# Obtaining Pitch Control for Unmanned Aerial Vehicle Through System Identification

Lucia Karens and Tawsiful Islam

***Abstract*—This study aimed to develop and evaluate a method to obtain a proportional-integral-derivative (PID) controller. The controller is for a control surface that controls pitch motion, by using data from flight tests with an unmanned aerial vehicle (UAV). Finding a suitable method to develop the controllers is essential to make the UAV autonomous, whilst being stable and controllable. Before developing the PID, data from test flights were used to model a transfer function for the control surface with MATLAB's toolbox for system identification. Thereafter, using the transfer function, the PID was developed by using MATLAB's toolbox for control systems. The whole method was evaluated by studying the rise time, settling time, and overshoot for the PID, and studying how well the transfer function fits with the flight data. The method of modeling the pitch motion with system identification and finding the PID gains has good potential to simplify the process of finding a PID controller. However, to acquire an accurate model for the pitch motion, which in turn can give a well-performing PID, an improved data sampling was suggested. Additionally, flight tests conducted before and after PID tuning, and in different conditions are recommended to be done in future studies. The flight test would work as a validation for the model to acquire a robust PID that performs as expected.**

***Sammanfattning*—Syftet med denna studie var att utveckla och utvärdera en metod för att hitta en proportionerlig integrerande deriverande (PID) regulator. Regulatorn är för en kontrollyta som kontrollerar tipprörelsen genom att använda data från flygtester med en drönare. Att hitta en lämplig metod för att utveckla regulatorer är nödvändigt för att göra drönaren autonom, samtidigt som den är stabil och kontrollerbar. Innan PID:n utvecklades användes data från flygtester för att modellera överföringsfunktionen för kontrollytan med MATLAB:s programvara för systemidentifiering. Därefter, genom att använda överföringsfunktionen, utvecklades PID:n med MATLAB:s programvara för reglersystem. Hela metoden utvärderades genom att studera stigtid, insvängningstid och översläng för PID regulatorn, samt studera hur väl överföringsfunktionen modellerar flygdata. Metoden för att modellera tipprörelsen och att hitta PID förstärkningarna har en god potential att förenkla processen av att hitta en PID regulator. Däremot för att få en precis modell för tipprörelsen, vilket i sin tur kan ge en välpresterande PID, föreslogs det att förbättra datainsamlingen. Dessutom rekommenderades det i framtida studier att flygtester genomförs i olika förhållande, både före och efter att PID regulatorn har hittats. Flygtesterna skulle fungera som en bekräftelse för modellen för att få en robust PID som presterar som väntat.**

***Index Terms*—pitch control, flight tests, unmanned aerial vehicles (UAV), system identification, PID control**

***Supervisors:* Mykola Ivchenko**

***TRITA number: TRITA-EECS-EX-2022:160***

## I. Introduction

Understanding the atmosphere of the Earth and the phenomena within it is fundamental to continuing to develop telecommunications and other satellite-dependent technologies. Observations from the ground, satellites, sounding balloons, and sounding rockets are among the possible ways to observe and understand the atmosphere. In the last decades, unmanned aerial vehicles (UAVs) have been successfully used as observational platforms in a wide range of situations [1]. In order to observe upper-atmospheric phenomena such as aurora borealis, sprites, and blue jets, research in the Space and Plasma Physics Department and the Aeronautics and Vehicle Engineering Department of KTH Royal Institute of Technology has given rise to the student student-driven project ALPHA [2]. The ALPHA project aims to develop a UAV that can fly above cloud level in order to take measurements and imaging of the high-altitude phenomena.

The ALPHA project is currently in its manufacturing phase. The half-scale UAV model is under construction, while the full-scale model is still in the modeling phase. Some parts of the UAV have been constructed already, and material tests are underway. The full-scale UAV will have a wingspan of around 4 meters and will be able to fly at around 10 km above sea level for several hours. [3]. A model of the ALPHA UAV is shown in Fig. 1.

Fig. 1. Current ALPHA design, courtesy of Victor Nan Fernandez-Ayala.

An important part of designing the ALPHA UAV is ensuring that it can fly autonomously. Part of the necessary work is to obtain an accurate model of the UAV. A model can be obtained through various methods, one of them is finding the model through an analytical method and taking help of simulations to find necessary parameters. However, previous studies have shown that simulations might not be suitable to model the true system without simplifications [4], [5]. An alternative method is modeling with system identification. System identification allows to model a system with observed input and output data from empirical work. This method is a common method in the industry to model various dynamic systems and has the advantage of being quick, adaptable, and convenient [6]. This study thus uses system identification with an available UAV.

The aim of this study is to develop and evaluate a method to obtain a PID control system for the ALPHA UAV that controls pitch motion. This study uses system identification to model pitch ('nose up and down') motion by collecting and analyzing data from flight tests that are done with an available UAV.

## II. THEORY: FUNDAMENTALS OF AERODYNAMICS

### A. Basics of Flight

Aerodynamics is the study of the motion of air, and the forces and moments that apply to objects moving through the air. What makes an aircraft fly is a combination of forces and moments acting on the body and the wings of the aircraft. These forces are mainly gravity, thrust, lift, and drag. Designing an aircraft is to take into account these forces to ensure that the model can fly. Properties such as the mass of the aircraft, the surface area and the profile of the wings, the position of the center of gravity, and the shape and the surface area of the tail, all are important to make a viable aircraft [7]. The ALPHA team has spent many hours designing an aircraft that is stable for flight.

An aircraft can move in three dimensions. To describe the orientation of an aircraft, the terms *pitch*, *yaw*, and *roll* are used. In the following paragraph, airplane nomenclature is used. As the ALPHA UAV is an aircraft with fixed wings, the same nomenclature can be used for it.

Pitching is the rotation around the axis that goes through the center of gravity and is parallel to the wings. Yawing is the rotation around the axis that goes through the center of gravity and points vertically from the underside of the fuselage (the body of the airplane) to the top of the fuselage. Rolling is the rotation around the axis that goes through the center of gravity, from the tail to the nose of the airplane. In order to control these movements, the airplane has control surfaces on the wings and the tail. These surfaces can be deflected to induce one or several of the three rotations. To put it simply, on a conventional tail, there are two horizontal control surfaces called the elevators. Deflecting the elevators causes the airplane to pitch. There is also a vertical control surface on the tail, called the rudder. The rudder serves to induce a yawing motion. The control surfaces on the wings have a different name depending on where they are situated. If there is only one control surface on each wing, it is called the aileron. Deflecting the ailerons causes a rolling motion. The control surfaces and the rolling, pitching, and yawing axes are shown in Fig. 2.

