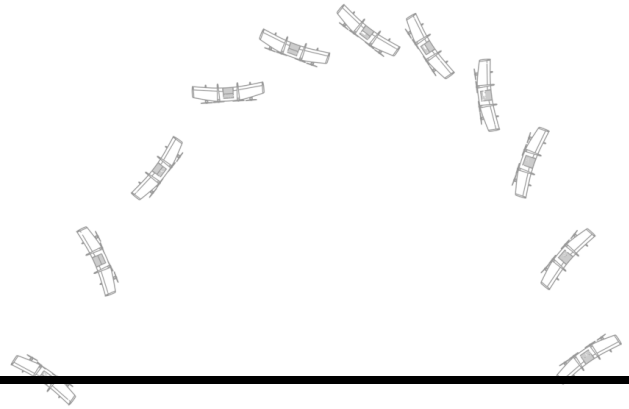
Master of Science Thesis

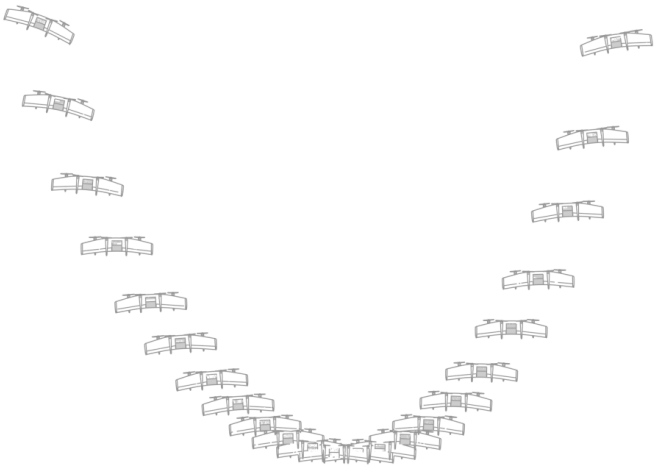


Aerobatic maneuvering of Autonomous Hybrid UAVs

Trajectory Tracking using INDI in the Control Frame

Koen Engelen

30 April 2023



Faculty of Aerospace Engineering · Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

Aerobatic maneuvering of Autonomous Hybrid UAVs

Trajectory Tracking using INDI in the Control Frame

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering at Delft University of Technology

Koen Engelen

30 April 2023

Faculty of Aerospace Engineering · Delft University of Technology

Copyright © Koen Engelen
All rights reserved.

Delft University Of Technology
Department Of
Control and Simulation

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"Aerobatic maneuvering of Autonomous Hybrid UAVs"** by **Koen Engelen** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 30 April 2023

Head of department: _____
dr.

Supervisor: _____
dr. E. Smeur

Reader: _____
ir.

Contents

Nomenclature

Abbreviations

CFD	Computational Fluid Dynamics
DOF	Degrees of Freedom
GNSS	Global navigation satellite system
IMU	Inertial measurement unit
INDI	Incremental Nonlinear Dynamic Inversion
NED	North-East-Down frame
PD	Proportional – Derivative (Controller)
PID	Proportional – Integral – Derivative (Controller)
RC	Radio Control
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing

Chapter 1

Introduction

Unmanned aerial vehicles (UAVs), such as quadrotors, are used in multiple applications varying from photography, package delivery, geographic mapping, crop monitoring, safety inspections, surveillance, reconnaissance, and many other applications [28]. With an increase in UAV applications, the mission profile of the UAV can vary based on the application itself. More tasks will require long endurance and extended range. Hybrid UAVs show promising features for these kind of applications as they can transition from hover to forward flight. This means they can be faster and more power efficient than quadrotors, while the ability to take off and land without a runway, remains.

The classifications of UAVs is broad and includes four main branches: multi-rotors, single-rotor helicopter, fixed-wing, and hybrid UAVs. The multi-rotor branch is well known for the quad-copter, one of the most recognizable UAVs as most commercially available drones fall in this category. This configuration provides good controllability and has the benefit that it can hover, take-off, and land vertically (VTOL). The single-rotor helicopter makes use of its main rotor to generate lift, while a secondary smaller rotor is used to control its heading. The helicopter has the same hover capabilities as the multi-rotors although mechanically more complex. The fixed-wing UAV is equipped with wings similar tho airplanes. The advantage of this type of UAV is the low power consumption of the vehicle. It is both able to maintain longer airtime as well as have a larger range. However, the downside is that it either needs to take-off on a runway or it needs assistance from contraptions such as a drone catapult. Hybrid UAVs combine fixed-wing with hover and VTOL capabilities, combining the advantages of both categories.

As mission profiles can vary between open areas and dense areas, such as urban and forest landscapes, the UAVs will require more dynamic trajectory control. With the emerging recreational use of UAVs, it has been shown that these aerial vehicles are able to perform extreme aerobatic maneuvers with the use of radio control (RC). From RC helicopters to fixed-wings and multi-rotors, performing aerobatic maneuvers with UAVs is difficult for pilots and requires practice as the UAV starts to behave non-linearly at extreme attitudes. Performing aerobatic maneuvers autonomously requires a method to tackle the control of the nonlinear behaviour of the aerodynamic surfaces. Regular PID control systems are

not designed to control these situations. The advantage of creating a controller capable of performing aerobatic maneuvers is that this will lead toward a more precise flight of autonomous UAVs. This will ultimately enable application of autonomous hybrid UAVs in dense environments.

Due to the challenges of maneuvering and the advantages of hybrid UAVs, the main goal of this work is to investigate suitable strategies to perform aerobatic maneuvers with autonomous hybrid UAVs. The scientific article, containing the methodology, results, and conclusion can be found in Part I. In Part II, the existing literature regarding this topic is discussed.

Part I

Scientific Article

Aerobatic Maneuvering of Autonomous Hybrid UAVs

Koen Engelen
TUDelft

Unmanned Aerial Vehicles (UAVs) are increasingly being used in various applications, which demand longer endurance, extended range, and high maneuverability. These requirements necessitate the development of effective control methods for Hybrid UAVs. In this paper, we propose an outer loop Incremental Nonlinear Dynamic Inversion (INDI) controller for Hybrid UAVs, based on an analytically derived control effectiveness to control the linear acceleration of the UAV. The control effectiveness is derived in a new frame that does not show singularities, technically allowing controlled flight at all attitudes. For trajectory tracking purposes, a Proportional Derivative (PD) controller is added. In simulation the proposed controller shows comparable results to already existing INDI controllers for hover and forward flight. When performing loop the loops it is shown that the proposed control system is able to handle high roll angles, while the already existing INDI controller crashed.

Nomenclature

Acronyms

ξ	=	position, $[m]$
C_D	=	Drag coefficient, $[-]$
C_L	=	Lift coefficient, $[-]$
G	=	Control effectiveness matrix
g	=	Gravitational acceleration, $[m/s^2]$
$GNSS$	=	Global navigation satellite system
IMU	=	Inertial measurement unit
$INDI$	=	Incremental nonlinear dynamic inversion
m	=	Mass, $[kg]$
NED	=	North-East-Down frame
PD	=	Proportional derivative
RC	=	Radio control
T	=	Thrust, $[N]$
UAV	=	Unmanned aerial vehicle
V	=	Airspeed, $[m/s]$
$VTOL$	=	Vertical take-off and landing

Subscripts

F_A	=	Aerodynamic force
F_G	=	Gravitational force
F_T	=	Thrust force
X_B	=	Body reference frame
X_C	=	Control reference frame
X_I	=	Inertial reference frame
X_W	=	Wind reference frame

I. Introduction

Unmanned aerial vehicles (UAVs), such as quadrotors, are used in multiple applications varying from photography, package delivery, geographic mapping, crop monitoring, safety inspections, surveillance, reconnaissance, and many other applications [1]. With an increase in UAV applications, the mission profile of the UAV can vary based on the application itself. More tasks will require long endurance and extended range. Hybrid UAVs show promising features for these kind of applications as they can transition from hover to forward flight. This means they can be faster and more power efficient than quadrotors, while the ability to take off and land without a runway, remains.

As mission profiles can vary between open areas and dense areas, such as urban and forest landscapes, the UAVs will require more dynamic trajectory control. With the emerging recreational use of UAVs, it has been shown that these aerial vehicles are able to perform extreme aerobatic maneuvers with the use of radio control (RC). From RC helicopters to fixed-wings and multi-rotors, performing aerobatic maneuvers with UAVs is difficult for pilots and requires practice as the UAV starts to behave non-linearly at extreme attitudes. Performing aerobatic maneuvers autonomously requires a method to tackle the control of the nonlinear behaviour of the aerodynamic surfaces. Regular PID control

systems are not designed to control these situations. The advantage of creating a controller capable of performing aerobatic maneuvers is that this will lead toward a more precise flight of autonomous UAVs. This will ultimately enable application of autonomous hybrid UAVs in dense environments.

Due to the challenges of maneuvering and the advantages of hybrid UAVs, the main goal of this work is to provide a suitable strategy for performing aerobatic maneuvers with autonomous hybrid UAVs.

Literature shows that acceleration control can be achieved with INDI for quadrotors [2]. For hybrid UAVs, this application shows singularities at roll angles of ±90 degrees [3]. This singularity occurs due to the mathematical derivation of the control derivatives at ±90 degrees roll, which are defined in the ZXY rotation order. Therefore, a controller is introduced that does not have singularities at different attitudes. Another approach provides a global controller for hybrid UAVs and shows that they are capable of performing maneuvers such as a vertical loop [4]. However, the control strategy is implemented in a way that maneuvers such as a knife edge are not possible as the UAV orients itself so that the desired acceleration is always perpendicular to the lateral axis.

This guidance controller differs from other controllers in that it makes use of a reference frame that is instanced from the body reference frame for determining the control effectiveness matrix of the controller. This reference frame, from hereon referred to as the control frame, has the advantage that the control effectiveness is not tied to the Euler angles that describe the orientation of the body frame with respect to the NED (North East Down) frame. As this controller is not reliant on the Euler angles, it is also not bound to the limitations induced by Euler angle singularities. Therefore, the UAV will be able to perform a maneuver that crosses these singularity points.

The main contribution of this article is a position control design that is capable of tracking trajectories at different velocity and attitude ranges. The controller relies on an approximate model of the aerodynamic forces that act on the UAV.

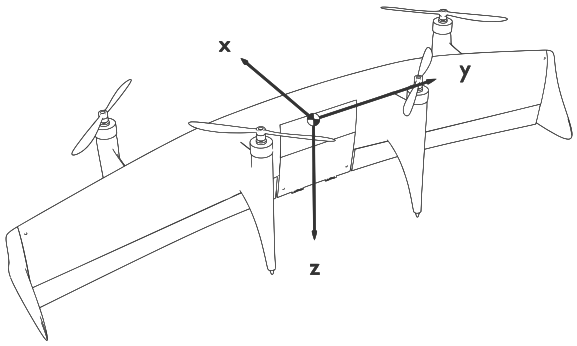


Figure 1. Representation of the quadshot with its body reference frame.

II. Methodology

In this section, the theory and implementation practices behind the proposed controller are discussed. First, the dynamical model and the derivation of the control law are described in subsection II.A. In subsection II.B the reference frames used throughout the paper are defined. Subsequently, the derivation of the control effectiveness is described in subsection II.C. Finally, the position controller is defined in subsection II.D.

The hybrid UAV used throughout this paper is shown in Fig. 1. The vectors in the figure are the axes of the body reference frame. The quadshot has a main wing with two elevons on the trailing edge. These elevons are actuated with two servo motors, allowing for creating both pitch and yaw moments with respect to the body frame. Additionally, the quadshot has 4 propellers similar to a quadrotor.

A. Incremental Nonlinear Dynamic Inversion

The advantages of INDI over conventional NDI includes the decreased knowledge required about the model and the improved robustness compared to NDI [5]. The difference between these methods is that INDI relies on incrementally changing the inputs to get to desired state based on the derivatives of the dynamic model, while NDI sets the inputs to the required values based on the dynamic model itself. Here, the increments are influenced by the inversion of the dynamic system so that a change in input can be expressed as a desired change in state. In the previous literature, several derivations for this linear acceleration control law were made using a Taylor

series expansion [6] and based on the estimation of external forces [2]. In this paper, we derive the control law through the Taylor series approximation of linear acceleration.

For this controller, the states to be controlled are the linear accelerations. The UAV is able to apply forces on itself by using its aerodynamic surfaces and its propellers, therefore, the translational dynamics of this system can be written as:

$$\ddot{\boldsymbol{\xi}} = \mathbf{g} + \frac{1}{m}\mathbf{F}_A(\boldsymbol{\eta}, \dot{\boldsymbol{\xi}}, \boldsymbol{w}) + \frac{1}{m}\mathbf{F}_T(\boldsymbol{\eta}, T) \tag{1}$$

Note that in Eq. 1 the reference frame is not specified. The equation remains valid for any inertial reference frame as long as the vectors change accordingly. Here, $\ddot{\boldsymbol{\xi}}$ represents the acceleration vector of the UAV. $\mathbf{g}$ and m represent the gravitational acceleration vector and the mass, respectively. The vector $\mathbf{F}_A$ represents the sum of all aerodynamic forces that work on the aerodynamic surfaces of the UAV, from which the propellers are excluded. This force is dependent on the attitude of the UAV $\boldsymbol{\eta}$ as well as the velocity $\dot{\boldsymbol{\xi}}$ and the wind velocity $\boldsymbol{w}$. The remaining term $\mathbf{F}_T$ is the acceleration vector caused by the thrust of all propellers. This acceleration depends on the attitude of the UAV and the magnitude of the thrust T. It should be noted that attitude η is the orientation difference between the body reference frame and the chosen inertial reference frame. Before elaborating on the reference frame in subsection II.B, the control law is derived by taking a first-order Taylor series expansion of Eq. 1, resulting in Eq. 2.

$$\begin{aligned}
\ddot{\boldsymbol{\xi}} = \mathbf{g} &+ \frac{1}{m}\mathbf{F}_A(\boldsymbol{\eta}_0, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}_0) + \frac{1}{m}\mathbf{F}_T(\boldsymbol{\eta}_0, T_0) \\
&+ \frac{\partial}{\partial \boldsymbol{\eta}}\frac{1}{m}\mathbf{F}_A\left(\boldsymbol{\eta}, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}_0\right)|_{\boldsymbol{\eta}=\boldsymbol{\eta}_0}(\boldsymbol{\eta} - \boldsymbol{\eta}_0) \\
&+ \frac{\partial}{\partial \dot{\boldsymbol{\xi}}}\frac{1}{m}\mathbf{F}_A\left(\boldsymbol{\eta}_0, \dot{\boldsymbol{\xi}}, \boldsymbol{w}_0\right)|_{\dot{\boldsymbol{\xi}}=\dot{\boldsymbol{\xi}}_0}(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_0) \\
&+ \frac{\partial}{\partial \boldsymbol{w}}\frac{1}{m}\mathbf{F}_A\left(\boldsymbol{\eta}_0, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}\right)|_{\boldsymbol{w}=\boldsymbol{w}_0}(\boldsymbol{w} - \boldsymbol{w}_0) \\
&+ \frac{\partial}{\partial \boldsymbol{\eta}}\frac{1}{m}\mathbf{F}_T\left(\boldsymbol{\eta}, T_0\right)|_{\boldsymbol{\eta}=\boldsymbol{\eta}_0}(\boldsymbol{\eta} - \boldsymbol{\eta}_0) \\
&+ \frac{\partial}{\partial T}\frac{1}{m}\mathbf{F}_T\left(\boldsymbol{\eta}_0, T\right)|_{T=T_0}(T - T_0)
\end{aligned} \tag{2}$$

By design, the Taylor series expansion creates an approximation of a function based on the derivatives of that function around a given point. In Eq. 1 that point is represented by the first three terms, which represents the initial acceleration $\ddot{\boldsymbol{\xi}}_0$. For the partial derivative terms of $\mathbf{F}_A$ that are derived with respect to the velocity $\dot{\mathbf{x}}$ and the wind velocity $\boldsymbol{w}$ it is difficult to estimate the effect these variables have on the aerodynamic forces. This means that the effect of these terms will be measured as disturbances in the accelerometer and will be accounted for in that manner. These terms are therefore set to zero to simplify the equation. The effect of these forces is therefore not considered in the controller and has to be countered by the incremental nature of INDI. Additionally, it should be noted that thrust is not modeled as a function of wind velocity or body velocity. In reality, the thrust is influenced by the wind velocity, but this requires wind-tunnel measurements of the propellers operating at different throttle levels and different freestream velocities. Therefore, the remaining partial derivatives are based solely on the attitude $\boldsymbol{\eta}$ and the magnitude of the thrust T. These two terms can be combined in the input vector $\mathbf{u} = [\boldsymbol{\eta}\ T]^T = [\phi\ \theta\ \psi\ T]^T$ as the attitude can be defined with roll, pitch, and yaw with respect to the reference frame. This results in the following simplified expansion:

$$\ddot{\boldsymbol{\xi}} = \ddot{\boldsymbol{\xi}}_0 + G(\boldsymbol{\eta}_0, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}_0, T_0)(\mathbf{u} - \mathbf{u}_0) \tag{3}$$

with $G(\boldsymbol{\eta}, \dot{\boldsymbol{\xi}}, \boldsymbol{w}, T)$:

$$G = \frac{1}{m}\begin{bmatrix} \frac{\partial}{\partial \phi}\left(\mathbf{F}_A(\phi, \theta_0, \psi_0, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}_0) + \mathbf{F}_T(\phi, \theta_0, \psi_0, T_0)\right)|_{\phi=\phi_0} \\ \frac{\partial}{\partial \theta}\left(\mathbf{F}_A(\phi_0, \theta, \psi_0, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}_0) + \mathbf{F}_T(\phi_0, \theta, \psi_0, T_0)\right)|_{\theta=\theta_0} \\ \frac{\partial}{\partial \psi}\left(\mathbf{F}_A(\phi_0, \theta_0, \psi, \dot{\boldsymbol{\xi}}_0, \boldsymbol{w}_0) + \mathbf{F}_T(\phi_0, \theta_0, \psi, T_0)\right)|_{\psi=\psi_0} \\ \frac{\partial}{\partial T}\mathbf{F}_T(\phi_0, \theta_0, \psi_0, T)|_{=_0} \end{bmatrix}^T \tag{4}$$

The final control law can be obtained by inverting Eq. 3, which results in Eq. 5.

$$\mathbf{u}_c = \mathbf{u}_f + G^+(\boldsymbol{\eta}_f, \dot{\boldsymbol{\xi}}_f, \boldsymbol{w}_f, T_f)(\boldsymbol{\nu} - \ddot{\boldsymbol{\xi}}_f) \tag{5}$$

Here G^+ is the pseudoinverse of the matrix G as it is a non-invertible matrix of size $(3x4)$. From hereon control inputs $\mathbf{u}_c$ can be computed based on the current control inputs $\mathbf{u}_f$, the control effectiveness G, the current acceleration $\ddot{\boldsymbol{\xi}}_f$ and the virtual control vector $\boldsymbol{\nu}$, which is the desired acceleration. The Control Effectiveness is elaborated further in subsection II.C with the reference frames specified in subsection II.B.

