Control of Fixed-Wing UAVs in Icing Conditions Using Nonlinear Model Predictive Control

Nadine Adelina Ghindaoanu

January 2023

Contents

Chapter 1

Introduction

1.1 Motivation

Unmanned Aerial Vehicles (UAVs), first developed and used by the military, are being increasingly used today for a variety of applications, having the advantage of usually being smaller than manned aircrafts and able to reach places that might be difficult to access otherwise. However, being a relatively new technology, it faces many challenges. An important challenge is being able to perform its tasks under severe weather conditions, a particularly dangerous one being in-flight icing [1]. In-flight icing has shown to lead to a degradation of the aerodynamic performance of the aircraft, as well as to a significant decrease in lift, increase in drag, and a deterioration of the stall limits[2], [3]. In addition, the degradation in performance is greater the smaller the UAV is, as icing will make up a larger part of the total weight of the aircraft. The severity of the in-flight icing also depends on the type of ice being formed, where complex ice geometries increase the performance degradation [3]. In addition, since UAVs do not have a pilot to identify the ice conditions, the controllers used must be able to handle the disturbances that come from icing and ensure the safety of the UAV.

When it comes to the controllers usually used for UAV control, an autopilot usually relies on an inner-loop PID controller, where the model is based on a linear approximation around trim states. This type of controller shows good performance when close to the trim states, but nonlinear effects are neglected. Using Nonlinear Model Predictive Control (NMPC) instead, the nonlinear effects could be included in the controller design. Due to the fast low-level dynamics in pitch and roll, NMPCs are typically too slow to be used, but a lot of progress has been made in the topic. The NMPC controllers developed by Reinhardt [4], for example, were shown to have equal or superior performance when tested together with the ArduPilot [5] controller, an widely used open-source drone software. What remains to be seen is whether NMPC could handle the hazards that UAVs are exposed to, more specifically the mentioned icing conditions. This would bring us a step closer to being able to implement a inner-loop NMPC controller on the UAV.

1.2 Previous Work and Contributions

Different control techniques have previously been developed for the purpose of controlling an UAV in icing conditions, such as the conventional PID control, Model Reference Adaptive Control (MRAC) by Högnadottir [6], as well as robust and gain-scheduled control by Kleiven [7]. When it comes to Nonlinear Model Predictive Control (NMPC), which is the topic of this thesis, it has previously been researched in detail by Reinhardt [4], as previously mentioned. The different NMPC controllers developed in his doctoral thesis were developed in Python using the package acados [8], and showed a great performance, but they have not been tested in icing conditions. This thesis aims to determine if the NMPC proposed by Reinhardt [4] is able to counteract the disturbances caused by icing. Its performance will also be compared to the one of the widely used PID controller, as well as to an adaptive controller's newer approach.

The PID controller used is based on Beard & McLain [9], while the MRAC controller is based on Lavretsky and Wise [10]. They were tuned for icing conditions by Högnadottir as explained in her thesis [6], and implemented in a Matlab/ Simulink simulator. This was first developed by Gryte [11], and later improved with the UAV's aerodynamic data found by Winter [12] and with Kleiven's work [7], which extended the model to account for asymmetric icing on the wings. However, the accretion of ice on the propeller, which is shown to result in a significant decrease in thrust and increase in torque [13],[14], had not been included in the model. In Müller's work [15], a model for the performance of a propeller in icing conditions has been developed by conducting experiments on an icing wind tunnel and analysing the performance of the iced propeller. As his experiments were performed on a Mejzlik 21x13E propeller, the parameters in his model have been slightly modified, following indications in his paper [15], to estimate the performance of the Aeronaut CamCarbon 14x8" (foldable), the propeller the Skywalker X8 uses.

The work presented in this thesis includes Müller's model in the simulator to account for the effect of ice on the propeller. The propulsion model is further improved by implementing the model proposed by Coates [16], which was shown to predict thrust better than the Fitzpatrick model [17],[18] previously used, with a root mean square error of $2.20 - 4.52$ percent, compared to Fitzpatrick's 6.56 percent [16]. Finally, the NMPC developed by Reinhardt is tuned and implemented in the mentioned Matlab/ Simulink simulator, tested and compared to the PID and MRAC controllers, with wind and icing conditions on both wings and propeller.

1.3 Report Outline

This thesis is organized in several parts. First, the theory used throughout the thesis will be introduced in Chapter 2, which includes the mathematical model of the UAV with its forces and moments, as well models for the environmental factors affecting the UAV. The NMPC is introduced in Chapter 3, together with the other controllers the NMPC will be compared to. In Chapter 4, the simulator and its implementation is presented, as well as the tuning of the controllers and the different simulation cases. Finally, the simulation results will be presented and discussed in chapters 5 and 6.

Chapter 2

Theory

This chapter presents and explains the theory necessary to understand the UAV system this thesis was working with, as well as the forces, moments and disturbances acting on it. The mathematical models explained here will be the background for the simulations, and some will also be used in the formulation of the controllers.

2.1 Coordinate Frames

The UAV model will be described by kinematic and dynamic equations of motion. These equations, however, can be expressed in different coordinate frames, and doing so will have its advantages depending on the controller used, as will be explained in the following sections. Therefore, the most common frames will be introduced, as well as the transformation between them. They are the body-fixed frame $\{b\}$, inertial frame NED $\{n\}$, stability frame $\{s\}$ and wind frame $\{w\}$. In this thesis, a vector $\boldsymbol{v}$ described in reference frame $\{a\}$ will be written as $\boldsymbol{v}^a$, and a rotation transforming vector $\boldsymbol{v}^a$ to vector $\boldsymbol{v}^b$ (described in reference frame $\{b\}$), will be given by the rotation matrix $\boldsymbol{R}_{ba}$, so that $\boldsymbol{v}^b = \boldsymbol{R}_{ba}\boldsymbol{v}^a$. Similarly, $\boldsymbol{v}^a = \boldsymbol{R}_{ab}\boldsymbol{v}^b = \boldsymbol{R}_{ba}^\top \boldsymbol{v}^b$ is the opposite rotation.

The position and attitude of the vehicle are usually expressed in **NED** (North-East-Down) frame, which is assumed to be the inertial frame for a small UAV. This is a local reference frame, so the position and orientation of the UAV are expressed relative to a fixed location on Earth, which is a good approximation if the UAV is limited to a small area. It is characterized by the $\boldsymbol{x}^n$ axis pointing to the North, the $\boldsymbol{z}^n$ axis pointing to the direction of gravity, and $\boldsymbol{y}^n = \boldsymbol{z}^n \times \boldsymbol{x}^n$. The linear and angular velocities of the aircraft, on the other hand, are usually represented in the **body**-fixed reference frame, which is fixed to the airframe of the vehicle. In this case, the $\boldsymbol{x}^b$ axis points forward through the nose of the UAV, the $\boldsymbol{z}^b$ axis points downwards, and $\boldsymbol{y}^b = \boldsymbol{z}^b \times \boldsymbol{x}^b$.

The coordinate frames **stability** and **wind** will also be used in this thesis, but in order to understand them, some aerodynamic concepts need to be introduced

first. These are the airspeed V_a, the angle of attack α, and the sideslip angle β, given by:

$$V_a = ||\boldsymbol{v}_r^b|| = \sqrt{u_r^2 + v_r^2 + w_r^2}, \tag{2.1a}$$

$$\alpha = \arcsin\left(\frac{w_r}{u_r}\right), \tag{2.1b}$$

$$\beta = \arcsin\left(\frac{v_r}{V_a}\right), \tag{2.1c}$$

where the vector $\boldsymbol{v}_r^b = [u_r, v_r, w_r]^\top$ is the relative velocity vector, given by $\boldsymbol{v}_r^b = \boldsymbol{v}_{nb}^b - \boldsymbol{R}_{nb}^\top \boldsymbol{v}_{nw}^n$, $\boldsymbol{v}_{nb}^b$ is the velocity vector of the UAV decomposed in body frame, and $\boldsymbol{v}_{nw}^n$ is the wind velocity vector in inertial frame.
The **stability** frame depends on the air surrounding the UAV, and its axes, starting in the body-fixed reference frame, are defined as a right-hand rotation about $\boldsymbol{y}^b = \boldsymbol{y}^s$, with rotation angle equal to the angle of attack α, as seen on Figure 2.1. On the other hand, starting in the stability frame, the **wind** frame is defined as a right-hand rotation about $\boldsymbol{z}^s = \boldsymbol{z}^w$, with rotation angle equal to the sideslip angle β, as seen on Figure 2.2.

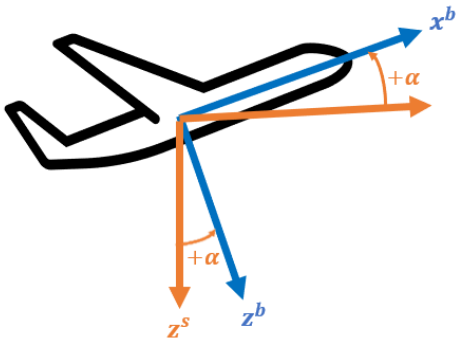

Figure 2.1: Rotation between the body and stability coordinate frames.

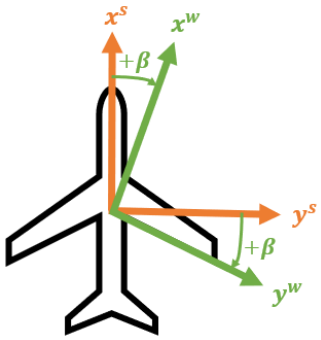

Figure 2.2: Rotation between the stability and wind coordinate frames.

The previously mentioned rotations can be described by rotation matrices. The rotation matrix that transforms a vector given in the body-fixed reference frame to the stability frame $\{s\}$ is given by $\boldsymbol{R}_{sb}(\alpha)$:

$$\boldsymbol{R}_{sb}(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}, \tag{2.2}$$

and the transformation from the stability frame to the wind frame $\{w\}$ is given

by $\boldsymbol{R}_{ws}(\beta)$:

$$\boldsymbol{R}_{ws}(\beta) = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

Then, the transformation from $\{b\}$ to $\{w\}$ is given by:

$$\boldsymbol{R}_{wb}(\alpha, \beta) = \boldsymbol{R}_{ws}(\beta)\boldsymbol{R}_{sb}(\alpha). \tag{2.4}$$

2.2 The UAV Model

In this section, the typical mathematical model that is used to describe the state of a fixed-wing UAV will be described. This model is composed of the kinematic and dynamic equations, which include a model for the forces and moments as well as the control surfaces. These equations are derived in [9] and are given by:

$$\dot{\boldsymbol{p}}^n_{nb} = \boldsymbol{R}(\boldsymbol{\Theta}_{nb})\,\boldsymbol{v}^b_{nb} \tag{2.5a}$$
$$\dot{\boldsymbol{\Theta}}_{nb} = \boldsymbol{T}(\boldsymbol{\Theta}_{nb})\boldsymbol{\omega}^b_{nb} \tag{2.5b}$$
$$\dot{\boldsymbol{v}}^b_{nb} = \frac{1}{m}(\boldsymbol{F}^b_a + \boldsymbol{F}^b_g + \boldsymbol{F}^b_t) - \boldsymbol{\omega}^b_{nb} \times \boldsymbol{v}^b_{nb} \tag{2.5c}$$
$$J^b\,\dot{\boldsymbol{\omega}}^b_{nb} = \boldsymbol{S}(\boldsymbol{J^b}\boldsymbol{\omega}^b_{nb})\boldsymbol{\omega}^b_{nb} + \boldsymbol{M}^b. \tag{2.5d}$$

The kinematics describe the relationship between position and velocity, and are given by (2.5a) and (2.5b). The vector $\boldsymbol{p}^n_{nb} = [p_n, p_e, p_d]^\top$ describes the inertial position of the UAV in the NED coordinate frame $\{n\}$, and $\boldsymbol{v}^b_{nb} = [u, v, w]^\top$ is the velocity vector in $\{b\}$, where u, v and w are the forward, lateral and vertical velocities, respectively. The vector $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$ contains the Euler angles, where ϕ, θ, and ψ are the roll, pitch and yaw angles, respectively, and are given in $\{n\}$. The rotational motion can be described by the angular velocity vector $\boldsymbol{\omega}^b_{nb} = [p, q, r]^\top$, given in $\{b\}$, where p, q, and r are the roll, pitch and yaw rates. The dynamic equations are given by (2.5c) and (2.5d), they are derived using Newton's second law of motion, and describe the relationship between motion and forces. The sum of all external forces acting on the UAV, in $\{b\}$, is given by $\boldsymbol{F}^b = \boldsymbol{F}^b_a + \boldsymbol{F}^b_g + \boldsymbol{F}^b_t$, and M^b is the total moment, given in $\{b\}$. They will be further described in section 2.4.1.
From (2.5) we also have $\boldsymbol{J}^b = (\boldsymbol{J}^b)^\top$, which denotes the inertia matrix and is given by

$$\boldsymbol{J}^b = \begin{bmatrix} J_{xx} & 0 & -J_{xz} \\ 0 & J_{yy} & 0 \\ -J_{xz} & 0 & J_{zz} \end{bmatrix}, \tag{2.6}$$

with the parameters in the inertia matrix given in Table 4.6. The matrices $\boldsymbol{R}(\boldsymbol{\Theta}_{nb})$

and $\boldsymbol{T}(\boldsymbol{\Theta}_{nb})$ are given by

$$\boldsymbol{R}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \tag{2.7}$$

$$\boldsymbol{T}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}. \tag{2.8}$$

Here the notation s_x and c_x is used as a compact form to write $\sin x$ and $\cos x$, respectively.

2.3 The UAV Model in Stability-Wind Frame Representation

For the formulation of the NMPC, based on Reinhardt's work [4], a model of the UAV in stability and wind frame representation is used instead of the model presented in Section 2.2. Doing so, the airspeed V_a, angle of attack α and sideslip angle β can be included in the state vector, as well in its OCP and constraint formulation. This way, both attitude and airspeed control can be achieved by the NMPC.

In order to obtain a globally unique and non-singular attitude representation for this model, the attitude is represented using the rotation matrix from the body-fixed frame $\{b\}$ to the inertial frame $\{n\}$, $\boldsymbol{R}_{nb}$, instead of the Euler angles that were used in (2.5), and is equivalent with $\boldsymbol{R}_{nb} = \boldsymbol{R}(\boldsymbol{\Theta}_{nb})$ given by (2.7). The matrix $\boldsymbol{R}_{nb}$ can be written as $\boldsymbol{R}_{nb} = [\boldsymbol{r}_x, \boldsymbol{r}_y, \boldsymbol{r}_z]$, where $\boldsymbol{r}_{\{x,y,z\}}$ are the axis of the body-fixed frame expressed in the coordinates of the inertial frame. Then, we define the state vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and the input $\boldsymbol{u} \in \mathbb{R}^{n_u}$ as:

$$\boldsymbol{x} = [V_a \ \beta \ \alpha \ \boldsymbol{r}_x^\top \ \boldsymbol{r}_y^\top \ \boldsymbol{r}_z^\top \ (\boldsymbol{\omega}_{nb}^s)^\top \ \delta_a \ \delta_e \ \delta_r \ \delta_t]^\top, \tag{2.9}$$

$$\boldsymbol{u} = [\dot{\delta}_a \ \dot{\delta}_e \ \dot{\delta}_r \ \dot{\delta}_t]^\top, \tag{2.10}$$

with $n_x = 19$ and $n_u = 4$. The states V_a, α and β are the airspeed, angle of attack and sideslip angle, respectively, and given by (2.1). The vector $\boldsymbol{\omega}_{nb}^s$ is the angular velocity decomposed in $\{s\}$ given by $\boldsymbol{\omega}_{nb}^s = \boldsymbol{R}_{sb}\boldsymbol{\omega}_{nb}^b$, with $\boldsymbol{R}_{sb}$ given by (2.2). The dynamic and kinematic equations for the state variables are given by:

$$\begin{bmatrix} \dot{V}_a \\ \dot{\beta} V_a \\ \dot{\alpha} V_a \cos\beta \end{bmatrix} = \frac{1}{m}(\boldsymbol{F}_a^w + \boldsymbol{R}_{wb}\boldsymbol{F}_T^b) + \boldsymbol{R}_{wb}\boldsymbol{R}_{nb}^\top \boldsymbol{g}^n - \boldsymbol{\omega}_{nb}^w \times \boldsymbol{v}_r^w \tag{2.11a}$$

$$\dot{\boldsymbol{R}}_{nb} = \begin{bmatrix} \dot{\boldsymbol{r}}_x & \dot{\boldsymbol{r}}_y & \dot{\boldsymbol{r}}_z \end{bmatrix} = \boldsymbol{R}_{nb}\boldsymbol{S}(\boldsymbol{R}_{sb}^\top \boldsymbol{\omega}_{nb}^s) \tag{2.11b}$$

$$\dot{\boldsymbol{\omega}}_{nb}^s = -\boldsymbol{\omega}_{bs}^s \times \boldsymbol{\omega}_{nb}^s + (\boldsymbol{J}^s)^{-1}(\boldsymbol{R}_{sb}\boldsymbol{\tau}^b - \boldsymbol{\omega}_{nb}^s \times \boldsymbol{J}^s\boldsymbol{\omega}_{nb}^s), \tag{2.11c}$$

where we notice that $\boldsymbol{\omega}_{nb}^{w}$ is the angular velocity vector decomposed in the wind frame $\{w\}$ and given by $\boldsymbol{\omega}_{nb}^{w} = \boldsymbol{R}_{wb}\boldsymbol{\omega}_{nb}^{b}$, with $\boldsymbol{R}_{wb}$ given by (2.4). The matrix $\boldsymbol{J}^{s}$ is the inertia matrix decomposed in the stability frame $\{s\}$, $\boldsymbol{\omega}_{bs}^{s}$ is the angular velocity of $\{s\}$ relative to the body-fixed reference frame and decomposed in $\{s\}$, and $\boldsymbol{v}_{r}^{w}$ is the relative velocity vector decomposed in $\{w\}$. They are given by

$$\boldsymbol{J}^{s} = \boldsymbol{R}_{sb}\boldsymbol{J}^{b}\boldsymbol{R}_{sb}^{\top}, \quad \boldsymbol{\omega}_{bs}^{s} = [0\ \dot{\alpha}\ 0]^{\top}, \quad \text{and} \quad \boldsymbol{v}_{r}^{w} = [V_a\ 0\ 0]^{\top}, \tag{2.12}$$

with $\boldsymbol{J}^{b}$ given by (2.6).