The pitching motion is called longitudinal, and the yawing and rolling motions are directional and lateral. It is easier to study the longitudinal motion of the airplane because it can be decoupled from the lateral and directional motions [7], [8]. This means that pitching motion can be controlled entirely by deflecting the elevators while rolling and yawing motions are connected to each other and can not be easily studied independently of one another. This study has chosen to focus

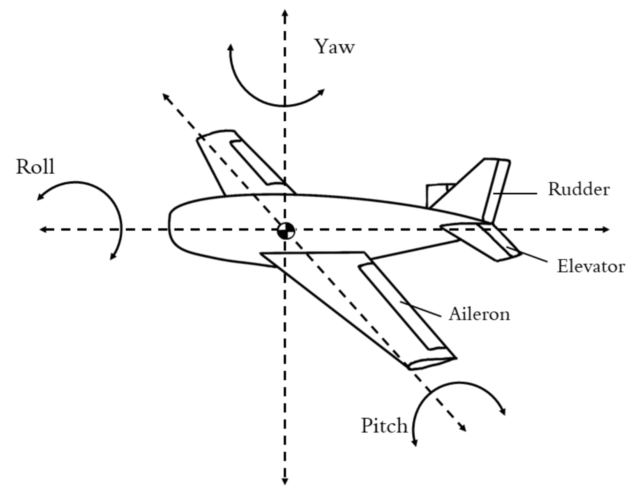


Fig. 2. Pitching, yawing and rolling axes for the aircraft.

on longitudinal control. Longitudinal control is control of the pitch, and of the pitch rate, which is the speed at which the pitch changes.

### B. Reference Frames

Several different coordinate systems are used in aerodynamics. It is important to use the right frame of reference when looking at the system responses of the aircraft.

*1) The Body Fixed Frame:* The frame is orthogonal. The origin point of the frame of reference is the center of gravity of the aircraft. The x-axis points forward through the nose of the aircraft and is also the roll axis of the aircraft. The z-axis points perpendicularly down and is the yaw axis. The y axis (pitch axis) is perpendicular to the xz-plane and points to the right in accordance with the right-hand rule.

*2) The North East Down (NED) Frame:* The frame is orthogonal. Its origin is a point on the surface of the Earth or the center of gravity of the aircraft. The x-axis follows the vector line that points to the magnetic North (tangential to the meridian). The z-axis points down towards the center of the Earth. The y-axis points East, tangential to the parallel. The frame is shown in Fig. 3.

The Body Fixed frame does not give information about the orientation or the position of the aircraft. To take these into account, a change of frame of reference is needed, from the Body Fixed frame to an external fixed frame, for example, the NED frame with origin at the center of gravity of the aircraft. Conversion from one frame of reference to the other is done by three consecutive rotations of the reference frame, with so-called Euler Angles [7].

The frame of reference used in this paper is mostly the NED frame since it is the frame used for obtaining equations of motion of the aircraft, and the relevant frame for collecting data.

### C. Equations of Motion

The equations of motion of an aircraft are obtained by taking into account all aerodynamic parameters and forces

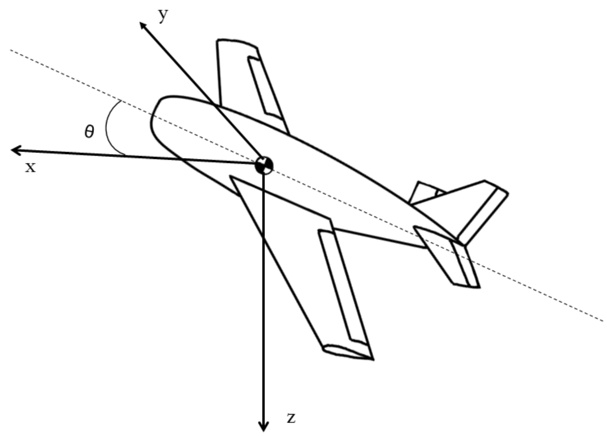


Fig. 3. North East Down frame for the aircraft. In this figure, the aircraft's nose is facing North and is pitching up at angle $\theta$.

acting on the aircraft. By using Newton's second law of motion and switching from the Body Fixed Frame to the NED reference frame, the equations of motion can be obtained [7]. The following Laplace transformed equations are linearized using the small-disturbance theory. This theory assumes that the aircraft only experiences small deviations around a steady flight condition. The equations are also somewhat simplified by assuming that controls are held fixed and that the propulsion is constant. Even though they are simplified, the equations give a good general description of the system [7]. The linearized and simplified equations of longitudinal motion for an airplane are of the form

$$\frac{\Delta q(s)}{\Delta \delta_e(s)} = \frac{As + B}{s^2 + Cs + D} \tag{1}$$

and

$$\frac{\Delta \theta(s)}{\Delta \delta_e(s)} = \frac{As + B}{s^3 + Cs^2 + Ds}, \tag{2}$$

where $\Delta q(s)$ is the change in pitch rate, $\Delta\theta(s)$ is the change in pitch, $\Delta\delta_e(s)$ is the deflection angle of the elevators. The coefficients $A, B, C, D$ depend on the aerodynamic parameters of the aircraft and on its physical properties. For instance, the mass of the airplane, the position of its center of gravity, the velocity, the lift and drag forces, the aerodynamic moments, the current position of the elevators, the current pitch of the aircraft, among others, are all contributing to the coefficients. Detailed equations are presented in [7]. Note that to switch from pitch rate to pitch, a simple multiplication by the integration factor $1/s$ is done. These equations are for so-called short-period motions, which are motions when a small deviation from equilibrium occurs.

## III. Theory: Aircraft

As the ALPHA UAV is still in the modeling phase, and the half-scale ALPHA UAV is in the assembling phase, they are not ready for flight testing. The flight testing is thus done with another radio-controlled aircraft, X-UAV Clouds [9]. This aircraft is commercially available and is marketed towards and used by amateur RC plane pilots. This plane was chosen for its size, which is close to the half-scale ALPHA UAV, and its availability. When it was made clear that the half-scale ALPHA UAV would not be manufactured in time for context L4a and L4b to perform necessary flight tests and other studies, group L4a and the ALPHA flying team acquired Clouds. Clouds was already in possession of the KTH Aeronautics Department. This study is based on data and analysis of Clouds.

Clouds is a fixed-wing plane made of EPO foam, with a V-tail and 1880 mm wingspan. The controllable surfaces of Clouds are the ailerons on the wings and the ruddervators on the tail. There are two propellers on the aircraft, each driven by a motor. The aircraft uses differential thrust, which means that the motors can act independently of one another.

Clouds has been outfitted with a custom electronic system by group L4a [10]. They have also, together with the flying team in ALPHA, made several modifications, including but not limited to, adding landing gear, adding a 3D-printed nose, and a belly plate to make the aircraft more resilient to crashes. An adhesive has also been used to ensure that every part of the aircraft is put in place, and to make the control surfaces more sturdy. The aircraft's appearance in April 2022 can be seen in Fig. 4.

Fig. 4. X-UAV Clouds on April 6th, 2022.

Clouds is a V-tail plane, while the ALPHA UAV has a more conventional T-tail. A V-tail and a conventional tail do not have the same control surfaces. While a conventional tail (and a T-tail) typically has a rudder and two elevators, the V-tail has two ruddervators, which serve as both rudder and elevator. One thing that should be noted when working with aircraft with V-tails is that V-tail dynamics are more complex than conventional tail dynamics. Fortunately, ruddervators can be approximated as elevators if the control surfaces are deflected at the same time and in the same direction [4]. This consequently allows one to use the equation of motion described in section II-C. Another argument for approximating the V-tail to a conventional tail is that the method is developed with the ALPHA UAV in mind, which has elevators and a rudder on its tail.