B. Reference Frames

In subsection II.A, we have seen that the control input can be obtained with the control effectiveness. This is a matrix containing the partial derivatives of the forces acting on the UAV with respect to the attitude and the thrust magnitude. It is necessary to select a reference frame from which these partial derivatives are obtained.

In this section, an overview of the reference frames used in this paper are discussed as well as the motivation behind the selection of the reference frame for determining the control effectiveness.

Fig. 1 shows an illustration of the Quadshot with the definition of its body reference frame, the first reference frame used in this paper. This reference frame is denoted by the subscript B. The second reference frame used is the North-East-Down (NED) reference frame, denoted by the subscript N.

If the NED reference frame is used to determine the control effectiveness, the Euler rotation sequence determines how the attitude of the drone is represented. As one can derive from the rotation sequences, only two of the three angles can be expressed with ranges defined with modulo 2π radians. The remaining angle only covers a range of π. Usually the former angles are ϕ and ψ, while the latter is θ. Specifically, when θ is $\pm\frac{\pi}{2}$ or $-\pm\frac{\pi}{2}$, the rotations around the ϕ and ψ become parallel, and the resulting orientation can be obtained by a single rotation around the combined axis. This means that there are multiple ways to describe the same orientation using different combinations of ϕ and ψ, which can lead to inconsistencies in computations. As the control effectiveness depends on the partial derivatives of these angles, it can be seen that singularities can occur at the point where the angle θ reaches $\pm\frac{\pi}{2}$. The exact location of this singularity is dependent on the rotation sequence of the attitude. In the case of the conventional ZYX rotation sequence, the singularity would occur at a pitch angle of $\pm\frac{\pi}{2}$ radians. For a hybrid UAV, this is highly inconvenient as the transition from hover to forward flight includes this pitch angle. Other sequences can be chosen, but the most useful is the ZXY sequence. This is due to the singularity occurring at a roll angle of $\pm\frac{\pi}{2}$ radians. For hybrid UAVs, this sequence is best as it supports the yaw and pitch with the cyclically defined ranges. Although this is the best sequence for the given scenario, it is just a method to move the underlying issue from one axis to another. To solve the underlying issue, a method should be used that does not create these singularities in the control effectiveness.

If non-inertial reference frames are used for the control effectiveness estimation such as the body reference frame, the virtual control vector $\boldsymbol{\nu}$ would also change along with the attitude. This makes non-inertial reference frames unsuitable for the control law described in subsection II.A. Similarly, the aerodynamic frame is not suitable, as the virtual control would change with a change in attitude.

Due to the disadvantages of the previously mentioned reference frames, a new reference frame is defined, from now on called the control reference frame, C. This reference frame is inertial, similar to the NED reference frame. The difference is that this reference frame is temporarily aligned with the body reference frame to determine the control effectiveness as illustrated in Fig. 2.

This results in the use of the control reference frame. It does not rotate with the UAV and is inertial. However, as mentioned originally, it is aligned with the body reference frame at the moment that control effectiveness needs to be determined. In subsection II.C the control effectiveness is determined based on this reference frame.

C. Control Effectiveness

Moving forward, an analytical derivation of the forces acting on the drone is performed. The control effectiveness is obtained by calculating the partial derivatives of these forces.

For any rotation, the rotation matrix of the body reference frame to the control frame is as follows:

$$\mathbb{T}_{CB} = \begin{bmatrix} c\theta c\psi - s\phi s\theta s\psi & -c\phi s\psi & s\theta c\psi + s\phi c\theta s\psi \\ c\theta s\psi + s\phi s\theta c\psi & c\phi c\psi & s\theta s\psi - s\phi c\theta c\psi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \tag{6}$$

Here $s\theta$ and $c\theta$ represent the sine and cosine of the angle θ respectively.

1. Thrust Force

Due to the fixed rotor configuration, the thrust force is always aligned in the opposite direction of the

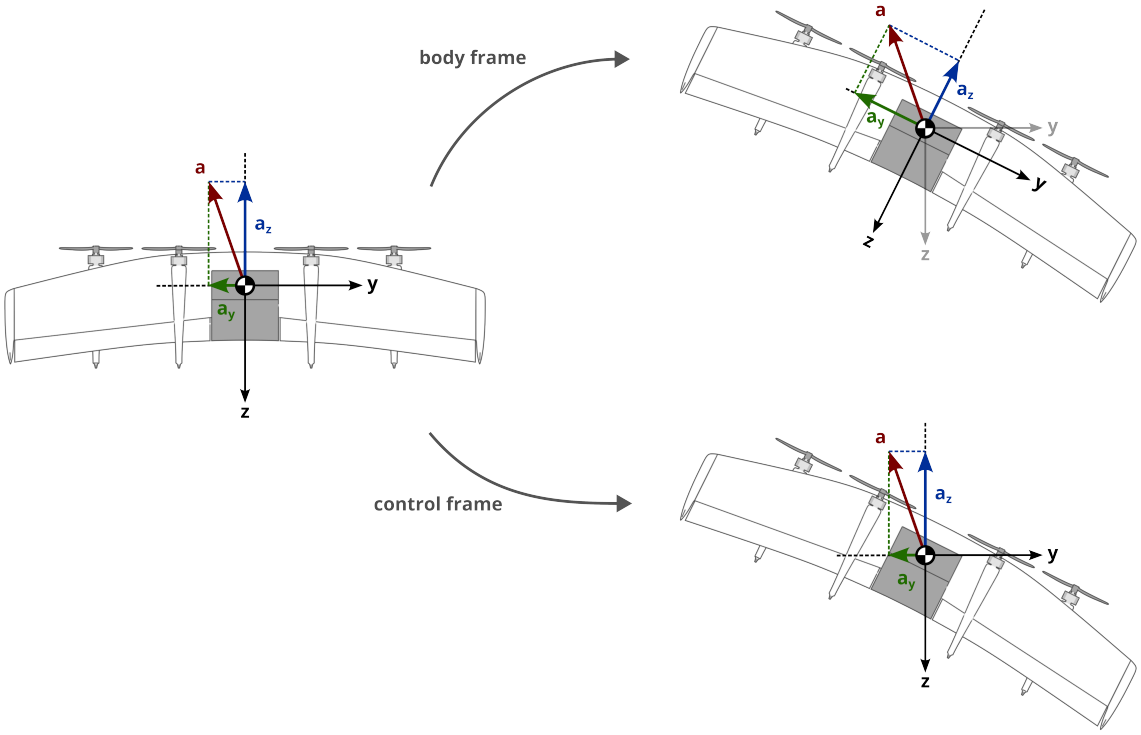


Figure 2. Difference between the control and body reference frame for a virtual displacement.

body z-axis. With this, the force vector can be easily expressed in the body frame:

$$\mathbf{F}_{T,B} = -\begin{bmatrix} 0 & 0 & T \end{bmatrix}^T \tag{7}$$

Transforming the thrust vector from the body to the control frame results in Eq. 8.

$$\mathbf{F}_{T,C} = \mathbb{T}_{CB}\mathbf{F}_{T,B} = -\begin{bmatrix} (s\theta c\psi + s\phi c\theta s\psi)T \\ (s\theta s\psi - s\phi c\theta c\psi)T \\ c\phi c\theta T \end{bmatrix} \tag{8}$$

Finally, the partial derivatives of the thrust force in the control frame can be used to obtain the control effectiveness as previously shown in Eq. 4. The partial derivatives are equated at the moment that the control frame is aligned with the body frame. This results in ϕ_0, θ_0, ψ_0 being equal to 0. The results of these derivations leads to the control effectiveness in Eq. 9.

$$\mathbf{G}_T(T) = \frac{1}{m}\begin{bmatrix} 0 & T & 0 & 0 \\ -T & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{9}$$

2. Aerodynamic Forces

The forces caused by the aerodynamic surfaces are best represented in the aerodynamic reference frame. The force $\mathbf{F}_A$ can be divided into the lift forces and drag forces of the wing. For the control effectiveness, partial derivatives of the aerodynamic forces are required with respect to the changes in attitude in the control frame. The forces themselves are defined in the wind reference frame. To express these forces in the control frame, the following transformations are applied:

$$\mathbb{T}_{CW} = \mathbb{T}_{CB}\mathbb{T}_{BW} \tag{10}$$

$$\mathbb{T}_{BA} = \begin{bmatrix} s\alpha & 0 & c\alpha \\ 0 & 1 & 0 \\ c\alpha & 0 & -s\alpha \end{bmatrix} \tag{11}$$

$$\mathbb{T}_{AW} = \begin{bmatrix} c\beta & -s\beta & 0 \\ s\beta & c\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

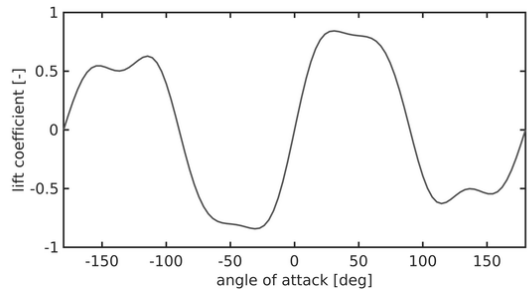


Figure 3. Lift coefficient approximation for a given angle of attack.

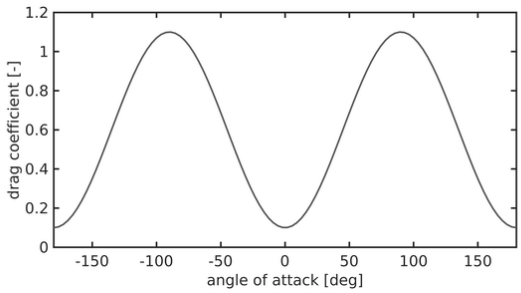


Figure 4. Drag coefficient approximation for a given angle of attack.

The aerodynamic forces that act on the UAV in the wind frame are shown in Eq. 13.

$$\mathbf{F}_{A,W} = -\begin{bmatrix} D \\ Q \\ L \end{bmatrix} \tag{13}$$

In the wind frame, D is the drag of the quadshot, which is parallel to the free-stream velocity. L is the lift caused by the wing and is defined as the force perpendicular to the free-stream velocity and the body y axis y_B. Q is the remainder force caused by the aerodynamics of the quadshot. The latter term will be dominated by the effects of the pylons and winglets of the quadshot. Since the effect of this is difficult to estimate, the Q term is neglected. This means that, in this model, lateral acceleration only occurs due to drag at higher angles of sideslip.

The lift and drag coefficients are both approximations based on aerodynamic surfaces with similar Reynolds numbers [7, 8]. These approximations can be seen in Fig. 3 and Fig. 4. The forces are then determined with the lift and drag equation shown in Eq. 14 and Eq. 15 respectively. The main focus of this approximation is that the lift coefficient has no local minimum near its stall point as this could result in the incremental controller settling on that minimum while stalling.

$$L = \frac{1}{2} C_L(\alpha) \rho {V_{xz}}^2 S \tag{14}$$

$$D = \frac{1}{2} C_D(\alpha) \rho V^2 S(\beta) \tag{15}$$

Since the control effectiveness requires the partial derivatives along different axes separately, the transformation matrices can be applied for rotations around the x, y, and z axes depending on the required partial derivative. As only one rotation is required for each partial derivative, the angle sequence is irrelevant. With this, the control effectiveness can be constructed as shown in Eq. 16. Eq. 17 shows the control effectiveness with the added row for sideslip control. The difference in sideslip angle can then be chosen to reduce the sideslip. For practical and efficient flights, minimizing the angle of sideslip is preferred. However, implementing this might require some additional changes to the control strategy. This is because the matrix becomes non-invertible if the angle of side slip can not be influenced by the control input and the partial derivatives are zero. To solve this, an additional step could be taken to evaluate the rank of the matrix before inverting it. If the rank is lower than the dimensions of the matrix itself, the matrix can be reduced before inverting it by removing the angle of sideslip. Although maneuvers with zero angle of sideslip and 90 degrees roll might be possible, these were not chosen to demonstrate this controller. Therefore, this control effectiveness matrix will not be used in the remainder of the paper as minimizing the sideslip prevents the UAV from performing the examined maneuvers discussed in section III. The latter option, Eq. 18, is used throughout the remainder of the paper. It should be noted that with the removed yaw control, the control effectiveness matrix becomes square and an inverse of this matrix exists. The same applies for the sideslip control.

$$\begin{bmatrix} \Delta a_{X_B} \\ \Delta a_{Y_B} \\ \Delta a_{Z_B} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{X_B}}{\partial \phi} & \frac{\partial a_{X_B}}{\partial \theta} & \frac{\partial a_{X_B}}{\partial \psi} & \frac{\partial a_{X_B}}{\partial T} \\ \frac{\partial a_{Y_B}}{\partial \phi} & \frac{\partial a_{Y_B}}{\partial \theta} & \frac{\partial a_{Y_B}}{\partial \psi} & \frac{\partial a_{Y_B}}{\partial T} \\ \frac{\partial a_{Z_B}}{\partial \phi} & \frac{\partial a_{Z_B}}{\partial \theta} & \frac{\partial a_{Z_B}}{\partial \psi} & \frac{\partial a_{Z_B}}{\partial T} \end{bmatrix} \begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta \psi \\ \Delta T \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \Delta a_{X_B} \\ \Delta a_{Y_B} \\ \Delta a_{Z_B} \\ \Delta \beta_i \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{X_B}}{\partial \phi} & \frac{\partial a_{X_B}}{\partial \theta} & \frac{\partial a_{X_B}}{\partial \psi} & \frac{\partial a_{X_B}}{\partial T} \\ \frac{\partial a_{Y_B}}{\partial \phi} & \frac{\partial a_{Y_B}}{\partial \theta} & \frac{\partial a_{Y_B}}{\partial \psi} & \frac{\partial a_{Y_B}}{\partial T} \\ \frac{\partial a_{Z_B}}{\partial \phi} & \frac{\partial a_{Z_B}}{\partial \theta} & \frac{\partial a_{Z_B}}{\partial \psi} & \frac{\partial a_{Z_B}}{\partial T} \\ \frac{\partial \beta_i}{\partial \phi} & \frac{\partial \beta_i}{\partial \theta} & \frac{\partial \beta_i}{\partial \psi} & \frac{\partial \beta_i}{\partial T} \end{bmatrix} \begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta \psi \\ \Delta T \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} \Delta a_{X_B} \\ \Delta a_{Y_B} \\ \Delta a_{Z_B} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{X_B}}{\partial \phi} & \frac{\partial a_{X_B}}{\partial \theta} & \frac{\partial a_{X_B}}{\partial T} \\ \frac{\partial a_{Y_B}}{\partial \phi} & \frac{\partial a_{Y_B}}{\partial \theta} & \frac{\partial a_{Y_B}}{\partial T} \\ \frac{\partial a_{Z_B}}{\partial \phi} & \frac{\partial a_{Z_B}}{\partial \theta} & \frac{\partial a_{Z_B}}{\partial T} \end{bmatrix} \begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta T \end{bmatrix} \tag{18}$$

D. Position Control

The trajectory is tracked by a simple PD controller that tracks the required acceleration based on the position and velocity error and a feed forward acceleration defined by the trajectory, as can be seen in Eq. 19.

$$\ddot{\boldsymbol{\xi}}_{com} = a_{ref} + K_d \left(\dot{\boldsymbol{\xi}}_{ref} + K_p \left(\boldsymbol{\xi}_{ref} - \boldsymbol{\xi} \right) - \dot{\boldsymbol{\xi}} \right) \tag{19}$$

This required acceleration is then used as a reference by the outer loop INDI controller. Based on this acceleration, the controller increases the reference attitude and thrust.

The INDI-based attitude controller utilized in this control system is not covered in this paper and was obtained through [9].

To obtain an initial estimate of the gains K_p and K_d, a linearized version of the control loop is used. This system is linearized around its hovering equilibrium. Due to the difference in the thrust dynamics and the angular rates, the thrust dynamics are omitted to design the gains. This is due to the fast response time of the thrust dynamics compared with the rotational dynamics. These simplifications result in the control loop in Fig. 5. As thrust dynamics is neglected, the gains can only be tuned for the transfer function of the pitch on the position tracking along the x direction and the roll on the position tracking on the y direction.

This linearized model was used to analyse the stability of the control system around its hover state. A pole zero plot was used to look at the poles of the transfer function and determine the gains so that the poles were in the open left half plane. These gains were used as initial values for the quadshot's indoor flights and were further tuned manually based on the quadshot's response to step inputs in all three directions. The pole zero plot of the selected gains can be seen in Fig. 7.

E. Instrumentation and State Estimation

For this paper, the controller was tested in different environments. These environments are: simulation, indoor, and outdoor. For each of these environments, different instrumentation was chosen to obtain the required measurements. For the indoor scenario, motion capture data is used to provide the quadshot with accurate position and attitude data. The acceleration and angular rates is obtained directly from the inertial measurement unit (IMU). As indoor there is practically no wind, instrumentation to measure wind velocity and direction were not equipped. For the outdoor scenario, Global Navigation Satellite System (GNSS) data are used to determine the position of the quadshot and the IMU is used for its attitude and acceleration. It should be noted that the unfiltered position data is directly used in the outer loop. Additionally the outdoor setup also makes use of an added pitot tube installed in front of the wing together with an angle of attack sensor. This angle sensor consists of a wind vane attached to a rotary encoder. The angle of sideslip is estimated through the lateral acceleration of the IMU. This is done by estimating the lateral wind velocity using Eq. 20, where a_y is the specific acceleration along the body Y axis. From here, the lateral wind velocity is combined with the wind measurement to obtain the angle of sideslip. In case of the simulation, the simulation data of the previously mentioned sensors was provided to the quadshot as is.

$$V_y \propto \sqrt{\|a_y\|} \cdot \frac{a_y}{\|a_y\|} \tag{20}$$

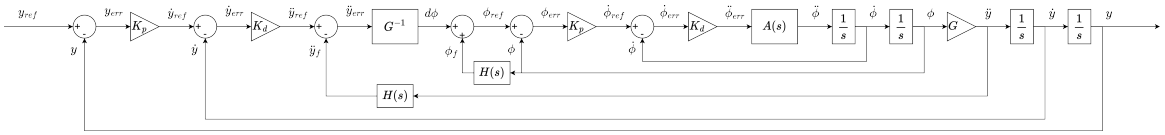


Figure 5. Linearized closed loop approximation.