2.4 Forces and Moments

The total forces and moments acting on the UAV will be described in this section. They are given by:

$$\boldsymbol{F}^{b} = \boldsymbol{F}_{g}^{b} + \boldsymbol{F}_{a}^{b} + \boldsymbol{F}_{t}^{b}, \tag{2.13}$$

$$\boldsymbol{M}^{b} = \boldsymbol{M}_{a}^{b} + \boldsymbol{M}_{p}^{b}, \tag{2.14}$$

where the total force in body-fixed reference frame $\boldsymbol{F}^{b}$ is given by the sum of the gravity force $\boldsymbol{F}_{g}^{b}$, the aerodynamic force $\boldsymbol{F}_{a}^{b}$ and the propulsion force $\boldsymbol{F}_{p}^{b}$. The total moment in body-fixed frame is equal to the sum of the aerodynamic moment $\boldsymbol{M}_{a}^{b}$ and the propulsion moment $\boldsymbol{M}_{p}^{b}$. In this case, however, the propulsion moment is assumed to be very small, and due to its modelling complexity in the NMPC problem, it was assumed to be negligible, so $\boldsymbol{M}_{p}^{b} \approx 0$.

2.4.1 Aerodynamic forces and moments

The aerodynamic forces in body-fixed reference frame are given by

$$\boldsymbol{F}_{a}^{b} = \boldsymbol{R}_{wb}^{\top}\boldsymbol{F}_{a}^{w} = \boldsymbol{R}_{wb}^{\top}\begin{bmatrix} -F_{drag} \\ F_{side} \\ -F_{lift} \end{bmatrix}, \tag{2.15}$$

with the drag force F_{drag}, side force F_{side} and lift force F_{lift} given by:

$$\begin{bmatrix} F_{drag} \\ F_{side} \\ F_{lift} \end{bmatrix} = \frac{1}{2}\rho V_a^2 S \begin{bmatrix} C_D(\alpha, q, \delta_e) \\ C_S(\beta, p, r, \delta_a, \delta_r) \\ C_L(\alpha, q, \delta_e) \end{bmatrix}, \tag{2.16}$$

where ρ is the air density and S is the wing area, given in Table 4.6. Based on Beard & McLain [9] and Winter [12], the aerodynamic coefficients are given by:

$$C_D = C_D(\alpha) + C_{D_q}(\alpha)\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e, \tag{2.17}$$

$$C_S = C_S(\beta) + C_{S_p}(\beta)\frac{b}{2V_a}p + C_{S_r}(\beta)\frac{b}{2V_a}r + C_{S_{\delta_a}}\delta_a + C_{S_{\delta_r}}\delta_r, \tag{2.18}$$

$$C_L = C_L(\alpha) + C_{L_q}(\alpha)\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e, \tag{2.19}$$

where b is the wingspan and c is the mean chord, given in Table 4.6. The aerodynamic moment vector $\boldsymbol{M}_a^b$, modelled in the body-fixed frame, is given by:

$$\boldsymbol{M}_a^b = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \frac{1}{2}\rho V_a^2 S \begin{bmatrix} bC_l(\beta, p, r, \delta_a, \delta_r) \\ cC_m(\alpha, q, \delta_e) \\ bC_n(\beta, p, r, \delta_a, \delta_r) \end{bmatrix}, \tag{2.20}$$

where the aerodynamic coefficients are given by:

$$C_l = C_l(\beta) + C_{l_p}(\beta)\frac{b}{2V_a}p + C_{l_r}(\beta)\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r, \tag{2.21}$$

$$C_m = C_m(\alpha) + C_{m_q}(\alpha)\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e, \tag{2.22}$$

$$C_n = C_n(\beta) + C_{n_p}(\beta)\frac{b}{2V_a}p + C_{n_r}(\beta)\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r. \tag{2.23}$$

2.4.2 Gravitational forces

The gravitational force acts on the $\boldsymbol{z}^n$ axis direction, and following Newton's second law it is equal to $g_z^n = mg$, where m is the mass of the UAV and g is the gravitational constant. In body-fixed reference frame, the gravitational force is then given by:

$$\boldsymbol{F}_g^b = \boldsymbol{R}_{nb}^\top m\boldsymbol{g}^n = \boldsymbol{R}_{nb}^\top \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \tag{2.24}$$

where $\boldsymbol{R}_{nb}$ is given by (2.7).

2.4.3 Propulsion forces

The propeller model used is based on Coates [16], and describes the propeller thrust T as a function of the thrust coefficient C_T:

$$T = \frac{\rho D^4}{2\pi^2}C_T\omega^2, \tag{2.25}$$

where ρ is the air density, ω is the propeller speed in rad/s and D is the propeller diameter. The propeller on the X8 Skywalker is the Aeronaut CamCarbon 14x8", which is 14 inches in diameter, giving $D = 0.3556\,m$. The thrust coefficient C_T is given as a first order approximation, which depends on the advance ratio J:

$$C_T(J) = C_{T,0} + C_{T,1}J, \tag{2.26}$$

$$J = \frac{2\pi V_a}{\omega D}, \tag{2.27}$$

where V_a is the airspeed, and $C_{T,0} = 0.126$, $C_{T,1} = -0.1378$. As the propeller force is generated in the body x-axis, it is given by:

$$\boldsymbol{F}_{prop}^{b} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\rho D^4}{2\pi^2} C_T(J)\omega^2 \\ 0 \\ 0 \end{bmatrix}. \tag{2.28}$$

In the previous work by Högnadóttir [6], a propeller model based on Fitzpatrick[17] and Beard & McLain [18] was used, this is also given in Appendix A.

2.5 Actuation

Throughout Section 2.4.1, the variables δ_a, δ_e, δ_r and δ_t have been recurrent in various equations. They are the aileron, elevator, rudder and throttle, respectively, and they are the typical control surfaces fixed-wing aerial vehicles have. By deflecting them, they alter the aerodynamic forces and moments of the UAV, can and subsequently control attitude. Although there is coupling between the states, δ_a mainly controls roll, while δ_e and δ_r mainly control pitch and yaw, respectively. The throttle, on the other hand, is responsible for altering the propulsion forces. When it comes to the Skywalker X8, it does not have a rudder, and it is equipped with a throttle and two pairs of elevons, replacing the aileron and the elevator. There is a direct relationship between aileron and elevator and the left and right elevons, δ_{el} and δ_{er}, which is given by

$$\begin{bmatrix} \delta_{er} \\ \delta_{el} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix}. \tag{2.29}$$

Because of this, δ_a and δ_e are still used when calculating the forces and moments, but they are mapped to the pair δ_{er} and δ_{el} when applying the actuation to the physical system.

2.6 Icing model

To simulate icing on the aircraft, two models are used: one to simulate asymmetric icing on the wings, and the other one for icing on the propeller.

2.6.1 Asymmetric icing on the wings

The asymmetric icing model for the wings was developed in Kleiven's thesis [7], by considering asymmetry in the aerodynamic forces and moments acting on the aircraft. In the model, the UAV is divided in a left and right side, so that the aerodynamic forces acting on it are given by

$$\boldsymbol{F}_k = \boldsymbol{F}_{k,r} + \boldsymbol{F}_{k,l}, \quad \text{for } k = drag, lift, side, \tag{2.30}$$

where $\boldsymbol{F}_{k,r}$ and $\boldsymbol{F}_{k,l}$ represent the drag, lift and side forces acting on the right and left side, respectively. On the other hand, the asymmetric aerodynamic moment is given by

$$\boldsymbol{M}_{a,asym} = \boldsymbol{M}_{a,0} + \sum_k (\boldsymbol{r}_k \times \boldsymbol{F}_{k,r} + \boldsymbol{l}_k \times \boldsymbol{F}_{k,l}), \qquad \text{for } \boldsymbol{F}_{k,r}, \boldsymbol{F}_{k,l} \notin \boldsymbol{M}_{a,0}, \tag{2.31}$$

where $\boldsymbol{M}_{a,0}$ is the symmetric aerodynamic moment, given in (2.20), and the second term describes the asymmetry. The vectors $\boldsymbol{r}_k$ and $\boldsymbol{l}_k$ represent the distance from the centre of mass to the point of attack on the right and left wing, respectively. The asymmetric forces and distances can be visualized in Figure 2.3, from Kleiven [7].

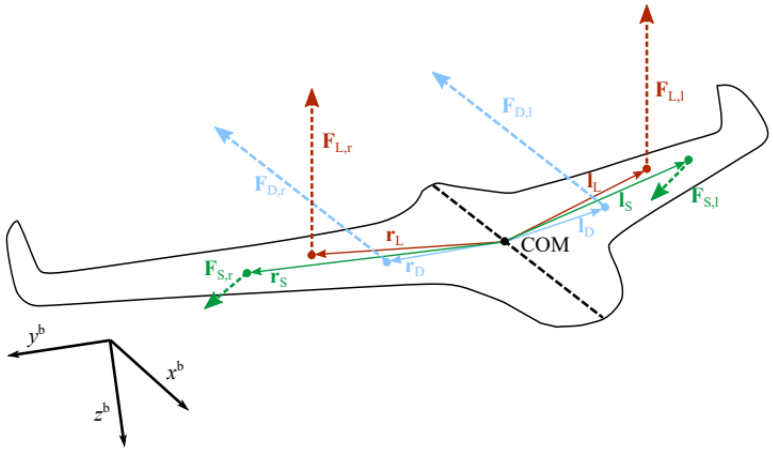

Figure 2.3: Asymmetric forces, from Kleiven [7].

2.6.2 Icing on propeller

Based on the work of Müller [15], a model for the thrust coefficient in icing conditions can be given by:

$$C_{T,iced} = C_T(J)(1 + min(TWC, TWC_{max})\Delta C_T(T)), \tag{2.32}$$

$$\Delta C_T(T) = \Delta C_{T,0} + \Delta C_{T,1} T + \Delta C_{T,2} T^2 \tag{2.33}$$

where $C_T(J)$ is given by (2.26), T is the temperature in $[^\circ C]$, TWC is the total water collected on the propeller, and TWC_{max} is the maximum amount of water collected on the propeller before an ice shedding event is expected. They are given by:

$$TWC = t\, LWC\, \omega \frac{D}{2}, \tag{2.34}$$

$$TWC_{max} = \frac{A_{max}}{\frac{D}{2}\omega^2}, \tag{2.35}$$

$$A_{max} = A_{max,0} + A_{max,1} T^2. \tag{2.36}$$

Eq. (2.36) describes the adhesion forces of the ice on the propeller, given by a second-order approximation and dependent on the temperature T. As the forces acting on the ice on the propeller are centrifugal and aerodynamic forces caused by the rotation of the propeller, they are proportional to the rotation rate, as seen in Eq. (2.35). In the same equation it can also be seen that A_{max} is calculated depending on the maximum amount of ice that can be collected on the propeller before ice shedding occurs. This relationship shows that there is an equilibrium between A_{max} and TWC_{max}, when the amount of ice collected is too large, the adhesion forces can no longer hold it, and the shedding occurs.
Similarly, icing also affects the power P and power coefficient C_P, given by the following set of equations, as described in Müller [15]:

$$P = C_P \rho n^3 D^5, \quad n = \frac{\omega}{2\pi} \tag{2.37a}$$

$$C_{P,iced} = C_P(J)(1 + min(TWC, TWC_{max})\Delta C_P(T)), \tag{2.37b}$$

$$C_P(T) = \Delta C_{P,0} + \Delta C_{P,1} T + \Delta C_{P,2} T^2, \tag{2.37c}$$

$$\Delta C_P(T) = \Delta C_{P,0} + \Delta C_{P,1} T + \Delta C_{P,2} T^2. \tag{2.37d}$$

2.7 Wind model

To simulate wind disturbances, the wind gusts are usually modeled by the von Karmen model, given in Beard and McLain [9]. In this thesis, similarly to Högnadóttir and Kleiven's previous work [6], [7], an approximation of the von Karman model is used, called the Dryden model. The model consists of six filters with white noise passing through, describing the wind effect on the transnational and rotational velocities, given based on the military specification MIL-F-8785C [19] as:

$$H_u(s) = \sigma_u \sqrt{\frac{2L_u}{\pi V_a}} \frac{1}{1 + \frac{L_u}{V_a} s}, \qquad H_p(s) = \sigma_w \sqrt{\frac{0.8}{V_a}} \frac{(\frac{\pi}{4b})^{\frac{1}{6}}}{L_w^{\frac{1}{3}}(1 + \frac{4b}{\pi V_a} s)},$$

$$H_v(s) = \sigma_v \sqrt{\frac{L_v}{\pi V_a}} \frac{1 + \frac{\sqrt{3}L_v}{V_a} s}{(1 + \frac{L_v}{V_a} s)^2}, \qquad H_q(s) = \frac{-\frac{s}{V_a}}{1 + \frac{4b}{\pi V_a} s} H_w(s),$$

$$H_w(s) = \sigma_w \sqrt{\frac{L_w}{\pi V_a}} \frac{1 + \frac{\sqrt{3}L_w}{V_a} s}{(1 + \frac{L_w}{V_a} s)^2}, \qquad H_r(s) = \frac{\frac{s}{V_a}}{1 + \frac{3b}{\pi V_a} s} H_v(s),$$

where b is the wingspan of the UAV, $\sigma_{\{u,v,w\}}$ are the turbulance intensities and $L_{\{u,v,w\}}$ are the turbulence scale lengths, given by

$$L_w = h, \qquad \sigma_w = 0.1 V_{20},$$

$$L_u = L_v = \frac{h}{(0.177 + 0.000823h)^{1.2}}, \qquad \frac{\sigma_u}{\sigma_w} = \frac{\sigma_v}{\sigma_w} = \frac{1}{(0.177 + 0.000823h)^{0.4}},$$

which are given for altitudes below 1000 feet, with V_{20} being the wind speed at $20\,ft \approx 6.1\,m$ [19].

Chapter 3

Controllers

3.1 Nonlinear Model Predictive Control (NMPC)

In Model Predictive Control, an optimization problem is formulated based on the reference to be tracked, with the key feature of being able to include constraints for the states in the state vector used in the problem formulation. After solving the optimization problem over a specific time horizon, it predicts the optimal states and input over the horizon length, and the fist input in the sequence is implemented. This allows for the predicted states to be adjusted and for the optimal input to be calculated every time.
In this section, the NMPC controller used in the thesis for the inner-loop control of roll, pitch and airspeed will be introduced and explained. The NMPC controller proposed is based on the work of Reinhardt [4]. The NMPC uses the dynamic model described in Section 2.3, where Eq. 2.11 is denoted by the continuous ODE $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$. This is discretized using an explicit Runge-Kutta method of order 4, becoming $\boldsymbol{x}(k+1) = \boldsymbol{f}_{RK4}(\boldsymbol{x}(k), \boldsymbol{u}(k))$, and used where applicable.

3.1.1 Reduced attitude parametrization

For the purpose of roll and pitch control, the reduced attitude vector $\boldsymbol{\Gamma} \in \mathbb{S}^2$ is used, following Reinhardt's work [4]. It is defined as the representation of the vertical axis of the inertial frame $\boldsymbol{e}_3 = [0,\ 0,\ 1]^\top$, expressed in the body frame:

$$\boldsymbol{\Gamma} = \boldsymbol{R}_{nb}^\top \boldsymbol{e}_3. \tag{3.1}$$

The reduced attitude vector can be parameterized as:

$$\boldsymbol{\Gamma}(\phi, \theta) = \begin{bmatrix} -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}^\top, \tag{3.2}$$

were roll is given by $\phi \in [-\pi,\ \pi]$ and pitch is given by $\theta \in [-\frac{\pi}{2},\ \frac{\pi}{2}]$. As seen, it is also independent of yaw.