The hardware and sensors used in Clouds according to [10] result in that the ruddervators and other servos in Clouds do not give feedback to the onboard controller. They act as "black boxes". This means that it is not possible to obtain data on the actual deflection angles of the ruddervators during flight. It is, however, possible to obtain data for the desired deflection angles sent by the onboard controller to the ruddervators. The ruddervators can be modeled by a simple transfer function [7]:

$$\frac{\delta_e}{\delta_c} = \frac{k}{\tau s + 1} \tag{3}$$

where $\delta_e$ is the actual deflection of the ruddervator acting as an elevator, $\delta_c$ is the deflection sent by the controller, and $k$ and $\tau$ are parameters of the servo and its motor. By taking into account this transfer function, the transfer function for the whole servo-dynamic equations of motion system can be modeled. The transfer function for the whole system is the product of (1) and (3) for the pitch rate, and of (2) and (3) for the pitch. The transfer functions of the pitch rate and pitch depending on the angle sent by the controller to the ruddervator are of the form

$$\frac{\Delta q}{\Delta \delta_c} = \frac{A's + B'}{s^3 + C's^2 + D's + E'} \tag{4}$$

and

$$\frac{\Delta \theta}{\Delta \delta_c} = \frac{A's + B'}{s^4 + C's^3 + D's^2 + E's}, \tag{5}$$

where $A', B', C', D', E'$ are coefficients resulting from the multiplications. The complete transfer function thus depends on both the aerodynamic properties of the aircraft and the properties of the ruddervator. The servo response is thus part of the whole system's response. Servos with a faster response time make the entire system respond faster.

## IV. Theory: Flight Control

### A. Obtaining the Transfer Function

To design a proportional–integral–derivative (PID) controller, the system that is studied must first be understood and made explicit. For an aircraft, this means obtaining the transfer functions for the servos and the transfer functions that depend on the aircraft's aerodynamics. There are several ways of obtaining the relevant transfer functions of the system: by simulating, using wind tunnels, and conducting test flights. Each method has its advantages and drawbacks.

Simulating an aircraft makes it possible to obtain aerodynamic parameters without having to fly the aircraft. This is usually done during aircraft design, to ensure that it is stable and flies well. A model of the aircraft is made using Computer-Aided Design (CAD) and then put through Computational Fluid Dynamics (CFD) software to obtain necessary aerodynamic parameters. To be accurate, these simulations use very fine 3D meshes and do heavy calculations. Simulating is often a very time-costly effort.

To use wind tunnels, the aircraft needs to be manufactured. It can be a full-scale aircraft or a smaller model. The wind tunnel then gives data about the aerodynamic properties of the aircraft, without needing to fly the aircraft. Wind tunnels are not easily accessible and are expensive and time-consuming. They do not give very accurate data for smaller UAVs, because of the low speeds, the non-traditional characteristics of the UAVs (lighter mass, unconventional geometry, and more), and the limitations regarding stall, among others [5].

Test flights are relatively easy to conduct and give realistic data. They are the best way to get the real parameters. But they need the aircraft to be fully constructed and designed, and they need time and a pilot. The data obtained might also be inexact due to noise from measuring instruments and atmospheric conditions. For UAVs, especially small ones, doing flight tests is considered to be an advantageous way to obtain the needed parameters [5], [11].

Since Clouds is already manufactured and can be used for test flights, it was decided to use the test flight method to obtain the needed parameters.

### B. Modeling with System Identification

Modeling the pitch motion with data from flight tests can be done with system identification. System identification is a tool to build a mathematical model for a dynamic system. For this study specifically, that would be the pitch motion that is controlled by the ruddervator. In the industry, modeling dynamic behaviors with system identification is an established method. This is mostly thanks to available software like MATLAB system identification toolbox, making it time-efficient compared with analytical methods. System identification is done by using observed data for the system from empirical tests to estimate the model through algorithms for identification. The computer goes through iterations of estimating the model by trying to fit the model to the data and validating it before presenting it to the user. The user's role is to filter data that may be unreliable or remove noise that may lead to an inaccurate model before letting the computer estimate a model. Moreover, the user also chooses the model structure, for instance how many poles and zeros the transfer function should have, or whether the model should be nonlinear or a state-space model. After the estimated model has been presented to the user, they have to do further analysis before accepting it [6], [12].

There are several decisions that need to be considered by the user when using system identification to estimate a model. The first is deciding which input and output signals to measure, to make sure they can describe the system that will be modeled. In addition, the user needs to consider if the signals are measured from an open-loop or closed-loop system. Data from an open-loop system have the advantage that the input is independent of the output. It is also easier to estimate and analyze the model if the observed data are from an open-loop system. There are cases where observing data from a closed-loop system might be needed due to the system being unstable if the output is not controlled. However, this should be avoided if possible as the model may be inaccurate due to the dependent signals [6], [12].

The second essential decision that should be made is the sampling of data, how many data points should be collected, and what the sampling rate should be. Both too fast and too slow sampling can be inconvenient. Too fast sampling leads to redundant data where new data does not provide further information. This is of interest if there is limited data storage space. Too slow sampling is in general worse than too fast sampling. Having few data points makes it difficult to determine the parameters for the model, which is why it should be avoided to determine a reliable model [6], [12].

### *C. PID Controller and Performance Specifications*

In order to have a stable system, the output from the system needs to be controlled, which can be done by using a PID controller. As the PID is a simple controller, it is easy to implement the PID when it uses the reference and output signal to calculate an error. The error is then used to control the input until the output matches the desired reference signal. On account of their simplicity, PID controllers are implemented in various circumstances, software, and even in modern-day aircraft. A block diagram of the closed-loop system studied in this paper is shown in Fig. 5. The open-loop system includes only the servo and the pitch dynamics.

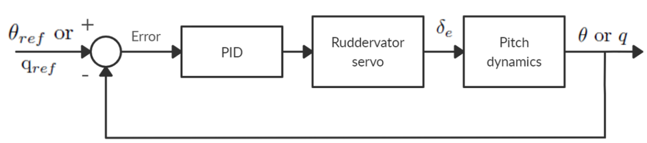


Fig. 5. Closed-loop block diagram for pitch and pitch rate.

The time dependent output signal $u(t)$ is controlled according to the following equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de}{dt} \tag{6}$$

where $e(t)$ is the error signal and is the difference between the reference signal and the output signal. The constants $K_p$, $K_i$, and $K_d$ are the gains for the proportional, integral, and derivative controllers in the PID. Obtaining these gains can be done with different analytical methods by working with root locus or using the Ziegler-Nichols method for instance [7]. However, analytical methods can be time-consuming and may have their drawbacks. In certain cases, it is advantageous to reduce a higher-order system to a second-order system, which makes calculations easier. However, the complexity of the system is not fully accounted for when designing the PID and the airplane might respond differently from what is expected. Other ways to design a PID can be done with software such as MATLAB. In MATLAB an application from the extension Control system toolbox [13] can be used by tuning a response from a transfer function and acquiring the gains from the tuned response numerically. With the application, the response from a higher-order system can be tuned directly, which allows the user to acquire gains for the PID that are more accurate than ones obtained analytically when reducing higher-order systems [14].

There are two different ways to analyze a controller's performance, either by doing it in the time domain or the frequency domain. Within the frame of this paper, the interest in performance lies in how fast the system responds rather than the cyclic behavior of the system. Therefore a time-domain analysis is used in this study. Time-domain specifications one can study are rise time $t_r$, settling time $t_s$, and peak overshoot $M_p$. The rise time of a response describes how fast the output changes from 10% to 90% of the desired output. The settling time is the time it takes for the system's output to stay within an interval of ±5% of the desired output. Maximum overshoot is given in percentage and is the size of the maximum output relative to the desired output. How to study the time specifications is shown in Fig. 6 [7].