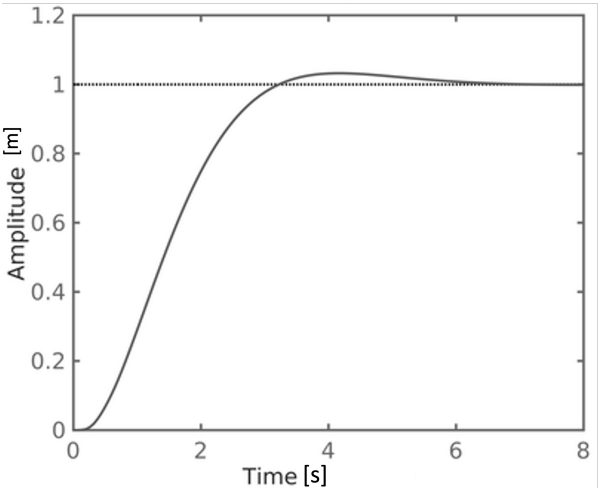


Figure 6. Step response of the linearized model.

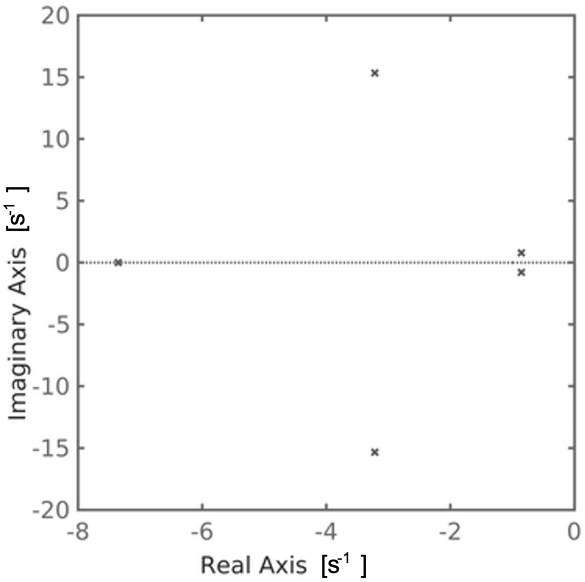


Figure 7. Pole zero plot of the linearized model.

III. Results & Discussion

Within this section, the results of the experiments are shown. For the simulation, the results are compared with an outer loop INDI controller based on [6, 10] from hereon called the baseline controller. To compare the baseline with the proposed controller, the same inner loop and and velocity controllers are used. The control effectiveness is based on the partial derivatives of the roll, pitch and thrust with respect to the accelerations in the NED reference frame. Since the baseline is based on a quadrotor and not a hybrid UAV, a comparison between the proposed and the baseline controller is made without the aerodynamic effects. Only when specified, the controller is simulated with aerodynamic forces and the control effectiveness is based on both thrust and aerodynamics.

The outer loop PD gains were initially estimated based on the linearized model and were fine-tuned in the cyberzoo, an indoor environment from the TUDelft available for UAV experiments. This indoor space is equipped with a motion capture system, that can provide feedback of the exact position and orientation of the UAV. The resulting values for K_p and K_d are 0.7 and 1.5, respectively. These gains are kept similar for both the proposed controller and the baseline controller.

Furthermore, the results are split in three categories; simulation, indoor, and outdoor results. Similar trajectories are selected for each category, as this approach can aid in extracting insights from the variations among these instances. The simulation has very little limitations as there are no unknown aerodynamic effects. The indoor and outdoor experiments both have a different set of trajectories as the indoor space is not large enough to perform large trajectories. Due to the differences between these scenarios, some trajectories will not be tested across all of them. In Table 1 an overview of the gathered results for every scenario is shown.

	Linearized	Simulation	Indoor	Outdoor
Step response	x	x	x	
Horizontal ellipse		x		x
Vertical ellipse		x		

Table 1. Scenarios that are tested for comparison.

The comparison consists out of three trajectories; namely the response to a step input, a horizontal ellipse and a vertical ellipse. The step response was chosen as it shows properties of the system behavior, such as the responsiveness and dampening. The horizontal ellipse was chosen as it is simple to define mathematically. The acceleration can be easily changed throughout the trajectory. Since the acceleration throughout the trajectory varies, it can be used to evaluate the general performance of both controllers. Lastly, the vertical ellipse is specifically designed to encounter the singularities around the ±90 degrees roll angle to evaluate the difference between the outer loop controllers.

Finally, all figures are presented in the NED frame unless otherwise specified.

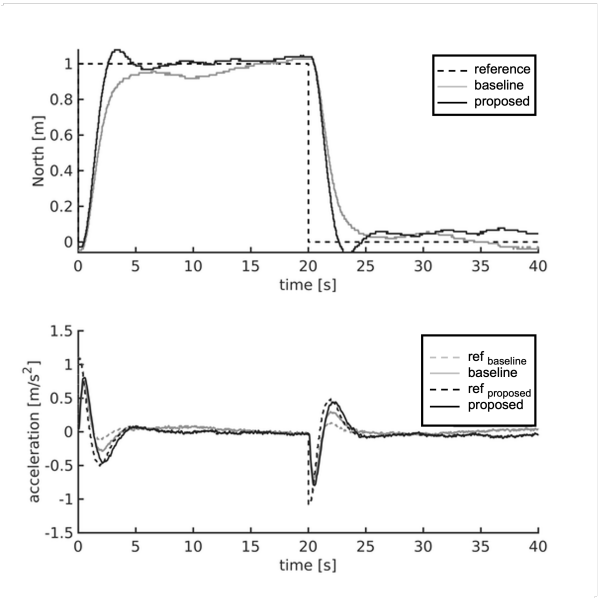


Figure 8. comparison on step response in simulation.

A. Step Response

For the first test, the response of the linearized closed loop approximation to a step input is shown in Fig. 6. A simulation of the step response for both controllers is shown in Fig. 8. It can be seen that both the proposed controller and the baseline controller converge to the reference position. With equivalent gains, the proposed controller displays a marginally quicker rise time and greater overshoot in comparison to the baseline controller. This disparity stems from the fact that, despite identical PD gains and inner loop gains, the outer loop control effectiveness differs. This discrepancy arises due to these values being obtained from separate reference frames, resulting in distinct behaviors. Both controllers track their own reference acceleration similarly for the given step response. However, the baseline method has some overshoot in the deceleration, which prevents the overshoot from occurring.

The controllers' response to a 2 meter step input in the cyberzoo can be seen in figure 9. It can be seen that the response to this input is similar to the simulation and linearized closed loop, with the exception that the cyberzoo has a step of 2 meter instead of 1 meter.

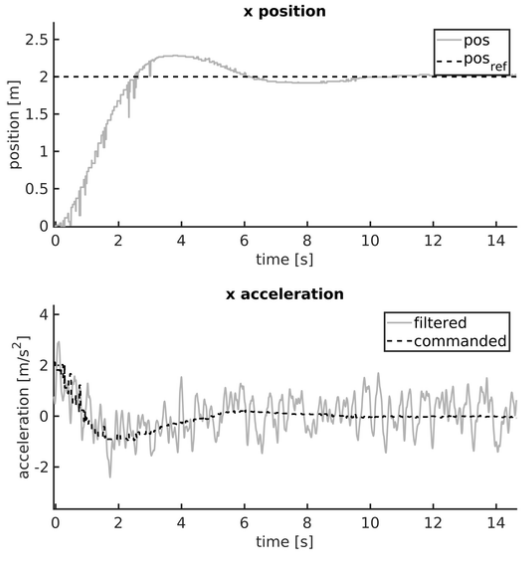


Figure 9. Position and acceleration of the quadshot reacting on a 2m step response in the cyberzoo.

B. Ellipse

The second test is an elliptical trajectory. The simulation results can be seen in Fig. 10 and Fig. 11, showing the northern and eastern tracking profile, respectively. An overview of the trajectory can be seen in Fig. 12. The dissimilarities between the baseline method and the proposed method are minimal, since both controllers exhibit comparable overshoots when operating at high speeds. This indicates that both controllers offer a similar level of performance.

This trajectory was also flown outside as shown in Fig. 13. Once the trajectory is started, the quadshot slowly starts to oscillate in the height. This caused the trajectory to be aborted as the height was changing too rapidly. The exact cause of these oscillations are not found and are not emerging in the cyberzoo. The main difference was that the cyberzoo uses optitrack to position the quadshot, while outside GNSS data were used. With GNSS the altitude is generally less accurate compared to the longitude and latitude. With this difference it might be useful to see if reduced height gains can fix this issue. During further testing and tuning outdoor, the oscillations were not mitigated.

In addition, the quadshot was only able to measure the freestream velocity accurately at low angles of attack. While in hover, the drone can't get an accurate measurement as the pitot tube is oriented perpendicular to the freestream velocity. Similarly, the angle of attack sensor requires airflow, hence the reading will not be accurate if there is a large angle of sideslip or there is not enough airflow to overcome the friction of the sensor. Also, the sideslip based on IMU data only provides a coarse estimate of the angle. Maneuvers that involve sideslip might improve with an actual angle of sideslip sensor.

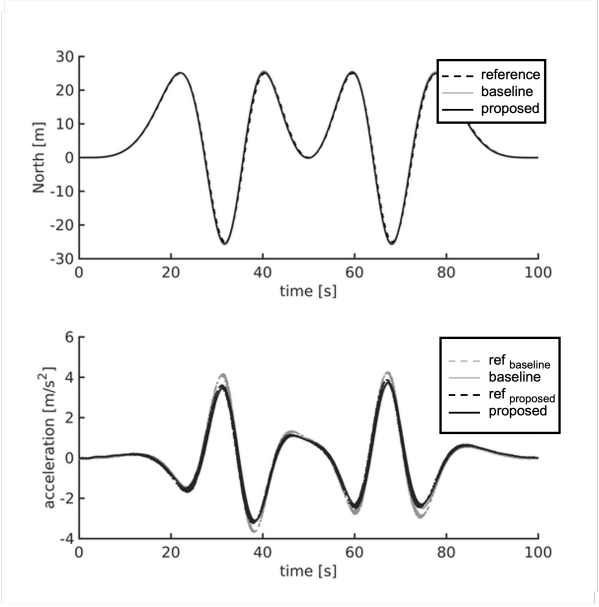


Figure 10. comparison on ellipse northern coordinates.

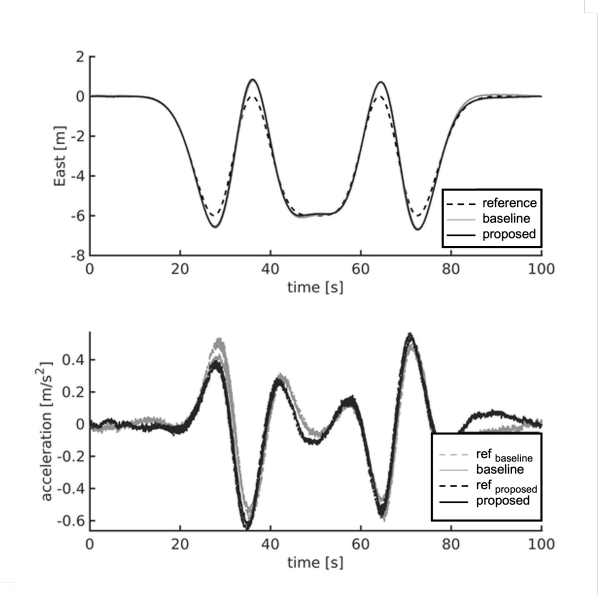


Figure 11. comparison on ellipse eastern coordinates.

C. Vertical Ellipse

The last test was set up specifically to demonstrate the advantage of the proposed controller. For this, a vertically oriented ellipse was used. The trajectory's velocities were chosen in such a way that the acceleration profile would require the UAV to accelerate downwards at the apogee of the trajectory. The tracking of this trajectory can be seen in Fig. 14 and Fig. 15 for the altitude and eastern location tracking, respectively. A side view of the trajectory is shown in Fig. 16. From the altitude data, it can be seen

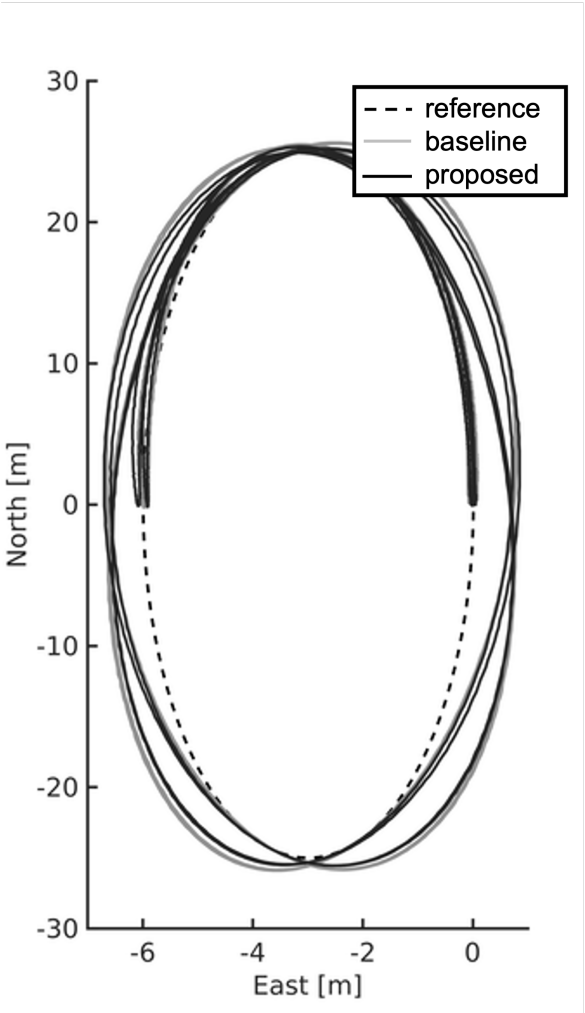


Figure 12. comparison on ellipse overview.

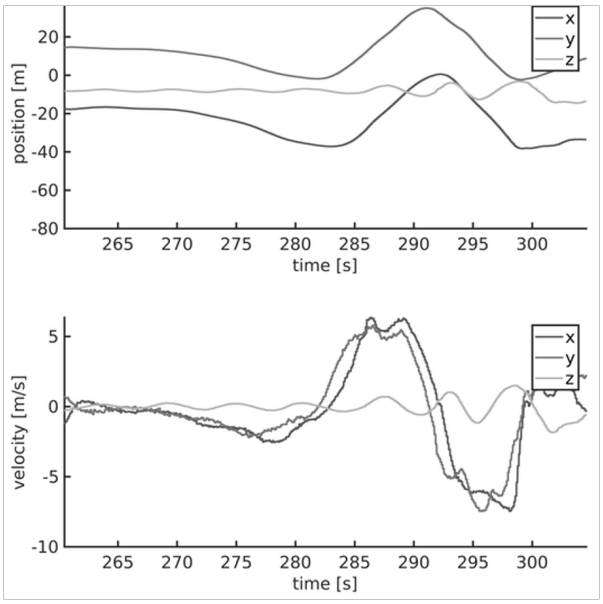


Figure 13. start of outdoor ellipse.

that the UAV starts accelerating upward. It should be noted that the trajectory is not initialised at the position of the quadshot and both controllers have to catch up with the reference. Near the apogee, the controllers have to accelerate downward and it can be seen that the downward acceleration of the baseline controllers saturates at $9.81\frac{m}{s^2}$, which means that it is in a free fall. However, this is not enough to meet the required acceleration and the baseline method overshoots the trajectory, creating a free fall parabola. The proposed method, however, has rolled 180^o allowing its thrust to accelerate the UAV downward. This was repeated several times with the new controller, however, the baseline controller crashed after the second apogee. A visualisation of the UAVs attitude during this maneuver using the proposed controller is shown in Fig. 17.

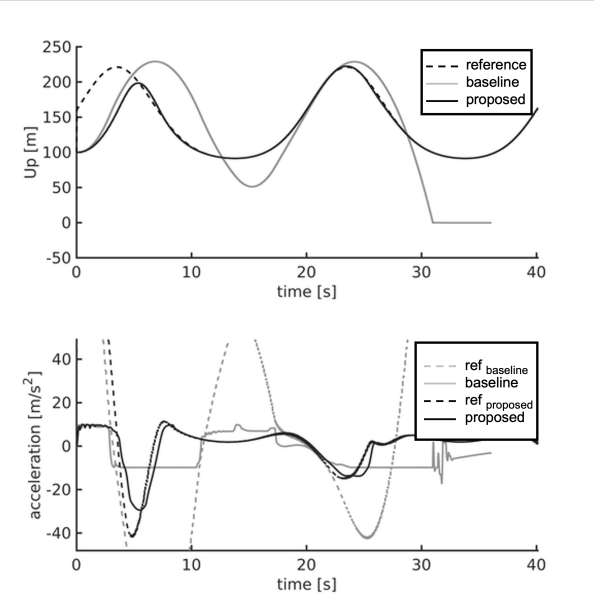


Figure 14. comparison on vertical ellipse altitude.

