excitation inputs were selected, one using the aileron for lateral excitation, and one using the elevator for longitudinal excitation. Even though modal analysis is a methodical way of determining the frequency of input signals, it did not yield especially good results in this project. Therefore the signal frequency was set to 5 Hz which worked well for both lateral and longitudinal excitation. Figure 5 illustrates these input signals, where a frequency of 5 Hz translates to the spacing between the initial deflection and the last deflection. This means that a signal frequency of 5 Hz results in a total excitation time of $T = \frac{1}{f} = \frac{1}{5} = 0.2$ seconds.

There are many ways to go about designing these input signals, for instance doublets and 1-2-1 maneuvers [17]. What has been used is part inspired by [17] and part trial and error. The aileron and elevator signals were both doublets, where the doublet frequency is the signal frequency explained above. There has been used a deflection of ±10° for both ailerons and elevators. As one can see from the input signals, the maneuvers were initialized with a 2 second trimmed flight at the beginning. This is in order to stabilize the system before doing a controlled excitation. After the excitation, the actuator settings return to the trim settings.

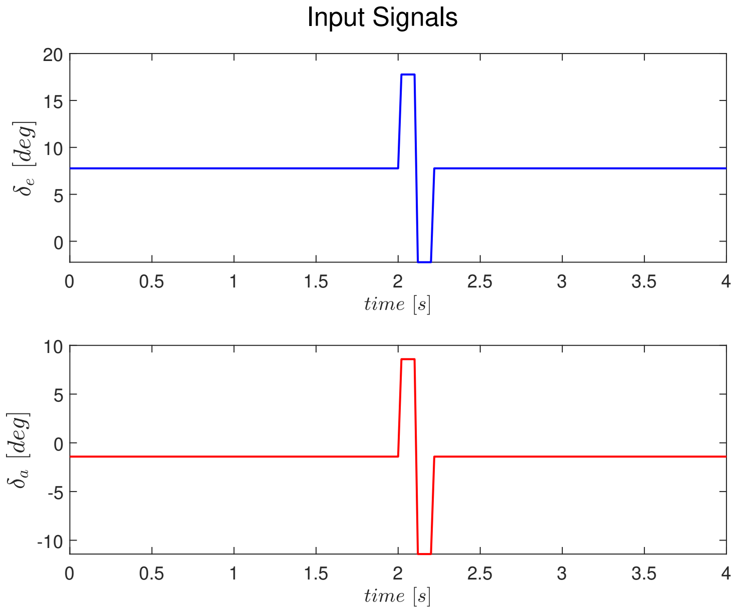


Figure 5: Aileron and elevator deflections used in SYSID.

4.2 Results

The results are highly dependent on the quality of the sensor data, how the optimization problem is initialized and solved, and the type of solver used for integrating the differential equations describing the dynamics. To initialize the optimization solver, there has to be an initial guess of the unknown parameters. In this study the initial guess was set to the true coefficient values plus a random number of magnitude relative to the magnitude of each coefficient. For instance the parameter C_{Y_r} has a magnitude of 10^{-2}, and the random number as a result is of magnitude 10^{-2}.

Both the generation of sensor data and the simulation of a the preliminary model was done by a Runge Kutta 4 integrator with a time-step of 0.02 seconds. The simulations and system identification method were both implemented in *matlab*. In the simulations, the flight trajectory was straight and level with constant speed until the excitation maneuvers occured. After the excitations, sensor data was collected for two seconds. This lead to a full simulation time of 4 seconds.

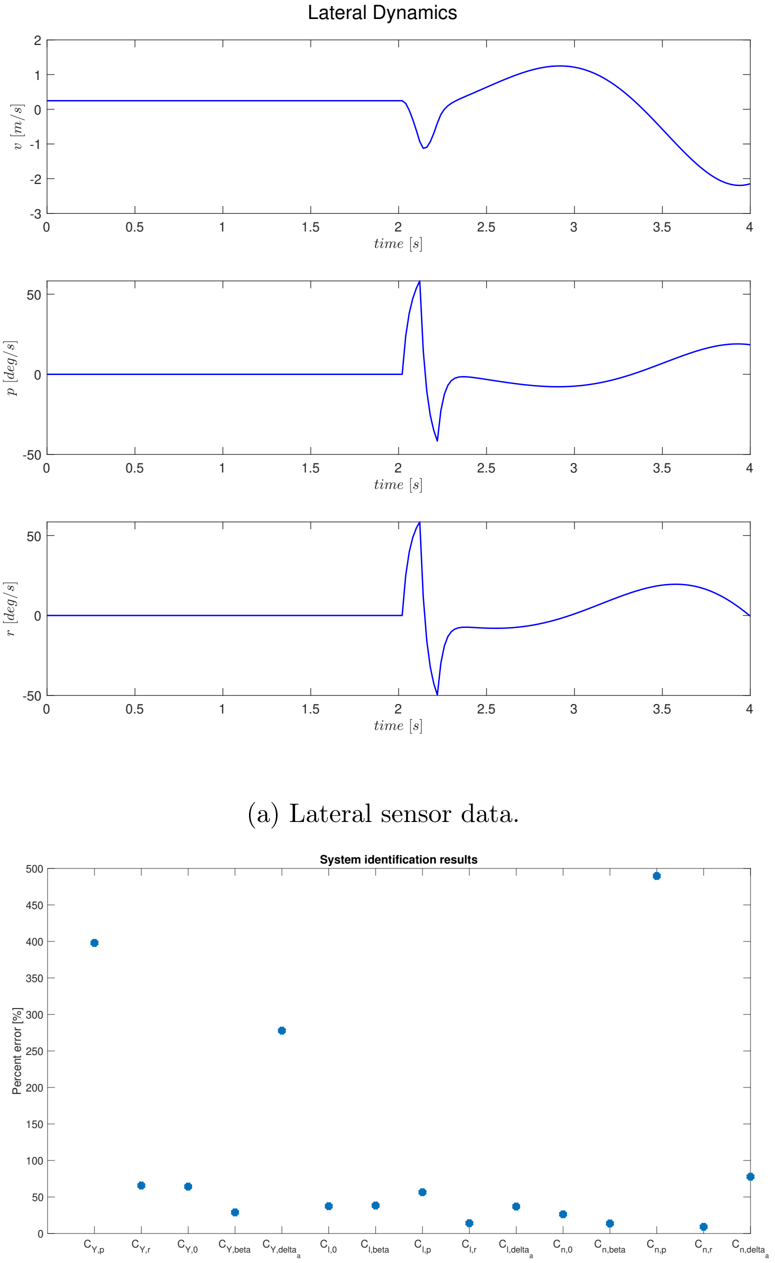
In order to determine the performance of the system identification algorithm, it is necessary to introduce an error measure. The error measure that has been used, is percent error. That is the relative error between the true and estimated parameters in terms of percentage. The formula for computing the percent error is [3]:

$$error = 100\% \cdot |\frac{v - v_{approx}}{v}|,$$

where v is the true value, and v_{approx} is the approximated value.

Lateral parameter identification

In figure 6a and 6b, the sensor data and the resulting parameter estimates are illustrated. The 15 coefficients have been identified with a varying degree of success, where most parameter estimates are within a percentage error of 100%. It seems as if the largest error occurs in the coefficients describing the impact of roll rate (p) and aileron (δ_a), indicating that those estimates are not accurate. Other than that, the identified parameters are relatively close to the true parameters.



(a) Lateral sensor data.

(b) Lateral system identification results.

Figure 6: Lateral SYSID.

In order to gauge how well the identified lateral model coincides with the sensor data, the identified parameters were applied to the equations of motion, and figure 7 describes the results. There are clear similarities between the lateral sensor data in 6a and the identified lateral model output in 7. Despite the large errors in some parameters, it seems as if the system dynamics are approximated to a satisfactory degree. However, this is an isolated test where the sensor data represents the states perfectly and only a small part of the state-space was simulated for a few seconds.

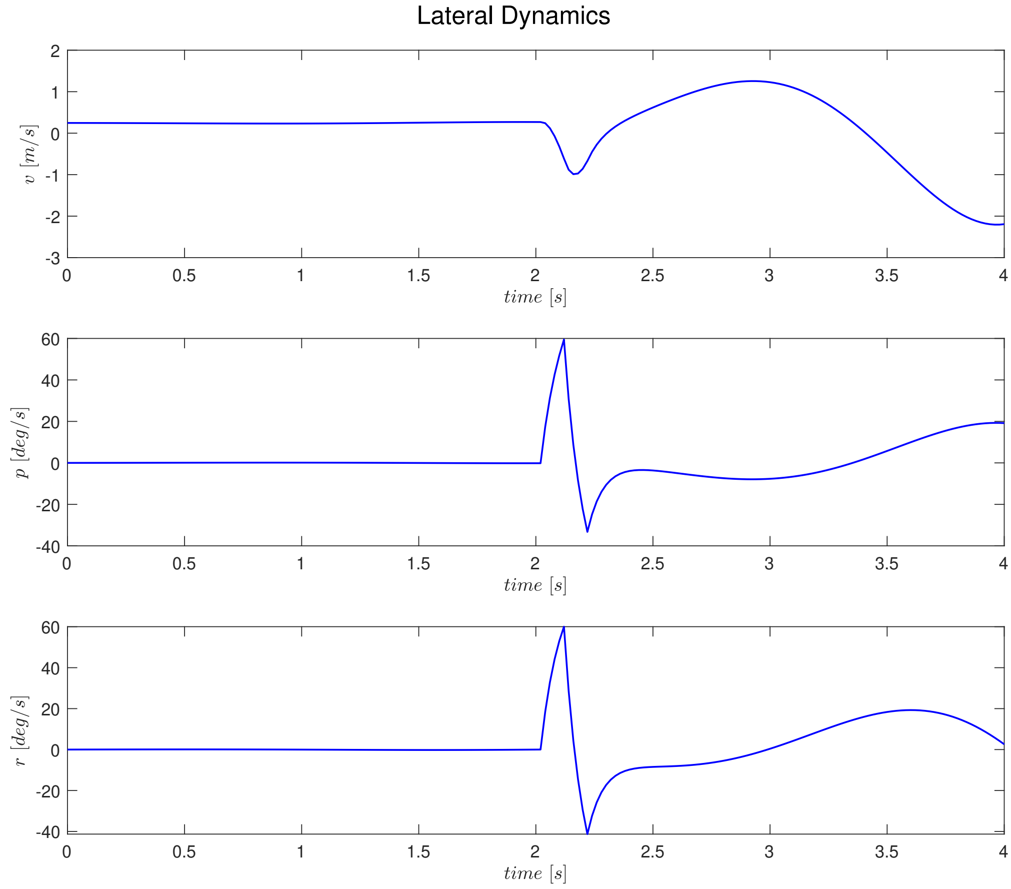
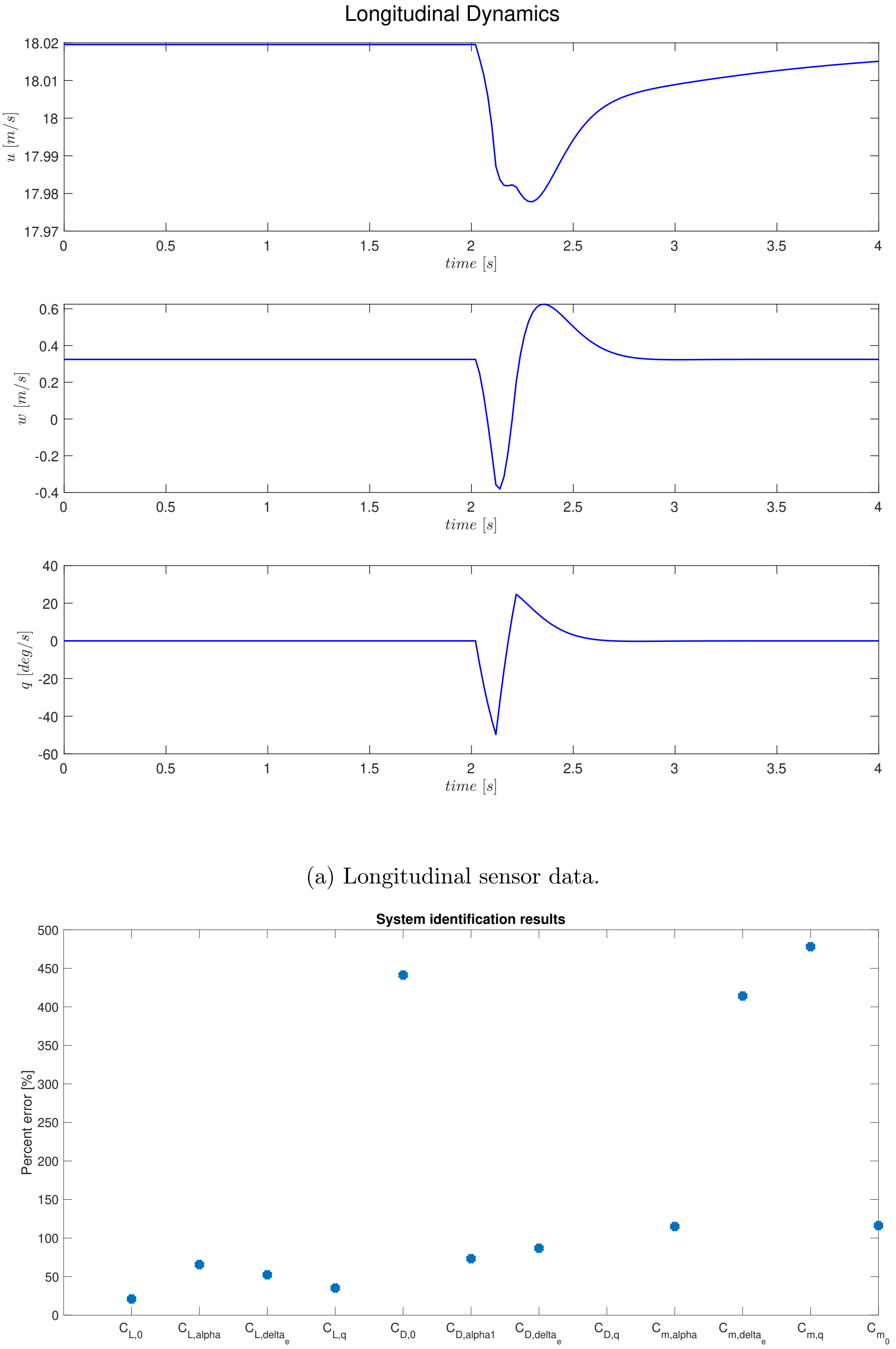