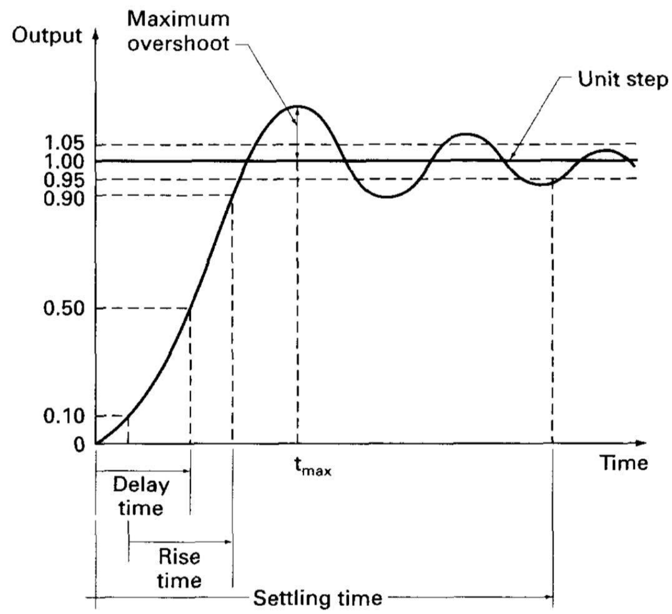


Fig. 6. Time response of an output showing how the time specifications are identified. Figure is retrieved from [7].

When designing a controller in general, it is preferred to have $t_r$, $t_s$, and $M_p$ as small as possible. Sometimes this may not be possible in practice due to the physical limitations of the system. In these cases, design choices are made such as tolerating a maximum of 5% overshoot or allowing the rise time or the settling time to be within a time interval.

## V. THEORY: FLIGHT TESTING

### *A. Flight Test and Flight Logs*

Flight testing is an integral part of determining whether an aircraft is fit for flight or not. Flight testing is conducted after the aircraft has been designed and gone through CAD and CFD simulations to ensure its static stability. Flight tests are also an alternative to obtaining data on how the aircraft responds to phenomena that are difficult to simulate, or non-linear. This is the case when wanting to find the transfer functions for the aircraft. The functions obtained from flight tests may look different from the functions with calculated aerodynamic parameters from simulations as simulations exclude or simplify non-linear phenomena.

When conducting flight tests, a proper pre-flight checklist and flight logging are needed to ensure that flight tests are streamlined and that relevant data are obtained and stored correctly. Flight logging is also necessary to document technical issues and possible crashes. Establishing flight protocols increases safety and improves the efficiency of conducting flight tests [15], [16].

### B. Autopilot

Contexts L4a and L4b discussed the benefits and disadvantages of creating an autopilot from scratch and using an already available autopilot. Creating an autopilot complex enough to control the entirety of the aircraft, conduct pre-planned missions, and fly the aircraft safely, is an enormous task that is beyond the scope of our project. Thus it was decided to use an already available autopilot called ArduPilot.

ArduPilot is free, open-source software that enables users to control and use UAVs, such as helicopters, planes, and others [17]. ArduPilot is widely used by both amateur UAV pilots and students, its versatility making it a valuable asset for piloting UAVs. The accessibility, relative simplicity, and versatility of the software make it ideal to use for the ALPHA project and is the reason why context L4a and L4b decided to use ArduPilot for piloting the Clouds prototype.

ArduPilot connects the aircraft with a ground control station (GCS), most often a computer. The aircraft itself carries sensors, output devices such as motors and servos, and a controller, which is a small computer-like device that takes inputs from the sensors and sends outputs to the output devices. The controller is run by a code that can be downloaded from ArduPilot to fit the specific type of UAV. The GCS is the interface between the user and the controller. ArduPilot's software is Mission Planner [18], which is a program that can be downloaded to any personal computer. Mission Planner makes it possible for the user to control their UAV, download and analyze output data, create detailed missions, and more.

One of the many tools available through Mission Planner is setting flight modes for the UAV. The flight modes that are included in Mission Planner come with presets for the settings on how the UAV can fly. One of the modes that allow the UAV to have assisted flying is called fly-by-wire A (FBWA). FBWA assists the pilot to fly the UAV by limiting how much the aircraft can roll and pitch. However, the elevators are still manually controlled. The maximum and minimum angles (in degrees) that the aircraft can roll and pitch are set by the pilot in Mission Planner. The throttle is manually controlled by the pilot when flying with FBWA [19].

## VI. Method

Designing a controller for the pitch of the aircraft was done in several steps: Obtaining the open-loop response of the system from flight tests, modeling the open-loop response with system identification and finding the PID gains that improved and stabilized the closed-loop response for the pitch and pitch rate. The method is implemented on the pitch angle and pitch rate separately from each other. The evaluation of the method was also done for the separate cases.

### A. Flight Testing

Flight testing has been done mainly by the ALPHA flying team, coordinated and led by Augustsson and Barsby by using X-UAV Clouds with the configuration and specifications according to their report. Before flying to collect data for the pitch or from other servo tests, flight tests were done to ensure that Clouds was trimmed and other parameters were set correctly. This is a necessary step to check that the aircraft was capable of flight, take-off, and landing in order to reduce the risk of crashes [10].

Thereafter, the test for pitching the aircraft with the ruddervators was performed. The test was done in FBWA flight mode to have manual control over the ruddervators without any interference from built-in controllers from Ardupilot. FBWA was also chosen to minimize the risk of stalling the aircraft and crashing, which can be caused by low airspeed when pitching too much [19]. The aircraft flew a straight line with no input for roll or yaw by the pilot. The initial speed before the test was approximately 15 m/s and no throttle inputs were made by the pilot during the test. The pitching was performed for ten seconds where the pilot pitched up and then pitched down. This was repeated immediately after and each pitching motion was executed for about 2-3 seconds.

A flight logging protocol was written, and a pre-flight checklist was created in collaboration with group L4a. The flight logging template is given in the Appendix. The tests that have been performed have been logged in flight logs and the data were saved in tlog files. The tlog files can be further analyzed; data from the tlog files were used to model the response from the ruddervators when changing the pitch.

### B. Determining Transfer Function from Open-loop Response

System identification was used by importing flight data from the test flights into an application for system identification that is included in the MATLAB system identification toolbox. The input was the signal sent from the remote control to the servos, and the output was the pitch of the aircraft. The tlog file that was stored after the flight test contained three relevant data sets for system identification. The data set named chan2_raw_mavlink_rc_channels_t contained the input signal from the remote controller, and the output signals were named pitch_mavlink_attitude_t for pitch and pitchspeed_mavlink_attitude_t for pitch rate. The input was a pulse-width modulation (PWM) signal, which controls the servos with electrical signals. The reason why PWM signals were used as input data was to account for the response time from the servos that affected the total response time. PWM signals are proportional to deflection angles and can be converted if the corresponding PWM signals for maximum deflection angles are known. However, with Clouds and tlog files, exact and true deflection angles are difficult to determine and therefore had not been chosen as the input signal in this study. The output data were in degrees and degrees per second. The data for the pitch changes were imported to the System Identification application. The application required that the two data sets have the same amount of data points and the same sample time. However, due to Cloud's hardware's setting on sampling data, the number of data points did not match and the signals were sometimes sampled at different rates. Therefore, simplifications and assumptions had to be made in order to determine the transfer function. The number of data points for the pitch angle and pitch rate was almost double compared to the data points for the PWM signals over the same time interval. To match the number of data points, every second data point for the pitch was removed. Thereafter the sample time was assumed to be the number of data points over the time interval. Besides input data, output data, and sample time, no further changes were done in the settings for the system identification.