3.1.2 Disturbance observer

Because of unmodelled dynamics, such as the icing models previously presented in Section 2.6, or other disturbances like the wind conditions, a disturbance observer is included with the purpose of obtaining offset-free attitude stabilization. The disturbance observer presented is based on Reinhardt [4], and affects airspeed V_a, sideslip angle β, angle of attack α and angular velocity ω_{nb}^b. The difference between the observed state and the one predicted by the controller at each time instant is given by:

$$\begin{bmatrix} \Delta V_a(t) \\ \Delta\beta(t) \\ \Delta\alpha(t) \end{bmatrix} = \begin{bmatrix} V_a(t) \\ \beta(t) \\ \alpha(t) \end{bmatrix} - \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times(n_x-3)} \end{bmatrix} \boldsymbol{x}^*(1|t-1) \tag{3.3}$$

$$\Delta\boldsymbol{\omega}_{nb}^b(t) = \boldsymbol{\omega}_{nb}^b(t) - \boldsymbol{R}_{bs} \begin{bmatrix} \boldsymbol{0}_{3\times12} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times4} \end{bmatrix} \boldsymbol{x}^*(1|t-1) \tag{3.4}$$

The disturbance estimates are then updated together with the NMPC by

$$\boldsymbol{d}_f(t) \leftarrow \boldsymbol{d}_f(t) + \boldsymbol{L}_f \begin{bmatrix} \Delta V_a(t) & \beta(t) & \alpha(t) \end{bmatrix}^\top, \tag{3.5}$$

$$\boldsymbol{d}_m(t) \leftarrow \boldsymbol{d}_m(t) + \boldsymbol{L}_m \Delta\omega_{nb}^b(t), \tag{3.6}$$

where $\boldsymbol{d}_f$ and $\boldsymbol{d}_m$ are the force and moment disturbances, respectively, and initialized as $\boldsymbol{d}_f(0) = \boldsymbol{d}_m(0) = \boldsymbol{0}_{3\times1}$, and with $\boldsymbol{L}_f$ and $\boldsymbol{L}_m$ given by

$$\boldsymbol{L}_f = \text{diag}(l_{V_a}, l_\beta, l_\alpha), \quad \boldsymbol{L}_m = \text{diag}(l_p, l_q, l_r). \tag{3.7}$$

The gains l_{V_a}, l_β, l_α, l_p, l_q, l_r are given in Table 4.3, and the disturbances are finally added to $\boldsymbol{f}$:

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) \triangleq \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \begin{bmatrix} \boldsymbol{d}_f^\top & \boldsymbol{0}_{1\times3} & \boldsymbol{d}_m^\top \end{bmatrix}^\top \tag{3.8}$$

3.1.3 Nonlinear Program

The reference vector to be tracked is defined as

$$\boldsymbol{r} \triangleq \begin{bmatrix} V_{a,ref} & \boldsymbol{\Gamma}_{ref}^\top \end{bmatrix}^\top, \tag{3.9}$$

where $\boldsymbol{\Gamma}_{ref}$ is parameterized using the references for roll and pitch, ϕ_{ref} and θ_{ref}, as shown in Eq. (3.2). The way the references angles have been generated in this thesis is explained in Section 4.1. Next, to formulate the NMPC scheme, the stage cost is defined as

$$l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{r}) = q_{V_a}(V_a - V_{a,ref})^2 + ||\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{ref}||_{\boldsymbol{Q}_\Gamma}^2 + ||\boldsymbol{u}||_{\boldsymbol{R}}^2, \tag{3.10}$$

which is a sum of quadratic terms, where q_{V_a} is the weighting scalar that penalizes the V_a error, and $\boldsymbol{Q}_\Gamma \in \mathbb{R}^\times$ and $\boldsymbol{R} \in \mathbb{R}^{n_u \times n_u}$ are the weighting matrices for attitude

error and input, respectively. The OCP over a prediction horizon T is then given by:

$$\min_{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)} \quad \int_0^T l(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\boldsymbol{r}(\tau))\,d\tau + \frac{1}{2}\boldsymbol{s}^\top \boldsymbol{P}\boldsymbol{s}, \quad t\in[0,T) \tag{3.11a}$$

$$\text{s. t.} \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{3.11b}$$

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{d}(0)), \quad t\in[0,T) \tag{3.11c}$$

$$\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{s}) \geq \boldsymbol{0}, \quad t\in[0,T) \tag{3.11d}$$

where $\boldsymbol{f}(\boldsymbol{x}(t),\,\boldsymbol{u}(t),\,\boldsymbol{d}(0))$ is the augmented system given in equation (3.8), and the term $\frac{1}{2}\boldsymbol{s}^\top\boldsymbol{P}\boldsymbol{s}$ was added to penalize the vector of slack variables $\boldsymbol{s} = [\boldsymbol{s}_{\underline{x}}^\top \;\boldsymbol{s}_{\overline{x}}^\top]^\top$ with the symmetric, positive-definite matrix $\boldsymbol{P}\in\mathbb{R}^{n_s\times n_s}$. All constraints on state $\boldsymbol{x}$ and output $\boldsymbol{u}$, as well as their respective slack variables, are included in the constraint $\boldsymbol{h}(x(t),\,\boldsymbol{u}(t),\,\boldsymbol{s})$. This is explained in Section 3.1.4. Discretizing 3.11 into N steps with shooting interval $\Delta t = T/N$ gives the nonlinear problem (NLP):

$$\min_{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)} \quad \sum_{k=0}^{N-1} l(\boldsymbol{x}(k|t),\boldsymbol{u}(k|t),\boldsymbol{r}(k|t)) + \frac{1}{2}\boldsymbol{s}^\top\boldsymbol{P}\boldsymbol{s}, \quad k\in[0,\ldots,N) \tag{3.12a}$$

$$\text{s. t.} \quad \boldsymbol{x}(0|t) = \boldsymbol{x}(t) \tag{3.12b}$$

$$\boldsymbol{x}(k+1|t) = f_{RK4}(\boldsymbol{x}(k|t),\boldsymbol{u}(k|t),\boldsymbol{d}(0|t)), \quad k\in[0,\ldots,N) \tag{3.12c}$$

$$\boldsymbol{h}(\boldsymbol{x}(k|t),\boldsymbol{u}(k|t),\boldsymbol{s}) \geq 0, \quad k\in[0,\ldots,N), \tag{3.12d}$$

where $\boldsymbol{x}(\cdot|t)\in\mathbb{R}^{n_x\times(N+1)}$ and $\boldsymbol{u}(\cdot|t)\in\mathbb{R}^{n_u\times N}$ are the predicted state and input sequence, respectively. The MPC scheme is based on solving the NLP given by Eq. 3.12 at time t, for the predictions $k\in[0,\ldots,N]$.

As the variables $[\delta_a,\delta_e,\delta_r,\delta_t]^\top$ are included in the state vector $\boldsymbol{x}$, the control input applied to the UAV is extracted from the optimal state after one shooting interval, given by $\boldsymbol{x}^*(1|t)$, obtained after solving the NLP. This is given by:

$$\boldsymbol{u}_{uav}(t) = \begin{bmatrix}\boldsymbol{0}_{n_u\times(n_x-n_u)} & \boldsymbol{I}_{n_u\times n_u}\end{bmatrix}\boldsymbol{x}^*(1|t) \tag{3.13}$$

3.1.4 Constraints

One of the key features of the NMPC is the possibility to add constraints on the state and output as a part of the NLP to be solved. This takes into consideration safety and the physical limits og the UAV. Let the constraints on the state $\boldsymbol{x}$ and the constraints on the input $\boldsymbol{u}$ to be given by $\mathcal{X}$ and $\mathcal{U}$, respectively:

$$\mathcal{X} \triangleq \{\boldsymbol{x}\in\mathbb{R}^{n_x} | \boldsymbol{h}(\boldsymbol{x},\boldsymbol{s}) \geq 0\} \tag{3.14}$$

$$\mathcal{U} \triangleq \{\boldsymbol{x}\in\mathbb{R}^{n_u} | \boldsymbol{u}-\underline{\boldsymbol{u}} \geq \boldsymbol{0} \wedge -\boldsymbol{u}+\overline{\boldsymbol{u}} \geq \boldsymbol{0}\} \tag{3.15}$$

When choosing the constraints on the state $\boldsymbol{x}$, limits on airspeed, sidelslip and angle of attack are set for safety reasons and to avoid stalling. The same is true

for the limits on the angular rates, and on the actuators, which express the true surface deflection limits of the elevons. All state constraints are given by:

$$\begin{aligned}
V_a - \underline{V}_a + \underline{s}_{V_a} &\geq 0 & -V_a + \overline{V}_a + \overline{s}_{V_a} &\geq 0 \\
\beta - \underline{\beta} + \underline{s}_\beta &\geq 0 & -\beta + \overline{\beta} + \overline{s}_\beta &\geq 0 \\
\alpha - \underline{\alpha} &\geq 0 & -\alpha + \overline{\alpha} &\geq 0 \\
p_s - \underline{p}_s &\geq 0 & -p_s + \overline{p}_s &\geq 0 \\
q_s - \underline{q}_s &\geq 0 & -q_s + \overline{q}_s &\geq 0 \\
r_s - \underline{r}_s &\geq 0 & -r_s + \overline{r}_s &\geq 0 \\
\delta_a + \delta_e - \underline{\delta}_{el} &\geq 0 & -\delta_a - \delta_e + \overline{\delta}_{el} &\geq 0 \\
-\delta_a + \delta_e - \underline{\delta}_{er} &\geq 0 & \delta_a - \delta_e - \overline{\delta}_{er} &\geq 0 \\
\delta_t - \underline{\delta}_t &\geq 0 & -\delta_t + \overline{\delta}_t &\geq 0,
\end{aligned} \tag{3.16}$$

where the slack variables ensure the feasibility of the quadratic problem by allowing constraint relaxation, and are given by

$$\boldsymbol{s} = [\boldsymbol{s}_{Va}^\top \quad \boldsymbol{s}_\beta^\top]^\top \geq 0. \tag{3.17}$$

Eq. (3.16) can be summarised by the function $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{s}) \geq 0$, and (Eq. 3.15) with (Eq. 3.16) can be written together as $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{s}) \geq 0$.

3.2 Roll and Pitch Control using PID and MRAC

In the previous work of Högnadóttir [6], there were two inner-loop controllers of pitch and roll presented: a proportional-integral-derivative (PID) controller and a model reference adaptive controller (MRAC). In this thesis, the NMPC will be compared to both, and they will therefore be introduced in this section. When it comes to airspeed control, a proportional-integral (PI) controller is used, which will be introduced in Section 3.3.

3.2.1 PID

The PID controller used in Högnadóttir's work [6] is based on Beard & McLain [9], and it is formed by two PID controllers, one for roll and one for pitch. The roll PID controller determines the aileron δ_a needed to drive ϕ to the commanded angle ϕ_{cmd}:

$$\delta_a = k_{p_\phi}(\phi_{cmd} - \phi) + \frac{k_{i_\phi}}{s}(\phi_{cmd} - \phi) - k_{d_\phi} p, \tag{3.18}$$

where the control gains k_{p_ϕ}, k_{i_ϕ} and k_{d_ϕ} are given in Table 4.5a.
Similarly, the controller for pitch is a PID controller which determines the elevator δ_e needed to drive θ to the commanded angle θ_{cmd}:

$$\delta_e = k_{p_\theta}(\theta_{cmd} - \theta) + \frac{k_{i_\theta}}{s}(\theta_{cmd} - \theta) - k_{d_\theta} q, \tag{3.19}$$

where the control gains k_{p_θ}, k_{i_θ} and k_{d_θ} are given in Table 4.5b.

3.2.2 Model Reference Adaptive Control (MRAC)

The MRAC controllers used in Högnadóttir's work [6] are based on Lavretsky and Wise [10], and the most important parts of the theory will be explained in this section.
A nonlinear system can be given on the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{\Lambda}(\boldsymbol{u} + \boldsymbol{\Theta}^\top \boldsymbol{\Phi}(\boldsymbol{x})), \tag{3.20}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the unknown state matrix, $B \in \mathbb{R}^{n\times m}$ is the known control matrix, $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ is the unknown control effectiveness matrix, $\boldsymbol{\Theta} \in \mathbb{R}^{N\times m}$ is constant and unknown and $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^N$ is the known regressor vector [10]. The objective is to track the reference model given by

$$\dot{\boldsymbol{x}}_{ref} = \boldsymbol{A}_{ref}\boldsymbol{x}_{ref} + \boldsymbol{B}_{ref}\boldsymbol{r}(t), \tag{3.21}$$

where $\boldsymbol{r}(t)$ is the commanded reference. This is achieved with the control law:

$$\boldsymbol{u} = \hat{\boldsymbol{K}}_x^\top \boldsymbol{x} + \hat{\boldsymbol{K}}_r^\top \boldsymbol{r} - \hat{\boldsymbol{\Theta}}^\top \boldsymbol{\Phi}(\boldsymbol{x}), \tag{3.22}$$

where $\hat{\boldsymbol{K}}_x$, $\hat{\boldsymbol{K}}_r$ and $\hat{\boldsymbol{\Theta}}$ are the controller gain matrices, given by:

$$\dot{\hat{\boldsymbol{K}}}_x = \text{Proj}(\hat{\boldsymbol{K}}_x, -\boldsymbol{\Gamma}_{\boldsymbol{x}}\boldsymbol{x}\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}), \tag{3.23}$$

$$\dot{\hat{\boldsymbol{K}}}_r = \text{Proj}(\hat{\boldsymbol{K}}_r, -\boldsymbol{\Gamma}_{\boldsymbol{r}}\boldsymbol{r}\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}), \tag{3.24}$$

$$\dot{\hat{\boldsymbol{\Theta}}}_x = \text{Proj}(\hat{\boldsymbol{\Theta}}_x, -\boldsymbol{\Gamma}_{\boldsymbol{\Theta}}\boldsymbol{\Phi}\boldsymbol{e}^\top\boldsymbol{P}\boldsymbol{B}). \tag{3.25}$$

Proj$(\cdot)$ is a projection operator, defined in [10], the symmetric and positive-definite matrices $\boldsymbol{\Gamma}_{\boldsymbol{x}}$, $\boldsymbol{\Gamma}_{\boldsymbol{r}}$ and $\boldsymbol{\Gamma}_{\boldsymbol{\Theta}}$ are the adaptive rates, and $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_{ref}$ is the error between the state and the reference.
When it comes to roll and pitch tracking specifically, two MRAC control schemes are developed, chosen as linear models with a bias term to capture the nonlinearities and unmodelled effects. From Beard & McLain [9], the roll dynamics are linearized as:

$$\dot{\phi} = p + d_{\phi_1}, \tag{3.26}$$

$$\ddot{\phi} = -a_{\phi_1}\dot{\phi} + a_{\phi_2}\delta_a + d_{\phi_2}, \tag{3.27}$$

where d_{ϕ_1} and d_{ϕ_2} are considered the disturbances of the system. The dynamics written on the same form as Eq. (3.20) are given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{\Lambda}(\boldsymbol{u} + \boldsymbol{\Theta}^\top \boldsymbol{\Phi}(\boldsymbol{x})), \tag{3.28}$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & a_1\end{bmatrix}\begin{bmatrix}\phi\\ p\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\lambda_1\left(\delta_a + [\theta_{bias_{roll}}][1]\right), \tag{3.29}$$

with $a_1 = -a_{\phi,1}$, $\lambda_1 = a_{\phi,2}$, $\theta_{bias,roll} = d_{\phi,2}$, where $(\boldsymbol{A}, \boldsymbol{B\Lambda})$ is controllable for all $a_{\phi,2} \neq 0$. Likewise, the linearized pitch dynamics [9] are given as

$$\dot{\theta} = q + d_{\theta_1}, \tag{3.30}$$

$$\ddot{\theta} = -a_{\theta_1}\dot{\theta} - a_{\theta_2}\theta + a_{\theta_3}\delta_e + d_{\theta_2}, \tag{3.31}$$

which on the same form as Eq. (3.20) they become

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B\Lambda}(\boldsymbol{u} + \boldsymbol{\Theta}^\top \boldsymbol{\Phi}(\boldsymbol{x})), \tag{3.32}$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_2 & a_3 \end{bmatrix} \begin{bmatrix} \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \lambda_2 \left(\delta_e + [\theta_{bias,pitch}][1] \right), \tag{3.33}$$

with $a_2 = -a_{\theta,2}$, $a_3 = -a_{\theta,1}$, $\lambda_2 = a_{\theta,3}$ and $\theta_{bias,pitch} = d_{\theta,2}$, where $(\boldsymbol{A}, \boldsymbol{B\Lambda})$ is controllable for all $a_{\theta,3} \neq 0$.

3.3 Airspeed Controller

The airspeed PI controller determines the throttle δ_t needed to drive V_a to the commanded airspeed $V_{a,cmd}$:

$$\delta_t = \delta_t^* + k_{p_V}(V_{a,cmd} - V_a) + \frac{k_{i_V}}{s}(V_{a,cmd} - V_a), \tag{3.34}$$

where δ_t^* is the throttle trim value, and it is given together with the control gains k_{p_V} and k_{i_V} in Table 4.1.

Chapter 4

Method

All simulations were performed in a the Matlab/ Simulink simulator, first developed by Gryte [11] and then extended by Högnadóttir [6]. The simulator included reference generation, the PID and MRAC controllers, as well as the physical model with the UAV dynamics, icing and wind models, and actuation. In this thesis, a NMPC controller has been added, as well as modifications in the thrust model and the effects of ice accretion on the propeller.

The simulator follows the classic control loop, where the reference is generated and the error between it and the state is found, which gets sent into the controller. The input from the controller is then sent to the physical model, where the new state is found. This chapter goes through how the reference is generated, the implementation and tuning of the controller as well as the implementation of the UAV and icing model, as seen in Figure 4.1. In addition, the simulations that were run to test the controllers will also be explained here.

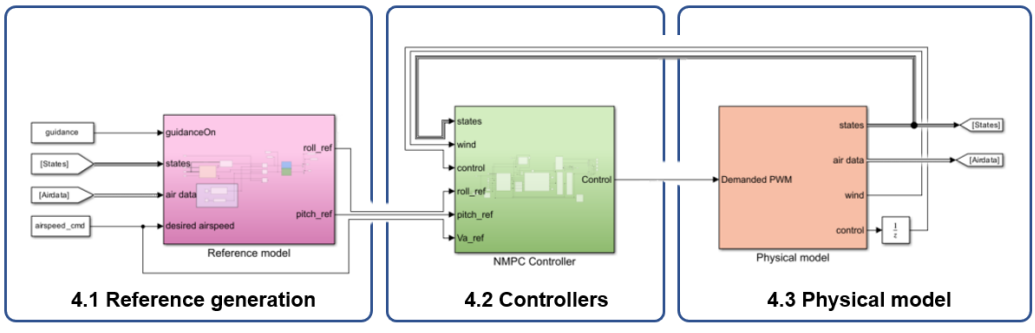

Figure 4.1: Overview of the simulator in Matlab/ Simulink, the subsections in which each part is explained.

4.1 Reference Generation

To compare the NMPC response to the previously developed PID and MRAC controllers, the references were also generated the same way as Högnadottir did in her work [6], and the methods are paraphrased in this section.