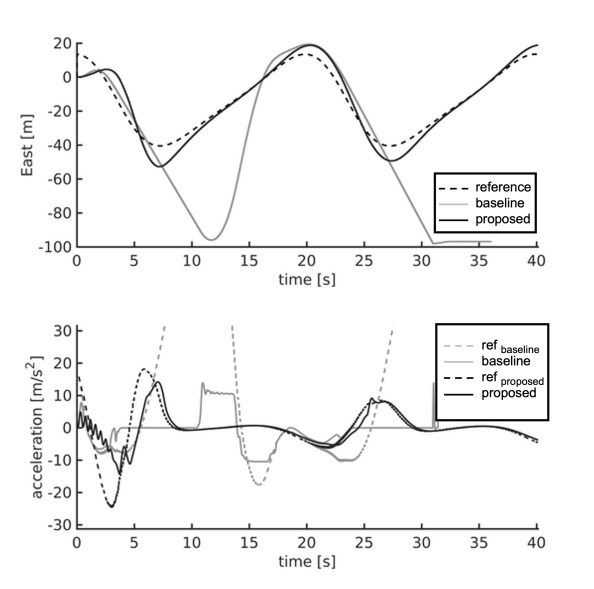


Figure 15. comparison on vertical ellipse eastern coordinates.

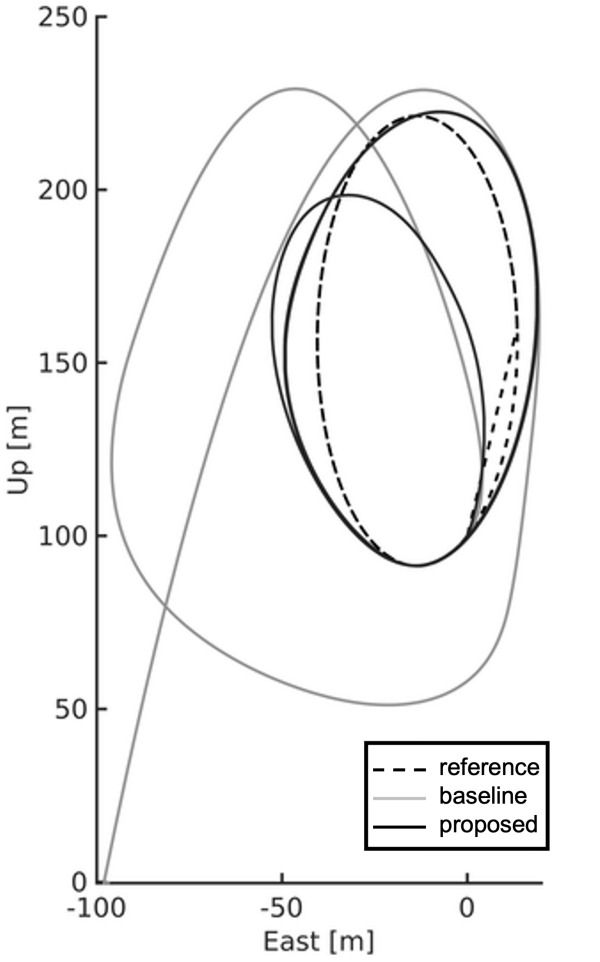


Figure 16. comparison on vertical ellipse overview.

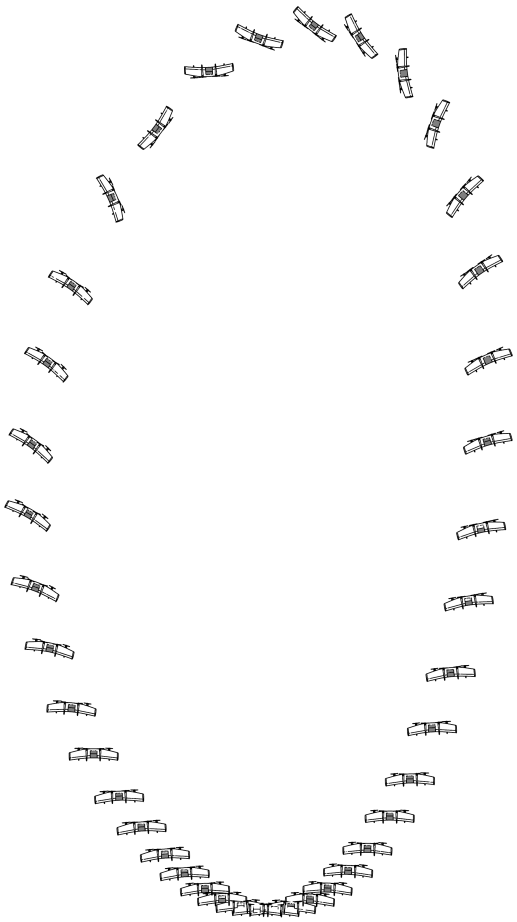


Figure 17. illustration of the vertical ellipse looping performed by the proposed controller. Quadshot is scaled up for visibility.

IV. Conclusion

In this paper, the control of a hybrid UAV has been demonstrated by using an outer loop Incremental Nonlinear Dynamic Inversion (INDI) controller that makes use of the control reference frame.

From the vertical ellipse, it can be seen that the proposed controller, using the control reference frame, has the additional benefit that it can track accelerations around the ±90 degrees roll angles. The control effectiveness of the thrust component proves to be simple to implement on the quadshot and could easily be implemented on other quadrotors. The only estimations required for the thrust are the actuator dynamics and the magnitude for a given thrust command. A linear fit of the thrust with respect to the thrust command is sufficient to track accelerations inside the cyberzoo. From simulation it is concluded that the proposed controller is able to maneuver past singularity while the baseline controller could not. Besides this advantage, both controllers have similar behaviours.

The contribution of the aerodynamics on the control effectiveness are more complicated. Equations that determine the forces' partial derivatives with respect to changes in attitude rely on the lift and drag coefficients relative to the angle of attack and angle of sideslip. Through practical experimentation, it was observed that the chosen sensors used for measuring airspeed had limited accuracy while in the hover state. Taking steps to rectify this issue could lead to improved control accuracy. Due to the incremental nature of INDI, the controller is less dependent on the parameters of the model and will incrementally correct for a mismatch between the model and the physical behavior. However, this is not confirmed outdoor. This due to oscillations in the altitude that kept the quadshot from tracking its acceleration. While the control effectiveness was tested in the cyberzoo as well, the cyberzoo test flight did not show these height oscillations. Due to the lack of wind in the cyberzoo, this means that the control effectiveness of the aerodynamics has not been verified in the field.

Future research could further examine the cause of the outdoor oscillations in height, as this could contribute to a better analysis of the accuracy of the aerodynamic control effectiveness. Though the cause is not determined, suggested improvements for this includes the use of a more extensive filter to obtain position data or more accurate separate gain tuning

in the altitude axis. Alternatively, onboard adaptive control effectiveness estimations such as least mean squares could be used to verify the control frame outdoor without the dependencies on the proposed control effectiveness.

References

[1] Zhang, R., Zhang, J., and Yu, H., "Review of modeling and control in UAV autonomous maneuvering flight," *IEEE International Conference on Mechatronics and Automation*, 2018. URL `https://ieeexplore.ieee.org/document/8484542/`, doi: 10.1109/ICMA.2018.8484542.

[2] Tal, E., and Karaman, S., "Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness," *IEEE Conference on Decision and Control*, 2018.

[3] Smeur, E., Bronz, M., and de Croon, G., "Incremental Control and Guidance of Hybrid Aircraft Applied to a Tailsitter UAV," *Journal of Guidance, Control, and Dynamics*, 2020. Doi: 10.2514/1.G004520.

[4] Ritz, R., and D'Andrea, R., "A global controller for flying wing tailsitter vehicles," *IEEE International Conference on Robotics and Automation*, 2017. URL `http://ieeexplore.ieee.org/document/7989318/`, doi: 10.1109/ICRA.2017.7989318.

[5] Wang, X., Van Kampen, E.-J., Chu, Q., and Lu, P., "Incremental Sliding-Mode Fault-Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 42, 2018, pp. 1–16. doi: 10.2514/1.G003497.

[6] Smeur, E. J., de Croon, G. C., and Chu, Q., "Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection," *Control Engineering Practice*, 2018. Doi: 10.1016/j.conengprac.2018.01.003.

[7] Ma, Z., Smeur, E., and Croon, G., "Wind tunnel tests of a wing at all angles of attack," *International Journal of Micro Air Vehicles*, Vol. 14, 2022. doi: 10.1177/17568293221110931.

[8] De Wagter, C., Remes, B., Smeur, E., van Tienen, F., Ruijsink, R., van Hecke, K., and van der Horst, E., "The NederDrone: A hybrid lift, hybrid energy hydrogen UAV," *Int. J. Hydrogen Energy*, Vol. 46, No. 29, 2021, pp. 16003–16018.

[9] Smeur, E. J., Chu, Q., and de Croon, G. C., "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles," *Journal of Guidance, Control, and Dynamics*, 2016. Doi: 10.2514/1.G001490.

[10] Smeur, E. J., de Croon, G. C., and Chu, Q., "Gust disturbance alleviation with Incremental Nonlinear Dynamic Inversion," *International Conference on Intelligent Robots and Systems (IROS)*, 2016. Doi: 10.1109/IROS.2016.7759827.

Part II

Literature Study

Chapter 1

Unmanned Aerial Vehicles

Over the last decade, the use of drones has increased drastically in every aspect due to the reduction in both the cost and size of processors and sensors. The design of these drones can vary depending on its use [8], but overall the most recognizable drone configuration is the quadcopter. The quadcopter is the most popular drone in the current consumer market due to its mechanical simplicity. The configuration allows for simple control and reduced cost compared to more complex mechanical systems such as the helicopter. For these advantages, several disadvantages can be thought of as well. The downside is the efficiency of the quadcopter compared to other UAVs such as fixed-wing UAVs. These types of UAVs are used when range or endurance is important for the mission of the UAV. Several applications of these types of UAVs exist today (border control). However, these UAVs are lesser-known due to their limited market share in the recreational sector. The disadvantage of these drones is that they often require a take-off lane and landing lane, which are not always available. Alternatives to take-off and landing lanes do exist but these also increase the operational cost such as catapults for take-off and catching nets for "landing".

Combining the best of both worlds, Hybrid UAVs provide a solution that allows UAVs to take-off and land vertically while being able to fly with a fixed-wing. This combines the ease of use of the quadrotor with the extended range and endurance and speed of a fixed-wing. From a historical point of view, one might notice that the advantages of VTOL aircraft were known decades ago. This can be seen by the aircraft that were developed in this same trend, i.e. the Harrier, the F-35B, and the V-22 Osprey. The concept of hybrid UAVs can be implemented in multiple ways leading to different designs. The amount of different designs for hybrid UAVs is staggering and will only increase over time. In [15], an overview of the current types of hybrid UAVs is given. These different designs offer different advantages and disadvantages such as cost, complexity, and efficiency. However, there are additional complications that are introduced with the control of hybrid UAVs. The main reason why hybrid UAVs are difficult to control is due to their large flight envelope.

A large flight envelope has a negative effect on the modeling of the system's behavior. Modeling the exact forces and moments of the aerodynamics for the entire flight envelope (including stall) is time-consuming and expensive, as it requires either wind tunnel tests or computed fluid dynamics (CFD). Besides that, a detailed model also requires detailed sensors that can measure all states, This might not always be possible as some sensor inputs such as the angle of attack sensor become inaccurate at low airspeed. Furthermore, the flight envelope causes large non-linearities in the lift, which makes it in turn more difficult to use the lift together with the thrust to obtain the desired acceleration. Finally, a large flight envelope leads towards an abundance of solutions to the guidance problem. If a reference trajectory was to be followed by the UAV it could do this in forward flight or in near hover mode. With both, these approached the UAV would have been able to follow the trajectory. However, the most suitable approach might depend on usage. This means that forward flight would be used for fast delivery and extended range, while the hover approach could be useful for achieving smaller positional error margins with respect to the reference trajectory, provided there is no external disturbance.

In section 1.1, the selected UAV model will be discussed. After that, the degree of under-actuation of the model is discussed in section 1.2. Finally, the importance of the attitude in aerobatics is discussed in section 1.3.

1.1 Quadshot

During the research, a quadshot [17] will be used to test the suggested method as these hybrid UAVs are readily available at the TU Delft for research purposes. This hybrid UAV model can be seen in figure 1.1. The quadshot has four rotors pointing towards the negative z-axis. It can be seen that the rotors on the positive x-axis are further apart in the y-axis compared to the other rotors. The main wing has two elevons that can produce a roll and a yaw moment if there is airflow over these surfaces. The UAV has two wing-lets and four landing rods which are shaped parallel with the airspeed in horizontal flight.

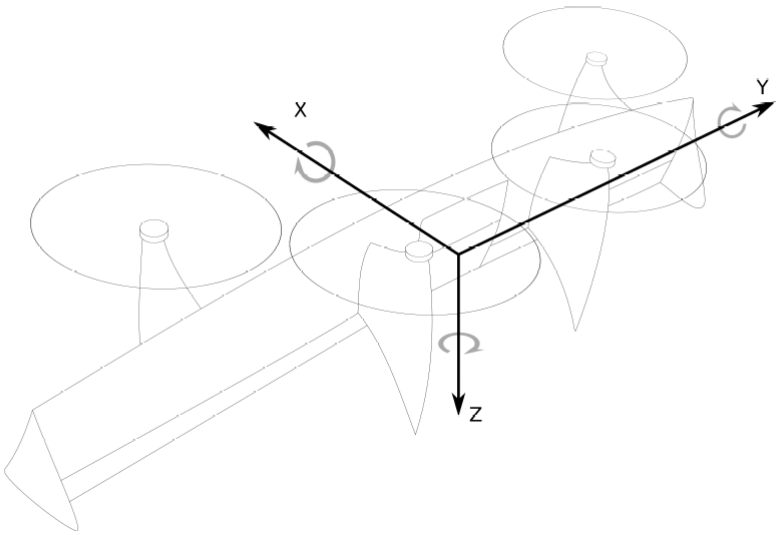


Figure 1.1: The quadshot UAV with its body axes from [19].

1.2 Under-Actuated Systems

Although the designs of the different types of UAVs are significantly different from each other, most of them share a common system property, i.e. under-actuation. The under-actuated nature is caused by the layout of these UAVs. For all rigid body UAVs, the UAVs have six degrees of freedom (DOF), three due to the translations, and three due to rotations in three-dimensional space. A system is under-actuated as the inputs can not control all degrees of freedom indidually. THis can be determined by the rank of the input matrix.

The quadshot is has a quadrotor configuration combined with a fixed-wing and two elevons. The elevons operate as both ailerons, elevators, and trailing edge flap, thus

having two separate inputs able to control yaw, roll, and lift. However, part of these actuators fulfill similar purposes as the rotors, although with different effectiveness. The Lift caused by the fixed-wing of UAVs acts in a different direction compared to the thrust and the hybrid UAVs therefor have one additional degree of freedom as the lift can be influenced by the elevons. This additional input is strongly dependent on the air velocity, angle of attack, and elevon deflections, but can be controlled.

Since there are 6 inputs and 6 degrees of freedom, the UAV can be fully actuated. However, the sideways acceleration of the quadshot can't be influenced by the inputs, meaning that there is a zero row in the input matrix. This in turn means that the rank of the input matrix has a maximum value of 5 since the last column has to be a linear combination of the previous inputs columns. The input matrix contains non-linear functions as the evelons can create lift depending on the velocity squared. This means that if the velocity over the wings is zero, the lift is zero and can not be controlled, leaving the input matrix with a rank of 4 similar to quadcopters. This means that in flight the UAV is only able to choose 5 states independently, while the 6th state is dependent on the previously defined states. An example of this is for forward horizontal flight, the three velocity components are defined. Additionally one could increase the roll angle of the hybrid UAV (comparable with pitch of fixed-wings) in order to maintain level flight, the UAV can use its elevons to decrease the lift. Similarly the yaw angle of the hybrid UAV (comparable with roll of fixed-wings) can be adjusted. This requires the final state of the quadshot, the pitch (comparable to yaw of fixed-wings), to change in order to maintain level flight.

The under-actuated nature of these UAVs doesn't prevent them from flying their desired path. With the control over the attitude and the thrust pointing in one direction, these UAVs are capable of reaching any desired location over time. From this point of view, it can be seen that attitude control plays a major role in maneuvering UAVs.

1.3 Attitude in Aerobatics

As discussed in 1.2, attitude can be controlled to achieve the desired accelerations. For a given path, the acceleration to track the path can be determined. However, this means that the attitude should allow for the accelerations to occur in the desired direction. This makes the attitude an important aspect of controlling UAVs. Although it might seem obvious that UAVs require to control their attitude for simple purposes such as hovering for quadrotors, aerobatic maneuvers impose additional difficulties. This is due to the non-linear effects within the aerodynamic model that are caused by extreme attitudes and high angle of attacks. Within regular applications of quadrotors, it is common that the pitch and roll only deviate slightly from their reference conditions. From this condition, the system is usually linearized and regular PID controllers are applied.

Besides the difficulties concerning the non-linearity, in aerobatics, the use of unusual attitudes can lead towards trajectories that might be solvable in a trivial manner if only the position is used as an input. An example of this is the knife-edge, where the aircraft's trajectory is preferably a straight line. During this maneuver, the fixed-wing UAVs' roll angle (or hybrid UAVs' yaw angle) will increase until its wings are perpendicular to the

horizon. The trivial solution for this trajectory would be to fly straight ahead and prevent the roll from happening in the first place, but that does not meet the aerobatics requirement. One could try to alter the trajectory to mimic a knife-edge over straight flight, or one could require the roll angle of the fixed-wing to be controlled separately. For non-aerobatic maneuvers with fixed-wing, the attitude is usually controlled so that the UAV maintains coordinated flight, and the side-slip is reduced. In comparison, coordinated flight is more efficient, as no side-slip increases the aerodynamic performance. Maneuvers such as the knife-edge, however, cannot be executed without side-slip and are generally executed when efficiency is not the main priority. Therefore, it is required to diverge from the coordinated flight attitudes to perform specific aerobatic maneuvers.