In the application, a transfer function could be estimated from the data by setting how many poles and zeros the transfer function would have. For the open-loop response, a second, third and fourth-order transfer function with none to four zeros were tested to see which of them fitted the flight data the best. The fit percentage was calculated by

$$fit = 100 * (1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}) \tag{7}$$

where $y$ was the original output from the flight data, $\hat{y}$ was the output calculated from the estimated transfer function, and $\bar{y}$ was the mean of the original output. The norm of the differences was calculated in the formula. The application gave a number in percentage of how well the model matched the flight data [20]. The fit is thus how well the model matches with the data.

### C. Designing a PID Controller

After a transfer function for the ruddervators had been estimated, a PID could be determined; to stabilize the system, or possibly make it respond faster. A controller for pitch and a controller for pitch rate were created separately. The pitch PID controller had the error of the pitch as input and the pitch rate PID controller had the error of the pitch rate as input.
The controller was selected to be a parallel PID to find the gains in (6) and was also chosen to be a one-degree-of-freedom PID in the PID tuner application. The gains for the PID were determined by using the PID tuner application, which is included in MATLAB control system toolbox. The response could be tuned by adjusting two sliders in the application. One slider adjusted the response time of the output and the second adjusted the transient behavior of the response, whether it should be robust or not. The criteria that were followed when designing the PID were inspired by the specifications Onuora et al. used [21]. The specifications that the PID needed to fulfill were prioritized in the following order that the:

1) overshoot $M$ would be less than 10%;
2) rise time $t_r$ would be less than 0.2 seconds;
3) settling time $t_s$ would be less than 0.5 seconds.

In order to consider physical limitations of how fast the aircraft could pitch, the times were also chosen to be close to the desired time limit, even though a faster time response could have been chosen. Behaviors like oscillation in the response were minimized as much as possible by first making the response more robust. If that would not work, the response time was increased.

## VII. Results

Several flight tests were conducted. Unfortunately, due to the many setbacks in making Clouds fly [10] and the large number of preliminary tests that were necessary for general flight; pitch collection could only be performed once in the desired conditions, at the beginning of April. During the following flight session, Clouds crashed irreparably.

From the successful flight test, the isolated pitch changes were performed for 10 seconds. From the flight test, 20 data points were obtained for the PWM input, and 40 data points were recorded for the pitch angle and pitch rate. In order to use the System Identification application, the pitch and pitch rate were undersampled by removing every second data point starting from the second data point. Doing this gave a sample time of 0.5 seconds. The comparison between the original data set and undersampled data set is shown in Fig. 7 for pitch angle and Fig. 8 for pitch rate. Both figures also show the PWM signals for the pitch motions during the flight test.

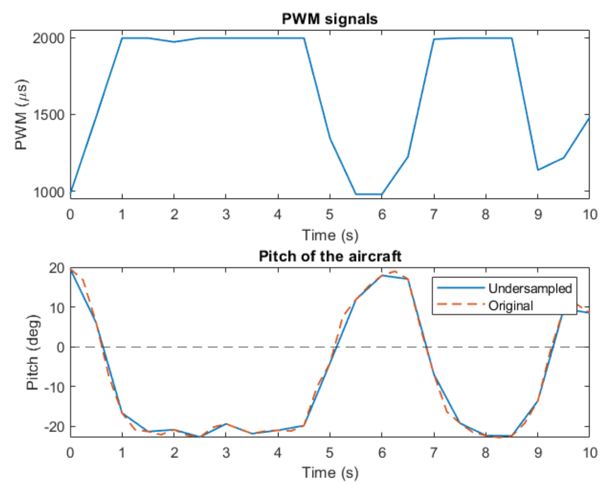


Fig. 7. Input PWM signals and output pitch angles for the aircraft during the flight test.

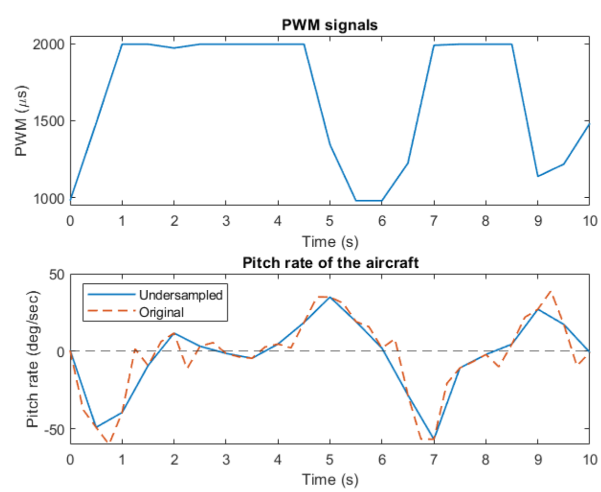


Fig. 8. Input PWM signals and output pitch rate for the aircraft during the flight test.

### A. PID for Pitch

After using system identification to model the pitch, different numbers of poles and zeros were tested to see if they fit the data from the flight test. Fig. 9 shows the tested models where the first number in the name stands for the number of poles and the second stands for the number of zeros. The graphs in the figure show the unit step response for the transfer functions that were approximated for the pitch $\theta$. The two transfer functions that had the best fit compared to the data from the flight test were the following:

$$\theta_1(s) = \frac{\begin{array}{r}-9.83 \cdot 10^{-5}s^4 - 0.059s^3 - 0.00455s^2 \\ -\,0.175s - 0.02\end{array}}{s^4 + 2.03s^3 + 4.49s^2 + 5.9s + 4.43} \quad (8)$$

and

$$\theta_2(s) = \frac{\begin{array}{r}-0.0108s^3 + 2.77 \cdot 10^5 s^2 \\ -\,7.12 \cdot 10^5 s - 6.55 \cdot 10^4\end{array}}{s^3 + 4.39 \cdot 10^6 s^2 + 2.10 \cdot 10^7 s + 1.26 \cdot 10^7}, \quad (9)$$

where $\theta_1$, with four zeros and four poles, had a fit of 80.6% and root mean square error of 3.01 degrees, and $\theta_2$, with three zeros and three poles, had a fit of 77.1% and root mean squared error of 3.56 degrees.

For the PID, $\theta_1$ was used to tune the response for the aircraft's pitch. The gains and the time specifications that were achieved with the PID tuner for the pitch and the unit step response are shown in Table I and Fig. 10.