4.1.1 Roll and Pitch Reference Model

In the following simulations, it is desired to drive roll and pitch to certain commanded angles, ϕ_{cmd} and θ_{cmd}. The reference model is described as a second order transfer function:

$$\frac{\phi_{ref}}{\phi_{cmd}}(s) = \frac{\omega_{n,\phi}^2}{s^2 + 2\zeta_\phi \omega_{n,\phi} s + \omega_{n,\phi}^2}, \tag{4.1a}$$

$$\frac{\theta_{ref}}{\theta_{cmd}}(s) = \frac{\omega_{n,\theta}^2}{s^2 + 2\zeta_\theta \omega_{n,\theta} s + \omega_{n,\theta}^2}, \tag{4.1b}$$

where $\omega_{n,\{\phi,\theta\}}$ and $\zeta_{\{\phi,\theta\}}$ are the natural frequency and the damping factor, respectively. They are design parameters chosen depending on the desired reference model response. Using Högnadottir's results [6], they are set as $\omega_{n,\phi} = \omega_{n,\phi} = 4$ and $\zeta_\phi = \zeta_\theta = 1$.

4.1.2 References from Guidance Controller

In addition to the simulations in which the UAV follows some commanded pitch and roll angles, a path-following scenario can also be simulated. In this case, the roll and pitch commanded angles are found by a guidance controller, based on the path, which is given as a set of waypoints. The guidance laws are given in the lateral and longitudinal directions. Once ϕ_{cmd} and θ_{cmd} are found from the guidance controller, the roll and pitch reference model described in Section 4.1.1 are used to find the references for the PID and MRAC controllers.

Based on Fossen [20], the propotional line-of-sight (LOS) guidance law is given by a course controller, which tracks a desired course angle χ_d and calculates the corresponding roll angle reference as:

$$\phi_{cmd} = k_{p_\chi}(\chi_{cmd} - \chi) + \frac{k_{i_\chi}}{s}(\chi_{cmd} - \chi). \tag{4.2}$$

The desired course angle χ_d is given by:

$$\chi_d = \chi_p - \tan^{-1}\left(\frac{y_e^p}{\Delta_{lat}}\right), \tag{4.3}$$

where the path-tangential angle χ_p and the cross-track error y_e are found by:

$$\chi_p = \text{atan2}(y_{k+1} - y_k,\, x_{k+1} - x_k) \tag{4.4}$$
$$y_e = -(x(t),\, x_k)\sin\chi_p + (y(t) - y_k)\cos\chi_p. \tag{4.5}$$

The current position of the UAV is given by $(x(t), y(t))$, while the position of the current waypoint and the next waypoint are given by (x_k, y_k) and (x_{k+1}, y_{k+1}), respectively.

Similarly, based on Nevstad [21] and You [22], a flight path angle controller, which tracks a desired flight path angle γ_d and calculates the corresponding pitch angle reference is given by:

$$\theta_{cmd} = -c_1(\gamma - \gamma_d) + \gamma_d + \alpha_{trim}, \tag{4.6}$$

where the desired flight path angle is given by:

$$\gamma_d = \gamma_p + \tan^{-1}\left(\frac{k_{ph} z_e + k_{ih}\int z_e d\tau}{\Delta_{lon}}\right). \tag{4.7}$$

The longitudinal look-ahead distance Δ_{lon} is given by $\Delta_{lon} = \sqrt{R_{max}^2 - z_e^2}$, while γ_p and the vertical cross-track error z_e are found by:

$$\gamma_p = \text{atan2}(-(z_{k+1} - z_k), L_{xy}), \tag{4.8}$$
$$z_e = S_{xy}\sin(\gamma_p) + (z(t) - z_k)\cos(\gamma_p). \tag{4.9}$$

S_{xy} and L_{xy} are the projection of the along-track distance and the projection of the path onto the xy-plane, respectively, and given by:

$$S_{xy} = \cos(\chi_p)(x(t) - x_k) + \sin(\chi_p)(y(t) - y_k) \tag{4.10}$$
$$L_{xy} = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}. \tag{4.11}$$

Finally, when using the guidance controller references, the airspeed controller is modified to include the feed-forward term $k_{p_{z_e}} z_e(t)$, and given by:

$$\delta_t = \delta_t^* + k_{p_V}(V_{a,cmd} - V_a) + \frac{k_{i_V}}{s}(V_{a,cmd} - V_a) + k_{p_{z_e}} z_e(t). \tag{4.12}$$

The tuning parameters used in the guidance controller come from the results of Högnadottir's work [6], and given in Table 4.1.

Parameter	Value
Δ_{lat}	33
k_{p_χ}	2
k_{i_χ}	0.1
k_{ph}	0.8
k_{ih}	0.1
R_{max}	20°
c_1	0.001
k_{p_V}	0.6
k_{i_V}	0.01
$k_{p_{z_e}}$	0.1
δ^*	0.44

Table 4.1: Tuning parameters of the guidance controller

4.2 Implementation of the Controllers

4.2.1 NMPC

The NMPC described in Section 3.1 was implemented in Python using the software package Acados [8], based on the controllers developed by Reinhardt in his doctoral thesis [4]. As it was implemented in Python, the UAV model used was slightly different, the difference being how the aerodynamic forces and moments were modelled, based on the model and parameters from Gryte [23], and not including the effects of icing on the aerodynamic coefficients. Therefore, the differences in the model are assumed to be disturbances, and the disturbance observer is expected to be able to handle them.

In Reinhardt's work [4], Eq. (3.11) was discretized using direct multiple-shooting with a forth-order explicit Runge-Kutta integrator. To solve the nonlinear problem in Eq. (3.12) a real-time iteration sequential quadratic programming (RTI SQP) is used together with the high-performance interior point method for the solutions of the quadratic problems, where the QP solver relies on the numerical subroutines of BLASFEO. All this is available in the Acados package mentioned. N was set to $N = 10$. Further, a C/C++ code was generated from the solution found on Python, from which an S-function, containing the NMPC, was implemented in Matlab, to be able to interface the NMPC with the simulator. This was done following the examples in the Acados documentation [24].

The constraints used in the NLP are given in Table 4.2, where the constraints on $\delta_\times$ are given by the real actuator limits. When it comes to the angle of attack α, the upper and lower bounds were chosen as the stall limit for a fully iced wing, suggested by Högnadóttir [6]. As the stall limit has not been clearly determined in the existing literature, with Winter [12] suggesting $\alpha_{stall} \approx 10$, the more restrictive $\alpha_{stall} \approx 4$ was used to push the NMPC and see its performance. When it comes to the airspeed upper bound, however, it was chosen as $\overline{V_a} = 45\,m/s$ due to

the NLP being infeasible with a lower $\overline{V_a}$ when simulated with severe wind gusts of $V_{wind} = 23\,m/s$. This is because the initial airspeed became $V_a = 41\,m/s$, with $V_{a,0} = 18\,m/s$. Ideally, $\overline{V_a}$ should be the same as the safety limit of $\overline{V_a} = 25\,m/s$, and the slack weight on V_a should allow for a higher initial airspeed, but due to the lack of time this issue was not fixed.

Variable	Value	
$\underline{V_a}$, $\overline{V_a}$	15, 45	m/s
$\underline{\beta}$, $\overline{\beta}$	-90, 90	deg
$\underline{\alpha}$, $\overline{\alpha}$	-4, 4	deg
$\underline{p^s}$, $\overline{p^s}$; $\underline{q^s}$, $\overline{q^s}$; $\underline{r^s}$, $\overline{r^s}$	-180, 180; -180, 180; -180, 180	deg
$\underline{\delta}_a$, $\overline{\delta}_a$	-35, 35	deg
$\underline{\delta}_e$, $\overline{\delta}_e$	-35, 35	deg
$\underline{\delta}_r$, $\overline{\delta}_r$	0, 0	deg
$\underline{\delta}_t$, $\overline{\delta}_t$	0, 1	-
$\underline{\delta}_{el}$, $\overline{\delta}_{el}$, $\underline{\delta}_{er}$, $\overline{\delta}_{er}$	-30, 20, -30, 20	deg

Table 4.2: Inequality constraints in the NMPC controller

When it comes to the tuning of the NMPC, it was fond by trial and error, starting with $\boldsymbol{Q} = \text{diag}[0.1, 300, 300, 300]$, and the $\boldsymbol{R}$ gains given in Table 4.3, which are lower relative to $\boldsymbol{Q}$, as it is often done in practice. With the disturbance observer turned off ($l_\times = 0$), a large offset error was observed in roll, pitch and airspeed, so tuning the disturbance observer was the first priority. As the offset error was larger on roll, the weight on the roll angular rate was set the highest, followed by a slightly lower weight on the pitch rate. When it comes to airspeed, a lower weight seemed to be enough to remove the offset error, and the final weights chosen are given in Table 4.3b.

During the further tuning of $\boldsymbol{Q}$, the attitude tracking was prioritized over the airspeed tracking, as the airspeed element can be tuned after a suitable attitude tuning is found. The airspeed weight corresponds to the first diagonal element of the weighting matrix $\boldsymbol{Q}$, while the weights on the elements of the reduced-attitude vector $\boldsymbol{\Gamma}$ are given by the other diagonal elements of $\boldsymbol{Q}$. Following Reinhardt's explanations in his work [4], higher weights on $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ would result on a tighter tracking of the pitch and roll angles, respectively, but they could also temporarily increase the cost function, and might require a longer horizon N to be able to converge. As the pitch response was good, the weight on it was lowered to improve the roll response, and the weight on airspeed was decreased and chosen such as the NLP was feasible under severe wind conditions. The resulting tuning is found in Table 4.3a.

(a) NMPC tuning matrices

Parameter	Value
$\boldsymbol{Q}$	diag[0.01, 50, 300, 300]
$\boldsymbol{R}$	diag[1, 1, 1, 1]
$\boldsymbol{P}$	diag[100, 100, 100, 100]

(b) Disturbance observer tuning parameters

Parameter	Value
l_{Va}, l_β, l_α	0.01, 0.01, 0.5
l_l, l_m, l_n	0.5, 0.1, 0.01

Table 4.3: Tuning parameters of the NMPC controller.

4.2.2 PID and MRAC Controllers

The PID and MRAC controllers were tuned and implemented based on Högnadóttir's work [6]. The PID controller for roll and pitch is given by equations (3.18) and (3.19), where the commanded angles ϕ_{cmd} and θ_{cmd} are set equal to the angles ϕ_{ref} and θ_{ref}, respectively, obtained after the reference generation explained in Section 4.1. In addition, anti-windup mechanisms were implemented in the simulator to prevent the integrators from winding up. The tuning parameters used were found in Högnadóttir's work [6], and can be seen in Table 4.4.

(a) Roll controller gains

Parameter	Value
k_{p_ϕ}	2.5
k_{i_ϕ}	2
k_{d_ϕ}	0.01

(b) Pitch controller gains

Parameter	Value
k_{p_θ}	2.5
k_{i_θ}	2
k_{d_θ}	0.01

Table 4.4: Tuning parameters of the PID controller.

When it comes to the MRAC controller, described in Section 3.2.2, the reference model $\boldsymbol{x}_{ref}$ is found as described in Section 4.1, where ϕ_{cmd} and θ_{cmd} are the commanded reference $\boldsymbol{r}(t)$ in the MRAC equations. The MRAC was tuned as described in Högnadóttir's work [6], and the resulting tuning parameters can be seen in Table 4.5.

(a) Roll adaptive rates

Parameter	Value
$\boldsymbol{Q}$	diag[3, 1]
$\boldsymbol{\Gamma}_x$	diag[12, 4]
$\boldsymbol{\Gamma}_r$	10
$\boldsymbol{\Gamma}_\Theta$	15

(b) Pitch adaptive rates

Parameter	Value
$\boldsymbol{Q}$	diag[4, 0.4]
$\boldsymbol{\Gamma}_x$	diag[6, 0.01]
$\boldsymbol{\Gamma}_r$	5
$\boldsymbol{\Gamma}_\Theta$	10

Table 4.5: Tuning parameters of the MRAC controller.

4.3 Implementation of the Physical Model

The physical model was implemented in the Matlab/ Simulink simulator by implementing the UAV equations of motion described in Section 2.2, together with the forces and moments described in Section 2.4.1, and using the physical parameters from Table 4.6. When it comes to the aerodynamic forces and moments, the asymmetric model described in Section 2.6.1 was implemented, to account for the effects of asymmetric icing on the wings. The aerodynamic coefficients were found based on Winter's work [12], where they are described in clean and iced state. As some of the coefficients were non-linear, they are given for a specific sideslip and angle of attack, and interpolated in the simulator by the Simulink block *n-D Lookup Table* [25], as done in Högnadóttir's work [6]. As the aerodynamic coefficients are given for the either clean or iced case, a linear interpolation is used, following Kleiven's work [7], to find the coefficients in an intermediate ice state:

$$C_k(\zeta) = \zeta C_{k,iced} + (1-\zeta)C_{z,clean},$$

where $\zeta \in [0,1]$ denotes the level of icing for each wing, with $\zeta = 0$ describing no ice on the wing (clean), and $\zeta = 1$ describing a fully iced wing.

Parameter		Value	
Air density	ρ	1.2250	kg/m^3
Mass	m	3.364	kg
Wing span	b	2.1	m
Mean chord	c	0.3571	m
Wing area	S	0.75	m^2
Propeller diameter	d	0.3556	m
	J_{xx}	0.335	kgm^2
	J_{yy}	0.140	kgm^2
	J_{zz}	0.400	kgm^2
	J_{xz}	-0.029	kgm^2

Table 4.6: Physical parameters of the Skywalker X8.

4.3.1 Propeller icing

The propeller icing model was implemented following Müller's explanations [15] and his help. For the development of the model, the wind-tunnel experiments Müller conducted were performed on the Mejzlik 21x13E propeller, and the parameters given in his work are valid for a propeller of the same size and material. The propeller the Skywalker X8 uses is a Aeronaut CamCarbon 14x8" (foldable), where the diameter is of 14 inches, as the name states. The propeller parameters therefore need to be adjusted to match the propeller used. When it comes to the thrust coefficients in the clean state, $C_{T,clean}$, the parameters are the same as the ones found in Coates' work [16]. The clean power coefficients $C_{P,clean}$ are taken

from [26], where the data is found from experiments with the Aeronaut CamCarbon 14x8" propeller, so it is assumed to be valid. The parameters for the clean coefficients are then found in Table 4.7.

(a) C_T parameters

Parameter	Value
$C_{T,0}$	0.126
$C_{T,1}$	-0.1378
$C_{T,2}$	0

(b) C_P parameters

Parameter	Value
$C_{P,0}$	0.032989
$C_{P,1}$	0.0866
$C_{P,2}$	-0.1623

Table 4.7: Parameters the clean thrust and power coefficients estimation.

When it comes to the adhesion force A_{max}, it depends on the material and surface finish of the propeller, so it is assumed to be the same, as both the Mejzlik and Aeronaut propellers are carbon fiber propellers. The ΔC_T and ΔC_P, however, depend on the twist and chord distribution of the propeller blades. The Mejzlik propeller is close to the propeller the UAV uses in terms of pitch to diameter ratio, so the twist distribution is expected to be similar. Because of lack of time and blade geometry data about the two propellers, the ΔC_T and ΔC_P found by Müller were assumed to be transferable as well. The final parameters used are given in Table 4.8.

(a) ΔC_T parameters

Parameter	Value
$\Delta C_{T,0}$	0.0233
$\Delta C_{T,1}$	0.0254
$\Delta C_{T,2}$	0.00140

(b) ΔC_P parameters

Parameter	Value
$\Delta C_{P,0}$	-0.00890
$\Delta C_{P,1}$	-0.0166
$\Delta C_{P,2}$	-5.79e-04

(c) A_{max} parameters

Parameter	Value
$A_{max,0}$	37.250
$A_{max,1}$	1223

Table 4.8: Parameters for the ΔC_T, ΔC_P and A_{max} estimation.

Looking at Eq. (2.32) and (2.34), describing the ice model for the propeller, we see that the performance depends on the time t the propeller has been under icing conditions, the temperature T, the liquid water content LWC, and the advance ratio J. The advance ratio will depend on the airspeed of the aircraft and rotation speed of the propeller, so it will be given by the simulator. As it is more interesting to simulate the worst performance degradation case, the C_T and C_P have been plotted for different temperatures and for a given time, using a standard $J = 0.6$ and $LWC = 0.44\,g/m^3$. Given the results in Figure 4.2, the worst final performance degradation was given at $T = -15°C$ after 100 seconds. Therefore, to add the effects of propeller icing to the simulator, the thrust model in Eq. (2.26) was implemented with the thrust coefficient given by Eq. (2.32), choosing $T = -15°C$, $LWC = 0.44\,g/m^3$, and $t = 200$s, to simulate the performance after the propeller has been in icing conditions for a longer period of time. The power coefficient was not implemented however, as it affects the propeller torque, which was assumed to be negligible as previously explained in Section 2.4.3. Given the case simulated

in Figure 4.2, the power coefficient, and subsequently the torque, were not shown to increase substantially, so the previous claim is assumed to remain valid.

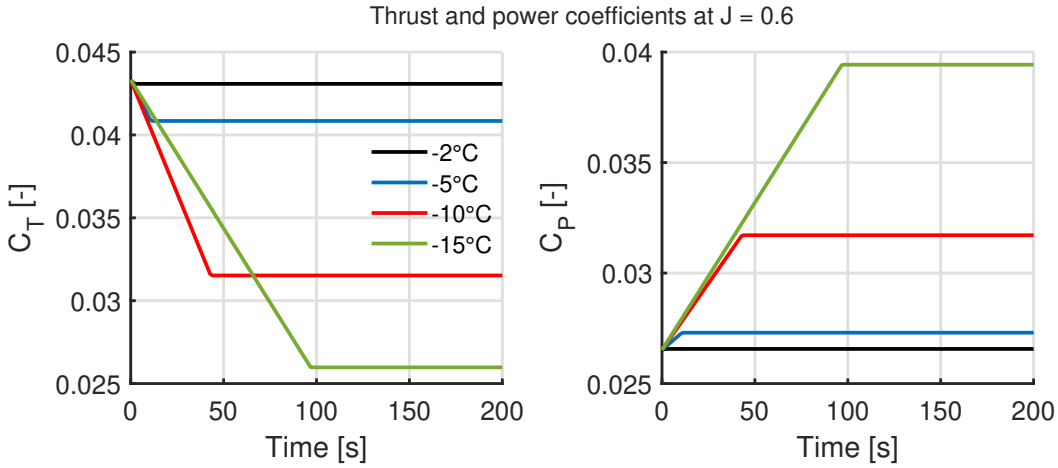

Figure 4.2: Thrust and power coefficient of the propeller for an advance ratio of $J = 0.6$ and LWC of 0.44 g/m^3.