Another maneuver a UAV could do is a looping. Here the goal is to rotate the UAV 360 degrees along its pitch axis while traversing along a circular path. If the UAV has a low velocity or the maneuver's duration is too high, it could hover its way along the path. In this case, the time aspect influences the UAVs performance. This is since the acceleration required to traverse a path is dependent on the speed with which the UAV traverses the desired path. Moreover, since the weight, lift, drag, and thrust cause the UAV's acceleration, the velocity impacts the UAV's attitude along the path. Therefore, it is required that the path contains additional information about the traversal time or velocity. Because if the path is designed without this, there is no way to design the path so that attitude will force the UAV to pitch 360 degrees. Different approaches to this are discussed in section 2.1

Chapter 2

Maneuvering Control Systems

Different control and trajectory planning methods that are able to perform maneuvers autonomously already exist. Although most of these control methods are used on quadcopters, helicopters, or fixed wings, these methods provide a good reference for the possible control methods of hybrid UAVs. In a recent paper [28], several state-of-the-art methods to model and control maneuvering flights for UAVs are compared. The approaches of the various methods diverge significantly as some of the research papers use traditional control methods including H-infinity or INDI, while other methods make use of neural network control or iterative learning. The latter methods, artificial intelligence, and iterative learning are discussed in section 3. In this section, the more traditional control methods are discussed together with the control architecture. In section 2.1, the control architecture of several papers are discussed. After that, the control methods are discussed in section 2.2. Finally, some implementations of these approaches are discussed in section 2.3. Note, for section 2.1 and section 2.3, only references were used of UAVs that were performing aerobatic maneuvers. Papers discussing nonlinear controllers were excluded from these sections and can be found in section 2.2.

2.1 Control Architecture

All types of UAVs used as a reference for this work suffer from under-actuated dynamics, irrespective from their designs. The overall resemblance between the different control architectures [24, 13, 4, 5] is shown in figure 2.1. Within this figure, the maneuvering generator passes the reference position and time, x_{ref} and t_{ref} respectively. the position controller passes the desired attitude q_{ref} to the attitude controller, which in turn sends the desired thrust T_{des} to the thrust controller. The desired thrust can be a single input or multiple inputs depending on the UAV type. It has to be noted that this is resemblance of all the control subsystems to get an idea of the workflow and not an actual representation of the subsystem. In reality, each of these subsystems has its feedback loops with additional subsystems to control the UAV.

Path/Trajectory Reference

Initially, a reference path or trajectory is created. This reference path or trajectory provides information to the position controller on the reference position. The main difference between the reference path and trajectory is that the path only stores three-dimensional constraints (position in three-dimensional space), while the trajectory stores a four-dimensional reference (position in three dimensions combined with a time reference). This means that the trajectory has a time reference associated with each point on the desired three-dimensional path. A generated trajectory has the benefit that the three-dimensional path can be easily be expressed as a function of time.

Designs that use a trajectory instead of a path have additional benefits as the position derivatives with respect to time can be calculated to find a reference velocity. This can also be done for higher-order derivatives such as acceleration, jerk, and snap. This means that the trajectory reference can provide more than just a reference position, i.e., it can also return the reference accelerations over the entire trajectory. For this reason, trajectories can provide more insight into the maneuvers of the UAV compared to reference paths.

Paths are sometimes used instead of trajectories for control architectures where the UAV's desired velocity is constant [13]. This velocity could also be used to obtain an estimated time of completion. However, that information is separated from the path itself.

Alternatively to both references mentioned previously, vector field guidance is a method that uses a vector field in three-dimensional space containing the desired velocities. This method is significantly different as the UAV will look at the velocity vector corresponding to its location. This vector field is designed around a path, and it is created to ensure that the vector field will converge the UAV towards the desired path irrespective of its location. Similar to the path following methods, this method has a reference velocity and a path. The path itself does not hold any time information, and the UAV itself will not use the path as a reference as this required information to follow the path is already included in the reference velocity.

The overall difference between path following and trajectory tracking is that path-following methods such as vector field guidance allow the UAV to follow the path more accurately. On the other hand, trajectory tracking methods have the benefit that UAVs with time delays will catch up because it is chasing a virtual target point that moves through space based on the time parameter. This has both advantages and disadvantages as the drone will try to catch up with the target point at the cost of possibly cutting off corners and, thus, leaving the desired path. This can be incredibly disadvantageous as the time delay is significantly high, and the path was carefully constructed around obstacles.

Position Controller

Within this section, the controllers in-between the trajectory, and the desired attitude are discussed. From the papers listed [24, 13, 4, 5], the common controller is the position controller. This controller usually translates the positional offset and the velocity offset to a specific acceleration. The specific acceleration is then used to determine the desired attitude. In [4, 5], the fixed-wing UAV has no acceleration controller and the position controller passes the desired attitude directly to the attitude controller. This approach is

based on the physical models discussed in [27]. For this to work, reference moments were already computed in the maneuver generator.

Attitude Controller

The attitude controller is responsible for obtaining the desired attitude by using the control surface deflections in [4, 5], while other methods make use of additional controllers due to the fact the attitude is determined by the rotors. This will eventually lead to a force distribution between the four propellers.

Thrust Controller

The thrust controller is responsible for obtaining the desired thrust for each propeller. There are many similarities between the different approaches [24, 13, 4, 5], all of them resulting in a single-input-single-output (SISO) system.

2.2 Control Methods for Nonlinear Systems

Apart from looking at the overall architecture, it is important to look at the controllers that can be used for these purposes. As mentioned before, the UAV will encounter nonlinear behavior at uncommon attitudes and high angles of attack. These nonlinearities occur from aerodynamic properties, gyroscopic affects and trigonometric relationships such as gravity. If these nonlinear behaviors of the system are neglected, this could result in unpredictable stability and a decrease in overall performance [6]. Therefore, the controller should be able to handle these nonlinearities. This is often done through a model-based approach that is created to cancel the nonlinear effect with a feed-forward component. The downside of model-based methods is that they require exact models of the nonlinear behavior, which is nearly impossible and increase costs. Approaches such as model reference adaptive control (MRAC) [29] and L1 adaptive control (L1-AV) [9] have been developed to estimate the model parameters. This is done so that the controller is less dependent on model behavior and can adapt to model uncertainties and disturbances. However, these methods are dependent on the adaptation rate. With low adaptation rates, the UAV might be unable to adapt to the new flight conditions. However, with high adaptation rates, the UAV might incorporate disturbances in the model. Therefore, adaptive control might impose disadvantages, especially in aerobatics where the changes of flight conditions can be rapid. In this section, controllers that cope with these nonlinearities are discussed.

Gain-Scheduling

Although all aircraft have nonlinearities in their dynamics model, linear control methods have been used for the flight controller of UAVs. These methods often linearize the nonlinear model around a flight condition within their operational flight envelope. The most common flight control systems use PID controller techniques to maintain the desired

outputs. The advantage of this is the known stability and robustness of the controller. However, this only holds for small deviations around that trim point. To further extend the operational range, gain-scheduling can be used. This method uses multiple trim points and interpolates between them to obtain the gains of the controller. With these trimmed operation points a linear control design method is used to derive a controller. One of the downsides is the fact that the controller needs to be linearized around several trim points depending on the range of operation, which requires accurate knowledge of the model. The main disadvantage of this approach is the overall performance and robustness, which can't be guaranteed with this control method as the nonlinear model needs to be linearized [12].

Nonlinear Dynamic Inversion

An alternative method, which does not require transition between multiple linearized operation points is Nonlinear Dynamic Inversion (NDI). This is a frequently used technique for multivariable control systems (MIMO) [25]. Through the feedback loops of the system, NDI is capable of partially linearizing the dynamics of the closed-loop system [26].

The disadvantage of NDI is that to obtain explicit cancellation of the nonlinear behavior, the method requires accurate knowledge of the nonlinear dynamics. As discussed at the beginning of section 1, these accurate models are difficult to obtain as they require wind tunnel tests, which are costly and need to be redone for every different UAV configuration. Besides this, model errors and parametric uncertainties introduce additional errors that prevent accurate estimation of the exact state of the model and make cancellation nearly impossible. Due to this main drawback, several control techniques have been designed to improve the robustness of NDI.

Incremental Nonlinear Dynamic Inversion (INDI) is a control technique based on NDI, which is capable of preserving the advantages of NDI while reducing the dependency on the dynamics model [25]. The dependency on the model is reduced since the INDI control method is based on sensor data. However, knowledge is required on the control effectiveness of the system's control inputs, which tells how much change in force or moment is the result of a change in control input. This is adjusted incrementally until the desired acceleration is obtained [18]. Even-though the control effectiveness needs to be known, an addition to INDI called adaptive INDI exists which estimates the control effectiveness of the UAV online [20]. As discussed in [26], INDI enhances the system's robustness toward model uncertainties and has already been implemented on several aerospace systems.

In [3], INDI has been used to perform transitioning maneuvers with a hybrid UAV. In [21] and [22], it is shown that this method outperforms traditional PID controllers based on position error when gust perturbations are introduced. Furthermore, the update rate of the position data shows that this approach should be capable of performing similar tasks outside with a standard GPS module.

Backstepping

In [12], a back-stepping approach was used to cope with nonlinearities of their fixed-wing UAV. According to [28], the convergence of back-stepping is better than traditional NDI. However, no direct comparison between back-stepping and INDI was found.

One of the advantages of back-stepping is that this method is able to achieve global asymptotic stability. The approach makes use of a recursive structure in which the errors of subsystems are reduced until the external control is stabilized and the recursive procedure is terminated. For this method to work, a Lyapunov function at each step has to be selected. The selection of this Lyapunov candidate function is important as an appropriate candidate function can reduce the complexity of the controller design process [11]. The disadvantage of this method is the increase in complexity as the process is required to repeat the differentiation of the virtual controls of all systems.

Similar to INDI discussed in section 2.2, sensor-based backstepping controllers have been developed which rely on sensor data instead of knowledge of the nonlinear model. Additionally, approaches are discussed in [16] to approximate the unknown nonlinear function with neural network identification models. With this method, the adaptive gains are updated online based on the results of the identification.

Sliding Mode

The final control method discussed in this section is sliding mode control. In [23], this method is discussed in order to autonomously dock an aircraft for mid-air refueling.

The sliding mode control is a form of discontinuous control known as variable structure control (VSC). The approach of this method is to switch between two structures to control the states towards the opposite region. Once the states have crossed the region it will change its control structure and converge towards the first region. This will result in an effective sliding of the states over a manifold representing the system's operational behavior. While the states are sliding on this manifold, the system high order system will be reduced to a lower order system.

The advantages of this control method are that it responds fast to external perturbations and has a low sensitivity towards model uncertainties. The main disadvantage of this method is that can induce high-frequency oscillations in the system. This phenomenon is called chattering and can lead to instability of the system and damage to the system itself.

2.3 Implementations

Aggressive Trajectory Tracking

In [24], An accurate quadcopter control method for aggressive trajectory tracking is discussed that makes use of INDI and differential flatness. This method goes further as it computes the acceleration, jerk and snap of the entire trajectory. In the paper it is shown how, with differential flatness, the trajectory can be used to derive the desired attitude rate and attitude acceleration. After that the linear and angular acceleration is tracked with INDI. Given that jerk and snap are required in this process, it is required that the trajectory itself is four times differentiable. This method allows for a yaw rate to be selected independent of the trajectory itself as the drone is symmetrical.

With this method, high accelerations and low tracking errors are achieved. The additional forces acting on the drone are compensated rapidly with the INDI implementation. Several aspects of this research might be interesting for the application on hybrid UAVs. Initially, it should be reflected if the model can be adjusted in order to account for the different configuration of the hybrid UAV. Besides that, the application of a variable yaw rate could be looked into. With a changing yaw rate and a controller, it is possible to control the yaw itself. However, the dynamics of a hybrid UAV and a quadcopter can vary at higher yaw rates as the aerodynamic surfaces will behave differently when subjected to rotational rates. Due to the 4 times differentiable trajectory requirement, it is possible to create a trajectory with sinusoidal functions. Furthermore, one might look into the trajectory generation of higher order polynomial splines.

Relaxed Roll Constraint

Another application uses a three-dimensional path following algorithm with a relaxed roll constraint. [13] This method uses the acceleration of a fixed wing UAV to follow a trajectory. This makes sure that the UAV is always accelerating in the desired direction. The trajectory following algorithm is designed in a way that the acceleration is always perpendicular to the velocity vector. This is done in order to maintain a constant velocity, The roll angle of this UAV is separated from the control loop and can therefore be controlled freely. Once the roll angle is adjusted, the UAV will try to obtain the desired acceleration with the new attitude. An example of this is that the UAV will try to compensate the weight of the UAV with lift of the rudder and thruster instead of its wings if the roll angle is 90 degrees. This approach has shown its capability of performing several maneuvers with a fixed wing UAV including knife-edge flight and the split-S maneuver. This approach has not been tested on hybrid UAVs and seems promising towards these types of maneuvers. The downside of this approach is that the trajectory following algorithm is used to maintain constant airspeed. It is necessary to look into the implications of changing the constant airspeed into a control parameter.

Agile Fixed-Wing UAVs

The approach discussed in [4] and [5] provides the capabilities of aerobatic maneuvers by applying simple control techniques to a control system that is based upon a physical model of the fixed-wing UAV. The physical model used to simulate the motion of the UAV consist out of a six degrees of freedom dynamics model that uses rigid body dynamics equations in combination with accurate techniques [27] to model aerodynamics, thruster dynamics and propeller slipstream effects.

Within this aerodynamics model, the wing surfaces of the aircraft are divided in smaller segments which are analysed. These wing surfaces can each produce their own lift, drag and moment around its aerodynamic center. Due to the agile maneuvering of the UAV, the aerodynamics can become complex. For this reason the aerodynamic model accounts for the entire flight envelope, partial flow over the wing surfaces of the UAV, large control surface deflections, and unsteady aerodynamic effects.

Similar to the aerodynamics, the thruster dynamics also have a great influence on the accuracy of the overall model. This thruster model makes use of the rotational velocity

of the propeller and the airflow in order to obtain the moments and forces delivered by the thruster. The airflow has influence on the performance and therefore the model was made in order to account for various situations including hover, axial flow, oblique flow, and reverse flow. Besides this, the gyroscopic effects and the effects of the batteries and brush-less motor on the thruster dynamics are included.

Finally, the aerodynamic model takes into consideration the effects of the propeller slip-stream on the aerodynamics. This effect causes an increased flow over the aerodynamic surfaces behind the propellers. This accounts for the axial velocity and the swirl velocity caused by the thruster.

Due to the four control inputs and the six degrees of freedom to be controlled, this system is under-actuated. Due to the physical properties of the fixed-wing, the control surfaces are used to control the attitude and the attitude is used to control the position. This cascaded control structure allows the tuning process to be simplified and modular (i.e. the attitude, position and thrust controller can be tuned separately from each other in that specific order).

With the control strategy discussed in this paper the fixed-wing was capable of performing the aerobatic maneuvers including: knife-edge, rolling harrier, hover, aggressive turnaround and transition between these maneuvers. The maneuvers were performed in simulation [4] and in flight [5]. The aerodynamic model of this approach seems expandable towards hybrid flight.

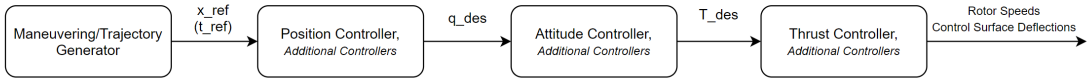


Figure 2.1: An simplified architecture of a maneuvering control system.

Chapter 3

Artificial Learning Systems

Vastly different from the methods mentioned in section 2 are the methods that are capable of training themselves. These learning based approaches to perform maneuvers show promising results and the applications of these methods are covering most fields of the UAV branch, i.e. fixed-wings, VTOLs such as quadcopters [14, 10] and helicopters [1, 2]. It is not difficult to see that, with the increasing material on this topic and the advancing hardware such as chips for complex computations, these methods will also provide solutions for maneuvering hybrid UAVs. In [7], a hybrid UAV (quadshot, see section 1.1) is trained with recurring neural networks in order to perform the transition from hover to forward flight. One of the disadvantages of these methods is that the trained networks will result in an effective 'black box', i.e. even-though the controller works it is difficult for a human to interpret the behavior of these complex controllers towards specific perturbations. With conventional controllers, the effect of perturbations can be investigated analytically.

Chapter 4

Conclusion

With the approaches discussed in section 2, the control architecture of a maneuvering UAV can be created. In general, the rotational rate of the UAV around its thrust vector can be seen as an input for the control system. In forward flight, it can be seen that with such a control system maneuvers such as knife-edge can be performed due to the body-fixed z-axis being a control input. This means that the UAV isn't able to perform coordinated flight when perfoming these kind of maneuvers. The other two axes are determined with the required attitude to obtain the desired acceleration at that point. This is logical as full control over the attitudes could result in a crashing plane due to the neglected positional errors. This means that to perform a looping, the UAV would have to get its attitude through the other controllers, which are in turn dependent on the position and its error with the reference position. Therefore, to perform both these maneuvers, the trajectory algorithm should return at least a trajectory where a path is a function of time, and a reference yaw when the UAV is not aiming for coordinated flight.

It seems that a decision can be made between simple control algorithms with a more detailed model and more complex model-based controllers such as INDI which offer simpler knowledge of the model. Controllers such as INDI offer the advantage of requiring less information about the model of the aircraft. This can lead to a reduction in time and cost as an accurate model requires wind tunnel testing or computed fluid dynamics. Besides this, an accurate model requires accurate sensors as well to estimate the state of the UAV. Since small UAVs are required to have lightweight sensors on board, not all desired levels of precision can be obtained. Furthermore, some sensors' accuracy is dependent on the flight envelope of the UAV. This combined makes options that require accurate models less appealing for hybrid UAVs and give the tendency to lean towards model-based approaches that require less model information such as adaptive INDI and sensor-based backstepping. Although direct implementations of back-stepping in aerobatic maneuvering have not been found yet, it might be interesting to compare the possibilities of this method with INDI. Similarly, Sliding mode control could be compared with INDI, although the solutions to the high-frequency oscillations of this method should be considered first.

Finally, neural networks have proven to be useful in a variety of fields. A trade-off should be made between the benefits that these approaches offer, such as the vastly proven

autonomous capabilities to perform aerobatic maneuvers, and the apparent black-box effect, which makes it difficult to predict how the system will behave to unknown inputs.

References

[1] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, June 2010. doi: 10.1177/0278364910371999.

[2] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. *NIPS Conference*, dec 2006. Retrieved from: https://papers.nips.cc/paper/3151-an-application-of-reinforcement-learning-to-aerobatic-helicopter-flight.

[3] Murat Bronz, Ewoud J Smeur, Hector Garcia de Marina, and Gautier Hattenberger. Development of a fixed-wing mini uav with transitioning flight capability. *AIAA Applied Aerodynamics Conference*, jun 2017. doi: 10.2514/6.2017-3739.