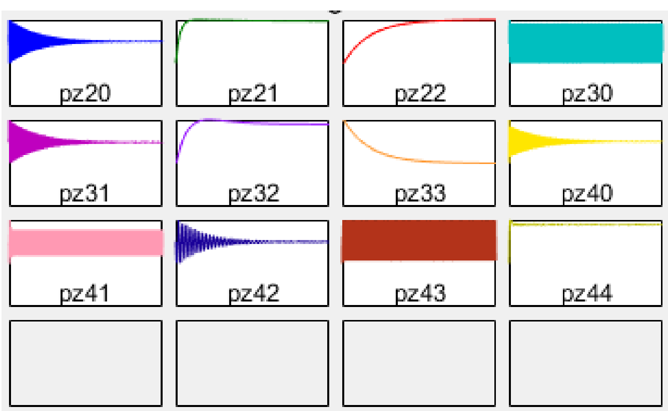


Fig. 9. Tested transfer function models and the shape of their unit step responses are shown in the screenshot from the System identification application. The number of poles is the first number and the number of zeros is the second number in the name of the transfer function. The y-axis shows the amplitude of the response and x-axis is the time. The scale between the graphs varies.

TABLE I
GAIN VALUES AND TIME SPECIFICATION FOR PITCH'S PID

| Parameter | Value |
|---|---|
| $K_p$ | -2581 |
| $K_i$ | -31 963 316 |
| $K_d$ | 0 |
| $t_r$ | 0.00058 s |
| $t_s$ | 0.00419 s |
| $M_p$ | 9.79% |

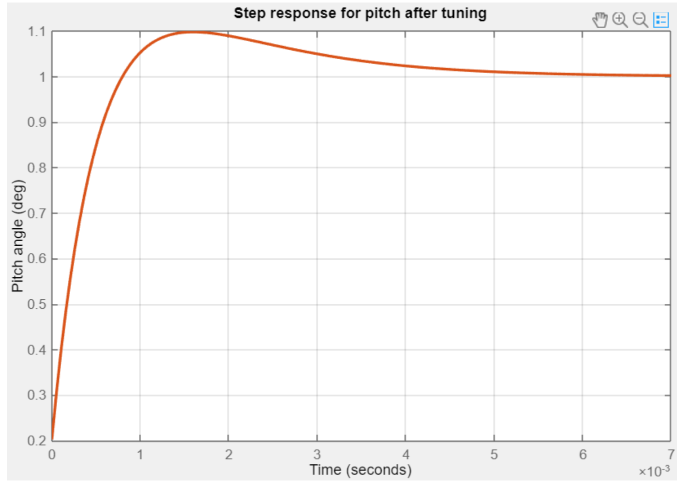


Fig. 10. Unit step response for the pitch after PID tuning.

### B. PID for Pitch Rate

Fig. 11 shows the tested models for the pitch rate. The graph for each transfer function also shows the unit step response. The two transfer functions that had the best fit for the pitch rate $q$ when compared to the data from the flight test were the following:

$$q_1(s) = \frac{-0.0475s^3 - 0.0289s^2 - 0.303s - 0.00421}{s^3 + 2.32s^2 + 5.51s + 12.8} \quad (10)$$

and

$$q_2(s) = \frac{\begin{array}{r}-0.0226s^4 - 0.0367s^3 - 0.0465s^2 \\ -\,0.0727s - 0.00235\end{array}}{s^4 + 0.698s^3 + 6.51s^2 + 0.976s + 7.14}, \quad (11)$$

where $q_1$, with three zeros and three poles, had a fit of 68.72% and root mean square error of 7.21 deg/sec. Transfer function $q_2$, with four zeros and four poles, had a fit of 75.87% and root mean square error of 5.57 deg/sec.

For the PID, $q_1$ was used to tune the response for the aircraft's pitch rate. The gains and the time specifications that were achieved with the PID tuner for the pitch rate and the unit step response are shown in Table II and Fig. 12.

TABLE II
GAIN VALUES AND TIME SPECIFICATIONS FOR PITCH RATE'S PID

| Parameter | Value |
|---|---|
| $K_p$ | 0 |
| $K_i$ | -1363 |
| $K_d$ | 0 |
| $t_r$ | 0.0371 s |
| $t_s$ | 0.314 s |
| $M_p$ | 0% |

## VIII. DISCUSSION AND ANALYSIS

### A. Flight Tests

Flight tests were more difficult to perform than anticipated. Hardware problems such as trimming failures; motor synchronization problems; difficulties in properly connecting all

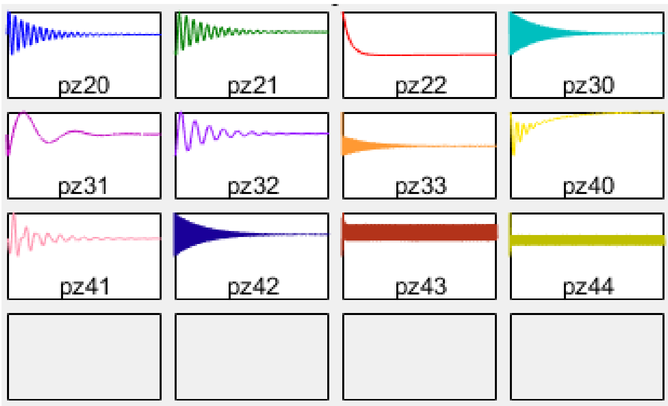


Fig. 11. Tested transfer function models and the shape of their unit step responses are shown in the screenshot from the System identification application. The number of poles is the first number and the number of zeros is the second number in the name for the transfer function. The y-axis shows the amplitude of the response and x-axis is the time. The scale between the graphs varies.

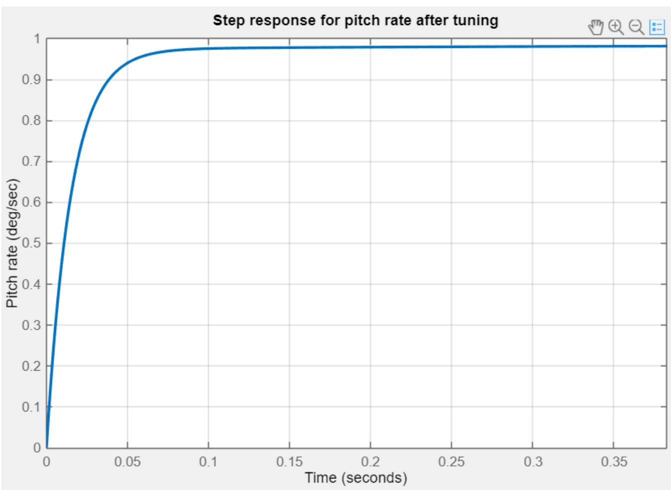


Fig. 12. Unit step response for the pitch rate after PID tuning.

electronic systems with each other and with Mission Planner; inappropriate propeller sizes; and others [10], led to a limited number of flight tests in the time of this study. This was outside of this group's control. In addition, the flight tests where Clouds managed to take off were divided into flights where the flying team learned how to pilot Clouds, flights that tested various flight modes from Ardupilot, and flight tests for collecting data on pitching, which were the flight tests of interest for this study. Pitching tests were not fully prioritized, which led to the result that only one test was performed before the crash. In addition, the flight tests were performed for the most part in FBWA mode, which is not entirely manual, because context L4a and the flying team wanted to minimize the risk of crashes as much as possible. For our study, it would have been better to have results from manual mode, where the pilot has full control of the aircraft, to get a fully accurate model of the system. Flying with FBWA was a compromise between flying fully manual and flying with aid from Ardupilot. As discussed earlier, FBWA does not correct pitching angles but limits them.