4.4 Simulations

This section describes the simulations that were run in order to test the performance of the NMPC, PID and MRAC controllers when affected by different disturbances, including ice accretion on the propeller and asymmetric icing on the wings, reduced airspeed and severe wind conditions. The path-following performance of the controllers is also tested by letting them receive their roll and pitch references from a guidance controller. To be able to objectively analyse the performance of each controller, the Integral Absolute Error (IAE) performance metric will be used and introduced in this section.

4.4.1 Performance Metric

The Integral Absolute Error (IAE) of the reference error of a variable a is given by:

$$IAE = \int_0^t |e_a(\tau)| d\tau. \tag{4.13}$$

In this case the controllers are following references for roll, pitch and airspeed, giving the reference errors $e_\phi = \phi_{ref} - \phi$, $e_\theta = \theta_{ref} - \theta$ and $e_{Va} = Va_{ref} - Va$. The IAE is used to evaluate the reference following performance of the controllers, where a high value indicates a larger error over the time interval. As in this case it is interesting to see not only what the final value is but also how it increases with time and with different disturbances, the IAEs for roll, pitch and airspeed of every controller have been plotted over the simulation time in all simulation cases.

4.4.2 Baseline simulation

The first simulation that is run serves as a baseline for the next simulations, therefore it is referred to it as the baseline simulation. It runs from 0s to 200s, with a reference in airspeed of $V_{a,ref} = 20\,m/s$ and the icing on the propeller turned on, as described on 4.3.1. In the time interval $t \in (0, 50)$s, an excitation for the MRAC controller is run so that its states converge to a good tuning, while the roll and pitch references for the NMPC and PID controllers are kept at 0° and 2.659° respectively. In the interval $t \in (60, 120)$s, the reference in pitch is set constant at $\theta_{cmd} = \theta_{trim} = 2.659°$ while roll changes between $\phi_{cmd} = 0°$ and $\phi_{cmd} = 30°$, following the reference model described in 4.1.1. In the last interval, $t \in (120, 200)$s, it is the pitch reference the one that changes between $\theta_{cmd} = 2.659°$ and $\theta_{cmd} = 30°$, while roll is kept constant at $\phi_{cmd} = 0°$. The sequence can be seen in Figure 4.3. In addition, an icing level sequence for the wings is run, where the roll and pitch are subject to 100% asymmetric icing during the intervals $t \in (100, 110)$s and $t \in (160, 170)$s.

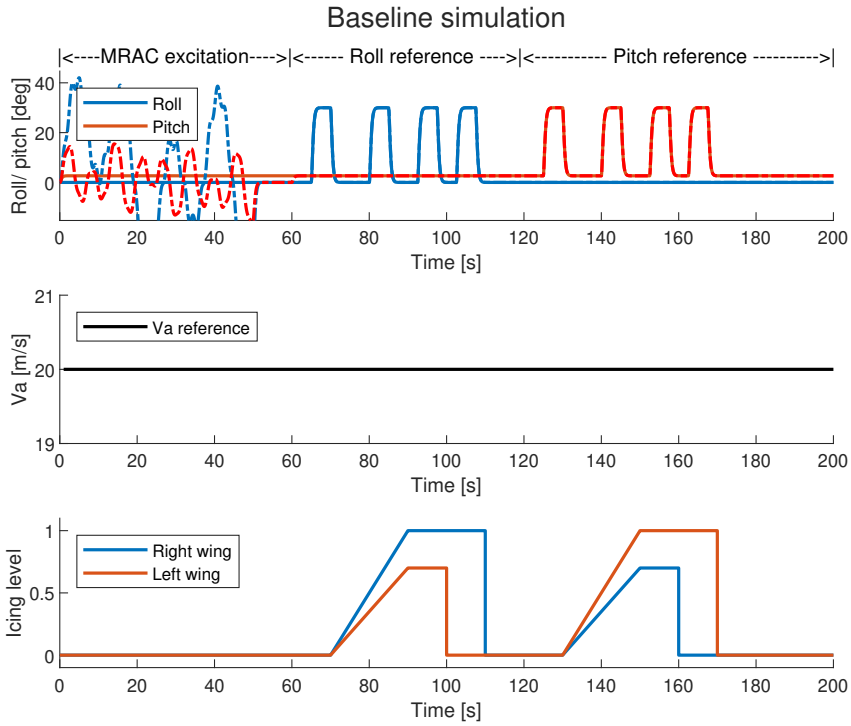

Figure 4.3: Roll, pitch, airspeed and icing level references in the baseline simulation.

4.4.3 Reduced airspeed and wind conditions

To test the controllers performance to disturbances, two simulations are performed: a reduced airspeed simulation, and a simulation with wind conditions. Both sim-

ulations use the same asymmetric icing sequence as described in the baseline simulation, as well as the same roll and pitch sequence. During the reduced airspeed simulation case however, the airspeed reference is set to $V_{a,ref} = 17\,m/s$. On the other hand, during the wind conditions simulation case, the airspeed reference is set back to $V_{a,ref} = 20\,m/s$, but a severe wind gusts of $V_{wind} = 23\,m/s$ are added to the simulation.

4.4.4 Guidance simulation

To test the controller's performance in a path-following scenario, a guidance simulation was performed, where the pitch and roll commanded angles were given by the guidance controller as described in Section 4.1.2. The path to follow was given as a set of waypoints, as shown in Figure 4.4. In this case, moderate wind conditions were simulated, with wind gusts of $V_{wind} = 15.4$ m/s, while the airspeed reference was set to $V_a = 20$ m/s. The asymmetric icing sequence was also modified to extend the interval in which the 100% asymmetric icing is present, lasting for $60\,s$ in total, and shown in Figure 4.5.

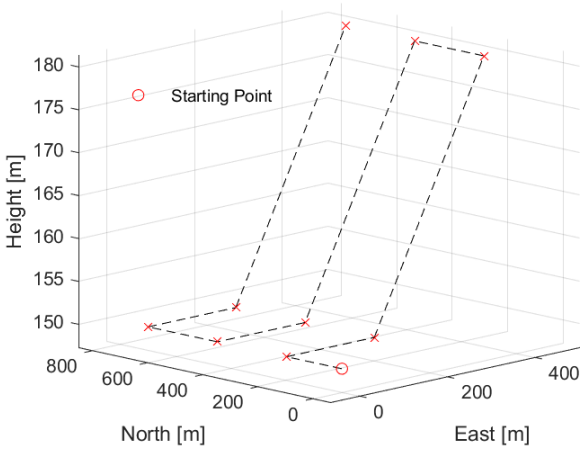

Figure 4.4: Path to be followed.

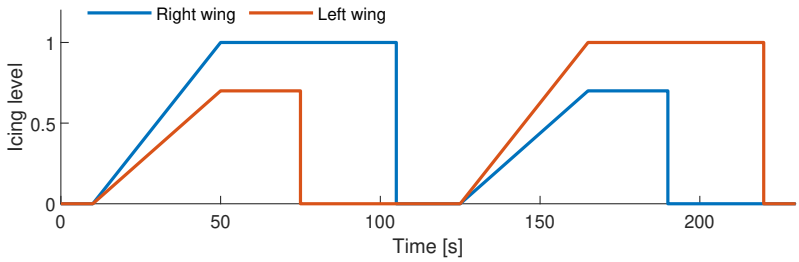

Figure 4.5: Icing level in the guidance simulation.

4.4.5 Additional simulations

After seeing the response of the previously described simulations, two additional simulations have been performed, to further push the limits to the amount of disturbances the controllers can handle based on the reference chosen. More specifically, noticing a poor performance of the PID during the 100% asymmetric icing interval in the reduced airspeed simulation, as seen in Section 5.2, it was interesting to see the response of the controllers when roll and pitch had the same reference, at the same time. The results for this new simulation are found in Section 5.4.

Additionally, it was also interesting to see whether the PID and MRAC controllers' tuning was outdated. The tuning used in this thesis was the same as the one Högnadóttir found in her work [6], as it showed a very good response, but the model in this thesis was slightly changed. The thrust model was updated from the Fitzpatrick model [18] to Coates' model [16], together with Müller's extension to include icing on the propeller. Therefore, additional simulations have been performed to determine how large the differences in performance of the PID and MRAC controllers are depending on the thrust model used, with the same tuning. For this purpose, the reduced airspeed and wind disturbance simulations presented in Section 4.4.3 have been run, first with the propeller model described in subsections 2.4.3 and 2.6.2, and then with the old propeller model described in [6]. The parameters used in both models are found in Appendix A and in Section 4.3, respectively. To assess the differences objectively, the IAE has been plotted for both cases as described in 4.4.1, and the simulation results are found in Section 5.4.

Chapter 5

Results

This chapter presents the results obtained after running the simulations described in 4.4. To assess the performance of the different controllers objectively, the Integral Absolute Error (IAE) is used as a performance metric, as described in 4.4.1. The IAE of the states being tracked (roll, pitch and airspeed) is therefore plotted for every simulation, and given as a part of the results.

5.1 Baseline simulation

The baseline simulation is run as described in Section 4.4.2, with a reference in the airspeed of $20\,m/s$ and no wind disturbances. Figure 5.1 shows the response of the first half of the simulation, with $t \in (60, 120)$, when the reference for roll oscillates between 0° and 30°, and the pitch angle is kept at a constant 2.6595°. It can be seen that the NMPC's airspeed tracking performance is significantly worse than the PID's and MRAC's, with a maximum airspeed error of $e_{Va} = 0.92\,m/s$, right after the asymmetric icing conditions at around $t \approx 111\,s$. During the most severe asymmetric icing conditions, between $t = 100\,s$ and $t = 110\,s$, the NMPC has the highest angle of attack, $\alpha = 4.35°$, at $t = 103.6\,s$, and the throttle increases during this time period as well. The NMPC's pitch tracking performance was staying close to its reference up until this point, not being affected by the roll reference as much as the PID and MRAC. However, during the severe icing conditions, the pitch performance is worsened slightly as well.

The roll simulation finishes at $t = 120\,s$, and in Figure 5.3 it can be seen that up until this point the NMPC's gave the best pitch tracking performance, but the worse roll and airspeed tracking, with the airspeed being particularly bad. The performance of the PID and MRAC are very similar, with the PID being better at tracking roll and airspeed, and the MRAC being better at tracking pitch than the PID.

Figure 5.2 shows the response of the baseline simulation between $t \in (120, 180)$, where the roll reference is kept at a constant 0°, and the pitch oscillates between 2.6585° and 30°. In this case, it can be seen that pitch reference affects the NMPC's

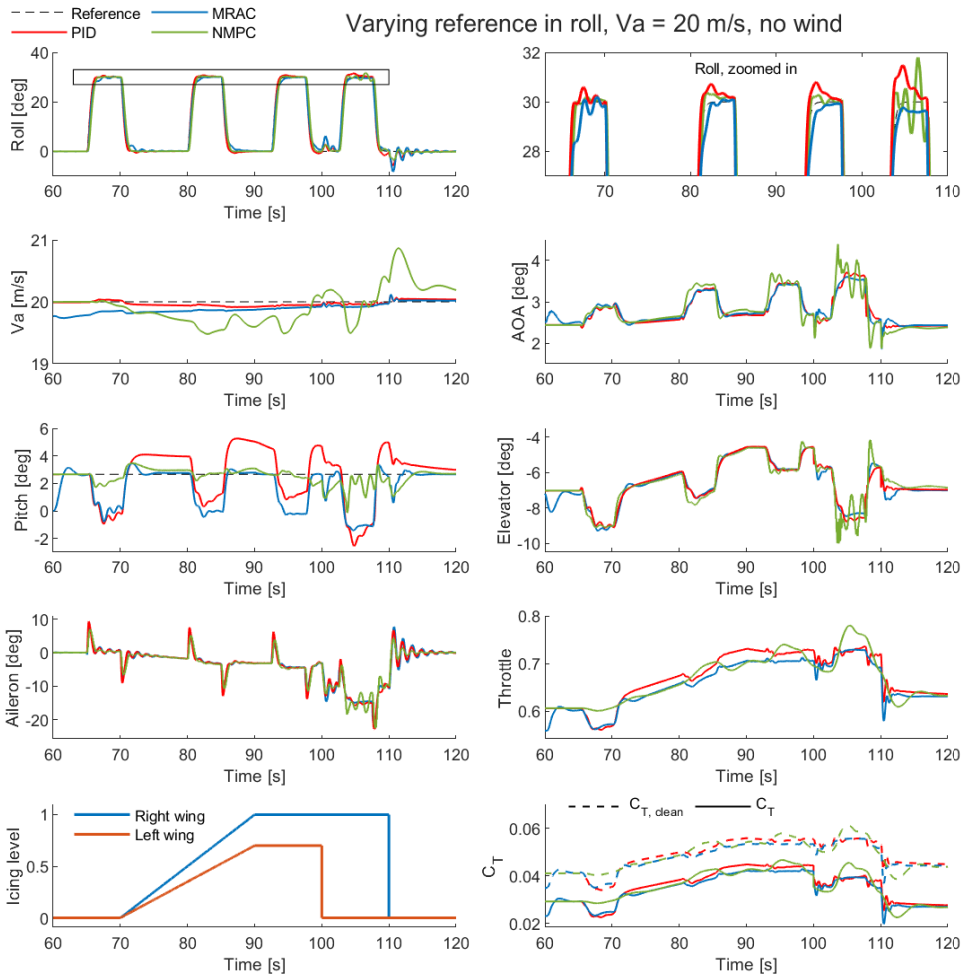

Figure 5.1: Baseline simulation for $t \in (60, 120)$, with a varying reference in roll from $\phi_{cmd} = 0°$ to $\phi_{cmd} = 30°$, constant $\theta_{cmd} = 2.659°$, constant $V_{a,ref} = 20\,m/s$, and no wind disturbances.

airspeed tracking performance more than the roll did, with oscillations throughout the simulation time, with the maximum airspeed error $e_{Va} = 1.65\,m/s$. The pitch reference affects the performance of the PID and MRAC airspeed tracking as well, to a smaller degree than the NMPC, but more than in Figure 5.1. In this simulation, the NMPC also keeps a lower angle of attack, with $\alpha_{max} = 4.86°$ at $t = 153\,s$ and $\alpha_{min} = -1.39°$ at $t = 158.1\,s$. The PID and MRAC, on the other hand, have a $\alpha_{max} = 5.96°$ and $\alpha_{max} = 6.42°$, respectively. When it comes to the roll tracking performance, the NMPC and PID perform similarly in this case, with an IAE increase in the interval $t \in (120, 180)$ of approximately 0.54 and 0.43, respectively, as seen in Figure 5.3.

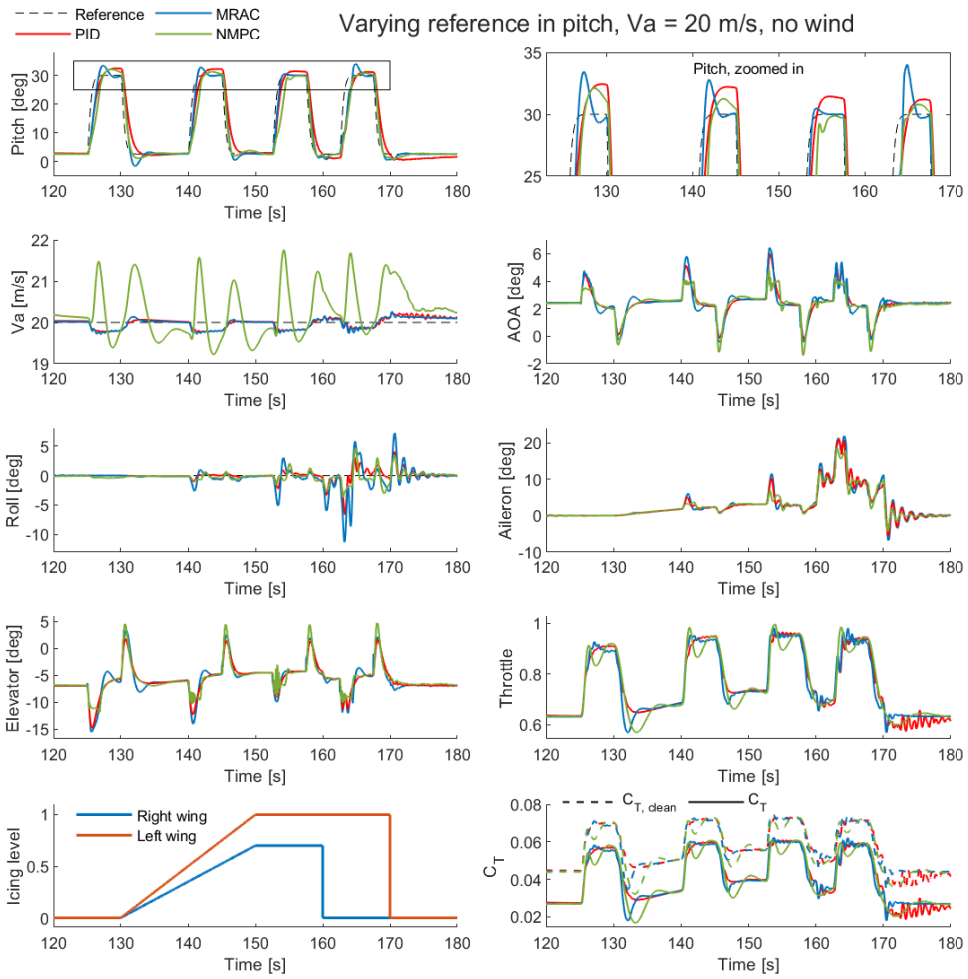

Figure 5.2: Baseline simulation for $t \in (120, 180)$, with a varying reference in pitch from $\theta_{cmd} = 2.659°$ to $\theta_{cmd} = 30°$, constant $\phi_{cmd} = 0°$, constant $V_{a,ref} = 20\,m/s$, and no wind disturbances.