Figure 7: Identified lateral model output.

Longitudinal parameter identification



(a) Longitudinal sensor data.

(b) Longitudinal system identification results.

Figure 8: Longitudinal SYSID.

In figure 8a and 8b, the longitudinal sensor data and identification results are illustrated. The first thing to notice is that the percentage errors are in general larger in the longitudinal identification process. It should be mentioned that there are 12 coefficients in the longitudinal dynamics, and one can therefore assume better convergence to the true parameters, given that the optimization problem contains fewer decision variables. This seems to not be the case, but that is not necessarily due to the number of coefficients. From figure 8b it is indicated that parameters describing various effects on the pitch moment is hard to estimate accurately. Both the effects on lift and drag were identified with percentage errors below 100%, with the exception of C_{D_0}. In the attempt of comparing the output of the identified longitudinal model to the sensor data, the integration solver was unable to simulate the system. This indicates that the estimates of the longitudinal parameters are not good enough to use in the model structure.

4.3 Discussion

Based on the results, it seems that the system identification algorithm works to some degree. The current implementation is the first iteration of such an algorithm, and there are for sure steps one can take to improve the performance. The largest influence on the performance is most likely the initial guess of the coefficients. As mentioned, in these tests the initial guesses are based on the true values plus a random value of a corresponding magnitude. These variations in initial guesses are assumed to have an effect on the ability to converge to an accurate solution, resulting in different percentage errors.

Another factor which may influence the performance of the algorithm, is the sensor data. As the identification algorithm minimizes the error between a data set and the output of a model with respect to the aerodynamic parameters, the aircraft should be excited enough such that the measurements carry some information about the aerodynamic behavior. On the other end, large excitations or chaotic dynamics may not carry much information about the aerodynamics as well. The information that is contained in a sensor data set will most likely influence how accurately the algorithm is able to estimate the parameters. Along with the type of dynamics captured by the sensors, the simulation time is also a factor in how well the minimization process develops. One could assume that a lengthy simulation with several different maneuvers captures more information regarding aerodynamics, and vice versa. There is a trade-off between simulation time and optimization time. The more lengthy the simulation, the more time consuming and complex the optimization routine becomes.

Continuing the reasoning in the above paragraph, the integrator time-step should affect the identification results as well. The tests conducted were done with a time-step of 0.02 seconds. By shortening the integrator time-step, perhaps more nuances in the dynamics would be easier to capture. Changing the time-steps may also affect the convergence time in the optimization solver to some degree.

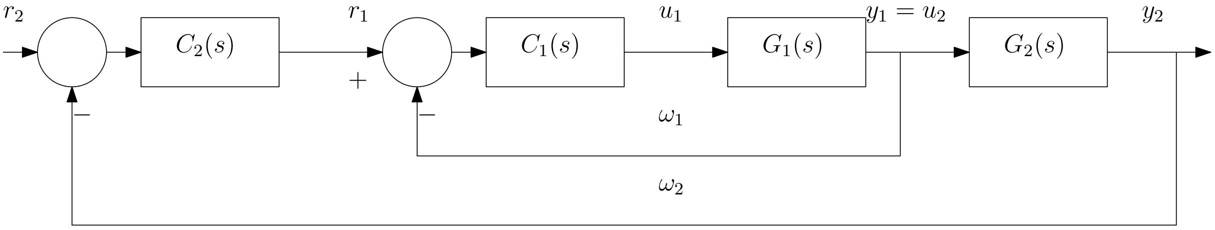
The system identification results are dependent on the initialization, sensor data, and integrator type used for simulation. One can argue that these factors affect each other. Given that the sensor data would carry a broad representation of the aerodynamics, the optimization would probably be less sensitive to the initial guess. Expanding on this idea, if the objective function had been convex, there would only be one optimal solution, regardless of initial guess. All things considered, the identification process yielded decent estimates of the aerodynamic coefficients, where the lateral parameters were most accurate.

5 Control Architecture

The control system includes a total of three control loops which has the objective to steer the Unmanned Aerial Vehicle (UAV) towards a predefined path by passing references on airspeed, altitude, course to a low-level autopilot. The control loops are based on successive loop-closure design, where the idea is to use nested loops with decreasing bandwidth from the innermost level and outwards. Another way to view the idea is to close several simple feedback loops in succession around the open-loop plant dynamics rather than designing a single (presumably more complicated) control system. The key step in closing the loops successively is choosing the bandwidth of the next loop. The inner loop should always be the fastest, and the next loop should be a factor of 5-10 slower than the preceding loop [4]. In figure 9, there are two loops in cascade. The inner loop operates at a frequency of ω_1, and the next loop operates at ω_2. In order to separate the frequencies we introduce a bandwidth separation variable W, and the bandwidth relationship between the inner and outer loop is determined by:

$$\omega_2 = \frac{1}{W}\omega_1$$

where $W = 5 - 10$. Using this separation, it can be assumed that for frequencies well below ω_1, the inner loop can be modeled as a gain of 1. This simplifies the design of the outer loop, and makes sure that the innermost loop does not affect the outer loop.



$$\omega_1 >> \omega_2$$

Figure 9: Illustration of successive loop closure.

5.1 Reference Models

The reference signals could take the shape of steps, where for instance the airspeed reference goes from 18 to 20 m/s. That jump can lead to a large controller input, which results in an abrupt change in dynamics. The potentially large controller input might destabilize or make the system harder to control and/or be physically not possible to achieve. This calls for reference models, which smoothes out these signals such that a feasible trajectory is fed into the controller. These models takes the shape of filters, where two different filters have been used in this project; a second-order low-pass filter for velocity references and a first-order low-pass filter cascaded with a mass-spring-damper system for position/attitude references. The models have been implemented as second- and third-order state-spaces, as illustrated in equation (28).

$$\dot{\mathbf{x}}_d = \mathbf{A}_d\mathbf{x}_d + \mathbf{B}_d r \tag{28}$$

For velocity reference models (2nd order LP filter) the states are $\mathbf{x}_d = [v_d, \dot{v}_d]^T$, and the state-space matrices are given by:

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}, \qquad \mathbf{B}_d = \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}$$

For position and attitude reference models (1st order LP filter and mass-spring-damper) the states are $\mathbf{x}_d = [y_d, \dot{y}_d, \ddot{y}_d]^T$, and the state-space matrices are given by:

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\omega_n^3 & -(2\zeta+1)\omega_n^2 & -(2\zeta+1)\omega_n \end{bmatrix}, \qquad \mathbf{B}_d = \begin{bmatrix} 0 \\ 0 \\ \omega_n^3 \end{bmatrix}$$

Speed and acceleration limitations, in addition to actuator limitations is important to take into consideration when tuning these different models. One has to make a compromise between accurate tracking and performance when setting the model bandwidth (ω_n). The bandwidth of the reference model must be lower than the bandwidth of the motion control system in order to obtain good tracking and stability [10]. On the other hand, a larger bandwidth results in better tracking. This is where the compromise is. A low bandwidth of $\omega_n = 1.44$ Hz, and critical damping $\zeta = 1$ was used on all reference models.

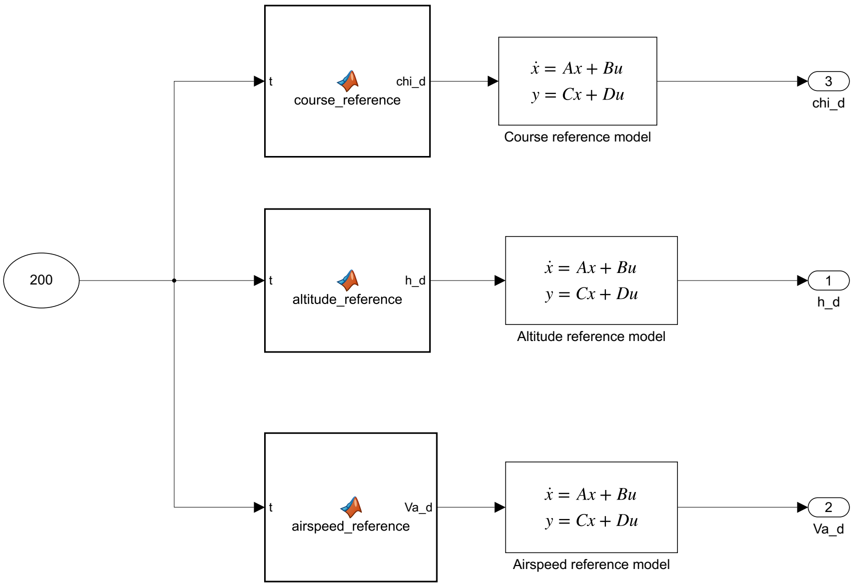


Figure 10: Reference models in *Simulink*.