The crash led to an abrupt end of the flight tests. The crash took place during the testing of one of the automatic flight modes of Ardupilot. The possible causes of the crash are explored more in detail in [10], but it seems to have been a mixture of flying too low, setting a more aggressive flying style than should have been used, and not setting a maximum speed for Clouds. The crash showed that while Ardupilot is an advanced tool that has many useful options, it does not prevent crashes from happening, and might even cause them if not used correctly. The crash was proof that in order to conduct safe flight tests, it is very important to understand Ardupilot's parameters and functions in great detail. Other conclusions are that flight tests which are for determining the core system's responses (without PIDs or other limitations or help from Ardupilot) should be performed before testing the autopilot functions. This might seem counter-intuitive since the aim of ALPHA is for the UAV to be autonomous. But understanding the system response first is necessary to develop a good PID that will then be a part of the autopilot.

### B. *PID for Pitch*

Regarding the data for the pitch, 20 data points were used in the end for modeling the transfer function for the pitch motion from the ruddervators, which resulted in a sample time of 0.5 seconds. This was a relatively good approximation compared to the actual timestamps. In Mission Planner, one could see the exact timestamp for the signals and see that the time between the data points varied between 0.4 to 0.5 seconds. By studying Fig. 7, the undersampling of the pitch motion was an acceptable simplification as the graph for the undersampled data almost matched the original. The smooth transition of pitch change was lost at several time stamps, for example at $t = 0$ s and $t = 5$ s in Fig. 7. The undersampling might have affected the coefficients in the transfer function, but not significantly.

After using system identification in MATLAB with the data from the flight test, two transfer functions had the best fit, and $\theta_1$ was chosen for PID tuning. It has a better fit and a smaller error than transfer function $\theta_2$, which is why it was chosen. The number of poles was in accordance to what was expected in (5), where the servo response time was taken into account.

The PID that could be acquired for the pitch was able to fulfill the criteria that were set for the controller, with a fast rise time and settling time, and overshoot that fulfilled the desired specifications. On the other hand, the values for the obtained gains were large and negative. The negative sign can be disregarded as this depends on the system's sign convention and the signals that were used. Regarding the values, they are in general smaller, as seen in other cases [4], [7], [21]. However, the values for the gains depend on the system. The system used in this study also had a different order compared to other studies, which have worked with second-order transfer functions. As the transfer function considered the servo's response, the PID for the pitch looked different when tuning. When using smaller gains, oscillations in the response were seen, which is an undesirable behavior in the system. When tuning, it was noticed that a faster response time was the solution to prevent the oscillation. However, one should also have in mind that the transfer function represented a PWM signal as input compared to other cases where the transfer function is using the deflection angle of the control surface as an input. As mentioned earlier, converting from PWM signals to deflection angles was not possible with the used hardware. This was mainly due to differences that could occur between expected servo angle and actual deflection when working with Clouds.

### C. PID for Pitch Rate

The number of data points for the pitch rate was similar to the number of pitch data points, and 20 data points were used to model the pitch rate. The sample time was also 0.5 seconds, which was also a good approximation as the pitch rate was recorded at the same time as the pitch angle. However, the issue with undersampling the pitch rate lead to data that did not represent the pitch rate well as shown in Fig. 8. This showed a case where the system identification method had the drawback of requiring the same number of data points for the input and output and that they needed to be recorded at the same time. In addition, Fig. 8 shows that the pitch rate changed drastically between positive and negative values. Due to the fast-changing values, a sample time of 0.5 seconds was too slow to record a good data set for the pitch rate. In this case, the simplification might not have been suitable to find the transfer function and required a faster sample time.

The transfer function that was chosen as the best for PID tuning was $q_1$. Compared to the fit that was acquired for the pitch's transfer function, $q_1$ only had a fit of 69%. This was a significantly worse result for modeling the pitch rate and was probably caused by the simplifications that were made. Even though the fittings were relatively bad and $q_2$ with 76% fit could also have been chosen, $q_1$ was chosen due to its order that matches better with what was expected as seen in (4). The root mean square error of 7.21 degrees/s was quite a large error in speed. This probably showed that the model was not a good model for the pitch rate with the data from the flight test.

Finding the PID for the pitch rate that would fulfill the requirement with no steady-state error was a challenge. As shown in Fig. 12, the response has a steady-state error and stayed under 1 deg/s past the settling time. As shown in Table II, only an integral controller was needed to tune the response for the pitch rate. However, what type of controller is needed depends on the system acquired. The question that should arise is whether the modeled system is accurate or not. The gain was also large and negative, which together with the gains in Table I probably depended on the system rather than on other errors. In general, this controller might not be the best and is probably not suitable for the UAV due to large errors and bad modeling.

### D. General Evaluation of the Method

In general, the method used in this report has a good potential for being a suitable method for tuning a PID for a UAV. This is mainly due to its simplicity and time efficiency when wanting to find an optimized PID.

Regarding the flight tests, the flight logging proved to be helpful in order to know where to find the data. Flight protocols enabled a streamlining of the flight testing procedures. A small team of three proved to be sufficient for flight testing. Nonetheless, the method lacked in clearly establishing which specific flight tests should be conducted, and in what order, as discussed in section VIII-A.

Regarding the PID, even though the method is simple and quick, the simplifications that were needed in order to find a transfer function presented some drawbacks of this method. Unsynchronized and undersampled data sets led to a result that may be unreliable when it came to the coefficients in the transfer functions and the time specifications of the PID. An improved data sampling could have shown the true pitch motion and pitch rate, which were lost when removing data points. On top of that, the speed of pitch changes depending on PWM could have been more accurate and might have shown some time delays in the system response, which were potentially lost with the undersampling. As shown in Fig. 7 and Fig. 8, no delays are apparent. The main reason why the data set needs to be correct and improved is that a bad data set did not give the best and most accurate model for the pitch motion. This thereafter affected the PID gains that were found from the PID tuner.

The transfer functions that had the best fit had the correct number of poles, but not the correct number of zeros if one assumed the servo's transfer function was included. It should be noted that the equations on which the theory is based are simplifications. The real transfer function for the longitudinal equations of motions may contain more poles and more zeros.

In this case, the system identification was relatively suitable. Other reasons why the system identification was not perfect, were that the fit for the pitch and pitch rate was under 90%. A fit better than 95% would allow one to believe the models were accurate with correct coefficients and small errors. As mentioned earlier, a correct model is needed in order to have a stable system, and the PID depends on the transfer function. The system identification method was highly time-efficient and multiple transfer functions could be tested to see how well they fitted the data in a matter of seconds. Compared to analytical methods, using the system identification toolbox was a time-efficient method. An analytical method with simulation could have given a more accurate model, but as discussed in section IV-A, it is not always possible to achieve correct and complete aerodynamic parameters from simulations alone.

The PID tuner was also easy to use and could be obtained within a few minutes. If given a correct and known transfer function, the gains could have been accurate. There is room for questioning whether the obtained values are correct considering they differ from other similar studies. In this study, there are strong suspicions that the values may be incorrect. It was difficult to determine whether they were correct or not since getting a good response was always possible when tuning in the application. Additional flights should have been performed in order to acquire a wide variety of data. The variation would work as a validation that would give a more accurate and robust transfer function. Another validation process would be to test the robustness of the PID by applying it to the aircraft and analyzing how well the aircraft responds with the PID. The robustness would also be tested as the aircraft's pitch motion can be affected by disturbance and turbulence. These flights can also be used to iteratively fine-tune the gains until a desirable response has been achieved.

### E. Future Improvements

To perfect the method and obtain the best possible results for the ALPHA UAV, several improvements are suggested.