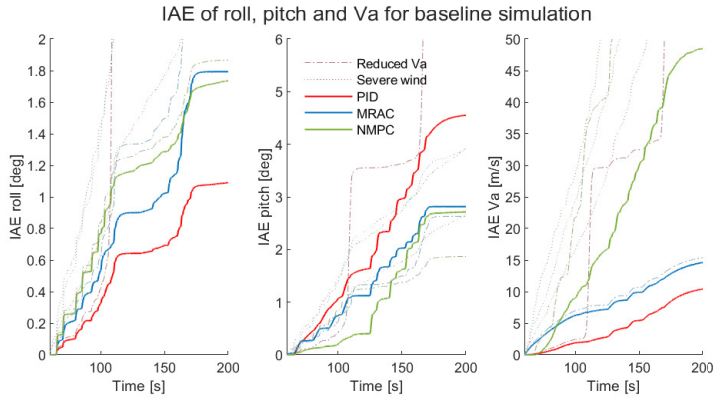

Figure 5.3: IAE of roll, pitch and airspeed throughout the baseline simulation, with a constant $V_{a,ref} = 20\,m/s$ and no wind disturbances. In grey, the responses with reduced airspeed and severe wind are shown for reference, but can be better seen in figures 5.6 and 5.9 respectively.

The results of the baseline simulation can be summarized by examining Figure 5.3. The best roll tracking performance was achieved by the PID controller, with the NMPC coming second due to the good performance in the second part of the simulation, when the MRAC's performance was the most affected by the changes in pitch reference. When it comes to pitch tracking performance, the NMPC was the best, followed closely by the MRAC. However, the NMPC's biggest weakness is the airspeed tracking performance, which is significantly worse compared to the PID and MRAC, with the PID having the best one.

5.2 Reduced airspeed simulation

The reduced airspeed simulation is run as described in Section 4.4.3, with the airspeed reference set to $V_{a,ref} = 17m/s$, and no wind disturbances. In the same way as with the baseline simulation, the response of the first half of the simulation, in the interval $t \in (60, 120)$, is shown in Figure 5.4. Here only the reference in roll changes, while pitch and airspeed remain constant. It can be see that, similar with the baseline case, the NMPC's airspeed response is significantly worse than that of the PID and MRAC, up until when the most severe ice asymmetry case is simulated. The PID performance is the worst in this time interval $t \in (100, 110)$, with a spike in both roll ($\phi_{max} = 60.65°$), airspeed ($Va_{max} = 24.69\,m/s$) and pitch ($\theta_{min} = -53.1°$). In the period $t \in (108, 113)$, the throttle also saturates to $\delta_t = 0$. The spike in pitch is experienced by both MRAC and NMPC, but to a much smaller degree. The angle of attack of the NMPC is also kept low, with the maximum at $\alpha_{max} = 4.52°$ around $t \approx 103$. The PID and MRAC have their maximum angles of attack at $\alpha_{max} = 6.49°$ and $\alpha_{max} = 6.22°$, respectively.

The overall performance up until this point can be seen in Figure 5.6 in the interval $t \in (60, 120)$. The PID's roll, pitch and airspeed performances greatly deteriorated after the severe asymmetric case. When it comes to the NMPC, its roll performance remains similar to the one of the baseline simulation, but both the pitch and airspeed performances are worsened, the airspeed performance remaining the worse out of the three controllers. The MRAC's roll and pitch performances are also deteriorated after the severe asymmetric case at $t = 100\,s$, but the airspeed performance remains almost the same.

The results of the reduced airspeed simulation in the interval $t \in (120, 180)$ are shown in Figure 5.5, where the reference in pitch was changing while the roll reference was kept constant. The same erratic behaviour of the PID controller in the severe asymmetric case can be seen here as well, with severe spikes in pitch ($\theta_{min} = -62.63°$), airspeed ($Va_{max} = 25.1\,m/s$) and roll ($\phi_{min} = -79.28°$). The PID's throttle saturates around this time also in this case. When it comes to the angle of attack, it is high with the PID and MRAC here as well, with their maximum at $\alpha_{max} = 7.59°$ and $\alpha_{max} = 8.19°$, respectively. The NMPC has a maximum angle of attack at $\alpha_{max} = 4.56°$ and a minimum angle of attack at $\alpha_{min} = -1.33°$. Its airspeed tracking performance remains poor, with a maximum airspeed of $Va_{max} = 22.44\,m/s$. Lastly, in this part of the simulation it can also be

seen that the MRAC's roll tracking performance is worsened compared to previous simulations.

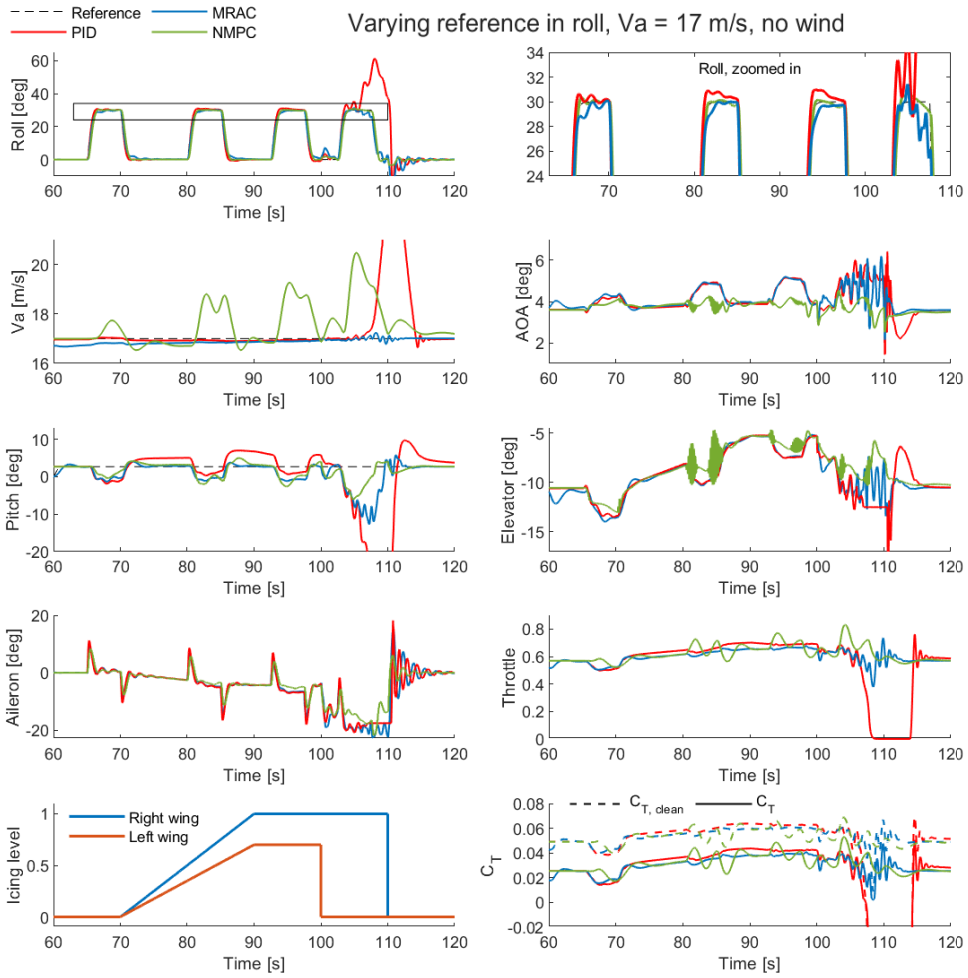

Figure 5.4: Reduced airspeed simulation for $t \in (60, 120)$, with a varying reference in roll from $\phi_{cmd} = 0°$ to $\phi_{cmd} = 30°$, constant $\theta_{cmd} = 2.659°$, constant $V_{a,ref} = 17\,m/s$, and no wind disturbances.

The results of the reduced airspeed simulation are summarized in Figure 5.6. It can be seen that the PID performance is overall severely impaired compared to the baseline simulation shown in Figure 5.3, and to the other two controllers, whose performance is not affected to the same degree. In this simulation, the MRAC becomes the best at tracking airspeed and pitch, while the NMPC becomes the best at tracking roll and remains the worst at tracking airspeed.

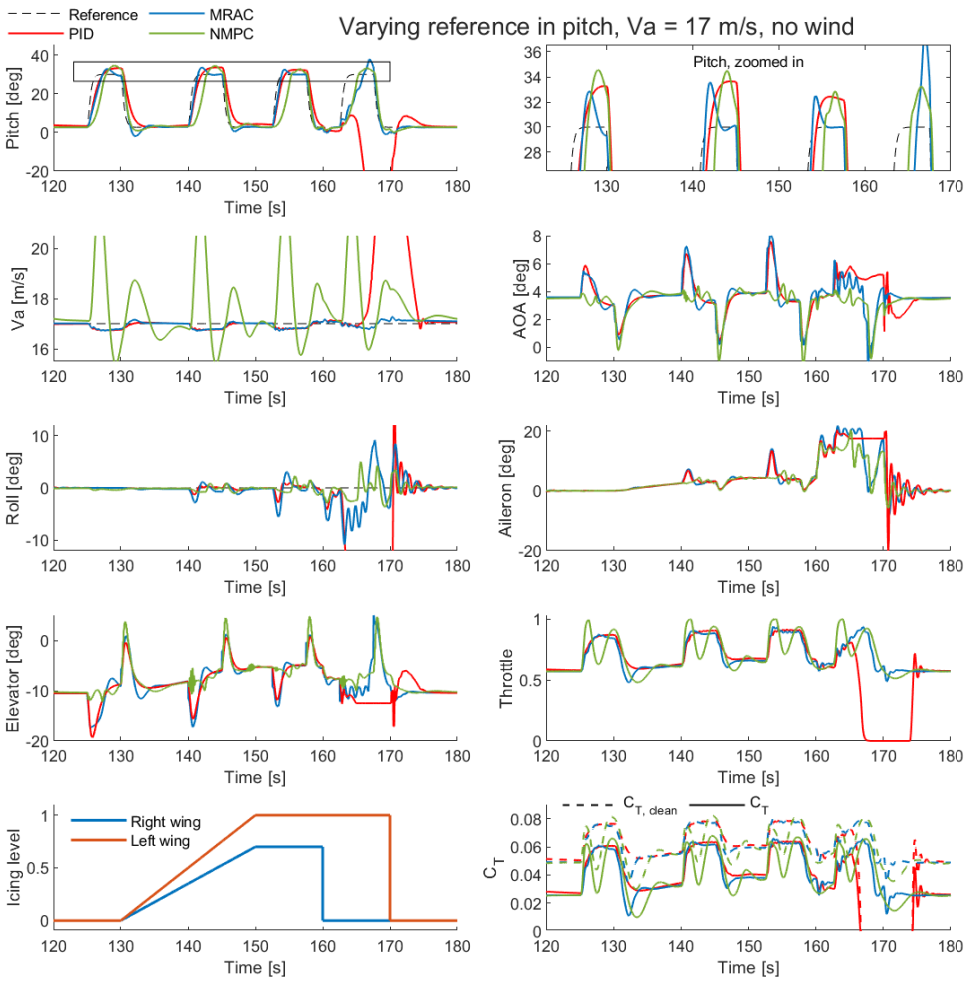

Figure 5.5: Reduced airspeed simulation for $t \in (120, 180)$, with a varying reference in pitch from $\theta_{cmd} = 2.659°$ to $\theta_{cmd} = 30°$, constant $\phi_{cmd} = 0°$, constant $V_{a,ref} = 17\,m/s$, and no wind disturbances.

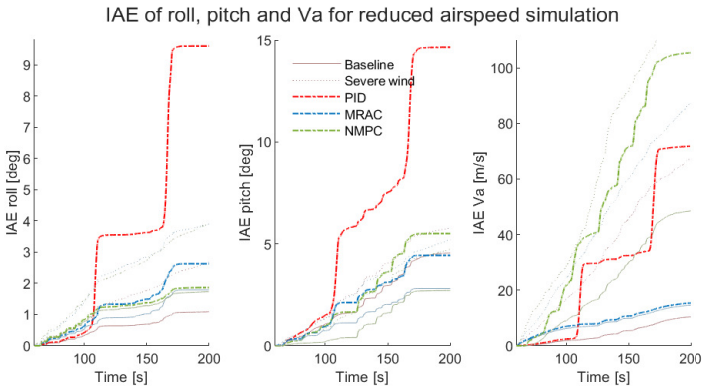

Figure 5.6: IAE of roll, pitch and airspeed throughout the reduced airspeed simulation, with a constant $V_{a,ref} = 17\,m/s$ and no wind disturbances. In grey, the baseline and severe wind responses are shown for reference, but can be better seen in figures 5.3 and 5.9 respectively.

5.3 Wind disturbance simulation

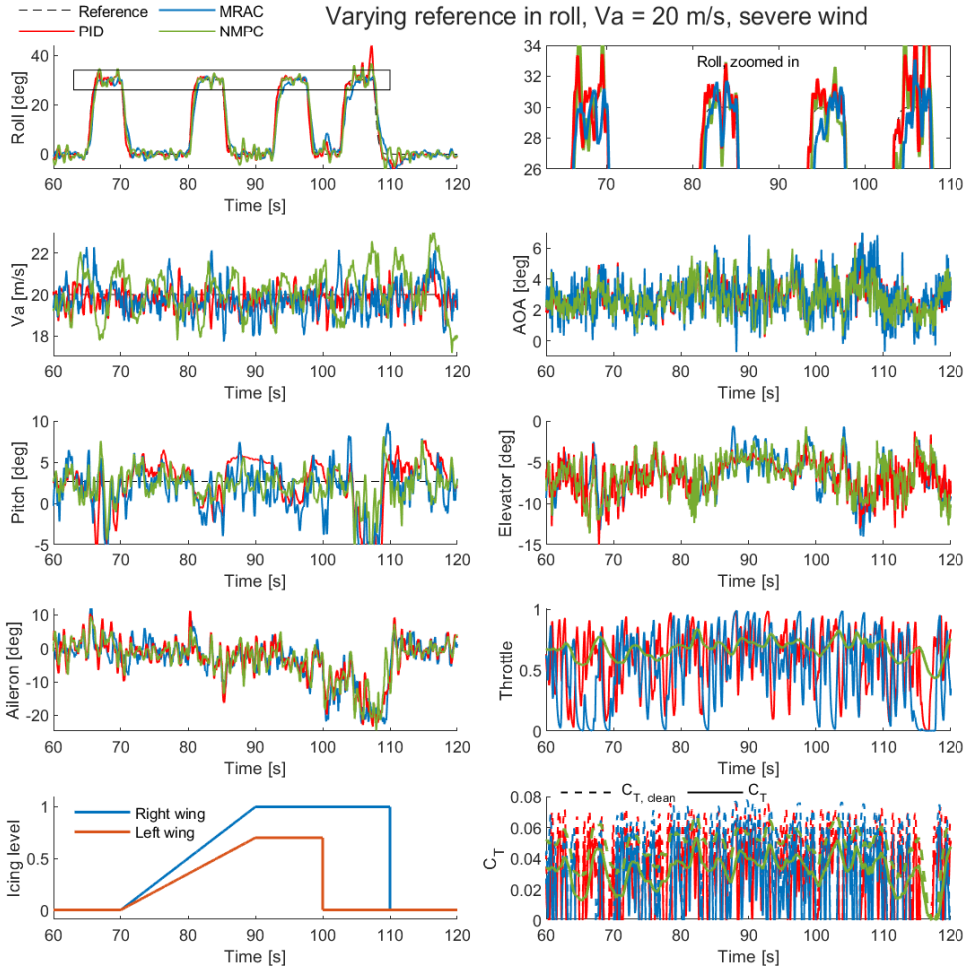

Figure 5.7: Wind conditions simulation for $t \in (60, 120)$, with a varying reference in roll from $\phi_{ref} = 0°$ to $\phi_{ref} = 30°$, constant $\theta_{ref} = 2.659°$, constant $V_{a,ref} = 20\,m/s$, and $V_{wind} = 23\,m/s$.

The wind disturbance simulation is run as described in Section 4.4.3, in the same way as the baseline simulation, but this time with severe wind gusts. The response of the first half of the simulation, in the interval $t \in (60, 120)s$, is shown in Figure 5.7. This time it can be seen that the wind conditions worsen the airspeed tracking performance in all three controllers, but the NMPC has the largest deviation of $e_{Va} = 3.38°$. The severe wind conditions also increases the angle of attack of the NMPC, which in the previous simulations had been kept low at around $\alpha_{max} \approx 4.5°$. In this simulation, the maximum angles of attack are given by $\alpha_{max} = 7.31°$, $\alpha_{max} = 6.34°$ and $\alpha_{max} = 6.17°$, and the minimums by $\alpha_{min} = -0.70°$, $\alpha_{min} = 0.23°$ and $\alpha_{min} = 0.43°$ for MRAC, PID and NMPC, re-

spectively. The overall response is noisier for all controllers, with the exception of the NMPC's throttle, which does not oscillate as much.

The results in the interval $t \in (120, 180)$, when the pitch reference changes and roll is kept constant, can be seen in Figure 5.8. In this case, the maximum angles of attack are given by $\alpha_{max} = 8.01°$, $\alpha_{max} = 7.03°$ and $\alpha_{max} = 6.19°$, and the minimums by $\alpha_{min} = -2.71°$, $\alpha_{min} = -2.01°$ and $\alpha_{min} = -2.22°$ for MRAC, PID and NMPC, respectively. Although the PID and MRAC airspeed tracking is worse compared to the previous simulations, the NMPC has the largest deviation here, with $e_{Va} = 4.53°$.