5.2 Controllers

The control systems in this thesis are a product of iterative design, where subsystems have come together throughout the project. There has been created three separate control systems; Course control, altitude control and airspeed control. These systems were designed in accordance with [4]. Altitude control and airspeed control both occurs in the longitudinal direction, and course control deals with the lateral dynamics. Therefore, the next iteration entailed creating two autopilots, one longitudinal and one lateral. Decoupling the longitudinal and lateral dynamics is reasonable when designing control systems for aircraft, as explained earlier in the thesis. When these two autopilots performed satisfactory in their separate domains, the next and final iteration was to connect them in order to simulate the full state-space with both autopilots running simultaneously. Inside the cascaded control loops are variations of PID control.

In addition to computing the input signals, it is advantageous to include a saturation element and actuator dynamics. This ensures that the different control systems act in a manner that is most similar to the physical systems. For instance, the aileron deflection is limited to $\pm 35°$ and the thrust is limited to no throttle to full throttle. In reality, the actuators does not react instantaneously, as there are both inertia to overcome and signal delays. The saturation issue has been dealt with by implementing a saturation block in *Simulink*. The input signal enters this block, and if the signal is larger than the maximum deflection, or smaller than the minimum deflection, the signal is limited to the max/min value. To model the impact of inertia in the actuators, the input signals were filtered through a lowpass filter. The filter is described in the equation below, where $K = 1$ and $\tau = 5$ has been used.

$$H_{act} = \frac{K}{\tau s + 1} \tag{29}$$

The various controllers were also fitted with anti-windup schemes to deal with eventual integrator windup. There are several ways of implementing anti-windup for different systems, but the method used in this project is based on subtracting an expression including the input signals from the error calculation. The idea is to subtract from the integrator exactly the amount needed to keep u at the saturation bound [4]. The mathematical representation and what was implemented in the various control systems is illustrated in equation (30).

$$I = \frac{1}{K_i}(u - u_{unsat}), \tag{30}$$

where u_{unsat} is the input signal before it is treated by the saturation block and the lowpass filter. u is the processed signal which is forwarded to the actuators. This I-value is fed back to the error computation and subtracted before the error is integrated.

5.2.1 Lateral Controller

The lateral controller deals with the rolling motion of the aircraft. A full diagram showing how the controller was implemented is shown in appendix A.1. The innermost loop consists of control of the roll angle (ϕ), and the outermost loop controls the course angle (χ). The output of the roll loop is passed on as input to the aileron control surface, and is given by equation (31).

$$\delta_a = K_{p_\phi}(\phi^d - \phi) - K_{d_\phi} p \tag{31}$$

Outside this loop, the course angle is controlled by outputting a desired roll angle (ϕ^d). The control law is given by equation (32).

$$\phi^d = K_{p_\chi}(\chi^d - \chi) - \frac{K_{i_\chi}}{s}(\chi^d - \chi) \tag{32}$$

The desired course angle is either determined by a reference signal or a guidance signal. This signal is then filtered through the related reference model for a smooth and physically feasible course trajectory. One can argue that the inner roll loop could make use of an integral term as well to compensate for eventual disturbances or unmodeled dynamics regarding the rolling motion. However, those two integrators can interfere with each other as an error might be integrated twice. In addition, the extra integrator would decrease the bandwidth of the inner loop, rendering the lateral control system in general slower. This could lead to an unstable system or at least worsen the performance of the controller. Disturbance rejection is therefore excluded from the roll loop, and is assumed dealt with by the outer integral term.

5.2.2 Longitudinal Controller

The longitudinal controller deals with the pitching motion and airspeed of the aircraft. The implementation of these two control systems is illustrated in appendix A.2. The pitching motion is controlled with successive loop-closure as the lateral control system, where the innermost loop controls the pitch angle (θ), and the outermost loop controls the altitude (h). The airspeed controller consists of a single PI-controller. The pitch loop outputs an input signal to the elevator control surface, and the control law is given by (33):

$$\delta_e = K_{p_\theta}(\theta^d - \theta) - K_{d_\theta} q \tag{33}$$

The desired pitch angle comes from the outermost control loop, which controls the altitude. The altitude control law is given by (34):

$$\theta^d = K_{p_h}(h^d - h) + \frac{K_{i_h}}{s}(h^d - h) \tag{34}$$

The desired altitude is given by a reference signal, which is fed to a reference model.

The airspeed can be controlled using both the pitch angle and thrust. For this particular application, airspeed control using thrust was the most natural choice. The control law which outputs the thrust input is given by (35):

$$\delta_t = K_{p_V}(V_a^d - V_a) + \frac{K_{i_V}}{s}(V_a^d - V_a) \tag{35}$$

For a full overview of the computation of the controller gains, see appendix B.6.

5.3 Gain Scheduling

If an aircraft is flying in cruise altitude and a maintained airspeed, a linear controller would probably suffice. The linear controller is only based on a particular operating condition, but aircraft do not fly in a single operating condition. It is therefore necessary to implement either a nonlinear controller which controls all relevant operating conditions, or use gain scheduling. Gain scheduling is a practical and powerful method for the control of nonlinear systems. A gain-scheduled controller is formed by interpolating between a set of linear controllers derived for a corresponding set of plant linearizations associated with several operating points [5]. This is partially the case for this system. The controller gains are determined by airspeed and the tuning of frequency and damping for different motions. Therefore the controllers are based on different operation points nonetheless. Gain scheduling methods are generally implemented in a way that is adapted to the problem at hand, and theoretical justification is hard to come by. In addition, one can only guarantee local stability and performance, as a global stability guarantee would require extensive testing.

The gain scheduling is performed on an exogenous scheduling parameter that characterizes the operating environment. For aircraft there are several parameters that accurately describe an operating environment. The mach number changes, dynamic pressure, center of gravity due to different payload placements, and angle of attack all characterize different operating environments. The number of scheduling parameters impacts the complexity of the interpolation. It is desirable to divide up the operating envelope enough to achieve a good performance everywhere, meaning more scheduling variables. The number of variables determines the dimension of the gain matrix, and many dimensions require many different tunings, which can become a large task.

In this project there has been implemented one scheduling parameter, which is airspeed. The airspeed is the most influential factor when determining the controller gains, and is therefore a natural choice when implementing a gain scheduled controller. The operating conditions have been split into six different operating points, i.e six different airspeeds:

$$\mathbf{V_a} = \begin{bmatrix} 15 & 17 & 19 & 21 & 23 & 25 \end{bmatrix}$$

For each airspeed interval, the airspeed reference is constant. Meaning that when the system is tuned for $V_a = 15$, the airspeed controller uses a setpoint of 15. Also, the lateral and longitudinal control systems were tuned separately. In practice this means that when tuning the lateral control system, the altitude was constant, and when tuning the longitudinal system, the course was constant. When it comes to the computation of the controller gains, the airspeed was set to the respective operating conditions. In addition, there was a tuning process of the system's frequency (ω_n) and damping behavior (ζ). Performing this tuning process in all of the operating conditions allowed for the gain schedule to be identified. The various controller gains for the various operating points can be found in table 3.

Table 3: Controller gains in relation to airspeed.

Controller gains	V_a **[m/s]**					
	15	**17**	**19**	**21**	**23**	**25**
K_{p_ϕ}	1.7	1.8	1.8	2.3	2.3	3.5
K_{d_ϕ}	0.019	0.039	0.035	0.048	0.046	0.076
K_{p_χ}	7.1	6.8	8.5	12.0	14.4	20.9
K_{i_χ}	10.2	13.5	18.8	33.8	44.4	85.6
K_{p_θ}	-2.3	-2.3	-3.5	-3.5	-3.5	-7
K_{d_θ}	-0.30	-0.26	-0.28	-0.25	-0.23	-0.30
K_{p_h}	0.20	0.20	0.22	0.22	0.22	0.20
K_{i_h}	0.25	0.28	0.41	0.45	0.49	0.95
K_{p_V}	1.65	1.67	2.00	2.01	2.02	3.33
K_{i_V}	1.43	1.65	2.62	2.91	3.20	6.52

These controller gains are the result of a tuning process which is described in [4], and continued to this thesis in the implementation presented in appendix B.6. In short, by tuning a set of tuning parameters, the gains were computed using these parameters and some aspects of the identified model. For the full overview of all tuning parameters across operating points, see table 4.

Table 4: Tuning parameters in relation to airspeed.

	V_a					
Tuning parameters	15	17	19	21	23	25
ζ_ϕ	0.9	0.9	0.9	0.9	0.9	1
e_ϕ^{max} [deg]	30	20	20	15	15	10
ζ_χ	0.9	0.707	0.707	0.707	0.707	0.707
W_χ	7	9	9	9	9	9
ζ_θ	1	1	1	1	1	1
e_θ^{max} [deg]	15	15	10	10	10	5
W_h	10	10	10	10	10	10
ζ_h	0.707	0.707	0.707	0.707	0.707	0.5
ζ_v	10	10	10	10	10	12

5.3.1 Discussion

Table 3 and 4 represent isolated tuning sessions, adapted to the various operating points. It is interesting to notice that while the tuning parameters were tuned, and the resulting controller gains were not paid attention to, there are patterns as airspeed increases/decreases. For instance, the magnitude of K_{p_θ} increases as the airspeed increases. In the lateral controller gains, the same trends seems to be the case. As airspeed increases, the gains increase in magnitude as well. This is clearly illustrated in the value of K_{p_χ} which has a value of 7.1 when $V_a = 15$ m/s, and 20.9 when $V_a = 25$ m/s.

The tuning parameters, displayed in table 4, shows that there is also a pattern as the airspeed increases/decreases. There is not as much differences in magnitude as with the controller gains, but as airspeed influences the computation of the gains directly, maybe that is not necessary. An interesting observation is that the roll damping ζ_ϕ goes from being slightly underdamped to critically damped as airspeed reaches large values. The opposite happens in the longitudinal system, where ζ_h goes from being underdamped to more underdamped as airspeed increases. Another interesting observation is that the separation variable between the outer and inner lateral control systems increases as airspeed increases, in order to continue accurate lateral control. This might indicate that the cascaded loops need more bandwidth separation as airspeed increases. In the longitudinal control system, that is not the case.

6 Guidance System

Guidance can be defined as the process of guiding the position of an object towards a given waypoint, which in general may be moving [16]. Guidance systems can be divided into two subcategories, trajectory tracking and path following. The main difference between them is their dependence of time. Trajectory tracking forces the UAV to a position on a path at a specific time, while path following forces the UAV to a position on a path regardless of time. It is the latter that has been implemented in this project. The position on a path is generated by a minimization scheme where the objective is to minimize the distance between the UAV and the path. In general the guidance system receives some waypoints in the horizontal plane which is then translated into commanded course angles to the lateral control system based on the ever-changing position of the aircraft. The waypoints are usually generated with a human interface. For this particular application, the human inputs desired positions in a local tangent plane which corresponds to north and east directions (p_n and p_e) in the inertial frame $\mathcal{F}^i$.

The guidance system which has been implemented into the control system is a Line-of-sight (LOS) guidance law combined with a path generator based on circles of acceptance. These two subcomponents will be discussed in more detail separately before they are tied together. LOS guidance is a scheme where the aircraft position (x^n, y^n), a stationary reference point (x_1^n, y_1^n), and a target point (x_2^n, y_2^n) forms a triangular relationship. See figure 11 for illustration.