First, data collection should be improved. The data sampling depends on the source code that is used by Ardupilot. The source code for Ardupilot is publicly available, which allows changing and synchronizing the sampling rate for the different sensors in the aircraft. A faster sample time would give an accurate model that records the small and drastic changes in pitch and pitch rate. A synchronized sampling between the PWM signals and the pitch angle helps to give a more accurate response time where the output's response might actually be different from what has been presented in Fig. 7 and 8.

Second, while accurate data help to find an accurate model for the pitch motion and pitch rate, data need to be collected more than once. Indeed, collecting data once may give a transfer function that is not robust and only coincidentally matches with the data from the flight test. To ensure the model is accurate and robust, more than one flight test is needed. They would also need to be conducted at different times of day and weather conditions to account for a wide range of aerodynamic disturbances. A robust model would then fit all different cases. This group proposes that at least three different test flights be conducted where pitch changes are recorded, to maximize the accuracy of the transfer function model.

Third, manual or FBWA flight tests should be prioritized before flight tests for autopilot functions. Flight tests should be clearly organized and the aspects to be tested should be determined in advance for more efficient tests.

Fourth, the PID gains should also be validated, which was not done in this study. After finding the PID gains from the PID tuner in MATLAB, the gains can be implemented in Ardupilot. A flight test can then determine how well the controller performs. After conducting flight tests in different conditions, the flight data can be used to analyze the time domain specifications for the controller and its robustness. If the controller does not perform as expected, new PID gains can be found again from MATLAB. However, there are risks involved with testing the PID in this way. If the system is unstable due to the controller, the aircraft is at risk of crashing. Therefore, the PID should be tested in MATLAB or other similar environments before being implemented. Testing for different input signals will show if the PID is robust or not. After these tests, the PID can be implemented into the aircraft. To further prevent crashing, a professional or experienced pilot should fly the aircraft in order to be able to take manual control in case the PID is unstable.

Fifth, an additional comparison could be made between the built-in PIDs in Ardupilot and the PID obtained through system identification. This would require converting the PWM signals to angles in degrees. If the system identification PID is better than the built-in one, it is pertinent to implement this study for the half-scale and the full-scale ALPHA UAV.

Sixth, flight test data could be filtered in order to reduce noise and uncertainties [4], [5], [11].

## IX. Conclusion

In conclusion, using system identification and PID tuner with MATLAB toolboxes have the potential to simplify the process of finding a PID for a UAV, thanks to its simplicity and time efficiency. However, in this study, it has been identified that a good set of flight data is needed to find an accurate transfer function as a model. An accurate transfer function will give good PID gains that will allow the controller to perform as expected. The proposed suggestions to improve the quality of the flight data and the model are to increase the sampling rate and perform various flight tests, before and after tuning, in different conditions. The various flight tests will allow the modeling of a robust transfer function and PID controller, and will also validate the model and PID gains. Other additional future improvements would be to compare the PID from this method with the built-in PID in Ardupilot to evaluate whether results from this study can be implemented on the ALPHA UAV.

## Appendix
## Flight Logging

## Acknowledgment

The authors would like to thank Mykola Ivchenko for his support, valuable insights, and patience with our ever-changing focus. We would like to thank Erik Barsby and Casper Augustsson Savinov from group L4a for the help in understanding Ardupilot and for being a great team to work with on the flight tests. We would also like to thank Andreu Matoses Gimenez for giving advice and engaging in useful discussions that helped us in coming up with the method. Finally, we would like to thank group L5; and other members of ALPHA.

## References

[1] E. W. Frew, J. Elston, B. Argrow, A. Houston, and E. Rasmussen, "Sampling severe local storms and related phenomena: Using unmanned aircraft systems," *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 85–95, 2012.

[2] ALPHA. (2022, Jan.) Alpha. [Online]. Available: https://www.kthaero.com/alpha

[3] M. Ivchenko, *Requirements for the ALPHA UAV v1*. Internal KTH ALPHA document, KTH, Stockholm, Sweden, Nov. 2021.

[4] M. Schmekel and L. Ringaby, "Simulation and control system design for autonomous gliding to a given location," BSc thesis, KTH, Stockholm, Sweden, May 2021.

[5] N. V. Hoffer, "System identification of a small low-cost unmanned aerial vehicle using flight data from low-cost sensors," *All Graduate Theses and Dissertations.*, 2014. [Online]. Available: https://digitalcommons.usu.edu/etd/4274

[6] L. Ljung, T. Glad, and A. Hansson, *Modeling and identification of dynamic systems*, 2nd ed. Lund: Studentlitteratur AB, 2016.

[7] R. C. Nelson *et al.*, *Flight stability and automatic control*, 2nd ed. New York: WCB/McGraw Hill, 1998.

[8] R. Mariani, "Stability," Lecture notes in Aerodynamics, KTH, Stockholm, Sweden, Feb. 2021.
[9] HOOAH Aviation Technology CO., LTD. (2016, Dec.) Fpv survey clouds. Changzhou, China. [Online]. Available: http://www.x-uav.cn/en/content/?474.html
[10] C. Augustsson Savinov and E. Barsby, "Building a fixed wing autonomous uav," BSc thesis, KTH, Stockholm, Sweden, May 2022.
[11] J. Shen, Y. Su, Q. Liang, and X. Zhu, "Calculation and identification of the aerodynamic parameters for small-scaled fixed-wing uavs," *Sensors*, vol. 18, no. 1, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/1/206
[12] L. Ljung, *System identification theory for the user*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.
[13] The MathWorks Inc. (2022, Apr.) System identification toolbox. [Online]. Available: https://se.mathworks.com/products/sysid.html
[14] The MathWorks Inc. (2022, Apr.) Control system toolbox. [Online]. Available: https://se.mathworks.com/products/control.html
[15] A. J. Keane, A. Sóbester, and J. P. Scanlan, *Small unmanned fixed-wing aircraft design: a practical approach*, 1st ed. New York: John Wiley & Sons, 2017.
[16] A. Sobron, D. Lundström, R. Larsson, P. Krus, and C. Jouannet, "Methods for efficient flight testing and modelling of remotely piloted aircraft within visual line-of-sight," in *31st Congress of The International Council of the Aeronautical Sciences (ICAS)*, Belo Horizonte, Brazil, Aug. 2018.
[17] ArduPilot Dev Team. (2022, Feb.) Ardupilot. [Online]. Available: https://ardupilot.org/ardupilot/index.html
[18] ArduPilot Dev Team. (2022, Mar.) Mission Planner Home. [Online]. Available: https://ardupilot.org/planner/index.html#home
[19] ArduPilot Dev Team. (2022, Apr.) FBWA Mode (FLY BY WIRE_A). [Online]. Available: https://ardupilot.org/plane/docs/fbwa-mode.html
[20] The MathWorks Inc. (2022, Apr.) Compare. [Online]. Available: https://se.mathworks.com/help/ident/ref/compare.html
[21] A. E. Onuora, C. C. Mbaocha, P. C. Eze, and V. C. Uchegbu, "Unmanned aerial vehicle pitch optimization for fast response of elevator control system," *International Journal of Scientific Engineering and Science*, vol. 2, pp. 16–19, 2018. [Online]. Available: http://ijses.com/wp-content/uploads/2018/03/577-IJSES-V2N3.pdf