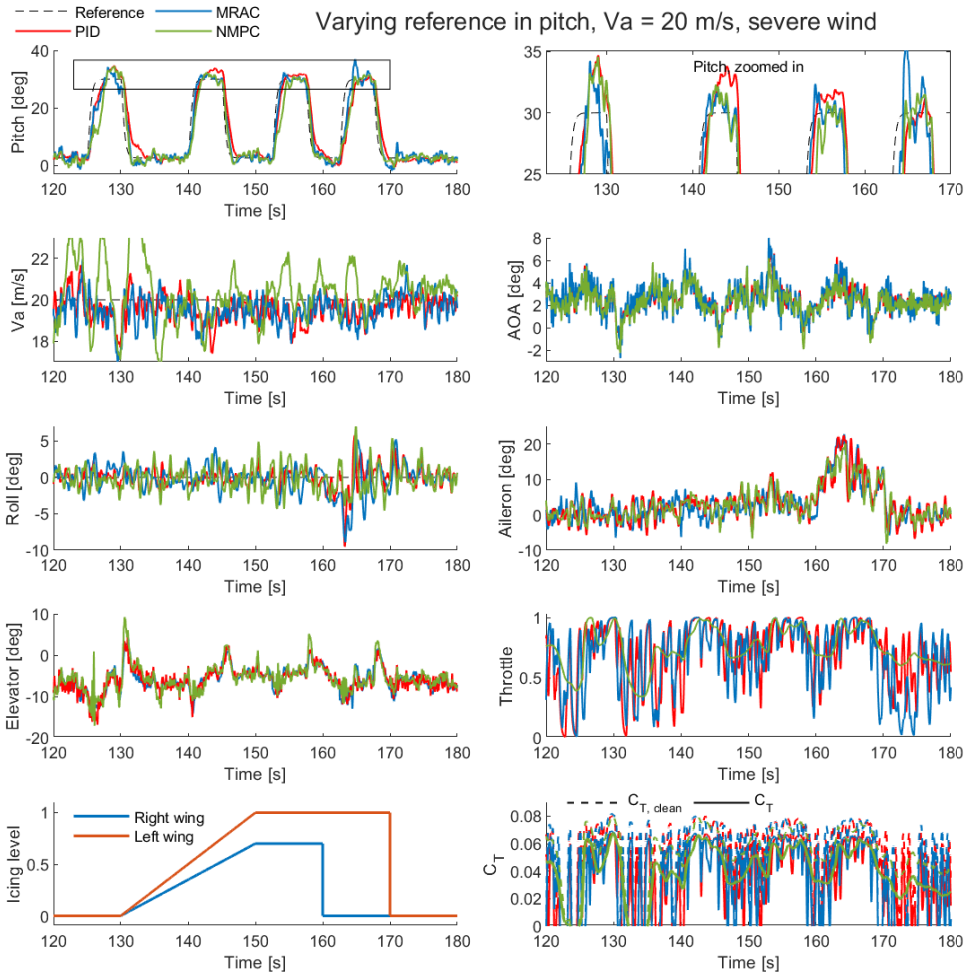

Figure 5.8: Wind conditions simulation for $t \in (60, 120)$, with a varying reference in roll from $\phi_{cmd} = 0°$ to $\phi_{cmd} = 30°$, constant $\theta_{cmd} = 2.659°$, constant $V_{a,ref} = 20\,m/s$, and $V_{wind} = 23\,m/s$.

With severe wind conditions, the response of all controllers is noisier but the effects of the 100% asymmetric icing case are not as severe in this case as they were in the reduced airspeed case. Looking at the IAE of roll, pitch and airspeed

in Figure 5.9, we see that the IAE increases in all cases as time passes, but the increases during the 100% asymmetric icing intervals are not as dramatic as they were in the previous simulations. The results show that the PID controller is the best at tracking roll and airspeed, the NMPC is the best at tracking pitch and the worst at airspeed, and the MRAC is somewhere in the middle.

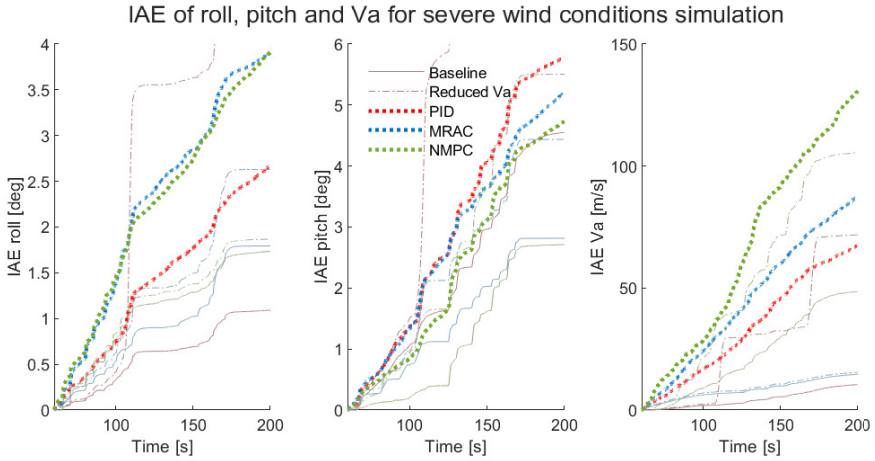

Figure 5.9: Baseline simulation for $t \in (60, 120)$, with a varying reference in roll from $\phi_{cmd} = 0°$ to $\phi_{cmd} = 30°$, constant $\theta_{cmd} = 2.659°$, constant $V_{a,ref} = 20\,m/s$, and $V_{wind} = 23\,m/s$. In grey, the baseline and reduced airspeed responses are shown for reference, but can be better seen in figures 5.3 and 5.6, respectively.

5.4 Additional simulations

To further compare the differences between controllers and the role tuning plays, two additional simulations have been performed.

First, after noticing the severely impaired PID performance of roll, pitch and V_a in the reduced airspeed simulation, during the interval with 100% asymmetric icing, a new test has been performed. This time ϕ_{cmd} and θ_{cmd} were changed at the same time, from $\phi_{cmd} = 0°$ to $\phi_{cmd} = 30°$ and from $\theta_{cmd} = 2.659°$ to $\theta_{cmd} = 30°$, respectively, while keeping $V_{a,ref} = 20\,m/s$. In Figure 5.10 it can be seen that the PID controller, although it follows the reference very well until the severe asymmetric icing interval, it crashes in this case, with δ_t saturating at 1. The MRAC roll and pitch response is somewhat worsened compared to the response in Section 5.2 as well, with the roll being over the reference of 30° and the pitch being under, but it manages to handle the severe asymmetric interval. In this simulation, however, the NMPC seems more consistent with the results obtained in Section 5.2.

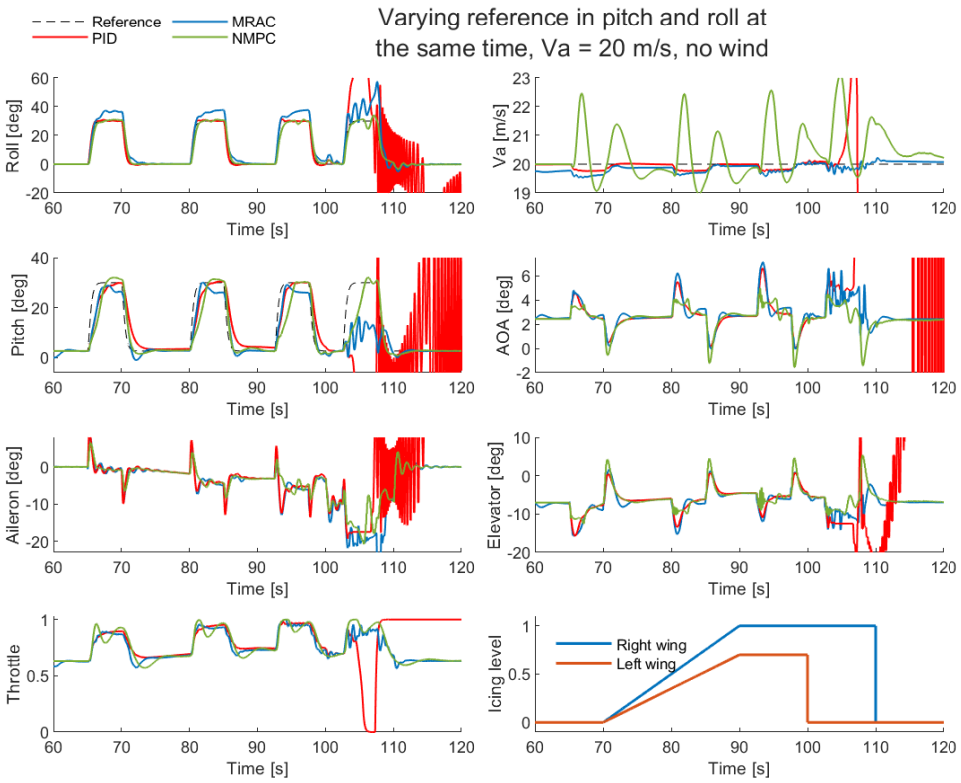

Figure 5.10: Roll and pitch reference change at the same time

Lastly, an additional simulation using the previous UAV model, given in Appendix A, has been run for the PID and MRAC controllers, to assess to what degree not updating their tuning might have resulted in a worse response. The reduced airspeed and wind disturbances simulations were repeated for this case, giving the results presented in Figures 5.11 and 5.12. In general it can be seen that the error is slightly smaller using the old thrust model, depending on the controller and the disturbance type. In the reduced airspeed case, the initial MRAC airspeed error is around $3\,m/s$ larger because the updated thrust model was used, but this error decreases with time. The roll and pitch response, as well as the PID's response, remain the same with both thrust models. In the severe wind conditions case we see a slight difference, with the controllers performing better when the old thrust model is used, except for the MRAC's airspeed performance, which turns out to be better with the new model.

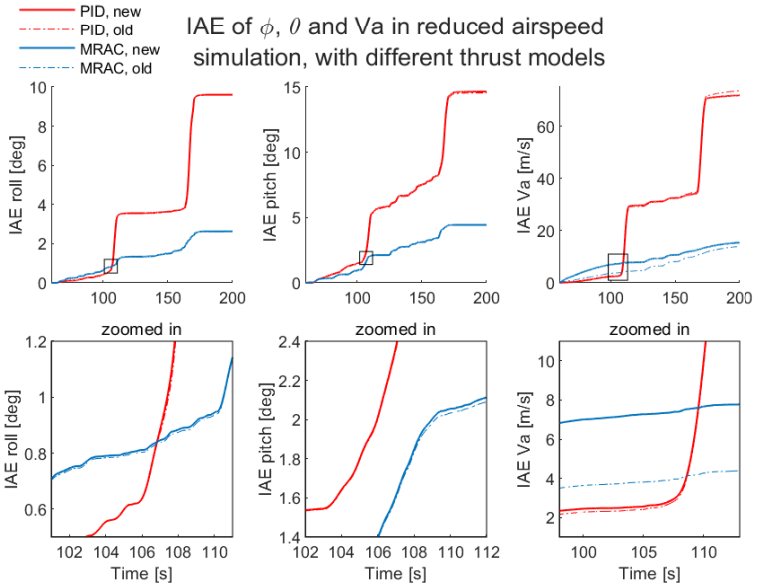

Figure 5.11: Roll and pitch reference change at the same time

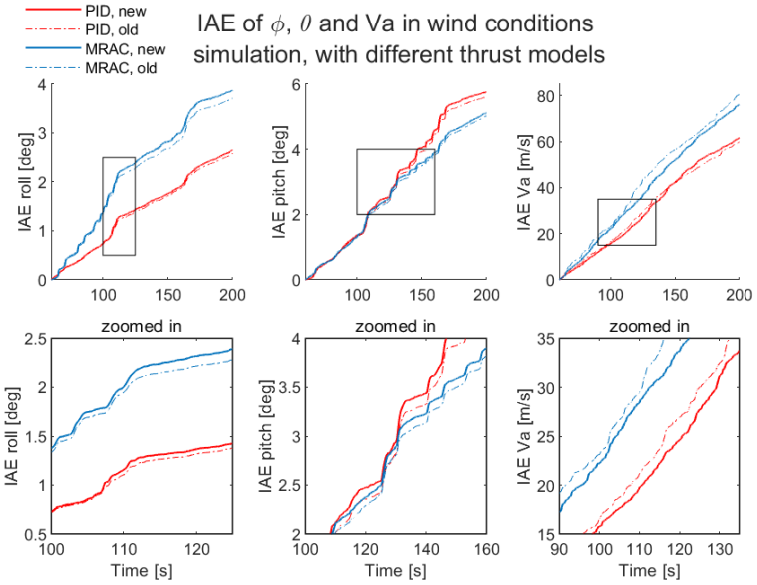

Figure 5.12: Roll and pitch reference change at the same time

5.5 Guidance simulation

In this simulation, the path-following performance of the three controllers is tested, where the references are given from the guidance controller, described in Section 4.1.2, and the simulation is run with moderate wind gusts, as described in 4.4.4.

The asymmetric icing test has been modified to be more severe, with 60s of 100% asymmetric icing in total, as seen in Figure 4.5. Figure 5.13 shows the results in 3D, while Figures 5.14 and 5.15 show the performance from the two sides.

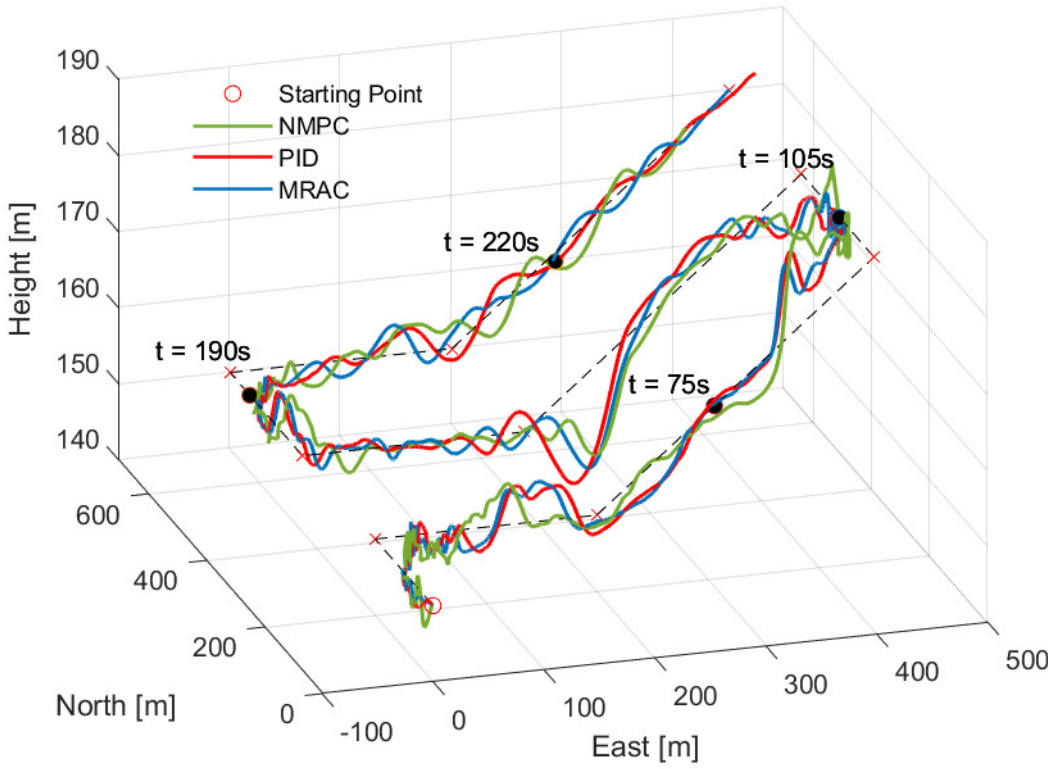

Figure 5.13: Path-following performance of the NMPC, PID and MRAC controllers in the guidance simulation, with moderate wind conditions.

In Figure 5.14 we see a very similar performance in all three controllers, following the path with no clear differences when it comes to how fast and accurate they take turns. Looking at figure 5.15 we can see however that the NMPC tends to oscillate more between heights, reaching $h = 186\,m$ at $342\,m$ on the North axis. All three controllers seem to have larger discrepancies in height when they come close to changing it, as seen in figures 5.13 and 5.15, where the PID controller drops to $h = 142\,m$ when the reference is $h_{ref} = 155\,m$ and getting to $h_{ref} = 150\,m$.

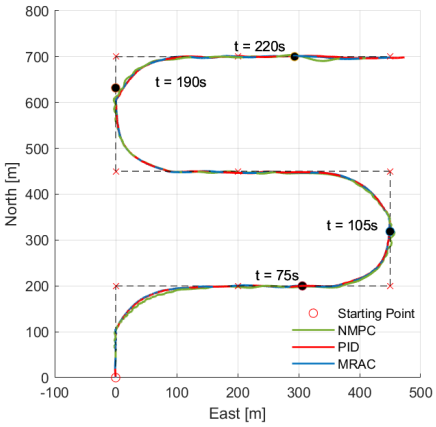

Figure 5.14: Guidance simulation results seen from the East-North plane.

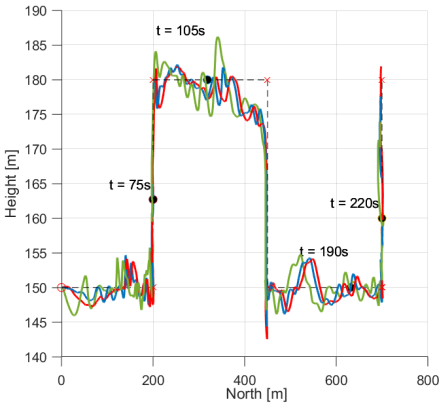

Figure 5.15: Guidance simulation results seen from the North-Height plane.