[4] Eitan Bulka and Meyer Nahon. Autonomous control of agile fixed-wing uavs performing aerobatic maneuvers. *IEEE International Conference on Unmanned Aircraft Systems*, June 2017. doi: 10.1109/ICUAS.2017.7991437.

[5] Eitan Bulka and Meyer Nahon. Automatic control for aerobatic maneuvering of agile fixed-wing uavs. *Journal of Intelligent Robotic Systems*, February 2018. doi: 10.1007/s10846-018-0790-z.

[6] Warren E Dixon, Aman Behal, Darren M Dawson, and Siddharth P. Nagarkatti. *Nonlinear Control of Engineering Systems: A Lyaponov-Based Approach*. Springer Science+Business Media, NewYork, 1 edition, 2003. doi: 10.1007/978-1-4612-0031-4.

[7] Alejandro Flores and Gerardo Flores. Transition control of a tail-sitter uav using recurrent neural networks. *Cornell University*, jun 2020.

[8] Mostafa Hassanalian and Abdessattar Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, may 2017. doi: 10.1016/j.paerosci.2017.04.003.

[9] Saeid Jafari, Petros A. Ioannou, and Lael Ruddy. What is l1 adaptive control. *AIAA Guidance, Navigation, and Control (GNC) Conference*, aug 2013. doi: 10.2514/6.2013-4513.

[10] Elia Kaufmann, Antonio Loquercio, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Deep drone acrobatics. *Robotics: Science and Systems*, jun 2020. doi: 10.15607/RSS.2020.XVI.040.

[11] Ki-Seok Kim and Youdan Kim. Robust backstepping control for slew maneuver using nonlinear tracking function. *IEEE Transactions on Control Systems Technology*, nov 2003. doi: 10.1109/TCST.2003.815608.

[12] Cao Lijia, Hu Xiaoxiang, Zhang Shengxiu, and Liu Yunfeng. Robust flight control design using sensor-based backstepping control for unmanned aerial vehicles. *Journal of Aerospace Engineering*, nov 2017. doi: 10.1061/(ASCE)AS.1943-5525.0000783.

[13] Sanghyuk Park. Autonomous aerobatic flight by three-dimensional path-following with relaxed roll constraint. *AIAA Guidance, Navigation, and Control Conference*, August 2011. doi: 10.2514/6.2011-6593.

[14] Robin Ritz and Raffaello D'Andrea. An on-board learning scheme for open-loop quadrocopter maneuvers using inertial sensors and control inputs from an external pilot. *IEEE International Conference on Robotics and Automation*, may 2014. doi: 10.1109/ICRA.2014.6907630.

[15] Adnan S. Saeed, Ahmad Bani Younes, Chenxiao Cai, and Guowei Cai. A survey of hybrid unmanned aerial vehicles. *Progress in Aerospace Sciences*, mar 2018. doi: 10.1016/j.paerosci.2018.03.007.

[16] Hongli Shi. A novel scheme for the design of backstepping control for a class of nonlinear systems. *Applied Mathematical Modelling*, oct 2010. doi: 10.1016/j.apm.2010.10.018.

[17] Pranay Sinha, Piotr Esden-Tempski, Christopher A. Forrette, Jeffrey K. Gibboney, and Gregory M. Horn. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. *IEEE Aerospace Conference*, April 2012. doi: 10.1109/AERO.2012.6187313.

[18] E.J.J. Smeur, M. Bronz, and G.C.H.E. de Croon. Incremental control and guidance of hybrid aircraft applied to a tailsitter uav. *Journal of Guidance, Control, and Dynamics*, feb 2020. doi: 10.2514/1.G004520.

[19] Ewoud Jan Jacob Smeur, QiPing Chu, Guido de Croon, and Bart Remes. Modelling of a hybrid uav using test flight data. *IMAV Conference and Competition*, August 2014. doi: doi.org/10.4233/uuid:2eed26cf-a7e8-4a6c-ac42-89a9608e4d2a.

[20] Ewoud J.J. Smeur, Qiping Chu, and Guido C.H.E. de Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance, Control, and Dynamics*, mar 2016. doi: 10.2514/1.G001490.

[21] Ewoud J.J. Smeur, Guido C.H.E. de Croon, and Qiping Chu. Gust disturbance alleviation with incremental nonlinear dynamic inversion. *International Conference on Intelligent Robots and Systems (IROS)*, oct 2016. doi: 10.1109/IROS.2016.7759827.

[22] Ewoud J.J. Smeur, Guido C.H.E. de Croon, and Qiping Chu. Cascaded incremental nonlinear dynamic inversion for mav disturbance rejection. *Control Engineering Practice*, apr 2018. doi: 10.1016/j.conengprac.2018.01.003.

[23] Zikang Su, Honglun Wang, Na Li, Yue Yu, and Jianfa Wu. Exact docking flight controller for autonomous aerial refueling with back-stepping based high order sliding mode. *Mechanical Systems and Signal Processing*, aug 2017. doi: 10.1016/j.ymssp.2017.08.036.

[24] Ezra Tal and Sertac Karaman. Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. *IEEE Conference on Decision and Control*, December 2018.

[25] R.C. van 't Veld, E. van Kampen, and Q. P. Chu. Stability and robustness analysis and improvements for incremental nonlinear dynamic inversion control. *AIAA Guidance, Navigation, and Control Conference*, January 2018.

[26] Xuerui Wang, Erik-Jan Van Kampen, QiPing Chu, and Peng Lu. Stability analysis for incremental nonlinear dynamic inversion control. *2018 AIAA Guidance, Navigation, and Control Conference*, January 2018. doi: 10.2514/6.2018-1115.

[27] Khan Waqas. Modeling dynamics of agile fixed-wing uavs for real-time applications. *International Conference on Unmanned Aircraft Systems*, jun 2016. doi: 10.1109/ICUAS.2016.7502599.

[28] Renshan Zhang, Jiyang Zhang, and Huangchao Yu. Review of modeling and control in uav autonomous maneuvering flight. *IEEE International Conference on Mechatronics and Automation*, August 2018. doi: 10.1109/ICMA.2018.8484542.

[29] Wenya Zhou, Kuilong Yin, Rui Wang, and Yue-E Wang. Design of attitude control system for uav based on feedback linearization and adaptive control. *Mathematical Problems in Engineering*, mar 2014. doi: 10.1155/2014/492680.

Part III

Appendices

Appendix A

Control Effectiveness Derivation

$$\begin{aligned}\frac{\partial a_{X_{Wing}}}{\partial \phi_v} &= \frac{1}{m}\frac{\partial}{\partial \phi_v}\left(-c(\alpha_v)L(\alpha_v, V_{XZ,v}) - s(\alpha_v)D(\alpha_v, V_{XZ,v})\right) \\ &= \frac{\rho S}{2m}\left(s(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_L(\alpha_v)V_{XZ,v}^2 - c(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 - c(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right) \\ &+ \frac{\rho S}{2m}\left(-c(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_D(\alpha_v)V_{XZ,v}^2 - s(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 - s(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right)\end{aligned} \tag{A.1}$$

$$\begin{aligned}\frac{\partial a_{Y_{Wing}}}{\partial \phi_v} &= \frac{1}{m}\frac{\partial}{\partial \phi_v}\left(s(\phi_v)s(\alpha_v)L(\alpha_v, V_{XZ,v}) - s(\phi_v)c(\alpha_v)D(\alpha_v, V_{XZ,v})\right) \\ &= \frac{\rho S}{2m}\left(c(\phi_v)\frac{\partial \phi_v}{\partial \phi_v}s(\alpha_v)C_L(\alpha_v)V_{XZ,v}^2 + s(\phi_v)c(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_L(\alpha_v)V_{XZ,v}^2\right) \\ &+ \frac{\rho S}{2m}\left(s(\phi_v)s(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 + s(\phi_v)s(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right) \\ &- \frac{\rho S}{2m}\left(c(\phi_v)\frac{\partial \phi_v}{\partial \phi_v}c(\alpha_v)C_D(\alpha_v)V_{XZ,v}^2 - s(\phi_v)s(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_D(\alpha_v)V_{XZ,v}^2\right) \\ &- \frac{\rho S}{2m}\left(s(\phi_v)c(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 + s(\phi_v)c(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right) \\ &= \frac{\rho S}{2m}\left(s(\alpha_v)C_L(\alpha_v)V_{XZ,v}^2 - c(\alpha_v)C_D(\alpha_v)V_{XZ,v}^2\right)\end{aligned} \tag{A.2}$$

$$
\begin{aligned}
\frac{\partial a_{Z_{Wing}}}{\partial \phi_v} &= \frac{1}{m}\frac{\partial}{\partial \phi_v}\left(-c(\phi_v)s(\alpha_v)L(\alpha_v, V_{XZ,v}) + c(\phi_v)c(\alpha_v)D(\alpha_v, V_{XZ,v})\right) \\
&= \frac{\rho S}{2m}\left(s(\phi_v)s(\alpha_v)C_L(\alpha_v)V_{XZ,v}^2 - c(\phi_v)c(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_L(\alpha_v)V_{XZ,v}^2\right) \\
&+ \frac{\rho S}{2m}\left(-c(\phi_v)s(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 - c(\phi_v)s(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right) \\
&+ \frac{\rho S}{2m}\left(-s(\phi_v)c(\alpha_v)C_D(\alpha_v)V_{XZ,v}^2 - c(\phi_v)s(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_D(\alpha_v)V_{XZ,v}^2\right) \\
&+ \frac{\rho S}{2m}\left(c(\phi_v)c(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 + c(\phi_v)c(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right) \\
&= \frac{\rho S}{2m}\left(-c(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_L(\alpha_v)V_{XZ,v}^2 - s(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 - s(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right) \\
&+ \frac{\rho S}{2m}\left(-s(\alpha_v)\frac{\partial \alpha_v}{\partial \phi_v}C_D(\alpha_v)V_{XZ,v}^2 + c(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \phi_v}V_{XZ,v}^2 + c(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \phi_v}\right)
\end{aligned}
\tag{A.3}
$$

Lift and drag contributions due to wing derived with respect to θ_v

$$\begin{aligned}\frac{\partial a_{X_{Wing}}}{\partial \theta_v} &= \frac{1}{m}\frac{\partial}{\partial \theta_v}\left(-c(\alpha_i)L(\alpha_v, V_{XZ,v}) - s(\alpha_i)D(\alpha_v, V_{XZ,v})\right) \\ &= \frac{\rho S}{2m}\left(-c(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \theta_v}V_{XZ,v}^2 - c(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \theta_v}\right) \\ &+ \frac{\rho S}{2m}\left(-s(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \theta_v}V_{XZ,v}^2 - s(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \theta_v}\right)\end{aligned} \tag{A.4}$$

$$\frac{\partial a_{Y_{Wing}}}{\partial \theta_v} = 0 \tag{A.5}$$

$$\begin{aligned}\frac{\partial a_{Z_{Wing}}}{\partial \theta_v} &= \frac{1}{m}\frac{\partial}{\partial \theta_v}\left(-s(\alpha_i)L(\alpha_v, V_{XZ,v}) + c(\alpha_i)D(\alpha_v, V_{XZ,v})\right) \\ &= \frac{\rho S}{2m}\left(-s(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \theta_v}V_{XZ,v}^2 - s(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \theta_v}\right) \\ &+ \frac{\rho S}{2m}\left(+c(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \theta_v}V_{XZ,v}^2 + c(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \theta_v}\right)\end{aligned} \tag{A.6}$$

Lift and drag contributions due to wing derived with respect to ψ_v

$$\begin{aligned}
\frac{\partial a_{X_{Wing}}}{\partial \psi_v} &= \frac{1}{m}\frac{\partial}{\partial \psi_v}\left(-c(\psi_v)c(\alpha_v)L(\alpha_v, V_{XZ,v}) - c(\psi_v)s(\alpha_v)D(\alpha_v, V_{XZ,v})\right) \\
&= \frac{\rho S}{2m}\left(+s(\psi_v)c(\alpha_v)C_L(\alpha_v)V_{XZ,v}^2 + c(\psi_v)s(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_L(\alpha_v)V_{XZ,v}^2\right) \\
&+ \frac{\rho S}{2m}\left(-c(\psi_v)c(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - c(\psi_v)c(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right) \\
&+ \frac{\rho S}{2m}\left(s(\psi_v)s(\alpha_v)C_D(\alpha_v)V_{XZ,v}^2 - c(\psi_v)c(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_D(\alpha_v)V_{XZ,v}^2\right) \\
&+ \frac{\rho S}{2m}\left(-c(\psi_v)s(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - c(\psi_v)s(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right) \\
&= \frac{\rho S}{2m}\left(+s(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_L(\alpha_v)V_{XZ,v}^2 - c(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - c(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right) \\
&+ \frac{\rho S}{2m}\left(-c(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_D(\alpha_v)V_{XZ,v}^2 - s(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - s(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right)
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
\frac{\partial a_{Y_{Wing}}}{\partial \psi_v} &= \frac{1}{m}\frac{\partial}{\partial \psi_v}\left(-s(\psi_v)c(\alpha_v)L(\alpha_v, V_{XZ,v}) - s(\psi_v)s(\alpha_v)D(\alpha_v, V_{XZ,v})\right) \\
&= \frac{\rho S}{2m}\left(-c(\psi_v)c(\alpha_v)C_L(\alpha_v)V_{XZ,v}^2 + s(\psi_v)s(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_L(\alpha_v)V_{XZ,v}^2\right) \\
&+ \frac{\rho S}{2m}\left(-s(\psi_v)c(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - s(\psi_v)c(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right) \\
&+ \frac{\rho S}{2m}\left(-c(\psi_v)s(\alpha_v)C_D(\alpha_v)V_{XZ,v}^2 - s(\psi_v)c(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_D(\alpha_v)V_{XZ,v}^2\right) \\
&+ \frac{\rho S}{2m}\left(-s(\psi_v)s(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - s(\psi_v)s(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right) \\
&= \frac{\rho S}{2m}\left(-c(\alpha_v)C_L(\alpha_v)V_{XZ,v}^2 - s(\alpha_v)C_D(\alpha_v)V_{XZ,v}^2\right)
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
\frac{\partial a_{Z_{Wing}}}{\partial \psi_v} &= \frac{1}{m}\frac{\partial}{\partial \psi_v}\left(-s(\alpha_v)L(\alpha_v, V_{XZ,v}) + c(\alpha_v)D(\alpha_v, V_{XZ,v})\right) \\
&= \frac{\rho S}{2m}\left(-c(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_L(\alpha_v)V_{XZ,v}^2 - s(\alpha_v)\frac{\partial C_L(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 - s(\alpha_v)C_L(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right) \\
&+ \frac{\rho S}{2m}\left(-s(\alpha_v)\frac{\partial \alpha_v}{\partial \psi_v}C_D(\alpha_v)V_{XZ,v}^2 + c(\alpha_v)\frac{\partial C_D(\alpha_v)}{\partial \psi_v}V_{XZ,v}^2 + c(\alpha_v)C_D(\alpha_v)\frac{\partial V_{XZ,v}^2}{\partial \psi_v}\right)
\end{aligned} \tag{A.9}$$

Appendix B

Quaternion Filter

The Quadshot makes use of a filtered orientation to estimate its control effectiveness. To prevent singularities from occurring, the orientation is filtered in quaternion representation instead of Euler representation. The filter applied to this orientation is the same second order Butterworth filter as the linear acceleration uses. However, quaternions are vectors on a 4D unit sphere that represent orientations in a 3D world, therefore, applying a linear low pass filter can result in a quaternion that exist inside the 4D unit sphere. This quaternion can then be re-normalized, but this can lead to a non-uniform rotational velocity. Therefore, an assessment is made of the validity of this approach by comparing it to the same orientation filtered in Euler representation.

The maximum error from the Euler based approach is roughly $6 \cdot 10^{-3}$ rad (≈ 0.4 deg) as can be seen in Figure B.1. The advantage of the quaternion filter can be seen in Figure B.2. Here the raw data changes from π to $-\pi$ on the yaw axis. It can be seen that the Euler filter smoothly transitions from π to $-\pi$, while in reality, the yaw angle just switched to the other side. The quaternion filter, similar to the raw data, goes to the point π and then immediately switches to $-\pi$. While it might seem as a singularity that the quaternion filter also has a steep change at $36.7s$, this is only due to its representation in Euler angles.