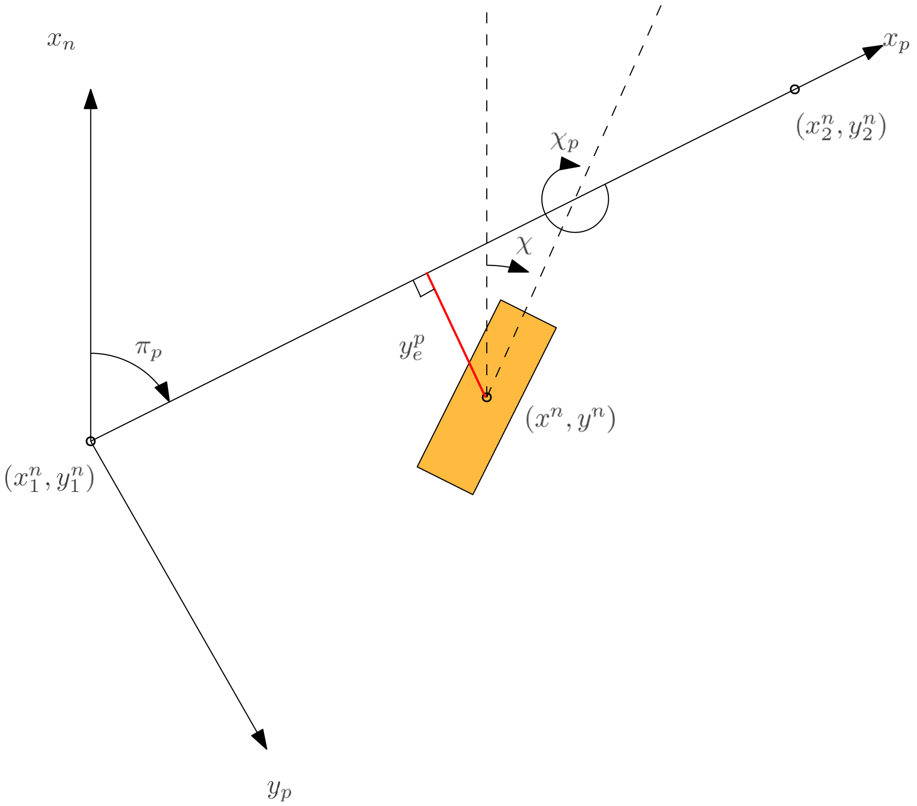


Figure 11: Illustration of LOS.

The idea is to minimize the cross-track error y_e^p, the red line in the figure. The computation of this error is done using the MSS toolbox [9]. In order to minimize this error, there has been implemented a proportional LOS guidance law, which takes the shape of equation (36), where the proportional gain K_p is a tunable parameter to achieve decent convergence. The output (χ_d) was then passed on to the lateral control system as a course reference to steer the aircraft to that particular course angle.

$$\chi_d = \pi_p - tan^{-1}(K_p y_e^p) \tag{36}$$

For the guidance system to function autonomously, the algorithm has to determine when to switch to the next waypoint, and that is where the path generator comes in. As mentioned, this system uses circles of acceptance in order to assess if the aircraft is close enough to the target point. If that is the case, then the original target point becomes the new reference point, and the next waypoint becomes the next target point. The straight lines between the reference points and target points serves as the path to be followed. This "close enough" assessment is determined by a circle of acceptance. One sets a specific radius around the target waypoint, and when the aircraft enters that circle it is considered arrived at the waypoint, and the algorithm then switches to the next point. And so it goes until the last waypoint has been flown through. This radius is necessary for an accurate trajectory because the aircraft's maneuverability is limited. For example, the aircraft is not able to change course from 5° to 10° instantly. Had the objective been to reach the target point exactly, the aircraft would have flown right by it, making it harder to reach the next point effectively. Equation (37) concretizes this switching idea and is obtained from [10]:

$$(x_{i+1}^n - x^n)^2 + (y_{i+1}^n - y^n)^2 \leq R_{i+1}^2 \tag{37}$$

When this inequality is true, the next waypoint should be selected. In this thesis, the radius has been set to $R_{i+1} = R = 200$m.

Figure 12 describes the complete guidance system, which is a subsystem of the autopilot. The inputs to the guidance system are the north and east positions in $\mathcal{F}^i$, together with a waypoint matrix. The position would come from an Inertial Navigation System (INS) or a Kalman filter for instance, but in this case it was fed back from the simulation. The waypoints are as mentioned a product of a human interface where a person inputs positions in the horizontal plane the aircraft is to fly through. The green block is where it is decided when it is time to switch to the next waypoint. This is where the circle of acceptance (37) was implemented. The variable "wp_{pos}" carries information about which waypoint the system is following throughout the simulation. The block outputs reference and target points, which the guidance block (orange) uses together with north and east positions to form a situational understanding such as illustrated in figure 11. Then the guidance block can minimize the cross-track error using the proportional LOS guidance law. The minimization process outputs a desired course angle which is then forwarded to the lateral control system.

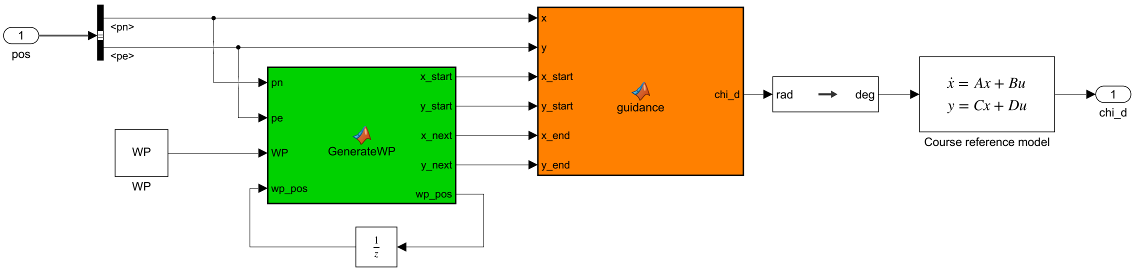


Figure 12: Guidance system implementation in *Simulink*.

The exact implementation of the two major subcomponents in the guidance system can be found in appendix B.1 and B.2. In addition, one can find the guidance system in relationship to the larger control system in appendix A.4.

6.1 Results & Discussion

In order to use the guidance system, there has to be a waypoint input such that the system knows where to guide the aircraft. This input is illustrated in figure 12 as the "WP"-block. Because the guidance system has been implemented in the horizontal plane, it is possible to determine horizontal waypoints, i.e north and east positions (p_n and p_e). These positions / waypoints are fed to the system through a matrix. The desired horizontal trajectory was initialized at $p_n^* = 0$ and $p_e^* = 0$, and from there followed three waypoints as described in the table below.

Table 5: Waypoints.

North (p_n)	p_n^*	1000	1500	2000
East (p_e)	p_e^*	300	-200	200

The desired trajectory (red dash line) and the actual position of the aircraft (blue line) are described in figure 13. For this particular set of waypoints and aircraft characteristics, the proportional gain was set to $K_p = 0.02$. It should be noted that more challenging waypoints calls for a different tuning, rendering this configuration adapted to the waypoints in table 5. The system was also quite sensitive to the K_p-gain. Changing this variable slightly, given the same waypoints, changed the convergence to the path significantly.

As figure 13 illustrates, this controller configuration resulted in a smooth and accurate path-following. The test case was performed with a constant airspeed of $V_a = 18$ m/s, and a constant altitude of $h = 100$ m. It can be seen that when the aircraft is approaching a waypoint, it starts going for the next one before it is reached. This feature results in a smooth flight with fewer sharp turns, which is desirable for commercial aircraft for instance. On the other side, if accuracy is appreciated more greatly than a smooth trajectory, these shortcuts may not be that desirable. A smaller circle of acceptance or other methods for switching to new waypoints are measures than can be taken if accuracy is more important than smooth convergence.

When studying the plot in detail, one can see a tendency of steady-state error. This is not ideal, but also not surprising. Because the controller only uses a proportional gain to control the cross-track error to zero, any steady-state error is not dealt with. In order to remove this error one needs to introduce integral action in the control law. This functionality is called ILOS, or integral LOS [10]. The steady-state error was not significant enough to disturb the path-following, but with more difficult waypoints, extending the system with ILOS might improve the performance of the system.

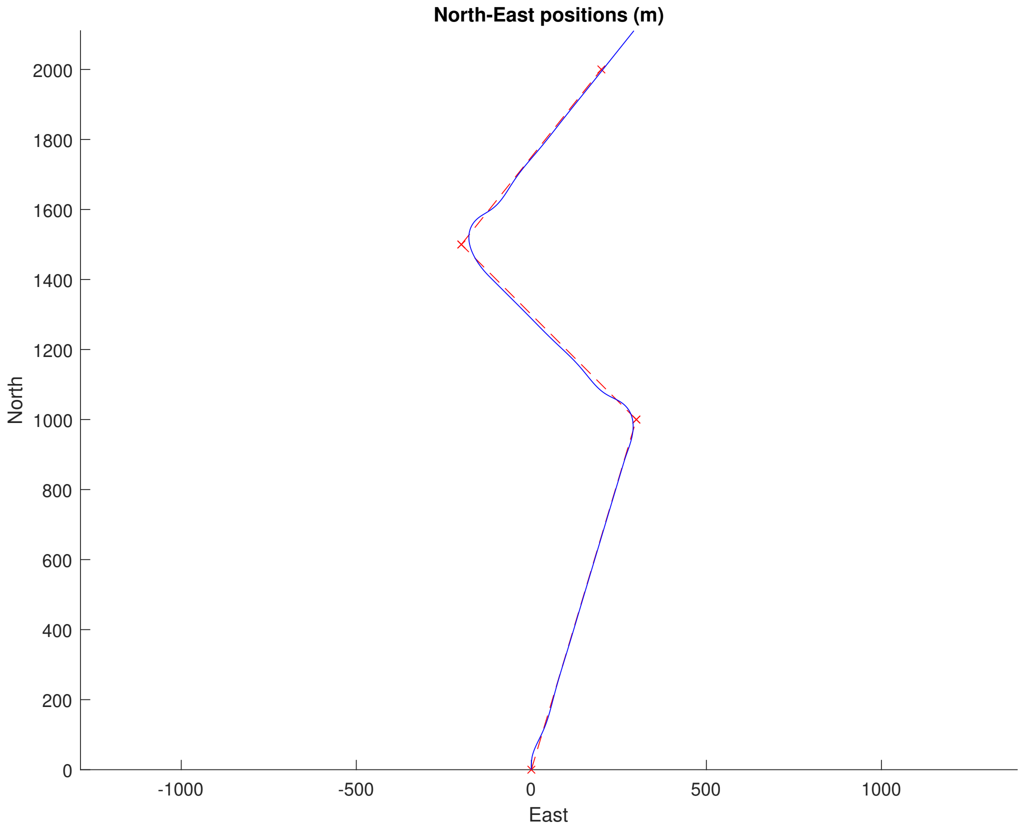


Figure 13: Guidance system test with waypoints.

7 Case Study

In order to determine the potential improvement by using gain scheduling, a case study has been conducted. The study takes two controllers, a nominal controller and a gain-scheduled controller, and compares the performance given some flight maneuvers. The nominal controller uses a control system configured to an airspeed of 18 m/s, while the gain-scheduled controller is configured as explained in section 5.3.

In order to compare the two controllers, it is important that the test scenario is the same for both simulations. The test scenario consists of a simulation lasting 140 seconds, where the full state-space is simulated. Relevant measurements (states) and variables will be recorded for plotting the data afterwards. The airspeed reference is varying from 15-25 m/s to test the performance across the operating conditions. While the aircraft experiences different airspeeds, the autopilots will be tested with altitude and course changes throughout the simulation. The aircraft will perform the same maneuvers with different airspeeds. Any change in performance due to airspeed is therefore more easily identified. The guidance system is not active during this test. This is because it is desirable to do some rapid movements to test the control systems, and the guidance system is more suited for nominal flight conditions and slower dynamics. For the simulations, the differential equations solver *ode45* has been used with variable step-size. This integrator is the main solver for *Simulink* simulations, and is in general suitable for nonstiff differential equations [15].