When it comes to the severe asymmetric icing case, happening at $t \in (75, 105)s$ and $t \in (190, 220)s$, it does not seem to affect the performance significantly, as the biggest height errors are not happening inside these intervals. This corresponds to the results found in the wind disturbance simulation, where as seen on Figure 5.9, the reference errors increase throughout the whole simulation time due to the wind conditions, with only slight additional increases during the 100% asymmetric icing interval.

Chapter 6

Discussion

In this chapter, the performance of the NMPC is analysed, discussing to what degree it is able to handle severe icing and wind conditions, by comparing it to the performance of PID and MRAC controllers. This is based on the simulation results presented in Chapter 5. Finally, the future work will be discussed as well in Section 6.5.

6.1 NMPC performance and tuning

To compare the performance of the NMPC presented in Section 3.1 to the PID and NMPC controllers developed in [6], several simulation cases have been tested, with varying commanded airspeeds, icing and wind conditions. In the attempt to compare the performances in an objective way, the Integral Absolute Error of the errors in roll ϕ, pitch θ and airspeed V_a has been used as introduced in Section 4.4.1.

When it comes to the NMPC performance, it was observed that it was significantly worse at tracking airspeed than the PID and MRAC controllers. In both the PID and MRAC case, airspeed was tracked using a PI controller, as shown in Section 3.3, while in the NMPC case, the airspeed is part of the reference vector used in the NMPC scheme for output tracking. Therefore, by solving the NLP described in 3.1, the controller tracks both airspeed and attitude. Given that both the airspeed V_a and angle of attack α are part of the state vector, and considering both their constraints when solving the NLP, there is a trade-off between them which can make the problem infeasible with certain tunings. Taking into consideration the stall angle, which in this thesis was assumed to be as low as $\alpha_{stall} = 4°$, and having that as an upper bound in the problem formulation, the airspeed needs to be able to increase if needed above the reference to be able to produce the same lift with a lower angle of attack. The airspeed tuning could probably be improved, but a much more aggressive one can make the problem become infeasible if the hard constraint on the angle of attack is kept. The current tuning therefore results in a poorer airspeed performance compared to the PID and MRAC controllers, but

also in a lower angle of attack, which was not taken into account by the other controllers.

During the severe wind disturbance simulation, the wind velocity is $V_{wind} = 23\,m/s$, which combined to the previous initial airspeed of $18\,m/s$ gives an initial airspeed of $Va = 41\,m/s$. To make the optimization problem feasible during this simulation, the upper bound of the airspeed was set to $Va_{max} = 45\,m/s$. It is however an unrealistic design choice, given that a normal value should have been $Va_{max} = 25\,m/s$ to stay within safety limits, so a better approach would have been to increase the slack weight on the airspeed variable, to make the controller try to reach the constraint first and the reference later.

When judging the performance of the NMPC, one point to remember is that there are differences between the UAV model the NMPC uses and the one implemented in the Matlab/ Simulink simulator, as mentioned in Section 4.2.1. The discrepancies between the observed states and their predictions are handled by the observer disturbance, whose tuning also has an effect on the performance. A higher weight on the angular velocity error can increase the roll and pitch performance, but when it comes to airspeed the same problem previously mentioned remains because of the constraints on the angle of attack.

The performance of the NMPC varied based on the different disturbances tested, but judging by the results it seems that it was generally less affected by the severe asymmetric icing interval in the simulations than the PID and the MRAC. During this interval, the PID's performance under low airspeed conditions was notably poor. It is safe to assume that a reason for that was the tuning, which could have attenuated the poor performance in the severe asymmetric icing interval if it was less aggressive. This however would decrease the tracking performance in the presence of high disturbances, such as in the severe wind conditions simulation. Furthermore, the tuning of the PID and MRAC controllers were left outside the scope of this project thesis, using the same tuning found by Högnadottir in her work [6]. However, with the updates in the UAV model to use Coates' propeller thrust model [16] with Müller's extension [15] to include the effects of icing, one can wonder if the PID and MRAC might have needed an updated tuning as well. This was checked in Section 4.4.5, but the results in Figure 5.11 and 5.12 showed negligible differences in performance. This is probably because although the new thrust model was found to be more accurate [16], it does not differ as much from the old one to significantly affect the overall response of the UAV, especially when compared to major disturbances like reduced airspeed or severe wind gusts. It can then be assumed that the PID and MRAC controllers are represented fairly.

6.2 Effects of asymmetric icing on the wings

All simulations were run with a varying level of icing and asymmetry on the wings, with an increase in icing level during a certain time interval, followed by an interval with 100% asymmetric icing on the wings, where one of the wings was fully iced while the other one was clean, assuming an ice shedding event had occurred.

The latter was shown to result in the most severe case, with the fastest performance degradation for all three controllers. The NMPC controller, however, seemed to be the least affected by the severe asymmetry case, since its IAE in Figures 5.6 and 5.9 seems to grow more linearly throughout the whole simulation time, as opposed to the bigger "steps" seen in the PID's and MRAC's IAEs in Figure 5.6, during the reduced airspeed simulation case. Generally, the biggest errors due to asymmetry can be seen in the roll performance of the three controllers, which is visibly affected during the interval with 100% asymmetric icing, and especially while the reference in pitch is changing. Since this interval simulates a sudden ice shedding event on the right wing while the left wing remains iced, it is clear that the weight difference would produce a roll moment, and therefore a larger deviation from the roll reference of 0°. Because of this, the coupling between roll and pitch increases during asymmetry, producing errors in each other's performances. During the icing intervals, an increase in throttle and angle of attack can also be observed, which can be explained by the increase of drag and decrease of lift as a result of the icing.

6.3 Effects of icing on the propeller

The effects of icing on the propeller have also been added to the simulations, by using the worst-case scenario of maximum performance degradation of the propeller, which was found at $-15°C$ in Section 4.3.1. The performance degradation, represented as the difference between the clean and iced thrust coefficient C_T, also depends on time and on the ratio between airspeed and the propeller's rotation rate, determined by δ_t. It changes therefore following equation 2.32. In the simulation results it can be seen that the difference between the clean and iced thrust coefficient is around 0.02. With the increase of ice on the propeller, the propeller generates less thrust because of a lower C_T, so the UAV has to compensate by increasing its propeller speed, and therefore δ_t, to maintain the same lift. Without ice on the propeller, the UAV would not need to use as much throttle. In addition to a decrease in thrust, the ice accretion on the propeller also increases the power coefficient, which increases the torque produced by the propeller. In the formulation of the NMPC, the effects of the propeller torque have been neglected, as they were small enough to be compensated for and too complex to model. With a maximum increase of power coefficient of 0.014 at $-15°C$ and advance ratio of $J = 0.6$, it is believed that it would not increase the propeller torque substantially, so its effects were assumed to be negligible in this case as well.

6.4 Stall angle

In the simulations performed, stalling of the UAV was taken into account in the formulation of the NMPC problem, by constraining the angle of attack to not go above the upper bound α_{stall}. The models of the PID and MRAC controllers did

not include stalling, and it has not been added in this thesis either, as the focus was to compare the NMPC to the existing models. Therefore, the PID and MRAC responses have to be analysed to verify that the angle of attack does not cross this limit. However, it is not clear what the maximum stall angle in the fully iced case is, as Winter's work [12] suggests that $\alpha \approx 10°$, while Högnadóttir's work [6] suggests an even lower $\alpha \approx 4°$. This could be due to the differences between the configurations used in their simulations. To test the NMPC advantages and limitations, the more conservative $\alpha \approx 4°$ was chosen as an upper bound for α. Although the predicted state of the NMPC never crossed this bound, the actual state did not always follow this limit, as there are discrepancies between the NMPC model and the one used in the simulator, described in Section 4.2.1. Nevertheless, the angle of attack was kept very low compared to the PID and MRAC, increasing the probability of recovery in the case the stall limit is crossed. The angle of attack was always well under 10°, and because the stall limit is not clearly determined in literature, and it depends on how iced the wing is, these results were assumed to be good enough. As mentioned in the previous section, the limit on the angle of attack did compromise the airspeed tracking performance, and possible improvements are discussed in the next section.

6.5 Future work

Before applying the solution proposed by the NMPC, many improvements should be made. With the current configuration, icing is treated as a disturbance and handled by the disturbance observer of the NMPC. A question that arises is whether this is good enough, or if the performance would be improved by including the effects of icing in the model of the NMPC. When it comes to that, the icing models used can also be improved. The iced propeller model proposed by Müller [15] was tested on a different propeller, and although the modifications made are assumed to be valid for the propeller the Skylwalker X8 uses, a more accurate model can be achieved through experiments with the correct propeller, or through increased knowledge about the two propellers' geometry and the relationship between each other. More experiments of the UAV in icing conditions is also needed to improve the icing model for the wings, where there is uncertainty about the aerodynamic coefficients between the iced and clean state of the wings. In addition, a better understanding about what the stall angle is in the fully iced state, as well as in different icing levels, would help improve the performance and validity of the results. In the NMPC case, the constraints on angle of attack could be changed accordingly, which would not limit the airspeed tracking performance as much as it was in this thesis. The airspeed tracking performance might also be improved with further tuning of the NMPC controller, and since the roll and pitch performance was on par with the one of the PID and MRAC controllers, a next step would be to implement the NMPC on the Skywalker X8.

Chapter 7

Conclusion

In this thesis, a NMPC controller developed by Reinhardt [4] for inner-loop control of roll, pitch and airspeed was implemented in a simulator in Matlab/ Simulink, first developed by Gryte [11] and expanded by Högnadóttir [6], with the objective to test the NMPC performance in icing conditions. The icing conditions include a model for the asymmetric icing on the wings, previously included in the simulator, with the addition of a model for the icing on the propeller, developed by Müller [15]. The NMPC was developed in acados for python, from which a C/C++ code was generated and implemented as an S-function in Matlab, together with a disturbance observer to handle the effects of icing. Its performance was compared to the one of the already well tuned PID and MRAC controllers, developed by Högnadóttir in her work [6].

The performance of the controllers was tested in different simulations, with asymmetric icing conditions on the wings and icing on the propeller. In addition to icing, the controllers were also tested with reduced airspeed or with severe wind conditions, and the results were analysed using the IAE of the tracking errors as a performance metric. When it comes to the performance of the NMPC compared to the PID and MRAC controllers, excepting its struggles tracking airspeed, the roll and pitch tracking performance were similar to the other controllers. The NMPC was able to handle the icing, reduced airspeed and severe wind conditions very well, as compared to the poor performance of the PID in the reduced airspeed case. It was generally the best at tracking pitch, and its roll performance was similar to the MRAC's, with the PID being superior in this case. When it comes to the 100% asymmetric icing intervals, the NMPC seemed to be less affected during certain simulation cases. The PID and MRAC were not further tuned in this thesis, but it was shown that their performance was very similar to the one they would have had if the modifications on the model had not been made.

However, whether the tuning used for the NMPC is the best one is questionable. The NMPC formulation has the advantage of allowing to set constraints on the states, which depending on how seriously they need to be followed, they can include slack terms to ensure the feasibility of the solution. Taking advantage of this feature, a stall limit has been set as the upper bound of the angle of attack

α, to include the effects of stalling in the NMPC. There are uncertainties in the literature about what this stall limit actually is for a fully iced wing, with Winter [12] suggesting $\alpha_{stall} \approx 10$ and Högnadóttir suggesting $\alpha_{stall} \approx 4$, so the latter was chosen to test the limits and features of the NMPC, and with no slack variable on α. This design choice might have been too harsh, considering the stall limit changes depending on the icing level, and it might have restricted the performance of the NMPC too much, resulting in the poor airspeed tracking performance when compared to the other controllers. It was discussed that this could be improved with better understanding of the stall limit, as well as with further tuning of the NMPC and the slack variables.

When it comes to further improvements of the model, in addition to more experiments to expand our knowledge on the aerodynamic coefficients and stall angle in intermediate ice stages, the icing on the propeller model also needs improvements. This is because the original model was developed for a propeller different than the one the Skywalker X8 uses, and, although slightly modified, it is uncertain to what degree the model parameters were transferable. A better understanding about the geometric differences between the two propellers, or experiments with the correct propeller, would increase the validity of the model. In addition, it was discussed that the NMPC could also be improved by including the icing effects in its model.

Bibliography

[1] R. Hann **and** T. A. Johansen, 'Unsettled Topics in Unmanned Aerial Vehicle Icing,' *SAE International, SAE EDGE Research Report EPR2020008*, 2020. DOI: https://doi.org/10.4271/EPR2020008.

[2] N. Gent **and** J. Cansdale, 'Aircraft Icing,' *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 358:2873-2911, 2000.

[3] M. Bragg, A. Broeren **and** L. Blumenthal, 'Iced-Airfoil Aerodynamics,' *Progress in Aerospace Sciences*, 41:323-362, 2005.

[4] D. P. Reinhardt, *On Nonlinear and Optimization-based Control of Fixed-Wing Unmanned Aerial Vehicles*. Doctoral thesis, Norwegian University of Science **and** Technology, 2022.

[5] *MS Windows NT kernel description*, https://ardupilot.org/, Accessed: 2021-12-03.

[6] S. Högnadottir, *Inner-Loop Adaptive Control of FixedWing Unmanned Aerial Vehicles in Icing Conditions*. M.S. thesis, Norwegian University of Science **and** Technology, 2022.

[7] R. Kleiven, *Robust and Gain Scheduled Flight Control of Fixed-wing UAVs in Wind and Icing Conditions*. M.S. thesis, Norwegian University of Science **and** Technology, 2021.

[8] *Acados solver for nonlinear optimal control*, https://github.com/acados/acados, Accessed: 2022-12-15.

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

[10] E. Lavretsky **and** K. A. Wise, *Robust and Adaptive Control: With Aerospace Applications* (Advanced Textbooks in Control and Signal Processing). London; New York: Springer, 2013.

[11] K. Gryte, *High Angle of Attack Landing of an Unmanned Aerial Vehicle*. M.S. thesis, Norwegian University of Science **and** Technology, 2015.

[12] A. Winter, *Systems identification, flight performance, and control of a fixed-wing UAV in icing conditions*. M.S. thesis, Norwegian University of Science **and** Technology, 2019.

[13] Y. Liu, L. Li, W. Chen, W. Tian **and** H. Hu, 'An experimental study on the aerodynamic performance degradation of a UAS propeller model induced by ice accretion process,' *Experimental Thermal and Fluid Science*, **jourvol** 102, **pages** 101–112, 2019, ISSN: 0894-1777. DOI: https://doi.org/10.1016/j.expthermflusci.2018.11.008.

[14] N. Müller, R. Hann **and** T. Lutz, 'UAV Icing: Numerical Simulation of Propeller Ice Accretion,' *AIAA Aviation 2021 Forum*, 2021. DOI: 10.2514/6.2021-2673.

[15] N. Müller **and** R. Hann, 'UAV Icing: A Performance Model for a UAV Propeller in Icing Conditions,' *AIAA Atmospheric and Space Environments Conference*, **june** 2022. DOI: 10.2514/6.2022-3903.

[16] E. M. Coates, A. Wenz, K. Gryte **and** T. A. Johansen, 'Propulsion System Modeling for Small Fixed-Wing UAVs,' **in***2019 International Conference on Unmanned Aircraft Systems (ICUAS)* 2019, **pages** 748–757. DOI: 10.1109/ICUAS.2019.8798082.

[17] P. Fitzpatrick, *Calculation of Thrust in a Ducted Fan Assembly for Hovercraft*. Technical report, Hovercraft Club of Great Britain, 2003.

[18] R. W. Beard, *Uavbook Supplement. Additional thoughts on propeller thrust model.* Technical report, 2014.

[19] *Flying qualities of piloted airplanes*. U.S. Military Specification MIL-F8785C, 1980.

[20] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. Second edition. Wiley-Blackwell, 2021.

[21] S. O. Nevstad, *Autonomous landing of fixed-wing UAV in net suspended by multirotor UAVs*. M.S. thesis, Norwegian University of Science **and** Technology, 2016.

[22] D. I. You, Y. D. Jung, H. M. S. S. W. Cho, S. H. Lee **and** D. H. Shim, 'A Guidance and Control Law Design for Precision Automatic Take-off and Landing of Fixed-Wing UAVs,' *AIAA Guidance, Navigation, and Control Conference*, Aug. 2012. DOI: 10.2514/6.2012-4674.

[23] K. Gryte, R. Hann, M. Alam, J. Roháč, T. A. Johansen **and** T. I. Fossen, 'Aerodynamic modeling of the skywalker x8 fixed-wing unmanned aerial vehicle,' **in***2018 International Conference on Unmanned Aircraft Systems (ICUAS)* 2018, **pages** 826–835. DOI: 10.1109/ICUAS.2018.8453370.

[24] *Matlab + Simulink and Octave Interface*, https://docs.acados.org/matlab_octave_interface/index.html, Accessed: 2022-11-01.

[25] *N-D Lookup Table*, https://se.mathworks.com/help/simulink/slref/ndlookuptable.html, Accessed: 2022-12-15.

[26] *UIUC Propeller Database - Volume 3*, https://m-selig.ae.illinois.edu/props/volume-3/propDB-volume-3.html, Accessed: 2022-12-05.

Appendix A

Fitzpatrick Propeller Model

In the previous version of the Matlab/ Simulink simulator used, the propeller force was based on Fitzpatrick [17] and Beard & McLain [18], and given by

$$\boldsymbol{F}_{prop}^{b} = \begin{bmatrix} \frac{1}{2}\rho S_{prop} C_{prop} V_d (V_d - V_a) \\ 0 \\ 0 \end{bmatrix}, \tag{A.1}$$

where S_{prop} is the area of the propeller, C_{prop} is the aerodynamic coefficient of the propeller, and V_d is the discharge velocity, given by

$$V_d = V_a + \delta_t (k_{motor} - V_a), \tag{A.2}$$

with δ_t being the throttle and k_{motor} the motor constant. The parameters used are given in Table A.1.

Parameter	Value	
ρ	1.2250	kg/m^3
S_{prop}	0.1018	m
C_{prop}	1	
k_{motor}	40	

Table A.1: Parameters for the Fitzpatrick propulsion model.