Besides looking at the error between the quaternion and Euler filtered orientations, an additional comparison is made between both filtered data-sets and the raw data itself. This can be done since the cutoff frequency is high compared to the rates of the raw data itself. Therefore, the data is not heavily influenced by the filter. Before comparing these values, the delay between raw data and filtered data is compensated by shifting the raw data. This delay is determined with the cross-correlation between raw and filtered data and is equal for both the Euler and quaternion scenario. The results of this can be seen in Table B.3. It can be seen that without singularities there is almost no difference between the root mean squared (RMS) error of the Euler angles and the quaternions. It can even be seen that the quaternion filter matches the raw data slightly better. These RMS errors are present due to the small oscillations that were filtered out, but it can be seen that this Butterworth filter does not significantly alter the raw data. There is a big difference

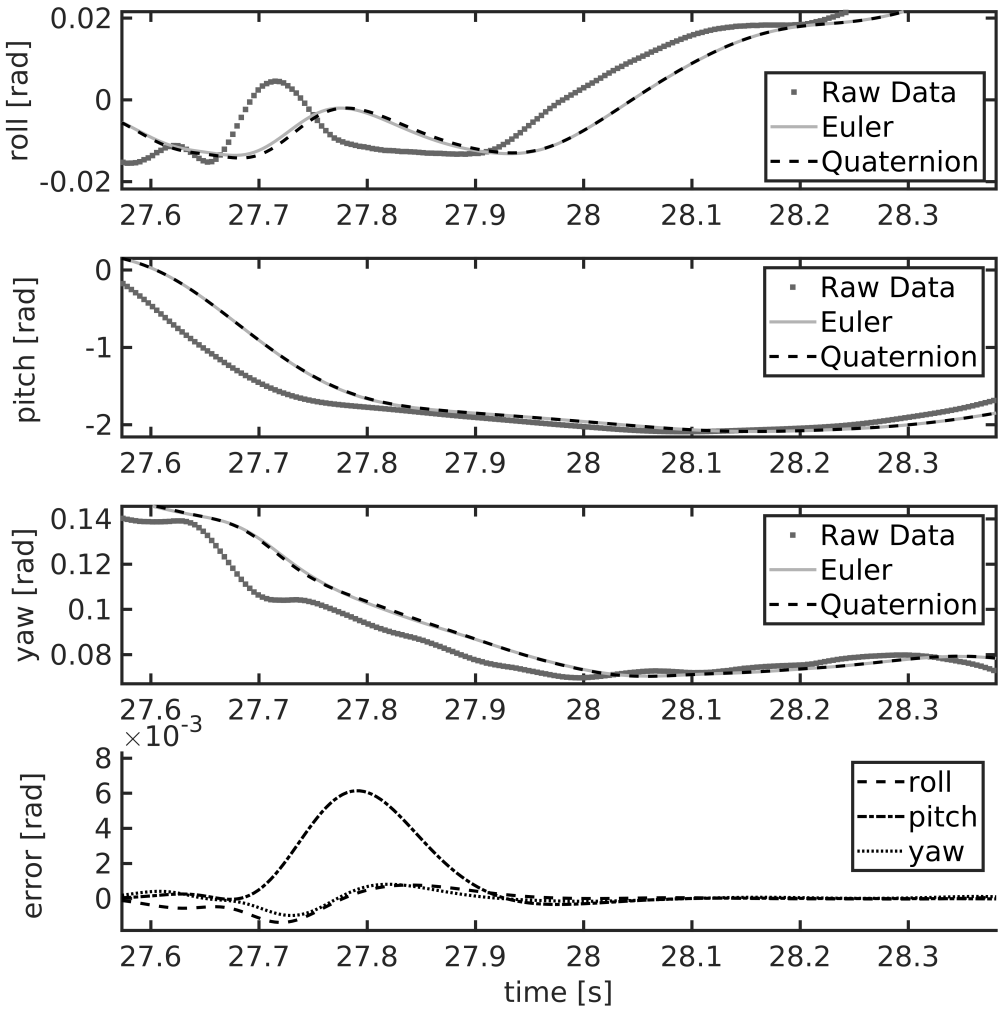


Figure B.1: Filtered orientations compared to raw data near the point where the error between the Euler and quaternion filter is largest

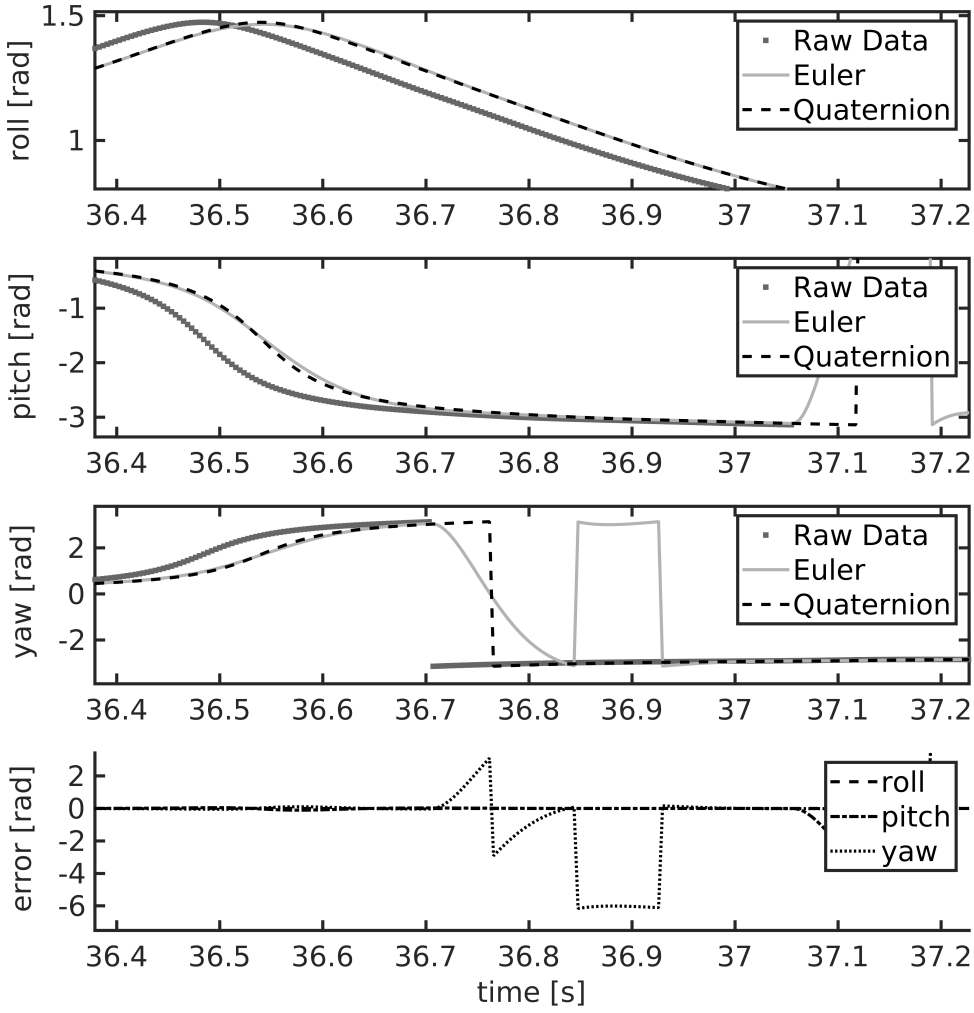


Figure B.2: Filtered orientations compared to raw data at a point of singularity

		Euler [rad]	Quaternion [rad]
1.0	**Roll**	0.0007	0.0007
	Pitch	0.0059	0.0057
	Yaw	0.0019	0.0018

Table B.1: Without singularities

		Euler [rad]	Quaternion [rad]
1.0	**Roll**	0.0029	0.0021
	Pitch	0.3402	0.0808
	Yaw	0.6697	0.0988

Table B.2: With singularities

Table B.3: RMS error of filtered orientations compared to the raw data time shifted to match the delay of the Butterworth filter

between the Euler and quaternion filtered data near the region where singularities occur. The Euler filtered data has significant errors in the pitch and yaw axes, while the error with quaternion filtered data still remains small. The increase in the quaternion RMS in the pitch and yaw is caused partially due to the comparison in Euler angles. This conversion caused singularity spikes in the quaternion filter as can be seen in Figure B.3. These spikes are not present in quaternion representation as the points π and $-\pi$ in yaw near each other in quaternion representation.

From the data gathered throughout this section it is concluded that the filter is accurate enough to implement it in the quadshot instead of using alternate and more accurate methods for quaternion filtering.

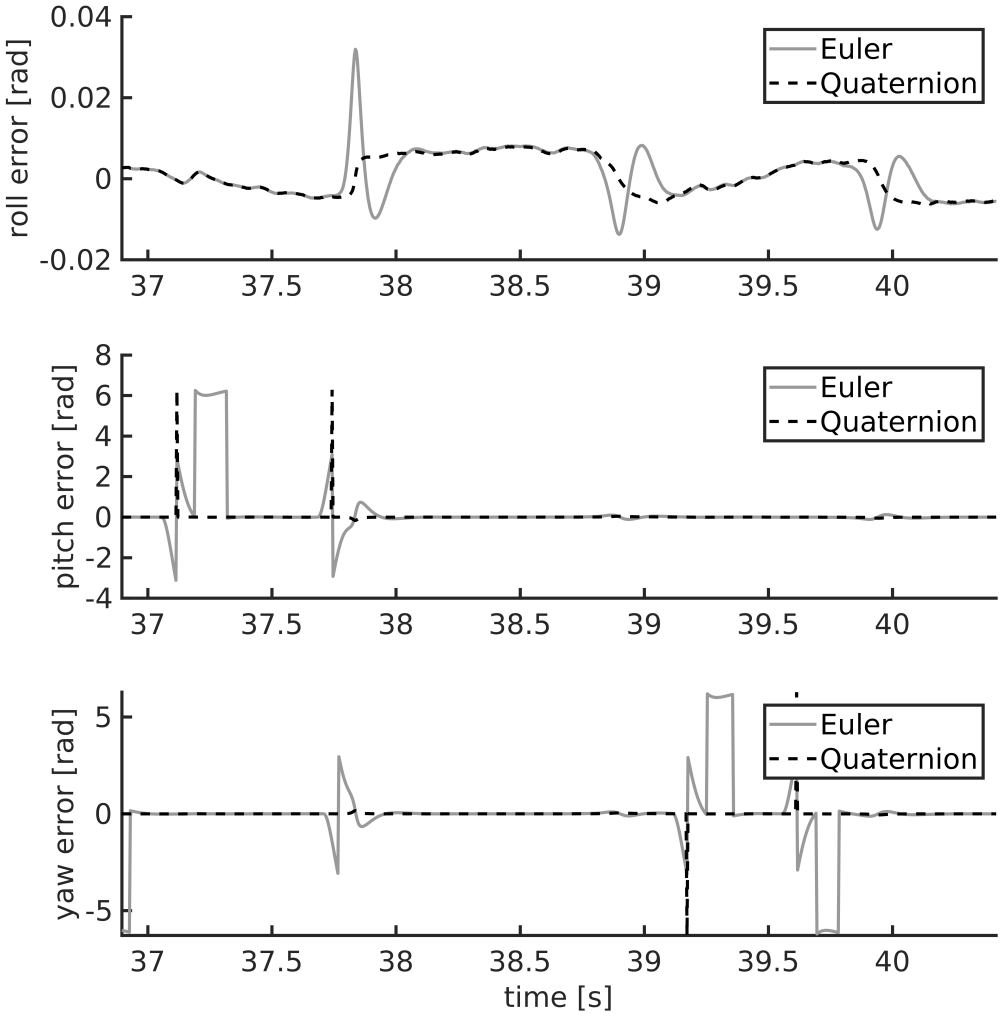


Figure B.3: RMS error of filtered orientations compared to raw data at points of singularity

Appendix C

Additional Trajectory Results

C.1 Linearized Model

The only results gathered from the linearized model are the step responses of the controller to a 1 meter input step. The position of the quadshot to this step input can be seen in figure C.1.

C.2 JSBSim Model

Figure C.2 shows the position and acceleration of the controller responding to a 1 meter step input. The results of this response compared to the linearized model are very similar, however, it should be noted that the derivative gain in the linearized model is lower, resulting in more overshoot.

The secondly, the controller is subjected to a circular trajectory. This was performed for 2 cases. One where the aerodynamics of the quadshot are neglected both in the simulation model and in the control effectiveness and one where the aerodynamics are included in the model and the control effectiveness. A top view of the trajectory for both the cases can be seen in C.4 and C.6. The position of the quadshot during these tests can be seen in C.3 and C.5.

The final trajectory tested in simulation is used to determine if the controller can track accelerations at high roll angles. The position and acceleration of the quadshot during the trajectory can be seen in figure C.8 and C.9 respectively. It can be seen that the quadshot is able to track the trajectory with some overshoot on the z position. This can also be seen in the acceleration as the filtered acceleration diverges from the commanded acceleration twice. These moments coincide with the spikes that can be seen in the y acceleration. The cause of this is due to an unrealistic acceleration defined in this trajectory. Near the peak, the acceleration increases rapidly so that the controller is forced to roll, however, the acceleration in the y axis is considerably smaller than the acceleration in the z axis. As

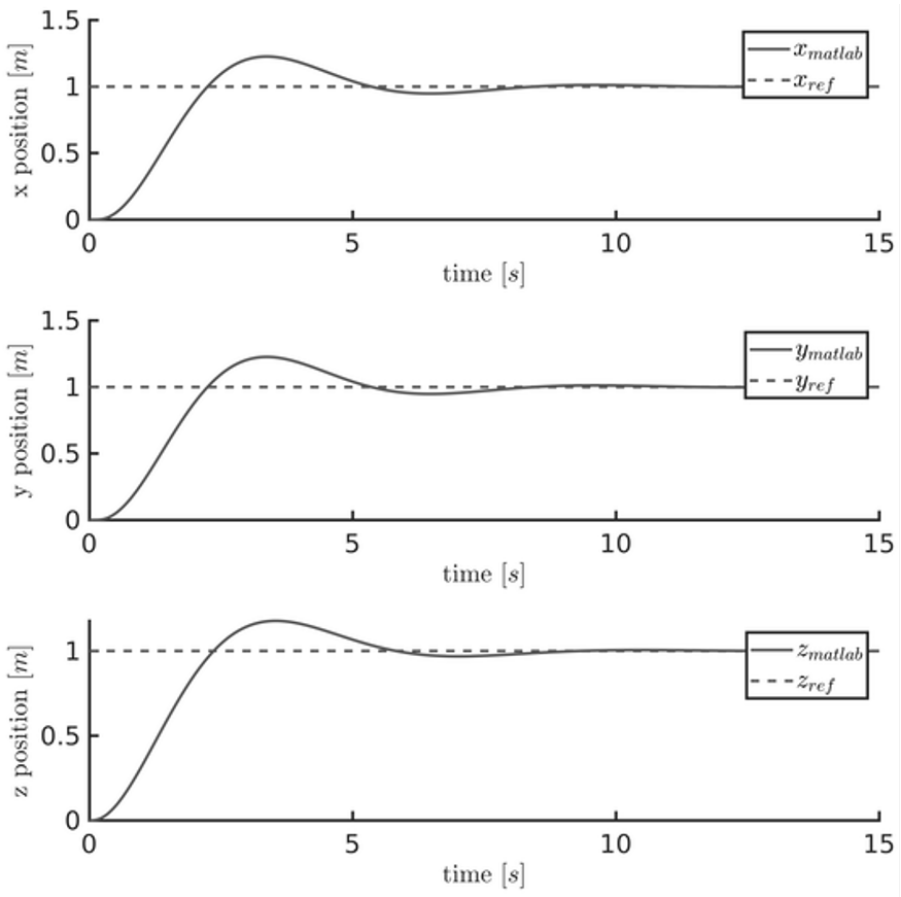


Figure C.1: Response to a step input on the reference positions with $K_p = 1$ and $K_d = 1$. (all t0059hree axes were performed individually)

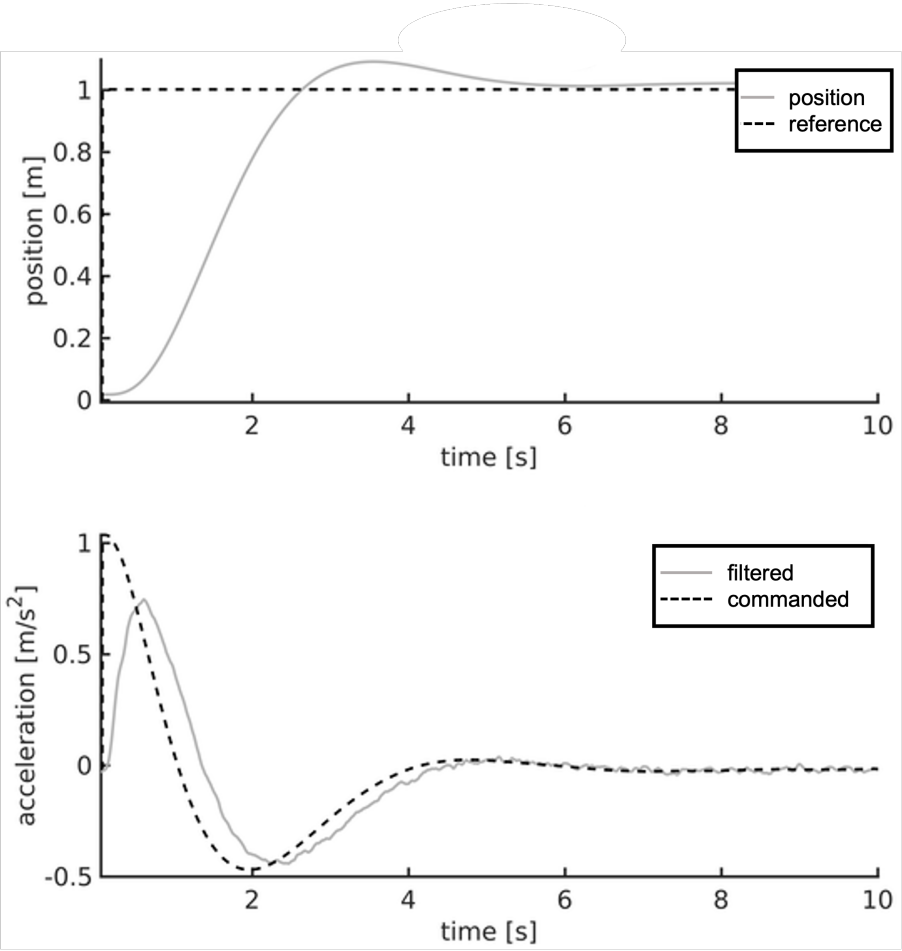


Figure C.2: Position and acceleration of the quadshot reacting on a 1m step response.

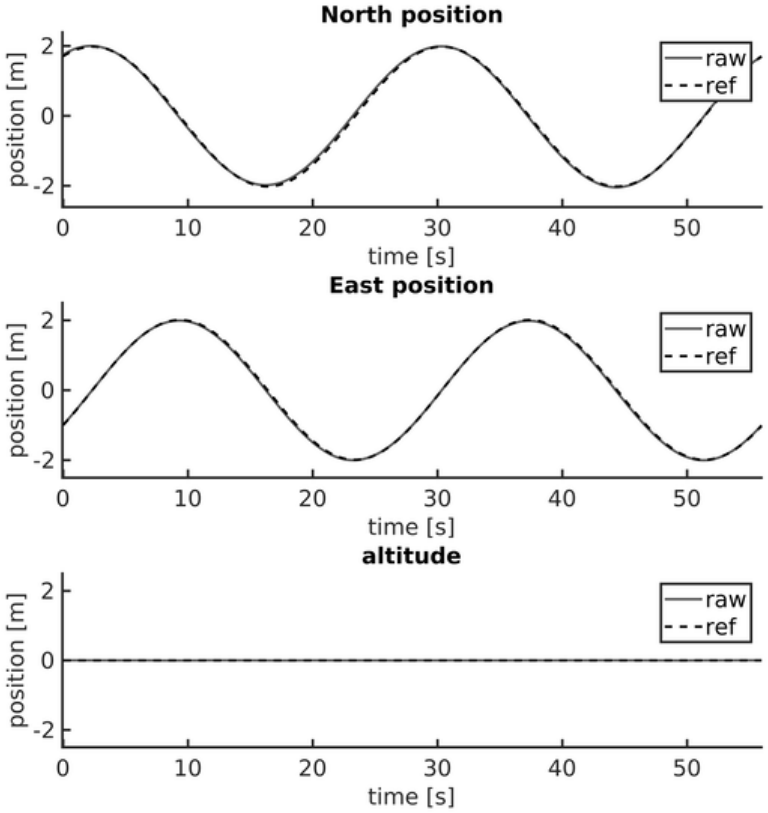


Figure C.3: Position of the quadshot following a circular trajectory with a velocity of 0.45m/s and no aerodynamics.