The test scenario consists of the following maneuvers. The aircraft flies with trim settings for 10 seconds in order to reach a stable flight. This is a level flight with constant altitude and airspeed. The simulation is divided into two tests, the first 60 seconds testing the lateral controller, and the last 60 seconds testing the longitudinal controller. In between those two tests there is a 10 second stabilization period. Throughout both tests, the airspeed starts at 15 m/s and increases by 2 for every 10 seconds, as illustrated in figure 14. For each new airspeed reference, there is a change of 25° in the course reference, or a change of 30 meters in the altitude reference. These maneuvers are done for a control system both with and without gain scheduling.

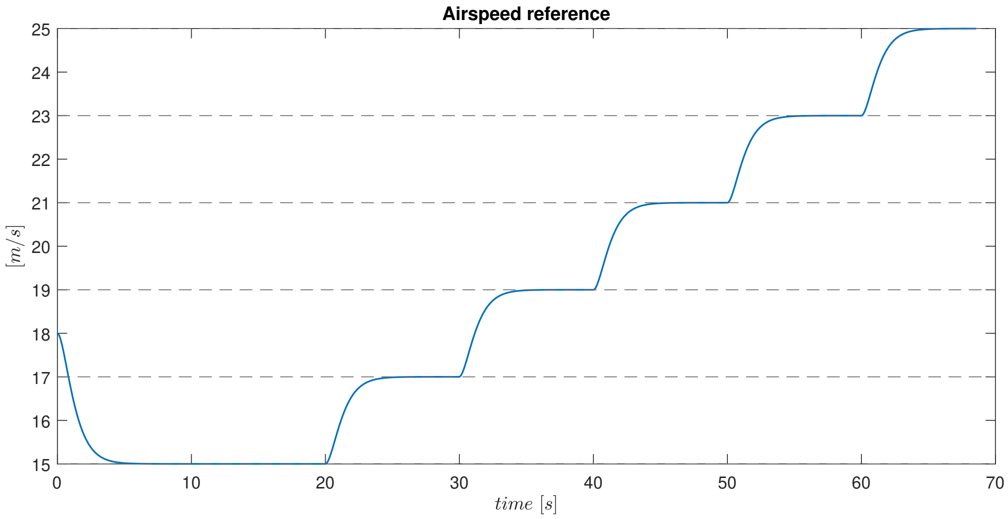


Figure 14: Airspeed reference used in case study.

7.1 Results & Discussion

The results from the case study are illustrated as desired and actual trajectories along with actuator deflections in figures 15, 16, 17, and 18. Within the plots, the simulation is divided into seven sectors. The first one being stable and trimmed flight, and the following six are the various airspeeds tested. All of the plots are set up in the same way, where the actual trajectory is illustrated by a blue line, the desired trajectory is illustrated by a green line. Along with trajectories, the actuator deflections as well as thrust are illustrated as red lines.

The test results from the nominal controller are displayed in figure 15 and 16. The lateral control system indicates a well-functioning controller, where the aircraft managed to follow the desired trajectory accurately. By studying the plot, one can see that there is a larger discrepancy when the airspeed is low contrary to when the airspeed is high. This is supported by larger aileron deflections in the beginning of the test. At 15 m/s the deflection was $\sim 3°$, and at 25 m/s it was just below 1°. This is most likely due to the fact that less deflection is needed to perform rolling motions in high speeds. There were also more oscillations in the aileron deflections at low airspeeds, which testifies to a more active control system at those speeds. It should also be noted that in the first tests conducted, the course reference was to change by 30° at every new airspeed reference, but that rendered the controller unstable.

Moving on the the nominal longitudinal controller, there were in general more oscillations in play both in the dynamics and elevator deflection. These oscillations seems to be present independently of airspeed, but seems to become more dominant as the airspeed increases. If one studies the elevator deflection at the end of the test, it is safe to assume that the oscillations would result in a bumpy and maybe unstable flight should the airspeed increase further. This tendency is also present in the airspeed control system, where an increasingly oscillating thrust force lead to an increasingly oscillating airspeed. However for the operating conditions relevant for this case study, the airspeed controller gets the job done nonetheless.

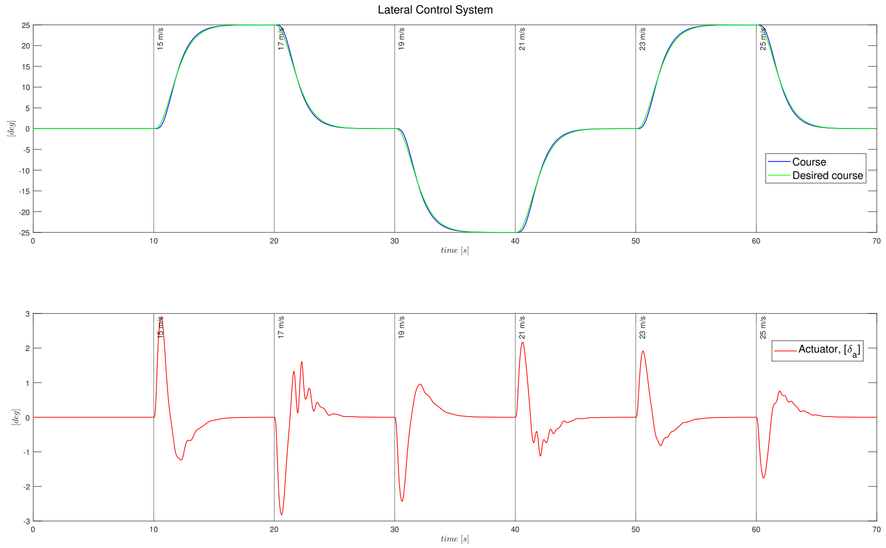


Figure 15: Lateral dynamics with nominal controller.

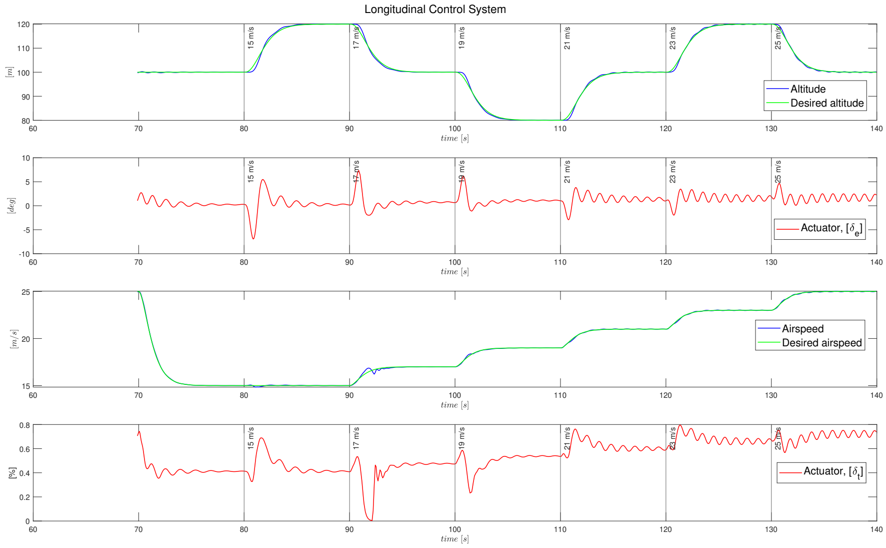


Figure 16: Longitudinal dynamics with nominal controller.

When gain scheduling was introduced to the control loops, we get what is illustrated in figure 17 and 18. Regarding the lateral control system, there is no apparent difference in performance in course control. The plot testifies that the actual course trajectory followed the desired trajectory accurately despite of different airspeeds. There was one improvement however, and that is slightly smoother aileron deflections. That does not affect the lateral dynamics to any significant degree, but resulted in a calmer rolling motion throughout the tests. Like the nominal controller, the aileron deflections decreased in size as the airspeed increased. It also seems like the gain scheduled controller yields more accurate course control when the airspeed increases.

It is when gain scheduling was introduced to the longitudinal control system that we see the biggest improvement. The nominal longitudinal controller was quite affected by oscillating dynamics and actuator deflections, which was not the case in the gain scheduling controller. As illustrated in figure 18, there were still some oscillations in the aftermath of a jump in reference, but they were dampened in a manner of seconds. This is especially visible in the plot describing elevator deflection. This has a direct influence on the altitude dynamics, which were oscillating when a new altitude reference was initiated and was smoothed out relatively quick. There is a pattern of less oscillations in general as the airspeed increased, which indicates a more accurate control of altitude when the airspeed is high. The airspeed controller also dealt with increasingly worse oscillations as airspeed increased in the nominal controller. In the case of the gain scheduling controller, these oscillations were significantly dampened, especially at high airspeeds.

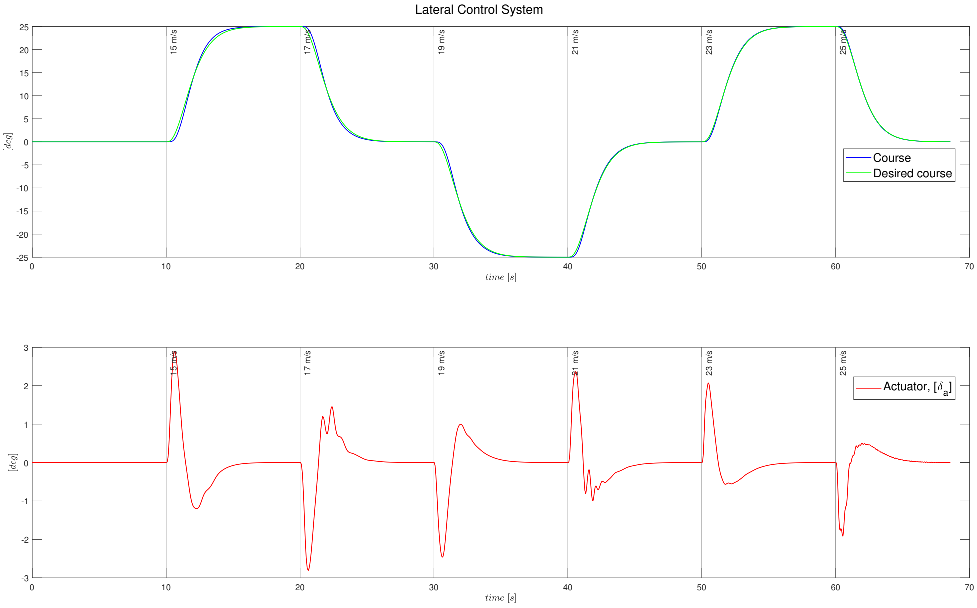


Figure 17: Lateral dynamics with gain scheduling.

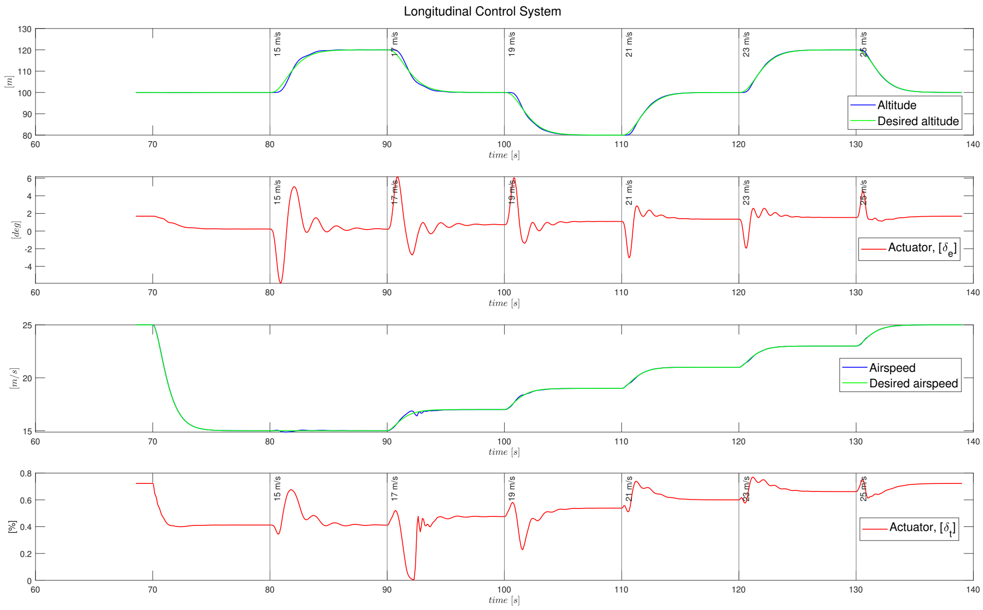


Figure 18: Longitudinal dynamics with gain scheduling.

The case study tested a lateral and longitudinal controller both with a nominal controller and a gain-scheduled controller. On a general note, the results seem to indicate that the controller with gain scheduling provided better flight performance. Interestingly, the lateral dynamics were considerably less affected by changes in airspeed than the longitudinal dynamics. That also meant that introducing gain scheduling to better the control was not particularly helpful. The difference between the nominal controller and the gain-scheduled controlled really came to light when studying the longitudinal dynamics. Unwanted oscillations present with nominal control were dampened with gain scheduled control. The altitude dynamics became smoother and better at following the desired trajectory, and the elevator deflections became smaller in magnitude as well as less oscillating when gain scheduling was used.

In figure 19, the three-dimensional flight path resulting from the maneuvers is displayed. As one can see, the first half dealt with course control, and the second half dealt with altitude control. Conducting lateral and longitudinal maneuvers simultaneously could possibly lead to different outcomes in performance, and is something that would be interesting to research further. However, both the lateral and longitudinal controller were active throughout the simulations.

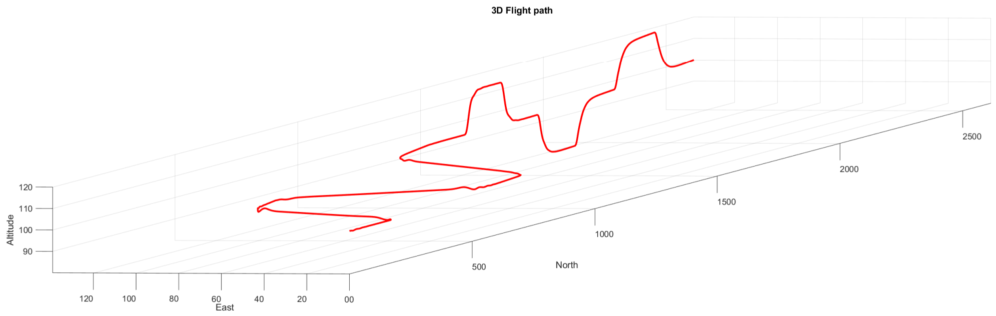


Figure 19: 3D visualization of case study.

8 Conclusion

Unmanned aerial vehicles have been a popular topic of research for some years now, and this thesis was focused in the direction of system identification and control of the Skywalker X8 aircraft.

System identification algorithms entails describing a dynamic system in mathematical terms. For this particular aircraft, there has been derived a preliminary model where the aerodynamic properties were to be identified. The "Output-error method" was used to perform this task, and showed ambiguous results. Many factors such as sensor data, integration solver, initialization, played a role in the outcome of the identification process. Given that these factors were similar throughout the tests, the longitudinal parameters seemed to be harder to identify accurately in comparison to the lateral parameters. However, in the lateral identification, the parameters describing the effect roll rate has on the dynamics were consistently inaccurate. One can argue that the initial guesses of the parameters and the type of integrator used has a significant impact on the results, and is something that should be looked into further. As a whole, the "Output-error method" provided decent results. Some aerodynamic coefficients were identified rather accurately, while others were a bit off.

Regarding control of the X8 aircraft, nested PID control loops has been implemented. There are two main control systems; lateral and longitudinal, which combined control pitch/altitude, roll/-course, and airspeed. In addition to the nested control loops, reference models have been utilized for a realistic reference trajectory. The controller gains are usually constant, but since this is in fact a nonlinear system, gain scheduling was implemented to potentially improve controller performance. Using airspeed as the scheduling variable, the performance was indeed improved when performing various flight maneuvers in various airspeeds. The most significant improvement occured in the longitudinal control system. With constant gains, oscillations became quite significant at higher speeds, but those were dampened out when gain scheduling was used. One can therefore argue that gain scheduling allows for better control across operating conditions than a nominal PID controller.

The control system was extended with a LOS guidance system on the outer loop. The objective of this system is to take a set of predefined waypoints, together with the ever-changing inertial position of the aircraft and output a desired course angle. This course command is then passed to the control system for accurate course following. The guidance system is based on LOS combined with a path generator whose purpose is to decide when to switch to a new waypoint. The proportional LOS guidance law is based on minimizing the cross-track error, and this algorithm must be tuned. Through testing, it showed that the system was quite sensitive to the tuning parameters and the waypoints it was to guide the aircraft through. However when adapting the tuning parameters to the waypoints in table 5, the system performed in a smooth and accurate manner. There was a tendency of steady-state error when the system was tested. This is expected due to the fact that potential modelling errors and disturbances are not dealt with by integral action in the controller. For this error to be removed, one could replace the proportional LOS guidance law with integral LOS, or ILOS. However, the steady-state error was almost non-existent as wind was not a factor, rendering the proportional LOS system effective.

One thing that these various aspects of unmanned aerial flight have in common is that they are somewhat case sensitive. One can design well-functioning systems that does their job in one operating condition, and does not work in another. However it is shown that ideas like gain scheduling reduces this case sensitivity. In addition, as more advanced guidance systems come to be, the act of guiding aircraft becomes increasingly more autonomous with tools such as LOS. This thesis also shines a light on the necessity and difficulty of identifying good mathematical models of nonlinear systems. One can most likely never mathematically describe aerial vehicle dynamics perfectly, but methods such as the "Output-error method" certainly makes the case that accurate models potentially can be identified to an acceptable degree.

8.1 Further work

Within system identification there are several topics that could be interesting to study further. In general what makes aerodynamic coefficients hard or easy to identify, but also how different factors affect the identification process. For instance, how different frequencies and shapes of input signals impact the results. Using modal analysis to uncover different excitation frequencies is something that has great potential as well. Regarding the sensor data, they have until now been modeled as "clear" signals without noise. The question of does noisy sensor signals affect the identification abilities to a significant degree, is something that could use more research. The same goes for noisy input signals, which directly affects the flight dynamics, and therefore the sensor data.

Sensor data and input signals is only one side of the story, the actual algorithm itself contains some unanswered questions as well. It would be interesting to study how the choice of integration solver and/or timestep affects the algorithm. Along with that, the question of how much the algorithm is dependent on initial guesses in proximity of the true parameter values is also interesting. This goes back to the optimization problem, where other objective functions, solvers and constraints may be better suited for system identification. Lastly, a comparison of various system identification methods could be very useful, as it is far from certain that the "Output-error method" is the superior one, if there even is a superior one.

The lateral and longitudinal control systems are designed to follow reference trajectories, and it would be interesting if their autonomy was extended. For instance, adding in functionality for take-off, cruise flight, and landing would decrease the human workload in everyday flight. An example of this is to convey to the control system that the aircraft should take off, and the system deals with the rest. Such autopilot features do exist today, and maybe some of these ideas are transferable to a small UAV such as the Skywalker X8. Regarding gain scheduling, this functionality should be extended to include more than one scheduling parameter. As mentioned, operating conditions are affected by for instance payload position, dynamic pressure and angle of attack. By extending the gain schedule to account for these parameters as well, one would increase the robustness of the different control systems. A study of to which degree these parameters affect the controller performance for UAVs would also be of great value.

The guidance system is the high-level system, located outside the lateral and longitudinal control systems. The responsibilities of this system could be extended in a couple of ways. For instance one could introduce features such as weather routing, obstacle avoidance and mission planning. Again, there exist examples of these systems in commercial and military aviation, and implementing them in small UAVs would increase their autonomy. Using ILOS instead of LOS was discussed earlier, and should be studied further in order to deal with steady-state error. This becomes more relevant in the presence of wind, which is also something that could be implemented to make the simulations more realistic.

A Simulink diagrams

A.1 Lateral Control System

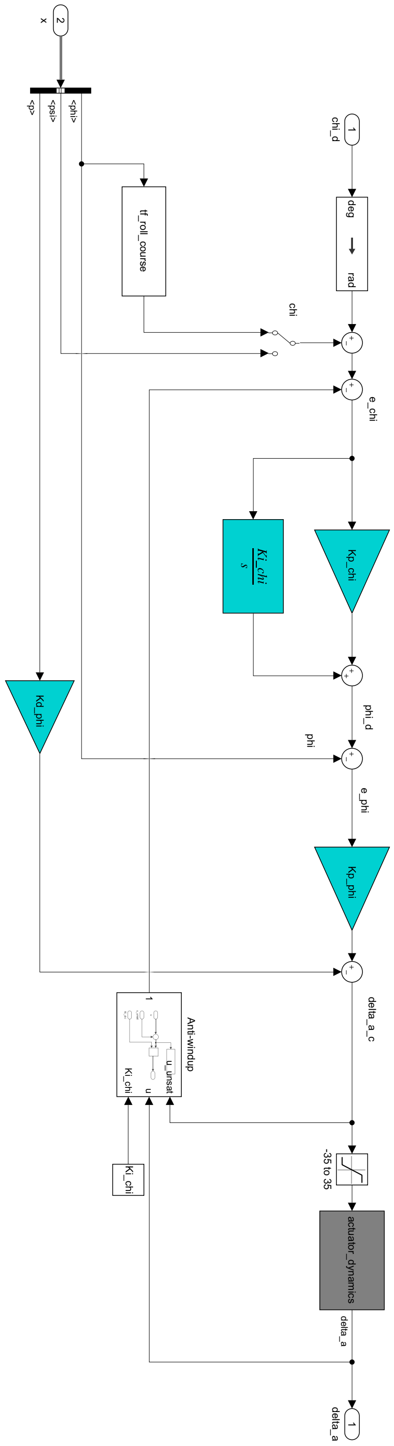


Figure 20: Lateral control with constant gains.