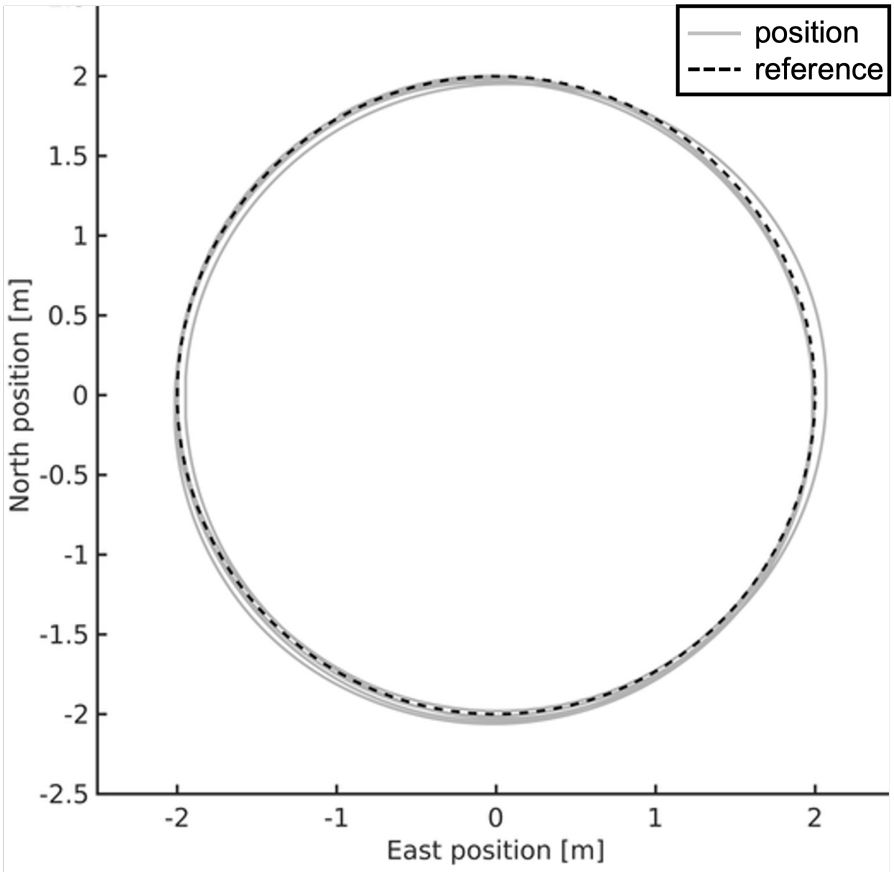


Figure C.4: Top view of the quadshot following a circular trajectory with a velocity of 0.45m/s and no aerodynamics.

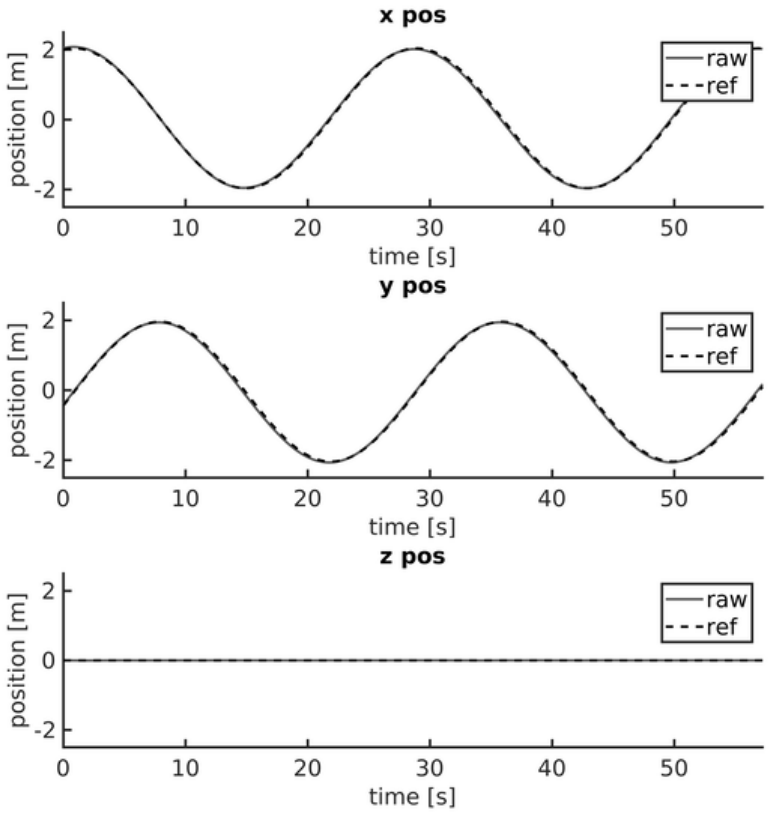


Figure C.5: Position of the quadshot following a circular trajectory with a velocity of 0.45m/s and aerodynamics.

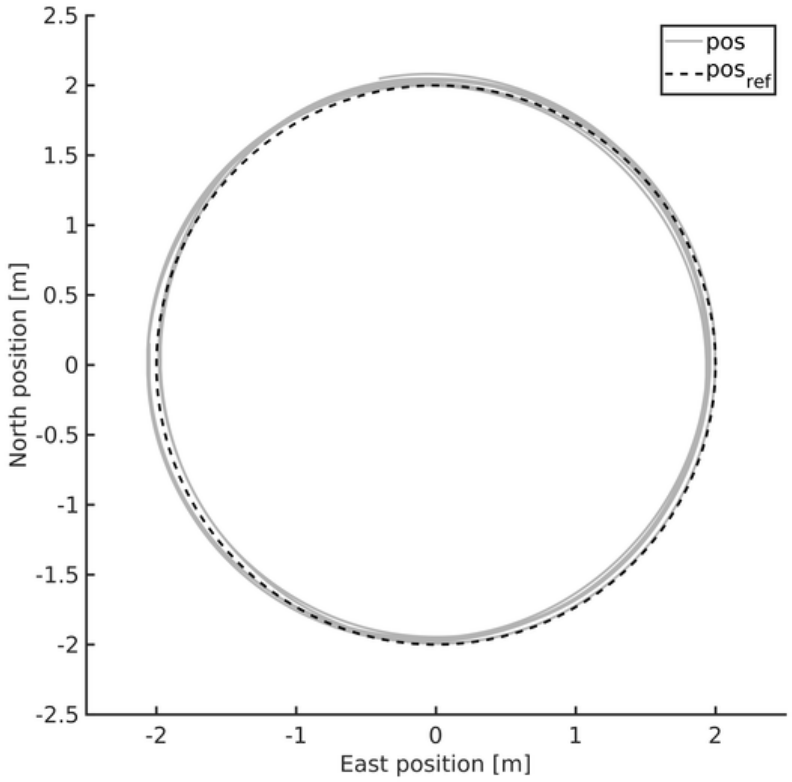


Figure C.6: Top view of the quadshot following a circular trajectory with a velocity of 0.45m/s and aerodynamics.

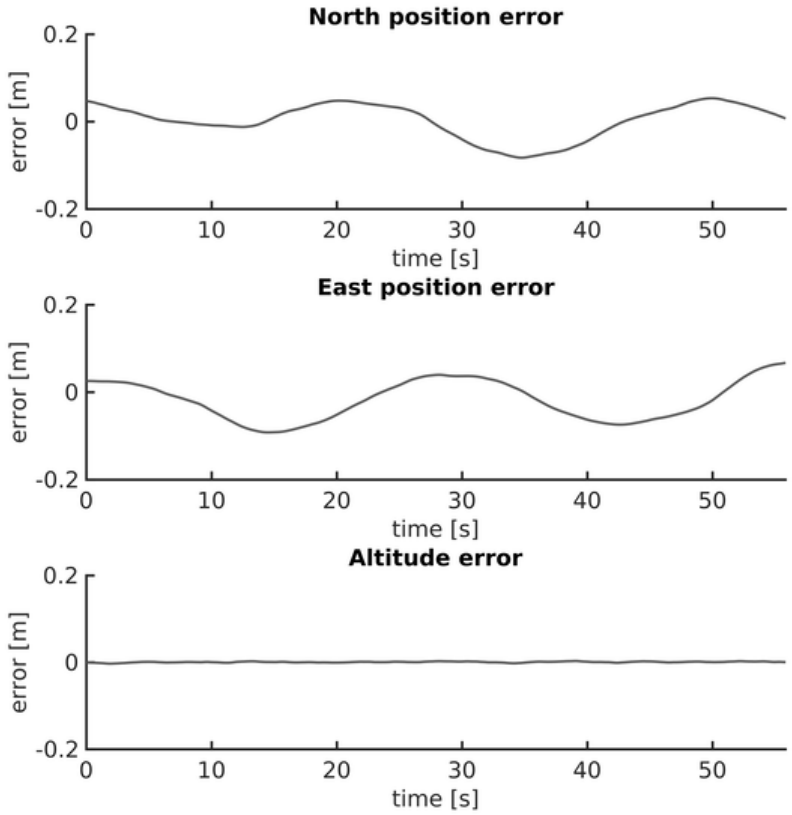


Figure C.7: Position error of the quadshot following a circular trajectory with a velocity of 0.45m/s.

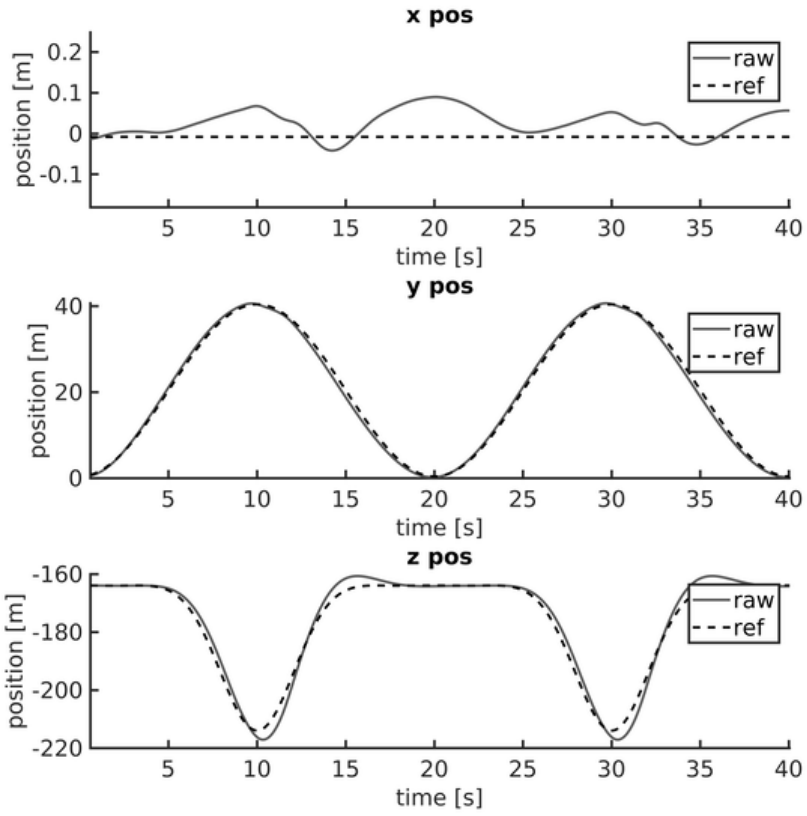


Figure C.8: Position of the quadshot following a trajectory with a drop forcing it to roll perform large roll angles.

the controller rolls, it forces its thrust vector along the y axis, were almost no acceleration is demanded. This forces the controller to reduce its thrust, while continuing the roll. After that the controller increases thrust again to meet the high acceleration requirement, after which it has to flip again resulting in the same error twice. This can be prevented by designing a trajectory with more achievable accelerations. The overall attitude of the quadshot during this maneuver can be seen in figure C.11. The high roll angles can be seen in figure C.10.

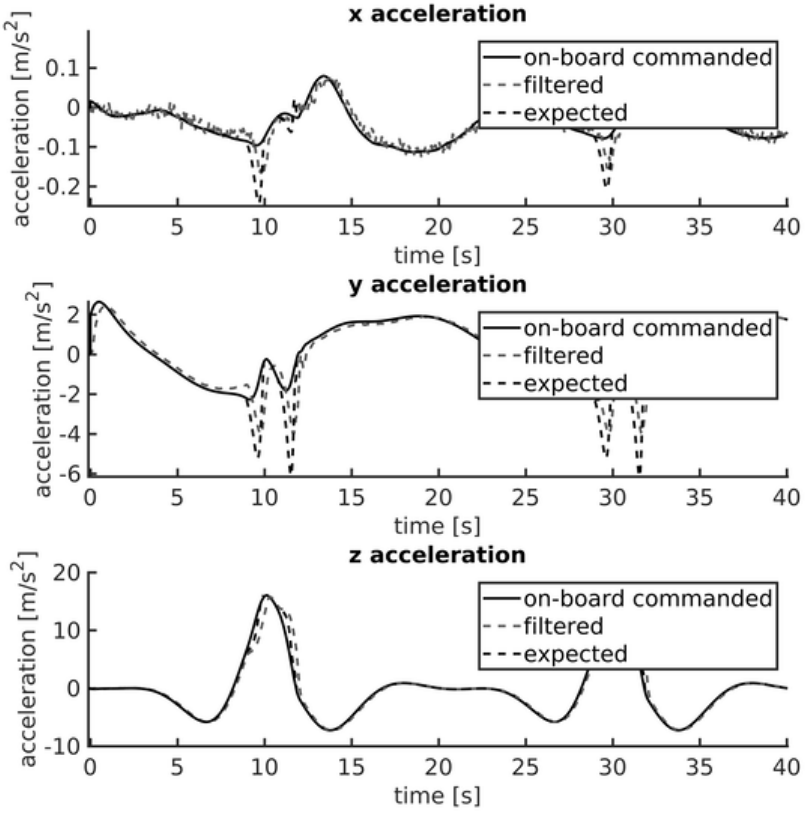


Figure C.9: Acceleration of the quadshot following a trajectory with a drop forcing it to roll perform large roll angles.

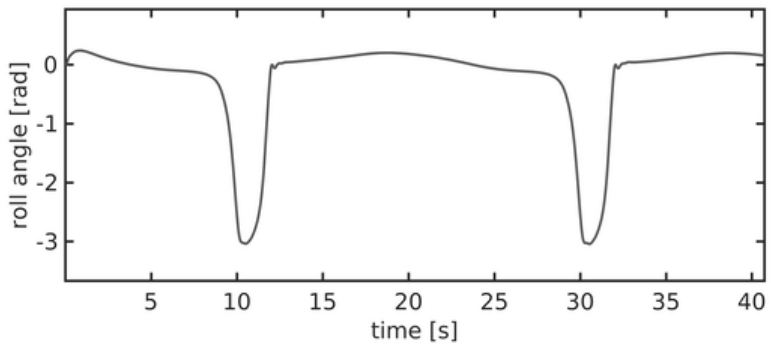


Figure C.10: Roll angle the quadshot following a trajectory with a drop forcing it towards large roll angles.

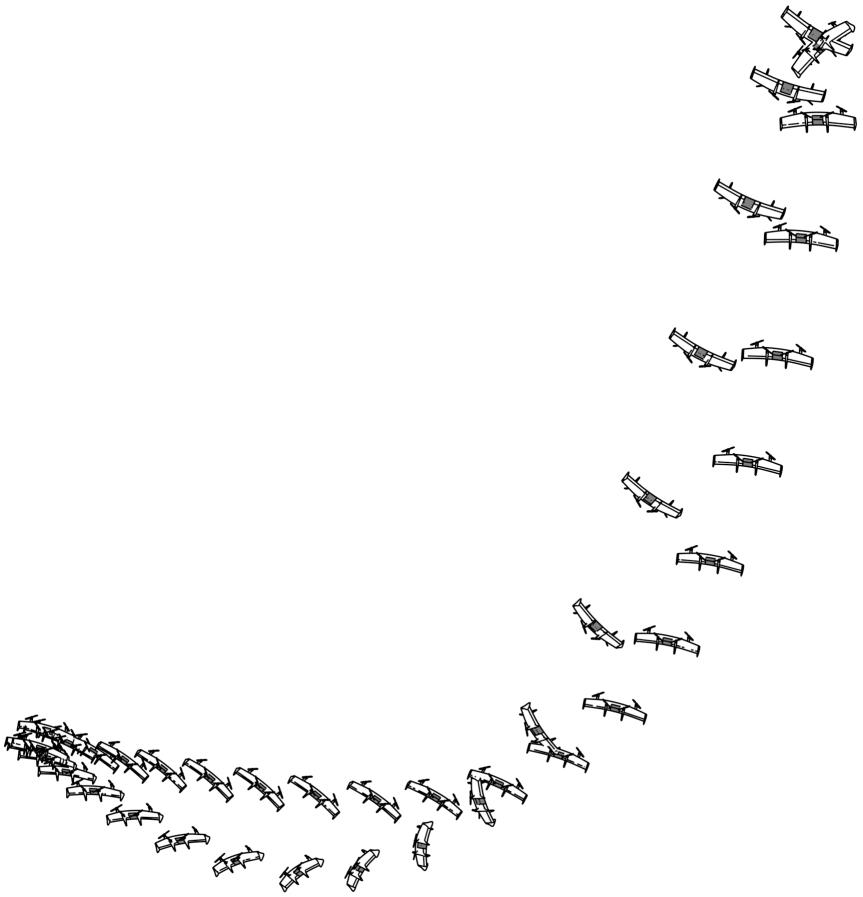


Figure C.11: Trajectory with high roll angles, quadshot scaled for visibility, simulated with aerodynamic surfaces.