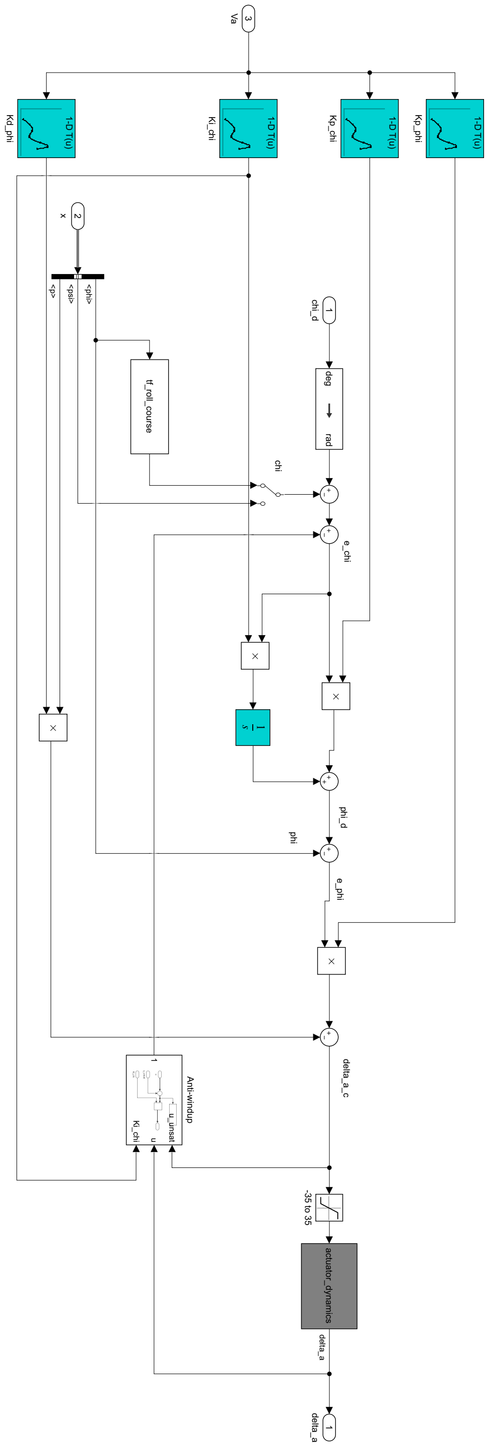


Figure 21: Lateral control with gain scheduling.

A.2 Longitudinal Control System

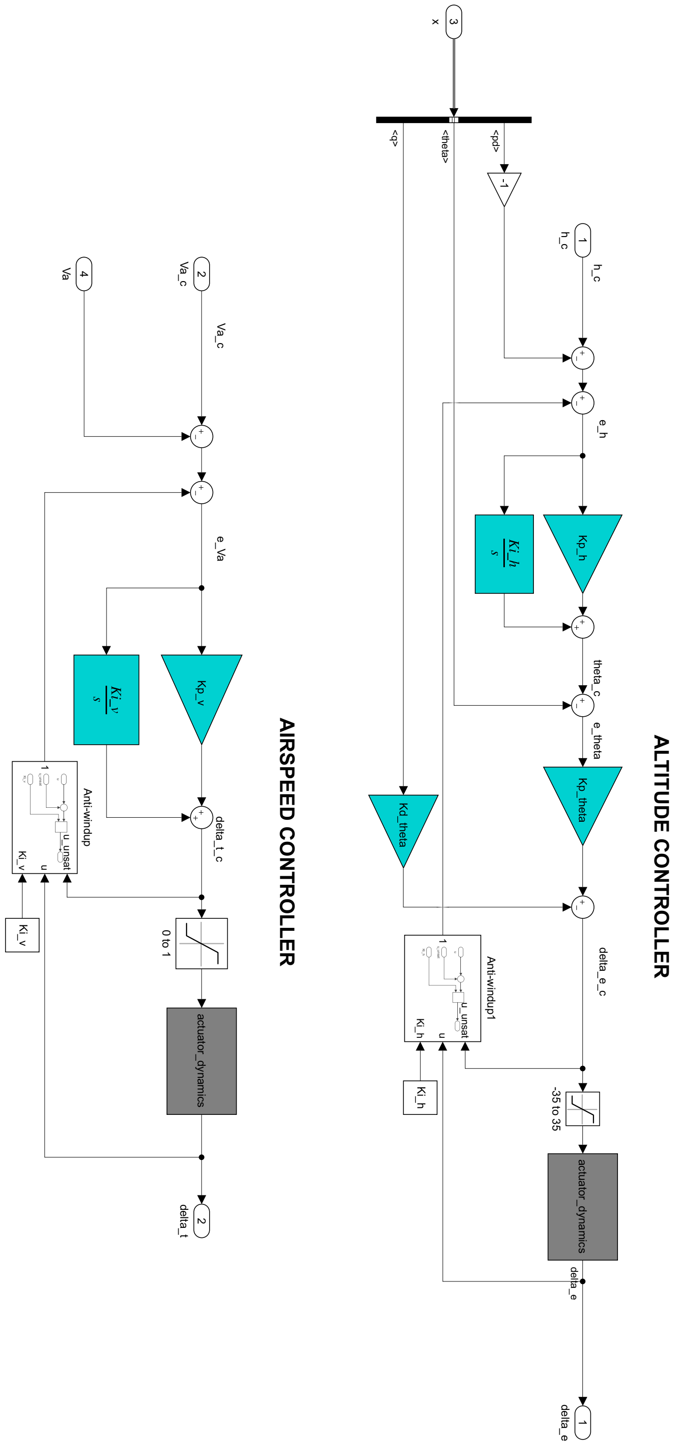


Figure 22: Longitudinal control with constant gains.

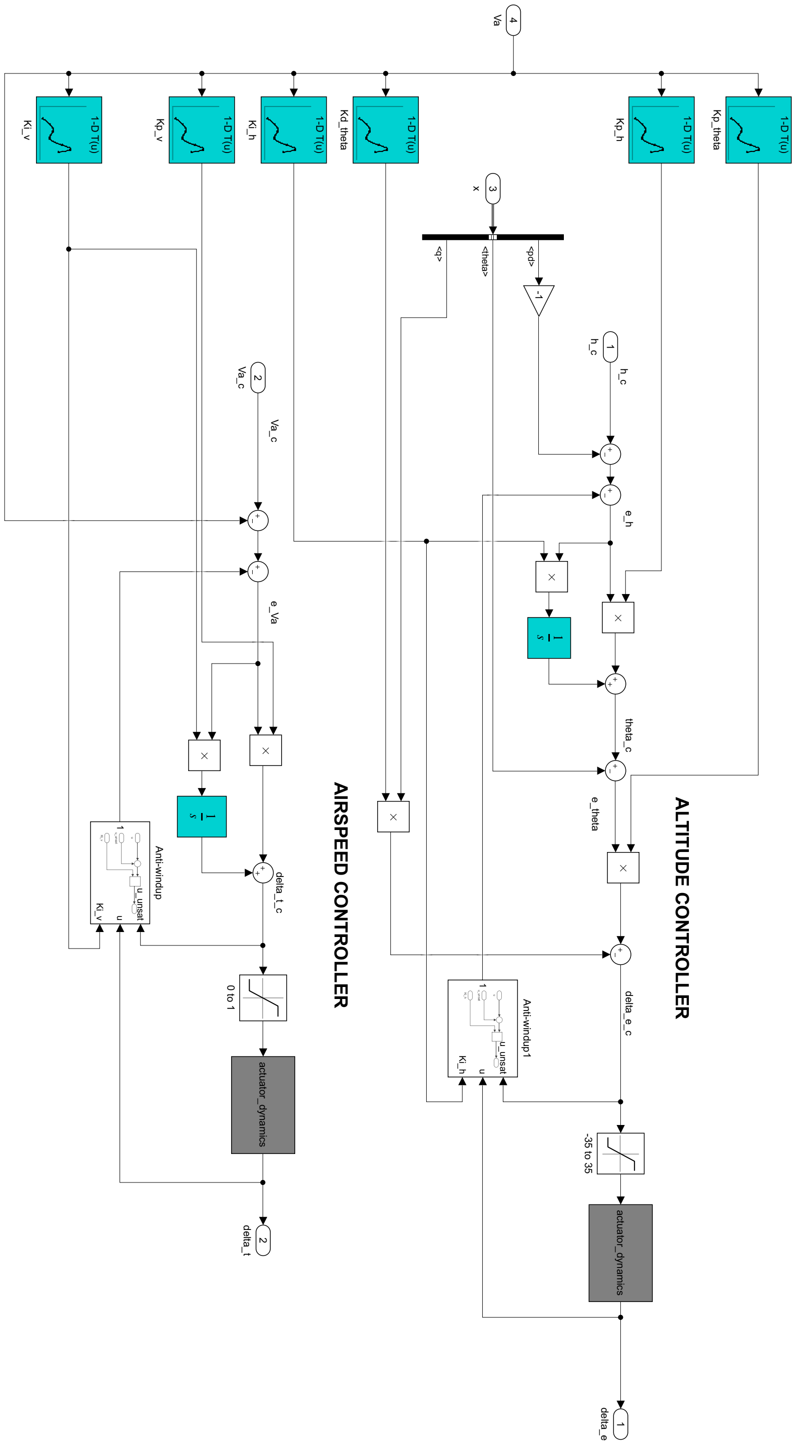


Figure 23: Longitudinal control with gain scheduling.

A.3 Autopilot

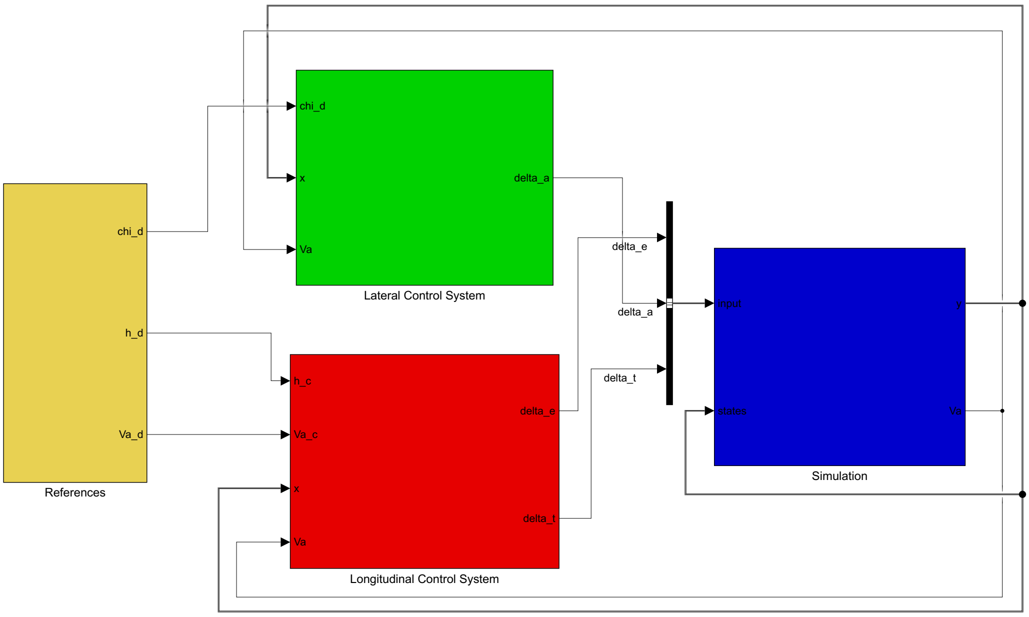


Figure 24: Overall control system.

A.4 Autopilot with Guidance System

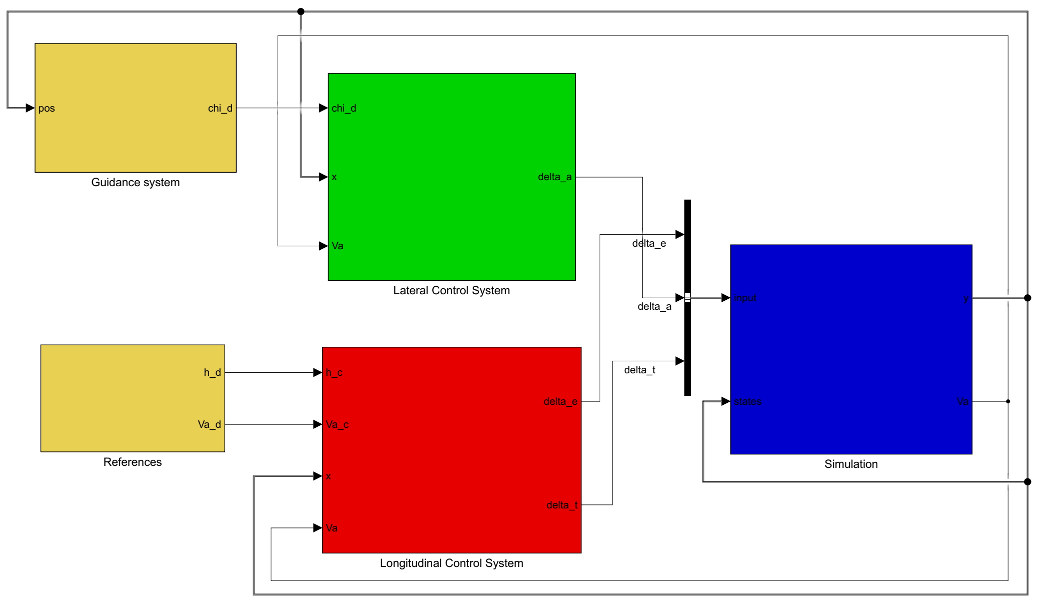


Figure 25: Overall control system with guidance system.

B Code

B.1 Generate trajectory

```
function [x_start,y_start,x_next,y_next,wp_pos] = ...
GenerateWP(pn,pe,WP,wp_pos)

if wp_pos == 0
    wp_pos = 1;
end

x_wp = WP(1,:);
y_wp = WP(2,:);

x_next = x_wp(wp_pos+1);
y_next = y_wp(wp_pos+1);

x_start = x_wp(wp_pos);
y_start = y_wp(wp_pos);

num_wp = length(x_wp);

R_i = 200; % Switching radius
coa = sqrt((x_next - pn)^2 + (y_next - pe)^2);
if coa <= R_i
    if wp_pos < num_wp-1
        wp_pos = wp_pos+1;
    end
end
end
```

B.2 Proportional Guidance Law

```
function chi_d   = guidance(x, y, x_start, y_start,...
x_end, y_end)

% Calculate cross track error
[~,~,y_e] = crosstrack(x_end, y_end, x_start, y_start, x, y);
% Path-tangential angle
pi_p = atan2(y_end-y_start,x_end-x_start);
% Proportional gain
Kp = 0.02;
% Desired course
chi_d = pi_p - atan(Kp*y_e);
end
```

B.3 SYSID Algorithm

```
e = y_data(:,1:200)-X_symbolic;

nlp = struct;
nlp.x = params;
nlp.f = 0.5*dot(e,e);
opts = struct;
opts.ipopt.print_level = 0;

solver = nlpsol('solver','ipopt',nlp,opts);
sol = solver('x0',param_guess);
sol_x = full(sol.x).*scale;
```

B.4 SYSID numerical results

Table 6: System identification results.

		True parameters	Estimated parameters	Percent error
Lateral	C_{Y_p}	-0.1374	0.4094	397.99 %
	C_{Y_r}	0.0839	0.0288	65.70 %
	C_{Y_0}	0.0032	0.0052	64.21 %
	C_{Y_β}	-0.2240	-0.1592	28.94 %
	$C_{Y_{\delta_a}}$	0.0433	0.1636	277.81 %
	C_{l_0}	0.0041	0.0026	37.38 %
	C_{l_β}	-0.0849	-0.0524	38.31 %
	C_{l_p}	-0.4040	-0.1756	56.54 %
	C_{l_r}	0.0555	0.0477	14.12 %
	$C_{l_{\delta_a}}$	0.1202	0.0758	36.93 %
	C_{n_0}	-0.0005	-0.0003	26.30 %
	C_{n_β}	0.0283	0.0244	13.70 %
	C_{n_p}	0.0044	-0.0170	489.65 %
	C_{n_r}	-0.0120	-0.0131	9.15 %
	$C_{n_{\delta_a}}$	-0.0034	-0.0008	77.74 %
Longitudinal	C_{L_0}	0.0867	0.1049	20.97 %
	C_{L_α}	4.0200	6.6580	65.62 %
	$C_{L_{\delta_e}}$	0.2780	0.4235	52.35 %
	C_{L_q}	3.8700	5.2307	35.16 %
	C_{D_0}	0.0197	0.1066	441.27 %
	$C_{D_{\alpha 1}}$	0.0791	0.1371	73.29 %
	$C_{D_{\delta_e}}$	0.0633	0.1183	86.87 %
	C_{D_q}	0	0	NaN
	C_{m_α}	-0.1260	0.0190	115.04 %
	$C_{m_{\delta_e}}$	-0.2060	0.6470	414.09 %
	C_{m_q}	-1.3012	4.9194	478.06 %
	C_{m_0}	0.0302	0.0653	116.21 %

B.5 Trim computation

```
x_dot_d = [0, 0, -Va_d*sin(gamma_d), 0, 0, 0, 0,...
0, Va_d*cos(gamma_d)/R_d, 0, 0, 0]';
statespace = 'full';
x_dot = equationsOfMotion(x,u,P,statespace);

sum = 0;
N = 12;
for i = 3:N
    sum = sum + (x_dot_d(i) - x_dot(i))^2;
end

f = sum;
nlp = struct;            % NLP declaration
nlp.x = x;               % decision vars
nlp.f = f;               % objective
```

B.6 Controller gains

```
% For roll controller
a_phi1 = -(1/2)*P.rho*Va^2*P.S_wing*P.b*P.C_p_p*P.b/(2*Va);
a_phi2 = (1/2)*P.rho*Va^2*P.S_wing*P.b*P.C_p_delta_a;

% For pitch controller
a_theta1 = -(P.rho*Va^2*P.c*P.S_wing*P.C_m_q*P.c)/(4*P.Jy*Va);
a_theta2 = -(P.rho*Va^2*P.c*P.S_wing*P.C_m_alpha)/(2*P.Jy);
a_theta3 = (P.rho*Va^2*P.c*P.S_wing*P.C_m_delta_e)/(2*P.Jy);

% For airspeed controller
a_v1 = P.rho*Va_star*P.S_wing*(P.C_D_0+P.C_D_alpha1*alpha_star+...
    P.C_D_delta_e*delta_e_star)/P.m + ...
    P.rho*P.S_prop*P.C_prop*Va_star/P.m;
a_v2 = P.rho*P.S_wing*P.C_prop*P.k_motor^2*delta_t_star/P.m;
a_v3 = P.g*cos(theta_star-chi_star);

% Roll loop
zeta_phi = 0.9;             % Design parameter
e_phi_max = deg2rad(20);    % Design parameter
delta_a_max = P.delta_a_max;
omega_n_phi = sqrt(abs(a_phi2)*(delta_a_max/e_phi_max));
Kp_phi = (delta_a_max/e_phi_max)*sign(a_phi2);
Kd_phi = (2*zeta_phi*omega_n_phi - a_phi1)/a_phi2;
Ki_phi = -0.5; % Trial and error, use rlocus when ideally
% Questionable to use integral term in inner loop...

% Course loop
W_chi = 9;           % Design parameter (bandwidth separation)
zeta_chi = 0.707;    % Design parameter
omega_n_chi = omega_n_phi/W_chi;
Kp_chi = 2*zeta_chi*omega_n_chi*Vg/P.g;
Ki_chi = omega_n_chi^2*Vg/P.g;

% Pitch loop
zeta_theta = 0.707;         % Design parameter
e_theta_max = deg2rad(20);  % Design parameter
delta_e_max = P.delta_e_max;
omega_n_theta = sqrt(a_theta2 + ...
    (delta_e_max/e_theta_max)*abs(a_theta3));
Kp_theta = (delta_e_max/e_theta_max)*sign(a_theta3);
Kd_theta = (2*zeta_theta*omega_n_theta - a_theta1)/a_theta3;
K_theta_DC = (Kp_theta*a_theta3)/(a_theta2 + Kp_theta*a_theta3);
```

```
% Altitude loop
W_h = 9;                    % Design parameter
zeta_h = 0.707;             % Design parameter
omega_n_h = omega_n_theta/W_h;
Kp_h = 2*zeta_h*omega_n_h/(K_theta_DC*Va);
Ki_h = omega_n_h^2/(K_theta_DC*Va);

% Airspeed loop
zeta_v = 9;
omega_n_v = omega_n_h*10;
Kp_v = (2*zeta_v*omega_n_v - a_v1)/a_v2;
Ki_v = omega_n_v^2/a_v2;
```

References

[1] Kostiantyn Abramov. *Skywalker X8 Reverse Enginnering (Process Overview)*. URL: https://grabcad.com/library/skywalker-x8-reverse-enginnering-process-overview-1.

[2] Joel A E Andersson et al. “CasADi – A software framework for nonlinear optimization and optimal control.” In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36.

[3] *Approximation error*. URL: https://en.wikipedia.org/wiki/Approximation_error.

[4] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[5] Christopher J. Bett. “9 - Gain-Scheduled Controllers.” In: *The Electrical Engineering Handbook*. Ed. by WAI-KAI CHEN. Burlington: Academic Press, 2005, pp. 1107–1114. URL: https://www.sciencedirect.com/science/article/pii/B9780121709600500864.

[6] Morten Breivik. “Topics in Guided Motion Control of Marine Vehicles.” PhD thesis. NTNU, 2010.

[7] Yuanyan Chen and Jim Zhu. “Pure Pursuit Guidance for Car-Like Ground Vehicle Trajectory Tracking.” In: Oct. 2017, V002T21A015.

[8] Bjarne Foss and Tor Aksel N. Heirung. *Merging Optimization and Control*. 2016.

[9] T. Fossen T.I. Perez. *Marine Systems Simulator (MSS)*. 2004. URL: https://github.com/cybergalactic/MSS. (accessed: 01.02.2021).

[10] T.I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2021.

[11] GregorDS. *Six degrees of freedom*. URL: https://commons.wikimedia.org/wiki/File:6DOF.svg.

[12] Kristoffer Gryte. *High Angle of Attack Landing of an Unmanned Aerial Vehicle*. 2015.

[13] Kristoffer Gryte et al. “Aerodynamic modeling of the Skywalker X8 Fixed-Wing Unmanned Aerial Vehicle.” In: June 2018, pp. 826–835.

[14] Eugene Morelli and Vladislav Klein. *Aircraft System Identification: Theory and Practice*. 2nd ed. Williamsburg, Virginia: Sunflyte Enterprises, 2015.

[15] Mark W. Shampine Lawrence F. Reichelt. “THE MATLAB ODE SUITE.” In: *SIAM Journal on Scientific Computing* (1997), pp. 1–22. URL: https://www.mathworks.com/help/pdf_doc/otherdocs/ode_suite.pdf.

[16] N. A. Shneydor. *Missile Guidance and Pursuit: Kinematics, Dynamics and Control*. Horwood Publishing Ltd., 1998.

[17] Benjamin M. Simmons, Hunter G. McClelland, and Craig A. Woolsey. “Nonlinear Model Identification Methodology for Small, Fixed-Wing, Unmanned Aircraft.” In: *Journal of Aircraft* 56.3 (2019), pp. 1056–1067. eprint: https://doi.org/10.2514/1.C035160. URL: https://doi.org/10.2514/1.C035160.

[18] Tobias Vaage. *Model Predictive Control for the Attitude Control of autonomous fixed-wing Unmanned Aerial Vehicles*. 2020.

NTNU

Norwegian University of
Science and Technology